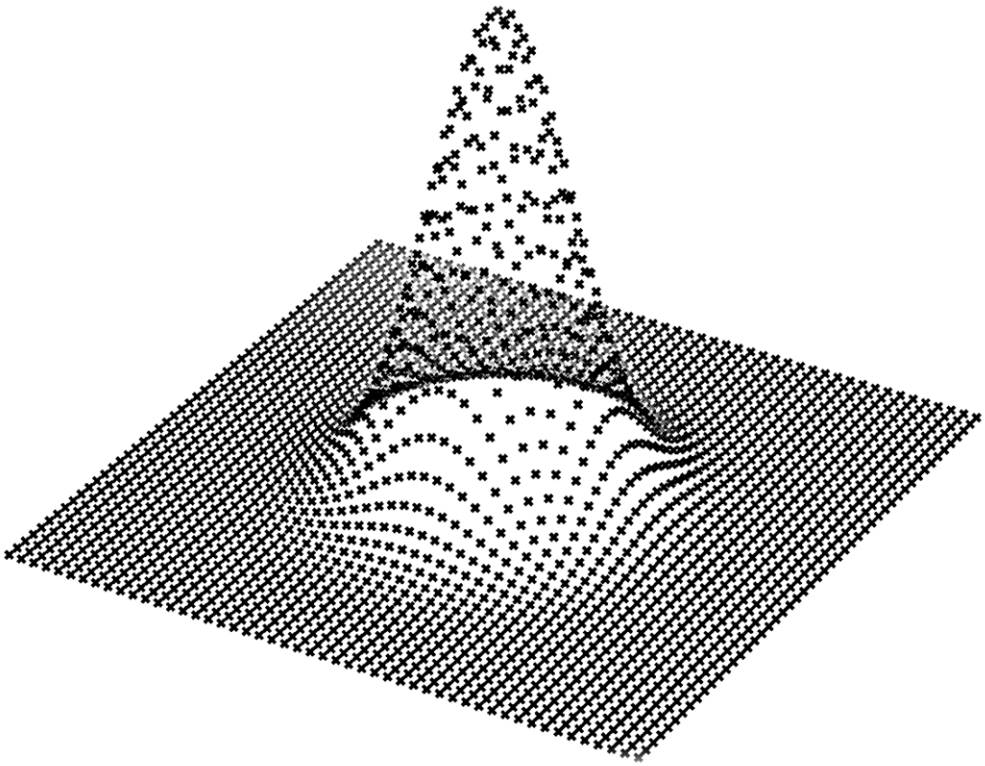
ARISTOTLE UNIVERSITY OF THESSALONIKI
FACULTY OF SCIENCES
DEPARTMENT OF MATHEMATICS

A thesis submitted in partial fulfilment of a
Master's degree in Science

CRYPTOGRAPHY BASED ON LEARNING WITH ERRORS

Florias Papadopoulos



Supervisor: Dimitrios Poulakis

THESSALONIKI 2023

Acknowledgements

I am deeply grateful to Professor Dimitrios Poulakis for his guidance and valuable advice throughout the entire process of researching and writing this thesis. Specifically, I would like to thank him for consistently allowing this paper to be my own work, but giving me valuable feedback when necessary. Moreover, I would also like to extend my thanks to Professor Konstantinos Draziotis for his insightful comments and suggestions.

In a more personal spirit, my heartfelt thanks go out to my friends and especially my girlfriend, for their patience and understanding during my periods of absence and seemingly endless amounts of panic. In addition, I would be remiss in not mentioning my family which has kept my spirits and motivation high during this process.

Lastly, I would like to extend my appreciation to the team behind ChatGPT, whose tool consistently proved invaluable in refining the clarity and polish of specific sections of this thesis, through a continuous back-and-forth collaboration.

"We become what we behold.
We shape our tools and then our tools shape us."
-Marshall McLuhan's contemporary John M. Culkin

Περίληψη

Η διπλωματική αυτή εξερευνά τον ταχέως εξελισσόμενο κόσμο της μετα-κβαντικής κρυπτογραφίας, εστιάζοντας σε κρυπτογραφικά σχήματα που βασίζονται σε πλέγματα. Η ανάλυση ξεκινά με μια εκτενή έρευνα θεμελιωδών εννοιών πάνω στα πλέγματα, ενώ συνεχίζεται με μια λεπτομερή αναφορά σε γνωστά προβλήματα πλεγμάτων που θεωρούνται «υπολογιστικά δύσκολα», καθώς και στους αλγορίθμους που επιχειρούν να τα επιλύσουν.

Ακολούθως, εστιάζουμε στο πρόβλημα Learning with Errors (LWE), πραγματοποιώντας μια σε βάθος έρευνα της θεωρίας και των σχετικών εφαρμογών του σε κρυπτογραφικά σχήματα. Καθώς όμως αυτά δεν είναι αρκετά αποδοτικά για να εφαρμοστούν σε πραγματικές συνθήκες, συνεχίζουμε την μελέτη μας με παραλλαγές τους που βασίζονται σε προβλήματα όπως το Ring-LWE και το Module-LWE, τα οποία είναι LWE προβλήματα σε πλέγματα με κάποια επιπλέον αλγεβρική δομή.

Τέλος, το κλείσιμο της εργασίας γίνεται με μια ενδελεχή διερεύνηση του Kyber, ενός κρυπτογραφικού σχήματος του οποίου η ασφάλεια βασίζεται στη δυσκολία επίλυσης υπολογιστικά δύσκολων προβλημάτων σε «module» πλέγματα. Ξεκινώντας με την παρουσίαση κάποιων σημαντικών κρυπτογραφικών τεχνικών που χρησιμοποιεί το Kyber, η έρευνα μας συνεχίζεται με μια λεπτομερή ανάλυση αυτού και της ασφάλειάς του, κλείνοντας με μια πρόσφατη εφαρμογή του, το υβριδικό κρυπτογραφικό σύστημα X25519Kyber768 που χρησιμοποιείται από τον φυλλομετρητή Google Chrome για την ενίσχυση της ασφάλειας του.

Abstract

This work navigates the evolving world of post-quantum cryptography, particularly focusing on lattice-based cryptographic constructions. Starting with a thorough exploration of fundamental lattice concepts, the study progresses to delve into well-known hard lattice problems and the algorithms attempting to solve them. Afterwards, the focus is shifted to Learning with Errors (LWE), providing a comprehensive examination of LWE theory, its applications, and a concise study of Ring-LWE, a variant that provides more efficient constructions. The conclusion unfolds with an in-depth exploration of CRYSTALS-Kyber, a cryptographic scheme whose security is based on the hardness of solving hard lattice problems in (module) lattices. Beginning with the necessary preliminaries, we then delve into a detailed analysis of Kyber and its security, finishing with a recent real-world application of the scheme, the X25519Kyber768 hybrid KEM that is used to fortify Google Chrome's security.

Preface

With the rapid evolution of quantum computing, which threatens the security of current cryptographic standards, coupled with NIST's (National Institute of Standards and Technology) move to define novel standards for digital signature generation, encryption and key-establishment protocols, there has been a noteworthy surge of interest in post-quantum cryptographic schemes. Lattice-based cryptographic constructions especially hold great promise for post-quantum cryptography as they boast very strong security proofs (based on worst-case hardness), competitively efficient implementations, as well as great simplicity, and even scalability. Thus, it should come as no surprise that, of the four algorithms that NIST chose in 2022 for its post-quantum cryptographic (pqc) standards, three of them were lattice-based.

Even more recently, on 24 August 2023, the first drafts of these standards were published. Among them, the draft of FIPS 203 [Nat23] outlining a cryptographic scheme called Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) Standard, derived from CRYSTALS-Kyber (Kyber, for short). CRYSTALS (CRYptographic SuiTe for Algebraic LatticeS) represents a suite of cryptographic tools submitted to NIST's post-quantum competition in 2017. This suite includes Kyber, a post-quantum key-encapsulation mechanism (KEM) that bases its security on hardness assumptions over a specific family of lattices called module lattices.

Particularly, Kyber's security relies on the hardness of solving the Learning with Errors (LWE) problem in module lattices (Module-LWE problem), which is proven to be at least as hard as solving some well-known hard problems in module lattices. Therefore, for someone already familiar with the basics of public key cryptography, the path towards fully understanding Kyber, one of the most important cryptographic schemes at the forefront of modern security, is quite straightforward: firstly, acquiring knowledge on lattices; followed by an exploration of the LWE problem, its variants and the cryptography that is based on them; ending on the specifics of Kyber itself and concepts surrounding its construction.
This is exactly the path that this thesis follows:

- **Part I (Lattices):** We start in *Chapter 1*, which navigates through the fundamental concepts of lattices, unveiling their intricacies and relevance in cryptographic protocols. From defining basic lattice structures to exploring lattice basis reduction algorithms, this chapter lays the groundwork for understanding the cryptographic significance of lattice-based problems. Subsequently, in *Chapter 2*, the complexity of well-known hard lattice problems is discussed, along with algorithms attempting to solve them.
- **Part II (LWE):** Building upon the lattice foundation, Part II delves into the realm of Learning with Errors (LWE), beginning in *Chapter 3* with a comprehensive exploration of LWE theory and applications, including the hardness of LWE and its cryptographic implications. In *Chapter 4*, this seamlessly transitions into a concise study of the theory behind Ring-LWE, a variant of LWE that serves as the foundation of more efficient primitives, and its applications.
- **Part III (CRYSTALS-Kyber):** Finally, we converge on Kyber by first looking into some necessary preliminaries for it in *Chapter 5*, including some essential concepts from cryptography (transformations from CPA to CCA schemes), optimizations for LWE-based schemes, and the Number Theoretic Transform. These set the stage for a detailed analysis of the Kyber scheme in the final chapter, *Chapter 6*, including an analysis of its security. Our research concludes with a real-world application of Kyber, the X25519Kyber768 hybrid key exchange, which was integrated into Google Chrome in August 2023, providing robust defence against prospective quantum attacks and showcasing the real-world implications of our research.

Keywords: Post-Quantum Cryptography, Lattices, Lattice-based Cryptography, Learning with Errors (LWE), CRYSTALS-Kyber, Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM), X25519Kyber768

Contents

Part I

Lattices

Chapter 1

Fundamental Concepts in Lattices

We commence our study with an introduction to the world of lattices, where the definitions provided are mainly taken from Galbraith's book [Gal18] and Draziotis' book [Δρα22].[1] However, we remark that, even though they represent vectors as rows in their work, we have opted to depict them as columns, aligning with the prevailing approach in most of our subsequent references (e.g. the introductory lectures of Micciancio [Mic12; Mic14; Mic21] that also follow this convention).

More precisely, on this chapter we systematically explore fundamental concepts in lattices, starting by core definitions, and progressing to in-depth discussions on determinants, dual lattices, and hard lattice problems. Subsequently, we establish certain important results around the Hermite and Minkowski bounds, illustrating their role in measuring the hardness of lattice problems. These are also applied on the family of q-ary integer lattices, which are directly related to cryptography. Lastly, we conclude with an overview of lattice basis reduction, and the Lagrange and LLL algorithms, setting the stage for a thorough understanding of important lattice algorithms in the following chapter.

1.1 Lattices and Hard Problems

Before delving into our discussion, it is important to highlight an inherent ambiguity associated with the term "lattice" in mathematics:

- In order theory and abstract algebra, lattices are defined either order-theoretically as a partially ordered set, or as an algebraic structure.
- In geometry and group theory, lattices are regular arrangements of points in Euclidean space, i.e. discrete subgroups of $\mathbb{R}^n$.

For our purposes, we focus exclusively on the latter interpretation.

1.1.1 Notation

The notation presented here will be used this way throughout the thesis, unless stated otherwise:

(i) For $x \in \mathbb{R}$, we use the following:
$\lfloor x \rfloor$ denotes the largest integer not greater than x, i.e. $\lfloor x \rfloor \doteqdot \max\{m \in \mathbb{Z} \,|\, m \leq x\}$.
$\lceil x \rceil$ denotes the smallest integer greater not smaller than x, i.e. $\lceil x \rceil \doteqdot \min\{m \in \mathbb{Z} \,|\, m \geq x\}$.
$\lfloor x \rceil$ denotes the integer closer to x with ties broken upward, i.e. $\lfloor x \rceil \doteqdot \lfloor x + \frac{1}{2} \rfloor$.
$\lceil x \rfloor$ denotes the integer closer to x with ties broken downwards, i.e. $\lceil x \rfloor \doteqdot \lceil x - \frac{1}{2} \rceil$.

(ii) Bold lower-case letters like $\mathbf{x}$ will be used to denote column vectors ($\mathbf{x}^T$ for row vectors), whereas bold upper-case letters like $\mathbf{A}$ will denote matrices. Horizontal concatenation uses a vertical bar, e.g. $[\mathbf{A}|\mathbf{Ax}]$. Functions can be applied entry-wise to vectors., e.g. for $\mathbf{x} = (x_1, \ldots, x_n)$, we have $\lfloor \mathbf{x} \rfloor = (\lfloor x_1 \rfloor, \ldots, \lfloor x_n \rfloor)$.

[1] The latter reference is written in Greek, whereas almost all other references from our bibliography are in English.

(iii) We denote by $||\mathbf{x}||$ the Euclidean norm of a vector $\mathbf{x} \in \mathbb{R}^m$, and note that some of the statements made in this and latter chapters also hold for other norms.

(iv) For a positive integer q, $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ denotes the quotient ring of integers modulo q, i.e. the collection of cosets $a + q\mathbb{Z}$ with the induced addition and multiplication operations. We often write $z \bmod q$ to denote the smallest positive number representing the coset $z + q\mathbb{Z}$, and $y \equiv z \pmod q$ to denote $y + q\mathbb{Z} = z + q\mathbb{Z}$.

(v) The standard asymptotic notation will be used, i.e. $o(\cdot), O(\cdot), \Omega(\cdot), \Theta(\cdot)$, etc. Tildes, like $\tilde{O}(\cdot)$, indicate suppression of logarithmic factors in the main parameter.

(vi) For a set S, we write $a \leftarrow S$ (or $a \xleftarrow{\$} S$) to denote that a is chosen uniformly at random from the set S.

1.1.2 Basic Definitions of Lattices

Definition 1.1. A subset $\mathcal{L} \subset \mathbb{R}^m$ is called a *lattice*, if there is a linearly independent set of vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ in $\mathbb{R}^m$ $(m \geq n)$ such that

$$\mathcal{L} = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{j=1}^{n} a_j \mathbf{b}_j : a_j \in \mathbb{Z} \right\}$$

The vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are called a *lattice basis*. The *lattice rank* is n and the *lattice dimension* is m. If $n = m$, then $\mathcal{L}$ is said to be a *full rank lattice* and is called an n-dimensional lattice. Furthermore, a *basis matrix* $\mathbf{B}$ of a lattice $\mathcal{L}$ is an $m \times n$ matrix formed by taking the columns to be basis vectors $\mathbf{b}_j$ $(1 \leq j \leq n)$. Thus, $\mathcal{L} = \mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\} = \mathbf{B}\mathbb{Z}^n$.

Remarks.

1. The vectors in a lattice form an Abelian group under addition, being a subgroup of $(\mathbb{R}^n, +)$.
2. If $\mathbf{0} \notin \mathcal{L}$, then $\mathcal{L}$ cannot be a lattice.
3. When $n \geq 2$, there are infinitely many choices for the basis of a lattice. Moreover, we can transition from one basis to another for the same lattice (more in the next subsection).

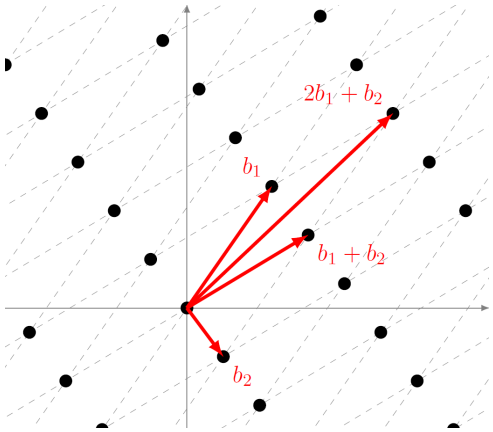


Figure 1.1: A lattice generated from $\mathbf{b}_1$ and $\mathbf{b}_2$ (from Chapter 14 of [Δρα22]).

For our study, a second equivalent definition, that has already been briefly mentioned, will also be useful. For full details and a proof, see Theorem 6.1 of [ST01].

Definition 1.2. A lattice $\mathcal{L}$ is any subset of $\mathbb{R}^m$ that is both:

(a) an *additive subgroup*: $\mathbf{0} \in \mathcal{L}$ and $-\mathbf{x}, \mathbf{x} + \mathbf{y} \in \mathcal{L}$ for every $\mathbf{x}, \mathbf{y} \in \mathcal{L}$; and

(b) *discrete*: every $\mathbf{x} \in \mathcal{L}$ has a neighbourhood[2] in $\mathbb{R}^n$ in which $\mathbf{x}$ is the only lattice point[3].

[2] In more precise terms, a subset is *discrete* if the set $\{\mathbf{x} \in \mathcal{L} : ||\mathbf{x}|| \leq r\}$ is finite, for any real $r > 0$.

[3] We interchangeably use the words *points* and *vectors* for elements of lattices.

Example 1.1. Some well-known lattices are the *integer lattice* $\mathbb{Z}^n$, the *scaled lattice* $c\mathcal{L}$, for $c \in \mathbb{R}$ and lattice $\mathcal{L}$, as well as the *checkerboard lattice* $\{\mathbf{x} \in \mathbb{Z}^n \mid \sum_i x_i \text{ is even}\}$. Moreover, we note that $\mathbf{0}$ is also a trivial lattice with rank 0.

Example 1.2. Let $B = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$ where $\mathbf{b}_1^T = (1, 2, 3)$, $\mathbf{b}_2^T = (1, 2, 1)$, $\mathbf{b}_3^T = (3, 1, 2)$.
Then, one obvious lattice is $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^n\} = \mathbf{B}\mathbb{Z}^n$, which is shown in the figure below.[4]

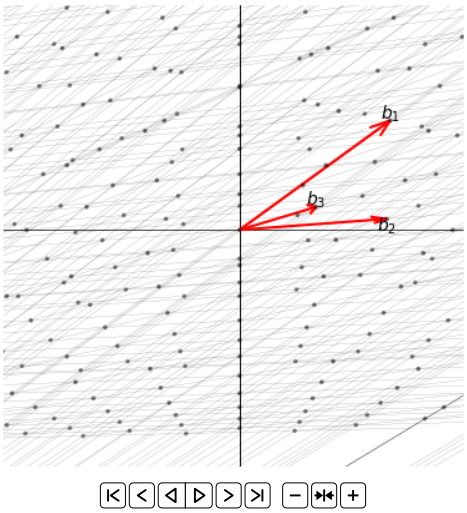


Figure 1.2: The three-dimensional lattice $\mathcal{L}(\mathbf{B})$ of Example 1.2 from several viewpoints.

Additionally, we introduce the concept of successive minima, a key notion for later sections:

Definition 1.3. For a lattice $\mathcal{L} \subset \mathbb{R}^m$ with rank n, the *successive minima* of $\mathcal{L}$ are denoted as $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$. These values are chosen such that, for $1 \leq i \leq n$, λ_i is the smallest possible positive number for which there exist i linearly independent vectors $\mathbf{u}_1, \ldots, \mathbf{u}_i \in \mathcal{L}$ satisfying $||\mathbf{u}_j|| \leq \lambda_i$ for $1 \leq j \leq i$.

Remark 1.1. It follows that $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ and also that $\lambda_1 = \inf_{\mathbf{x} \in \mathcal{L}/\{\mathbf{0}\}}\{||\mathbf{x}||\}$.
Moreover, it can be proven that there is at least one vector $\mathbf{z} \in \mathcal{L}$ such that $||\mathbf{z}|| = \lambda_1$ (proof in Theorem 14.3.1 of [Gal18]) and that λ_1, which is also called *first successive minima*, is independent of the choice of basis for the lattice $\mathcal{L}$. Thus, we also denote λ_1 as $\lambda_1(\mathcal{L})$, underscoring its specificity to each lattice $\mathcal{L}$.

1.1.3 Fundamental Domains and the Determinant of a Lattice

When the dimension m and the rank n of a lattice $\mathcal{L}$ satisfy $m > n$, , it is sometimes convenient to project the lattice $\mathcal{L}$ into $\mathbb{R}^n$ using the construction of the following proposition:

Proposition 1.1. Let $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ be an $m \times n$ basis matrix for a lattice $\mathcal{L}$ where $m > n$. Then, there is a linear map $P : \mathbb{R}^m \to \mathbb{R}^n$ such that $P(\mathcal{L})$ is a lattice of rank n and $||P(\mathbf{u})|| = ||\mathbf{u}||$, $\forall \mathbf{u} \in \mathcal{L}$. Moreover, $\langle \mathbf{b}_i, \mathbf{b}_j \rangle = \langle P(\mathbf{b}_i), P(\mathbf{b}_j) \rangle$, for all $1 \leq i < j \leq n$.
If the linear map is represented by an $n \times m$ matrix $\mathbf{P}$ so that $P(\mathbf{u}) = \mathbf{Pu}$, then a basis matrix for the image of $\mathcal{L}$ under the projection P is the $n \times n$ matrix $\mathbf{PB}$, which is invertible.

Proof: See Lemma 16.1.5 of [Gal18], noting again that there the vectors are rows, not columns.

[4] To enhance comprehension of 3D objects, we have chosen to add multiple views within the three-dimensional space. However, these diverse perspectives may not be accessible on all devices for viewing.

Proposition 1.2. Two $m \times n$ matrices $\mathbf{B}$ and $\mathbf{B}'$ generate the same lattice $\mathcal{L}$ if and only if $\mathbf{B}' = \mathbf{BU}$, where $\mathbf{U}$ is a *unimodular matrix*.[5]

Proof: See Lemma 16.1.6 of [Gal18].

Definition 1.4. The *fundamental parallelepiped* of a lattice $\mathcal{L}(\mathbf{B})$, where $\mathbf{B}$ is an $m \times n$ matrix, is defined as the set $\mathcal{P}(\mathbf{B}) = \{\mathbf{Bx} \,|\, \mathbf{x} \in [0,1)^n\}$.

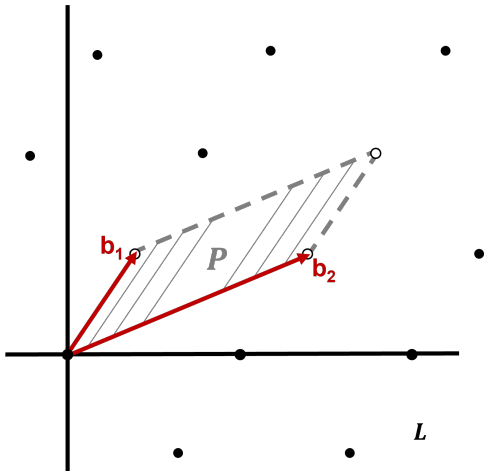


Figure 1.3: Fundamental parallelepiped of a 2D lattice $\mathcal{L}$ generated from $\mathbf{b}_1$ and $\mathbf{b}_2$.

Our main motivation for the creation of the construction of Proposition 1.1 was the preservation of lengths by linear maps, which can be extended to the preservation of volumes. This concept becomes useful when defining and computing the volume of a non full rank lattice:

Definition 1.5. The *determinant* or *volume* of a lattice $\mathcal{L}$ is the volume of the fundamental parallelepiped of any[6] basis $\mathbf{B}$ for $\mathcal{L}$, symbolized by $\det(\mathcal{L})$. Moreover,

- for $n = m$, we have $\det(\mathcal{L}) = |\det(\mathbf{B})|$.
- for $n < m$, we have $\det(\mathcal{L}) = |\det(\mathbf{PB})|$, where $\mathbf{P}$ is the matrix from Proposition 1.1.

Proposition 1.3.

(a) Let $\mathcal{L}$ be a lattice in $\mathbb{R}^m$ of rank n with basis matrix $\mathbf{B}$. Then, $\det(\mathcal{L}) = \sqrt{\det(\mathbf{B}^T\mathbf{B})}$.

(b) Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be an ordered basis for a lattice $\mathcal{L}$ in $\mathbb{R}^m$ and let $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ be its Gram-Schmidt orthogonalisation[7]. Then, $\det(\mathcal{L}) = \prod_{i=1}^{n} ||\mathbf{b}_i^*||$.

Proof: See Lemma 16.1.12 for (a) & Lemma 16.1.14 for (b) of [Gal18].

1.1.4 Dual Lattices and Dual Bases

Another concept that will prove useful later is that of dual lattices and their respective bases. However, we first remind the notion of the *Euclidean inner product* in $\mathbb{R}^n$: for two vectors $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{R}^n$, we have $\langle \mathbf{x}, \mathbf{y} \rangle = x_1 y_1 + \ldots + x_n y_n$.

Definition 1.6. Let $\mathcal{L} \subseteq \mathbb{R}^m$ be a lattice and write $V \subseteq \mathbb{R}^m$ for the $\mathbb{R}$-vector space spanned by the vectors in $\mathcal{L}$. The *dual lattice* of $\mathcal{L}$ is $\mathcal{L}^* = \{\mathbf{y} \in V \,|\, \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z} \text{ for all } \mathbf{x} \in \mathcal{L}\}$.

[5] An $n \times n$ matrix with integer entries and determinant ± 1 is called a unimodular matrix.
[6] The determinant is independent of the choice of basis matrix $\mathbf{B}$ and projection $\mathbf{P}$ (Lemma 16.1.9 of [Gal18]).
[7] For more information on G-S orthogonalisation, see Section 14.2 of [Δρα22] and Section A.10.2 of [Gal18].

Remark 1.2. The concept of dual lattices parallels the duality in vector spaces:
In the context of euclidean vector spaces, the dual is the set of linear functions $\phi : V \to \mathbb{R}$, represented by a vector $\mathbf{v} \in V$ such that $\phi(\mathbf{x}) = \langle \mathbf{v}, \mathbf{x} \rangle$. Similarly, for lattices, the dual is the set of linear functions $\phi : V \to \mathbb{Z}$, represented as vectors in $\text{span}(\mathcal{L})$, replacing $\mathbb{R}$ with $\mathbb{Z}$.

Example 1.3. The dual lattice of $\mathbb{Z}^n$ is $\mathbb{Z}^n$ and the dual of $c\mathcal{L}$ is $c^{-1}\mathcal{L}^*$, for $c > 0$ and lattice $\mathcal{L}$.

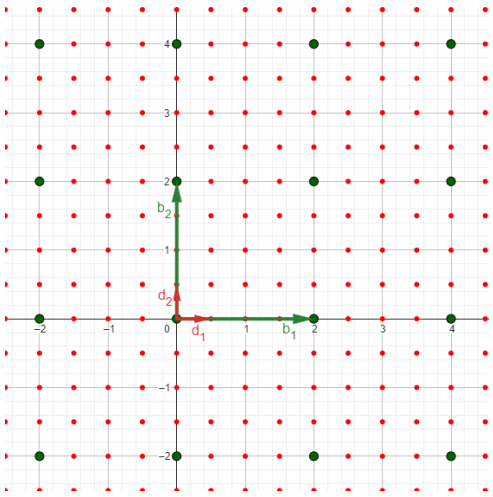


Figure 1.4: The lattices $c\mathcal{L}$ (green points) and $(c\mathcal{L})^* = c^{-1}\mathcal{L}^*$ (red points) for $c = 2$ & $\mathcal{L} = \mathbb{Z}^2$.

Proposition 1.4. For every lattice $\mathcal{L}$ with a basis $\mathbf{B}$, the dual lattice $\mathcal{L}^*$ has a basis $\mathbf{D} = \mathbf{B}(\mathbf{B}\mathbf{B})^{-1}$. Moreover, the determinant of the dual lattice is given by $\det(\mathcal{L}^*) = 1/\det(\mathcal{L})$.

Proof: See Theorem 3 & Proposition 6 of the "Introduction to lattices" notes from [Mic12].

1.1.5 Hard Problems on Lattices

We now define the computational and decision variants of two well-known hard problems on lattices. Note also that in most cryptographic applications only full rank lattices are used (i.e. $n = m$) and the d used in the definitions below is usually taken as 1.

Definition 1.7. Let $\mathcal{L}$ be an n-dimensional lattice.

1. **Shortest Vector Problem (SVP).** Given a basis matrix $\mathbf{B}$ for $\mathcal{L}$, compute a nonzero vector $\mathbf{u} \in \mathcal{L}$ such that $||\mathbf{u}||$ is minimal (i.e. $||\mathbf{u}|| = \lambda_1(\mathcal{L})$).
2. **Decisional Shortest Vector Problem (GapSVP)** Given a lattice basis matrix $\mathbf{B}$ and a real number $d > 0$, determine whether $\lambda_1(\mathcal{L}) \leq d$ or $\lambda_1(\mathcal{L}) > d$.
3. **Closest Vector Problem (CVP).** Given a basis matrix $\mathbf{B}$ for $\mathcal{L}$ and a point $\mathbf{w} \in \mathbb{R}^n$, compute $\mathbf{u} \in \mathcal{L}$ such that $||\mathbf{w} - \mathbf{u}||$ is minimal.
4. **Decision Closest Vector Problem (GapCVP).** Given a basis matrix $\mathbf{B}$ for lattice $\mathcal{L}$, a point $\mathbf{w} \in \mathbb{R}^n$, and a real number $d > 0$, determine whether there exists a vector $\mathbf{u} \in \mathcal{L}$ such that $||\mathbf{w} - \mathbf{u}|| \leq d$ or not.

Of particular importance to lattice cryptography are the *approximation* variants of these problems, which are parametrized by an *approximation factor* $\gamma \geq 1$ that is typically taken to be a function of the lattice rank n, i.e. $\gamma = \gamma(n)$.

Definition 1.8. Let $\mathcal{L}$ be an n-dimensional lattice and fix a $\gamma(n) = \gamma > 1$.

1. **Approximate SVP (SVP_γ).** Given a basis matrix $\mathbf{B}$ for $\mathcal{L}$, compute a nonzero vector $\mathbf{u} \in \mathcal{L}$ such that $||\mathbf{u}|| \leq \gamma\lambda_1(\mathcal{L})$.

2. **Approximate GapSVP (GapSVP$_\gamma$).** Given a lattice basis matrix $\mathbf{B}$ and a real number $d > 0$, determine wheter $\lambda_1(\mathcal{L}) \leq d$ or $\lambda_1(\mathcal{L}) > \gamma d$.[8]
3. **Approximate CVP (CVP$_\gamma$).** Given a basis matrix $\mathbf{B}$ for $\mathcal{L}$ and a point $\mathbf{w} \in \mathbb{R}^n$, compute $\mathbf{u} \in \mathcal{L}$ such that $||\mathbf{w} - \mathbf{u}|| \leq \gamma ||\mathbf{w} - \mathbf{y}||$, for all $\mathbf{y} \in \mathcal{L}$.
4. **Approximate GapCVP (GapCVP$_\gamma$).** Given a basis matrix $\mathbf{B}$ for lattice $\mathcal{L}$, a point $\mathbf{w} \in \mathbb{R}^n$, and a real number $d > 0$, determine whether the distance $dist(\mathbf{w}, \mathcal{L})$ of the point from the lattice is $\leq d$ or $> \gamma d$, where $dist(\mathbf{w}, \mathcal{L}) = \min\{||\mathbf{x} - \mathbf{w}|| : \mathbf{x} \in \mathcal{L}\}$.

Finally, we also mention three more hard problems which have a strong connection to lattice cryptography. We start with the *Bounded Distance Decoding Problem* (BDD) which asks to find the lattice point that is closest to a given target vector $\mathbf{w} \in \mathbb{R}^n$, where the target is promised to be "rather close" to the lattice. This promise and the uniqueness of the solution when $\alpha < 1/2$ are the main differences between BDD$_\alpha$ and CVP$_\gamma$, wherein the target can be an arbitrary point.

Definition 1.9. (Bounded Distance Decoding Problem (BDD$_\alpha$))
Fix $0 < \alpha < 1$. Given a basis matrix $\mathbf{B}$ for an n-dimensional lattice $\mathcal{L}$ and a vector $w \in \mathbb{R}^n$ such that there is a lattice point $\mathbf{u} \in \mathcal{L}$ with $||\mathbf{w} - \mathbf{u}|| \leq \alpha \lambda_1(\mathcal{L})$, compute $\mathbf{u}$.

The definitions of the other two problems, uSVP and SIVP, are sourced from [Pei16], where more detailed information is available about them. Furthermore, regarding the exact complexity (depending on the value of γ) of these and the previous problems, a more in-depth exploration will be conducted in the next chapter.

Definition 1.10. Let $\mathcal{L}$ be an n-dimensional lattice and fix a $\gamma(n) = \gamma > 1$.

1. **unique Shortest Vector Problem (uSVP$_\gamma$).** Suppose that $\mathcal{L}$ has a "unique" shortest vector, which means that the length of a shortest nonzero vector $\mathbf{v} \in L$ is at least a factor smaller than the lengths of all lattice vectors not parallel to $\mathbf{v}$, i.e. $\lambda_2(\mathcal{L}) \geq \gamma \lambda_1(\mathcal{L})$. Given a basis matrix $\mathbf{B}$ of $\mathcal{L}$, find the shortest nonzero vector in $\mathcal{L}$.
2. **Approximate Shortest Independent Vectors Problem (SIVP$_\gamma$).** Given a basis $\mathbf{B}$ of a lattice $\mathcal{L}$, output a set $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\} \subset L$ of n linearly independent lattice vectors where $||\mathbf{s}_i|| \leq \gamma \lambda_n(\mathcal{L})$, for all i.

1.1.6 The Hermite and Minkowski Bounds

We have observed that there are numerous choices for a basis in a given lattice $\mathcal{L}$. A fundamental challenge in lattice theory is to compute a "nice" lattice basis for $\mathcal{L}$; specifically one where the vectors are relatively short and close to orthogonal. This is directly correlated to solving some of the hard problems mentioned above as a "nice" basis can be a useful tool in computing a shortest vector or solving CVP (using Babai's Algorithm, as we will see in a latter chapter).

As an example, in the figure below we compare a "nice" basis $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2]$ (in green) with a "bad" basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2]$ (in blue) of a two-dimensional lattice $\mathcal{L}$.

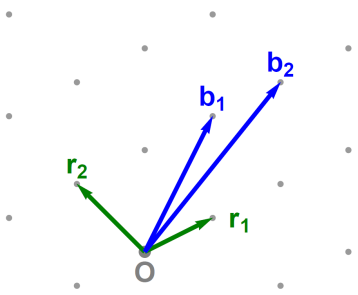


Figure 1.5: Comparing bases in a two-dimensional lattice (from [Laa15]).

[8] If $\lambda_1(\mathcal{L})$ falls between d and γd, either answer is acceptable. Alternatively,this version can be considered as a *promise problem*, where the input $\mathbf{B}$ is guaranteed to satisfy one of the two cases.

How orthogonal is a basis of a lattice? In general, the optimal basis would have vectors that are mutually orthogonal (perpendicular) to each other. While it's unusual for a lattice to possess an orthogonal basis, if it does, the determinant is straightforwardly the product of the lengths of the basis vectors. Nonetheless, even with non-orthogonal bases, we can still establish an upper bound for $\det(\mathcal{L})$ through the inequality below:

Theorem 1.1. (Hadamard's Inequality)
Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be any basis for a lattice $\mathcal{L}$. Then, $\det(\mathcal{L}) \leq ||\mathbf{b}_1|| \cdots ||\mathbf{b}_n||$, with equality if and only if $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are pairwise orthogonal.

Proof: Briefly, this holds because the volume of a parallelepiped is bounded by the product of its side lengths. For more see Theorem 1.4.1 of [Sil20b].

Therefore, it is evident that the degree to which it deviates from being an equality serves as a measure of the non-orthogonality of a basis.

Moreover, to further enhance our view in the matter, we can turn to a well-known theorem by Minkowski, which asserts that every lattice possesses at least one "reasonably" orthogonal basis. An overview of its proof is included after some important remarks.

Theorem 1.2. (Minkowski's Theorem)
There is a constant γ so that for all lattices $\mathcal{L}$ of dimension n:

(a) There is a nonzero vector $\mathbf{u} \in \mathcal{L}$ satisfying $||\mathbf{u}|| \leq \gamma^{1/2} \det(\mathcal{L})^{1/n}$.

(b) There is a basis $\mathbf{u}_1, \ldots, \mathbf{u}_n$ for $\mathcal{L}$ satisfying $||\mathbf{u}_1|| \cdots ||\mathbf{u}_n|| \leq \gamma^{n/2} \det(\mathcal{L})$.

Remark 1.3. Particularly, the above inequalities hold true for

$$\gamma = \frac{4}{\pi} \cdot \Gamma\left(\frac{1}{2}n + 1\right)^{2/n} \approx \frac{2n}{\pi \mathrm{e}}.$$

Here $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ is the gamma function. The approximation, which is valid for large n, comes from Stirling's formula: $\Gamma(z+1) \approx \sqrt{2\pi z}\left(\frac{z}{e}\right)^z$. As we will see in the overview of the proof, the gamma function appears because the volume of a unit ball in $\mathbb{R}^n$ is $\pi^{n/2}/\Gamma\left(\frac{1}{2}n + 1\right)$.

Definition 1.11. The first part of the theorem above states that there is a constant γ so that for every n-dimensional lattice $\mathcal{L}$, the shortest nonzero vector in $\mathcal{L}$ has length at most $\gamma^{1/2} \det(\mathcal{L})^{1/n}$. The smallest such γ is denoted γ_n and called *Hermite's constant*, i.e.

$$\gamma_n \doteqdot \sup_{\text{Lattices } \mathcal{L} \subset \mathbb{R}^n} \left\{ \lambda_1(\mathcal{L})^2 \det(\mathcal{L})^{-2/n} \right\}.$$

Remark 1.4.

(a) From the remark above we have $\gamma_n \lesssim 2n/\pi e$.
A refined estimate due to Blichfeldt [Bli29] states that $\gamma_n \leq \frac{2}{\pi}\Gamma\left(\frac{1}{2}n + 2\right)^{2/n} \approx n/\pi e$.

(b) The exact value of γ_n is known only for $n \leq 8$ and $n = 24$; see [Sil20b] for details.

Overview of Minkowski's Theorem proof. Several approaches exist for proving Minkowski's Theorem. In [Sil20b], a proof using Voronoi cells, commonly employed in lattice theory, is presented. An alternative approach can be found in Prof. Silverman's lecture in [Sil20a]. However, in this thesis, we primarily follow the proof outlined in [Mic14], proving one theorem and its corollary, and then diverging from there to prove Minkowski's theorem.

Theorem 1.3. (Blichfeldt's Theorem) Given a lattice $\mathcal{L}(\mathbf{B})$ and a set $S \subseteq span(\mathbf{B})$, if $vol(S) > \det(\mathcal{L})$ then S contains two points $\mathbf{z}_1, \mathbf{z}_2 \in S$ such that $\mathbf{z}_1 - \mathbf{z}_2 \in \mathcal{L} \backslash \{\mathbf{0}\}$.

Proof. First, we make three helpful remarks about domains of the form $\mathcal{P}(\mathbf{B}) + \mathbf{x}$:

(i) Each so-called *translated fundamental domain* $\mathcal{P}(\mathbf{B}) + \mathbf{x} \doteqdot \{\mathbf{B}\mathbf{v} + \mathbf{x} \,|\, \mathbf{v} \in [0,1)^n\}$ for $\mathbf{x} \in \mathcal{L}$, contains exactly one point of $\mathcal{L}$.

(ii) The translated fundamental domains cover $\mathbb{R}^n$, i.e. $\cup_{\mathbf{x}\in\mathcal{L}}(\mathcal{P}(\mathbf{B})+\mathbf{x})=\mathbb{R}^n$.
(iii) Each of the (translated) fundamental domains has volume $\det(\mathcal{L})$, by definition.

Now, consider the sets $S_\mathbf{x}=S\cap(\mathcal{P}(\mathbf{B})+\mathbf{x})$, where $\mathbf{x}\in\mathcal{L}$. These sets form a partition of S, i.e. they are pairwise disjoint and $S=\bigcup_{\mathbf{x}\in\mathcal{L}}S_\mathbf{x}$, as can clearly be seen in the first and second part of Figure 1.6. In particular, we have $vol(S)=\sum_{\mathbf{x}\in\mathcal{L}}vol(S_\mathbf{x})$.

Notice that the shifted sets $S_\mathbf{x}-\mathbf{x}=(S-\mathbf{x})\cap\mathcal{P}(\mathbf{B})$ are all contained in $\mathcal{P}(\mathbf{B})$, as shown in the third part of the illustration for the two-dimensional case. Our goal is to prove that the $S_\mathbf{x}$ cannot all be mutually disjoint. Since $vol(S_\mathbf{x})=vol(S_\mathbf{x}-\mathbf{x})$, we have

$$vol(\mathcal{P}(\mathbf{B}))=\det(\mathcal{L})<\mathrm{vol}(S)=\sum_{\mathbf{x}\in\mathcal{L}}vol(S_\mathbf{x})=\sum_{\mathbf{x}\in\mathcal{L}}vol(S_\mathbf{x}-\mathbf{x}).$$

As $S_\mathbf{x}-\mathbf{x}\subseteq\mathcal{P}(\mathbf{B})$, the inequality above implies that these sets cannot be disjoint (i.e. there exist two distinct vectors $\mathbf{x}\neq\mathbf{y}\in\mathcal{L}$ such that $(S_\mathbf{x}-\mathbf{x})\cap(S_\mathbf{y}-\mathbf{y})\neq\emptyset$.

Let $\mathbf{z}$ be any vector in the (non-empty) intersection $(S_\mathbf{x}-\mathbf{x})\cap(S_\mathbf{y}-\mathbf{y})$ and define

$$\mathbf{z}_1=\mathbf{z}+\mathbf{x}\in S_\mathbf{x}\subseteq S,\quad \mathbf{z}_2=\mathbf{z}+\mathbf{y}\in S_\mathbf{y}\subseteq S.$$

These two vectors satisfy $\mathbf{z}_1-\mathbf{z}_2=\mathbf{x}-\mathbf{y}\in\mathcal{L}\backslash\{\mathbf{0}\}$.

□

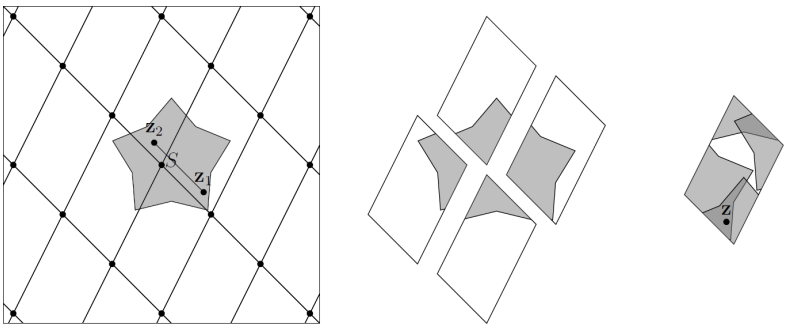


Figure 1.6: Illustration of Blichfeldt's theorem (from [Mor4]).

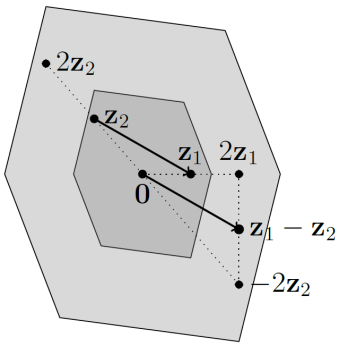


Figure 1.7: Illustration of Minkowski's convex body theorem (from [Mor4]).

Corollary 1.1. (Minkowski's Convex Body Theorem)
Let $\mathcal{L}(\mathbf{B})$ be a full-rank lattice. If $S\subseteq span(\mathbf{B})$ is a symmetric, convex[9] body of volume $vol(S)>2^n\det(\mathcal{L})$, then S contains a nonzero lattice point.

Proof. Consider the set $S/2=\{\mathbf{x}:2\mathbf{x}\in S\}$. When $n=1$, the volume of $S/2$ is half the volume of S, and for $n=2$, it becomes a quarter of the volume of S. On the same spirit the volume of $S/2$ for $S\subset\mathbb{R}^n$ satisfies

$$vol(S/2)=2^{-n}vol(S)>\det(\mathcal{L}).$$

Hence, by Blichfeldt's theorem, there exist $\mathbf{z}_1,\mathbf{z}_2\in S/2$ such that $\mathbf{z}_1-\mathbf{z}_2\in\mathcal{L}\backslash\{0\}$. Moreover, by definition of $S/2$, we have that $2z_1,2z_2\in S$. Since S is symmetric, we also have $-2z_2\in S$ and by convexity,

$$\mathbf{z}_1-\mathbf{z}_2=\frac{2\mathbf{z}_1-2\mathbf{z}_2}{2}\in S.$$

Thus, from the above, $\mathbf{z}_1-\mathbf{z}_2$ is a nonzero lattice vector contained in the set S (see Figure 1.7).

□

[9]A subset S of $\mathbb{R}^n$ is considered to be *convex* if any linear combination $a\mathbf{x}_1+(1-a)\mathbf{x}_2,(0\leq a\leq 1)$ is also included in S for all pairs of $\mathbf{x}_1,\mathbf{x}_2\in$S.

Having proved the corollary, we now have all the tools to prove Minkowski's theorem:

Proof of (a), Theorem 1.2.
Let $\mathbb{B}_R^n = \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x}|| \leq R\} \subset \mathbb{R}^n$ be an n-dimensional ball of radius R.
It can be proven that $vol(\mathbb{B}_R^n) = vol(\mathbb{B}_1^n) \cdot R^n$, where $\mathbb{B}_1^n$ is the n-dimensional unit ball.
Moreover, if n is reasonably large, the volume of $\mathbb{B}_1^n$ is

$$vol(\mathbb{B}_1^n) = \frac{\pi^{n/2}}{\Gamma(\frac{1}{2}n+1)} \approx \left(\frac{2\pi e}{n}\right)^{n/2} \cdot \frac{1}{\sqrt{\pi n}},$$

where the approximation is computed similarly to the one for γ_n before.
Thus, if we take $R \approx \sqrt{2n/\pi e} \cdot \det(\mathcal{L})^{1/n}$, then we have $vol(\mathbb{B}_R^n) \gtrsim 2^n \det(\mathcal{L})$ and can therefore use Minkowski's convex body theorem to prove that $\mathbb{B}_R^n$ contains a nonzero lattice point.

Proof of (b), Theorem 1.2.
This can be proven inductively using (a), as mentioned in [Sil20a].

1.1.7 Heuristics and Assumptions: from Theory to Practice

Gaussian Heuristic (GH). In lattice cryptography, one should keep in mind that the lattices used typically are full rank, high-dimensional and with integer values. For these lattices, we are interested in solving the previously mentioned challenging problems. Therefore, it is valuable to obtain estimates that provide insights into the problems' complexity within these lattice structures. For instance, in the case of SVP, the previous results give us an upper bound on the length of the shortest nonzero vector (through the first part of Minkowski's theorem). However, practical estimations supported by experimental data would be even more valuable for real-life applications, as they provide a better estimate of the security of systems based on these problems.

That's where the Gaussian Heuristic comes in, which is a well-known heuristic that works "reasonably well" in practice for these random integer lattices (though it is not universally applicable). While one might try to rigorously justify it through probability theory, such details are beyond the scope of this thesis. Instead, we only present the "heuristic proof" of [Sil20b] in an effort to explain the concept.

In general, as mentioned in [AD21], the *Gaussian Heuristic* predicts that the number $\mathcal{L} \cap S$ of lattice points inside a measurable body $S \subset \mathbb{R}^n$ is approximately equal to $vol(S)/vol(\mathcal{L})$. When applied to Euclidean n-dimensional balls it leads to the following predictions for SVP and CVP:

Definition 1.12. (Gaussian Heuristic)
Let $\mathcal{L}(\mathbf{B}) \subset \mathbb{R}^n$ be a random lattice, with n sufficiently large.
SVP: We expect that the smallest nonzero vector in $\mathcal{L}$ satisfies

$$\lambda_1(\mathcal{L}) = \min_{\mathbf{u} \in \mathcal{L}/\{\mathbf{0}\}} ||\mathbf{u}|| \approx \sqrt{\frac{n}{2\pi e}} \det(\mathcal{L})^{1/n}$$

CVP: Let $\mathbf{t} \in \mathbb{R}^n$ be a random target point. Then we expect

$$\min_{\mathbf{u} \in \mathcal{L}} ||\mathbf{u} - \mathbf{t}|| \approx \sqrt{\frac{n}{2\pi e}} \det(\mathcal{L})^{1/n}$$

Heuristic "Proof". The proofs of each prediction are similar, so we explain only the CVP one.

Taking into account Blichfeldt's theorem and Minkowski's convex body theorem, we expect that if we take a random, symmetric and convex region in $\mathbb{R}^n$, whose volume significantly exceeds $\det(\mathcal{L})$, then that region is likely to contain a point of $\mathcal{L}$. However, if its volume is significantly smaller than $\det(\mathcal{L})$, then it probably won't contain a point of $\mathcal{L}$.

Let now $\mathbb{B}_R^n(t) = \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x} - \mathbf{t}|| \leq R\}$ be a ball of radius R centered at $\mathbf{t}$. This ball can be our region and by performing a computation similar to the one for the proof of Minkowski's theorem, we get:

$$vol(\mathbb{B}_R^n(t)) \approx \det(\mathcal{L}) \iff R \approx \sqrt{\frac{n}{2\pi e}}\,\det(\mathcal{L})^{1/n}$$

which makes us expect that if R is larger than this value, then at least one point is contained inside $\mathcal{L} \cap \mathbb{B}_R^n(t)$ and if R is smaller, none are contained in the intersection. Therefore, the solution to the CVP problem of finding $\min_{\mathbf{u}\in\mathcal{L}} ||\mathbf{u}-\mathbf{t}||$ is likely to be roughly equal to $\sqrt{n/2\pi e}\cdot\det(\mathcal{L})^{1/n}$.

Remark 1.5. We note that the upper bound $\sqrt{n/\pi e}\,\det(\mathcal{L})^{1/n}$ we get by using Blichfeldt's refined bound for γ_n, and is true for all lattices, is only $\sqrt{2}\approx 1.4$ times larger than the expected length of $\lambda_1(\mathcal{L})$ for a lattice due to GH, $\sqrt{n/2\pi e}\,\det(\mathcal{L})^{1/n}$, which is what we anticipate for most lattices (but not all).

Hermite Factors. Apart from GH, there is another essential parameter that has proven to be useful in practice. It is a parameter associated with the basis matrix $\mathbf{B}$ of an n-dimensional lattice $\mathcal{L}$ and this is one of the reasons why it can be used implicitly as an efficiency indicator of lattice basis reduction algorithms, which aim to provide an improved lattice basis.

Definition 1.13. Let $\mathcal{L}(\mathbf{B})$ be a lattice of rank n with basis matrix $\mathbf{B}$. Then, we define the *Hermite Factor (HF)* of the lattice $\mathcal{L}$ for matrix $\mathbf{B}$ as

$$\mathrm{HF}(\mathcal{L},\mathbf{B}) = \frac{||\mathbf{b}_1||}{\det(\mathcal{L})^{1/n}}.$$

In fact, since $\mathbf{b}_1\mathbf{b}_1^*$ and $\det(\mathcal{L}) = \prod_{i=1}^n ||\mathbf{b}_i^*||$, HF only depends on $\mathbf{B}^*$ and thus we can write $\mathrm{HF}(\mathcal{L},\mathbf{B}) = \mathrm{HF}(\mathbf{B}^*)$, where $\mathbf{B}^*$ is the Gram-Schimdt basis of $\mathbf{B}$.

Remark 1.6. According to [Δρα22], Hermite proved on 1850 that the Hermite Factor of a lattice $\mathcal{L}$ with matrix $\mathbf{B}$ is a function of n, thus the quantity $\lambda_1((L)/\det(\mathcal{L})^{1/n}$ is also only depended on n, and therefore $\gamma_n \doteqdot \sup_{\mathcal{L}\subset\mathbb{R}^n}\left\{\lambda_1(\mathcal{L})^2/\det(\mathcal{L})^{2/n}\right\}$ is also only depended on n.

Next, we briefly mention how γ_n and HF relate to the basis reduction algorithms:

Proposition 1.5. (Hermite's Inequality)
For $n > 2$:

$$\gamma_n < \left(\frac{4}{3}\right)^{(n-1)/2}$$

Proof. An overview of the proof can be found in [Δρα22] and its references.

The LLL (Lenstra-Lenstra-Lovász) lattice basis reduction algorithm that we examine in the next section represents an algorithmic version of the previous proposition. Likewise the next inequality (which is a generalization of Hermite's) is translated into a practical, algorithmic approach by the BKZ (Block Korkin-Zolotarev) algorithm, a refinement of the LLL algorithm.

Proposition 1.6. (Mordell's Inequality)
For $2 \le k < n$:

$$\sqrt{\gamma_n} < \sqrt{\gamma_k}^{(n-1)/(k-1)}$$

Proof. For more information the interested reader is referred to [Ngu10] and its references.

Finally, we highlight the indirect manner with which the Hermite Factor is connected to the estimation of the algorithmic efficiency of lattice reduction algorithms. This is accomplished through a value called Root Hermite Factor, which has become the standard tool for measuring the output quality of a lattice reduction algorithm.

Definition 1.14. Let $\mathcal{L}(\mathbf{B})$ be a lattice of rank n with basis matrix $\mathbf{B}$ and assume $||\mathbf{b}_1||/\det(\mathcal{L})^{1/n} = O(\delta^n)$. Then, we define the *Root Hermite Factor (RHF)* of the lattice $\mathcal{L}$ for matrix $\mathbf{B}$ as the real number δ such that

$$||\mathbf{b}_1|| = \delta^n \det(\mathcal{L})^{1/n}.$$

Thus, $\delta^n = \mathrm{HF}(\mathcal{L},\mathbf{B})$. In addition, RHF is also symbolized as $\mathrm{RHF}(\mathcal{L},\mathbf{B})$.

Remark 1.7. Clearly, $\delta \geq 1$. Moreover, as we expound more in latter chapters, a lattice basis reduction algorithm's efficacy improves as it produces values of δ that draw nearer to 1.

Geometric Series Assumption (GSA). Another heuristic assumption widely used in lattices is the Geometric Series Assumption (GSA), first presented by Schnor and Euchner in [Sch03].

Definition 1.15. We state that the *Geometric Series Assumption (GSA)* is true for a lattice $\mathcal{L}$ with basis matrix $\mathbf{B}$, if

$$\frac{||\mathbf{b}_i^*||}{||\mathbf{b}_1||} = r^{i-1}$$

for some $r \in (0, 1)$. Equivalently, GSA is true if the sequence $\{||\mathbf{b}_1^*||, ||\mathbf{b}_2^*||, \ldots, ||\mathbf{b}_n^*||\}$ is a geometric progression with common ratio $r < 1$.

One notable result is the lemma below, which connects GSA to RHF:

Lemma 1.1. Let $\mathcal{L}$ be a lattice with its basis matrix $\mathbf{B}$ and assume that the Root Hermite Factor δ exists. Then, the Geometric Series Assumption of the basis $\mathbf{B}$ is true with ratio $r \approx \delta^{-2}$.

Proof. The proof is quite simple, see Lemma 14.3.2 of [$\Delta\rho\alpha$22].

Corollary 1.2. Let $\mathcal{L}$ be a lattice with a basis matrix $\mathbf{B}$, assume that the Root Hermite Factor δ exists and that the Geometric Series Assumption is true. Then, for $i \geq 1$, we have

$$||\mathbf{b}_i^*|| \approx \delta^{n-2(i-1)} \cdot \det(\mathcal{L})^{1/n}$$

As we elaborate more in the following chapter, this last corollary is useful for enumeration algorithms, which belong to a particular group of algorithms that solves SVP by systematically enumerating all lattice points in a bounded region of space.

1.1.8 Lattices Used in Cryptography

On the last part of the section, we present a family of lattices that is pivotal to lattice cryptography, the q-ary m-dimensional integer lattices. These lattices possess a notable theoretical property: solving some hard lattice problems over *random instances* of these lattices is, in an asymptotic sense, just as challenging as solving hard problems for *any* lattice. This is the celebrated worst-case to average-case reduction line of research [Ajt96; Reg09b] that marked a significant milestone in the establishment and advancement of lattice-based cryptography. We explore this topic in greater depth in Part II of this thesis.

Definition 1.16. (q-ary Lattice)
Let q be a positive integer. A q-ary lattice $\mathcal{L}$ of dimension m is a lattice satisfying $q\mathbb{Z}^m \subseteq \mathcal{L} \subseteq \mathbb{Z}^m$.

Particularly, we are mostly interested in the following two kinds of q-ary lattices, as they are used in various cryptographic constructions:

Definition 1.17. (Parity check Lattice / Kernel q-ary Lattice)
Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a matrix with coefficients in $\mathbb{Z}_q$. Then, we call a *parity check lattice* the following:

$$\mathcal{L}_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{x} \equiv \mathbf{0} \bmod q\}.$$

Furthermore, another way to view the above set is as the kernel of a mapping from $\mathbb{Z}_q^n$ to $\mathbb{Z}_q$.

Definition 1.18. (Row-generated lattice / Image q-ary Lattice)
Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a matrix with coefficients in $\mathbb{Z}_q$. Then, we call a *row-generated lattice* the following:

$$\mathcal{L}_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \mathbf{y} = \mathbf{A}^T\mathbf{s} \bmod q, \text{ for some } \mathbf{s} \in \mathbb{Z}^n\} = \mathbf{A}^T\mathbb{Z}^n + q\mathbb{Z}^m.$$

Remark 1.8.

(i) For readers with some knowledge of coding theory, we briefly explain the names given above: the "parity check" lattice corresponds to the code whose parity check matrix is $\mathbf{A}$, whereas the "row-generated" lattice corresponds to the code generated by the rows of $\mathbf{A}$.

(ii) The above lattices are dual to each other, up to normalization. More precisely, we have $\mathcal{L}_q^{\perp}(\mathbf{A}) = q\mathcal{L}_q(\mathbf{A})^*$ and $\mathcal{L}_q(\mathbf{A}) = q\mathcal{L}_q^{\perp}(\mathbf{A})^*$.

As we will see in Part II, the lattice $\mathcal{L}_q(\mathbf{A})$ is often used as the foundational lattice in various cryptographic constructions. Thus, it is valuable to ascertain the potential range of the shortest vector within this lattice. We take a few steps on this direction, presenting the initial results below. However, the more complete picture will be given in a latter chapter.

Lemma 1.2. For $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{A}' \in \mathbb{Z}_q^{m\times n}$, we have

(a) $\dim \mathcal{L}_q^{\perp}(\mathbf{A}) = m \quad \& \quad \dim \mathcal{L}_q(\mathbf{A}') = m.$

(b) $\det(\mathcal{L}_q^{\perp}(\mathbf{A})) \leq q^n \quad \& \quad \det(\mathcal{L}_q(\mathbf{A}')) \geq q^{m-n}.$

(c) If q is prime, and $\mathbf{A}$, $\mathbf{A}'$ are invertible in $\mathbb{Z}_q$, the above inequalities are equalities.

Proof: See Lemma 4 in Lecture 9 of [DD18]. Less formally, helpful directions can also be found on crypto-stackexchange "How to prove the inequalities of q-ary lattice determinant?" [link].

Using theorems from the previous subsection and the lemma above, we can calculate some upper bounds for the length of the shortest nonzero vector, and provide an estimation of it according to the Gaussian Heuristic More precisely, as per the lemma, for a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with coefficients in $\mathbb{Z}_q$, the lattice has dimension m and $\det(\mathcal{L}_q^{\perp}(\mathbf{A})) \leq q^n$, thus getting:

(a) According to the upper bound from Minkowski's theorem: $\lambda_1(\mathcal{L}_q^{\perp}(\mathbf{A})) \leq \sqrt{\frac{2m}{\pi e}} \cdot q^{n/m}$

(b) According to the refined upper bound from Blichfeldt: $\lambda_1(\mathcal{L}_q^{\perp}(\mathbf{A})) \leq \sqrt{\frac{m}{\pi e}} \cdot q^{n/m}$

(c) According to the Gaussian Heuristic estimate: $\lambda_1(\mathcal{L}_q^{\perp}(\mathbf{A})) \approx \sqrt{\frac{m}{2\pi e}} \cdot q^{n/m}$.

1.2 Lattice Basis Reduction

As we mentioned earlier, the aim of lattice basis reduction is to take a lattice basis and transform it into a "nice" one, i.e. one that contains vectors that are short and close to orthogonal. To this end, the Lagrange and LLL algorithms are discussed, along with some useful results on their efficiency.

1.2.1 Lagrange Algorithm

We begin by delving into the so-called "Gauss-Lagrange" algorithm, for which more details can be found in [Δρα22; Gal18] and [Ngu10]. Starting now, we will call it *Lagrange's algorithm* as, according to [Ngu10], it is incorrectly attributed to Gauss [Gau01] though it was first stated by Lagrange in [Lag73]. Lagrange's algorithm is a lattice basis reduction algorithm for two-dimensional lattices that reduces the basis to the "optimal" shortest vectors for the lattice.

Let $\mathcal{L}$ be a lattice and $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2]$ be its basis matrix. We remind that due to $\mathbf{B}$ being a basis matrix, $\mathbf{b}_1, \mathbf{b}_2$ are linearly independent (this will be usefu in proofs). In the following we give a first definition for a "Lagrance reduced" basis, and later present a simpler, equivalent one.

Definition 1.19. An ordered basis $\mathbf{b}_1, \mathbf{b}_2$ of $\mathcal{L}$ is called *Lagrange reduced* (also called an L-basis) if and only if, for every $q \in \mathbb{Z}$,

$$||\mathbf{b}_1|| \leq ||\mathbf{b}_2|| \leq ||\mathbf{b}_2 + q\mathbf{b}_1||.$$

Theorem 1.4. Let λ_1, λ_2 be the successive minima of $\mathcal{L}$. If $\mathcal{L}$ has an ordered basis $\{\mathbf{b}_1, \mathbf{b}_2\}$ that is Lagrange reduced, then $||\mathbf{b}_i|| = \lambda_i$ for $i = 1, 2$.

Proof. See Theorem 17.1.2 of [Gal18], and Theorem 14.4.1 of [Δρα22].

Hence, this theorem establishes that a Lagrange reduced basis consists of vectors of minimal length. Furthermore, we note that this result holds true for any norm, although the exact algorithm we introduce to obtain a Lagrange reduced basis only works for the Euclidean norm. Let's now prove the equivalent simplified version of a Lagrange reduced basis:

Lemma 1.3. An ordered basis $\mathbf{b}_1, \mathbf{b}_2$ of $\mathcal{L}$ is *Lagrange reduced* if and only if

$$||\mathbf{b}_1|| \leq ||\mathbf{b}_2|| \leq ||\mathbf{b}_2 - \mathbf{b}_1||, ||\mathbf{b}_2 + \mathbf{b}_1||.$$

Proof. A detailed proof can be found in Lemma 14.4.2 and Corollary 14.4.2 of [Δρα22].
As a slight sketch of the proof we mention one key fact needed for the converse direction (as the forward is trivial): setting $F(\mu) = ||\mathbf{b}_2 \pm \mu\mathbf{b}_1||^2$, the graph of this function is a parabola, having a minimum for $-1 < \mu < 1$. Thus, $||\mathbf{b}_2 \pm \mathbf{b}_1|| \leq ||\mathbf{b}_2 + q\mathbf{b}_1||$ for $q \in \mathbb{Z}\backslash\{-1, +1\}$.

□

Before demonstrating an algorithm for transforming a basis to a Lagrange reduced one, we highlight an interesting geometric property inherent to Lagrange reduced bases:

Lemma 1.4. Let $\{\mathbf{b}_1, \mathbf{b}_2\}$ be an ordered basis of $\mathcal{L}$ such that $||\mathbf{b}_1|| \leq ||\mathbf{b}_2|| \leq ||\mathbf{b}_2 - \mathbf{b}_1||, ||\mathbf{b}_2 + \mathbf{b}_1||$ (or equivalently let $\{\mathbf{b}_1, \mathbf{b}_2\}$ be Lagrange reduced). Then,

(a) If θ is the angle of the vectors $\mathbf{b}_1, \mathbf{b}_2$, then

$$|\cos(\theta)| \leq \frac{||\mathbf{b}_2||}{2||\mathbf{b}_1||}$$

(b) It is true that

$$\frac{|\langle \mathbf{b}_1, \mathbf{b}_2 \rangle|}{||\mathbf{b}_1||^2} \leq \frac{1}{2}$$

Proof. A detailed proof can be found in Lemma 14.4.1 of [Δρα22].

Remark 1.9. From (a) we get that $\theta \in [\pi/3, 2\pi/3]$ due to

$$|\cos(\theta)| \leq \frac{||\mathbf{b}_2||}{2||\mathbf{b}_1||} \leq \frac{||\mathbf{b}_1||}{2||\mathbf{b}_1||} \leq \frac{1}{2}.$$

Therefore, the angle between two vectors of a Lagrange reduced basis is between 60° and 90°.

Sketch of the Lagrange Algorithm. As mentioned earlier in the proof of Lemma 1.3, the function $F(\mu) = ||\mathbf{b}_2 - \mu\mathbf{b}_1||^2 = B_2 - 2\mu\langle \mathbf{b}_1, \mathbf{b}_2 \rangle + \mu^2 B_1$, where $B_1 = ||\mathbf{b}_1||^2$ and $B_2 = ||\mathbf{b}_2||^2$, is a parabola. To find its minimum, we differentiate with respect to μ, and find that $F(\mu)$ is minimized at $\mu = \langle \mathbf{b}_1, \mathbf{b}_2 \rangle / B_1$.

Keeping that in mind, and having the algorithm presented alongside, we now explain its workings. Our goal is, starting from a basis $\{\mathbf{b}_1, \mathbf{b}_2\}$ to transform them into a Lagrange reduced basis $\{\mathbf{b}'_1, \mathbf{b}'_2\}$ that satisfies the inequality:

$$||\mathbf{b}'_1|| \leq ||\mathbf{b}'_2|| \leq ||\mathbf{b}'_2 \pm \mathbf{b}'_1||.$$

This process can be explained in two parts, which we then we continue iterating until reaching termination. Note also that the notation $\mathbf{b}_1^{(\cdot)}$ and $\mathbf{b}_2^{(\cdot)}$ is used to track the evolving basis throughout the iterative process until it terminates.

*** Lagrange Algorithm ***

(Input) A basis $\{\mathbf{b}_1, \mathbf{b}_2\}$.
(Output) A Lagrange reduced basis.

Step 1. Compute $B_1 = ||\mathbf{b}_1||^2$ and $\mu = \langle \mathbf{b}_1, \mathbf{b}_2 \rangle / B_1$.
Step 2. Compute $\mathbf{b}_2 \leftarrow \mathbf{b}_2 - \lfloor \mu \rceil \mathbf{b}_1$ and $B_2 = ||\mathbf{b}_2||^2$.
Step 3. While $B_2 < B_1$, do:
- Swap $\mathbf{b}_1$ and $\mathbf{b}_2$ and set $B_1 \leftarrow B_2$.
- Compute $\mu \leftarrow \langle \mathbf{b}_1, \mathbf{b}_2 \rangle / B_1$.
- Compute $\mathbf{b}_2 \leftarrow \mathbf{b}_2 - \lfloor \mu \rceil \mathbf{b}_1$ and $B_2 \leftarrow ||\mathbf{b}_2||^2$.

Step 4. Return $\{\mathbf{b}_1, \mathbf{b}_2\}$.

First, we compute $\mathbf{b}_2^{(1)} \leftarrow \mathbf{b}_2 - \lfloor \mu^{(1)} \rceil \mathbf{b}_1$ and $B_2^{(1)} \leftarrow ||\mathbf{b}_2^{(1)}||^2$. This can be considered a first "size reduction" step for the algorithm.

· **Inequality Check**: We now compare the squared lengths of $\mathbf{b}_1$ and $\mathbf{b}_2^{(1)}$, which are $B_1 = ||\mathbf{b}_1||^2$ and $B_2^{(1)}$, respectively. If $B_2^{(1)}$ is smaller, we swap them. This gives us a new basis, $\{\mathbf{b}_1^{(2)}, \mathbf{b}_2^{(2)}\}$ which satisfies the first part of the inequality.
· **Size Reduction**: To satisfy the second part of the inequality, we have to reduce $\mathbf{b}_2^{(2)}$ using $\mathbf{b}_1^{(2)}$ as much as we can, and therefore our goal is to find $x \in \mathbb{Z}$ such that $||\mathbf{b}_2^{(2)} - x\mathbf{b}_1^{(2)}||$ is minimized. Our previous insight tells us that the minimum in $\mathbb{R}$ occurs at $\mu^{(2)} = \langle \mathbf{b}_1^{(2)}, \mathbf{b}_2^{(2)} \rangle / B_1^{(2)}$. To keep our new vector within the lattice, we select $x = \lfloor \mu^{(2)} \rceil$.[10] Thus, our new $\mathbf{b}_2^{(3)}$ is $\mathbf{b}_2^{(2)} - \lfloor \mu^{(2)} \rceil \mathbf{b}_1^{(2)}$.
· **Iterative Process**: We check if $B_2^{(3)} < B_1^{(2)}$ for the new basis $\{\mathbf{b}_1^{(2)}, \mathbf{b}_2^{(3)}\}$. If true, we swap the vectors and repeat the process. This continues until the inequality is achieved, resulting in an L-basis, $\{\mathbf{b}_1', \mathbf{b}_2'\}$.

- **Termination**:
(a) In cases where the while loop in the algorithm doesn't initiate,we end up with two vectors $\mathbf{b}_1$ and $\mathbf{b}_2' = \mathbf{b}_2 - \lfloor \mu \rceil \mathbf{b}_1$. Therefore, this basis is Lagrange reduced as we have $||\mathbf{b}_1|| \leq ||\mathbf{b}_2'|| \leq ||\mathbf{b}_2' \pm q\mathbf{b}_1||$ for every $q \in \mathbb{Z}$ due to $\lfloor \mu \rceil$ minimizing the length function (or due to (b) from Lemma 1.4, providing $\lfloor \mu \rceil = 0$ if the basis is already reduced).
(b) On the other hand, if the while loop in the algorithm is executed, the process always leads to a Lagrange reduced basis because lattices are discrete, and the initial vectors are within a circle defined by the maximum of their lengths. With each iteration, this circle shrinks, ensuring that the process concludes naturally after a finite number of steps.

For an even more detailed analysis, the interested reader is referred to [Δρα22].

Remark 1.10. We note that, if B such that $||\mathbf{b}_1||^2, ||\mathbf{b}_2||^2 \leq B$ then, the Lagrange algorithm has complexity $O(\log(B)^2)$ (see Theorem 17.1.10 of [Gal18], along with the remarks after it). Moreover, for further details on generalizing the algorithm to $n > 2$ dimensions and adapting it for different norms, the interested reader is referred again to [Gal18] and its references in the end of Section 17.1.

Example 1.4. Set $\mathbf{b}_1 = (3.1, 1.2)$ and $\mathbf{b}_2 = (1.3, 3.9)$. We would like to compute the Lagrange reduced basis of $\{\mathbf{b}_1, \mathbf{b}_2\}$, following the steps of the algorithm above. Additionally, an illustration of the these two basis can be found below.

Step 1. We have $B_1 = ||\mathbf{b}_1||^2 = 11.05$ and $\mu^{(1)} = \langle \mathbf{b}_1, \mathbf{b}_2 \rangle / B_1 \approx 0.79$, with $\lfloor \mu^{(1)} \rceil = 1$.
Step 2. We compute $\mathbf{b}_2^{(1)} = \mathbf{b}_2 - \lfloor \mu^{(1)} \rceil \mathbf{b}_1 = (1.8, -2.7)$ and thus $B_2^{(1)} = ||\mathbf{b}_2^{(1)}||^2 \approx 10.53$.
Step 3. We have $B_2^{(1)} < B_1$, thus we initiate the while loop:
- We swap $\mathbf{b}_1$ and $\mathbf{b}_2^{(1)}$. Thus $\mathbf{b}_1^{(2)} = (1.8, -2.7)$ with $B_1^{(2)} = 10.53$ and $\mathbf{b}_2^{(2)} = (3.1, 1.2)$.
- We compute $\mu^{(2)} = \langle \mathbf{b}_1^{(2)}, \mathbf{b}_2^{(2)} \rangle / B_1^{(2)} \approx 0.22$ with $\lfloor \mu^{(2)} \rceil = 0$.
- We get $\mathbf{b}_2^{(3)} = \mathbf{b}_2^{(2)} - \lfloor \mu^{(1)} \rceil \mathbf{b}_1^{(2)} = (3.1, 1.2)$ and thus $B_2^{(3)} = ||\mathbf{b}_2^{(3)}||^2 \approx 11.05$.

Now we check if $B_2^{(3)} < B_1^{(2)}$ and, as it is false, the while loop ends.
Step 4. The algorithm returns the Lagrange reduced basis

$$\{\mathbf{b}_1' = \mathbf{b}_1^{(2)} = (1.8, -2.7), \mathbf{b}_2' = \mathbf{b}_2^{(3)} = (3.1, 1.2)\}.$$

Note that the angle between the vectors $\mathbf{b}_1', \mathbf{b}_2'$ is $\theta \approx 77.47^o$, because

$$cos(\theta) = \langle \mathbf{b}_1', \mathbf{b}_2' \rangle / ||\mathbf{b}_1'|| \cdot ||\mathbf{b}_2'|| \approx 0.217.$$

[10] In [Δρα22] this is chosen to be $\lceil \mu \rfloor$, but we decided to follow the choice of [Gal18].

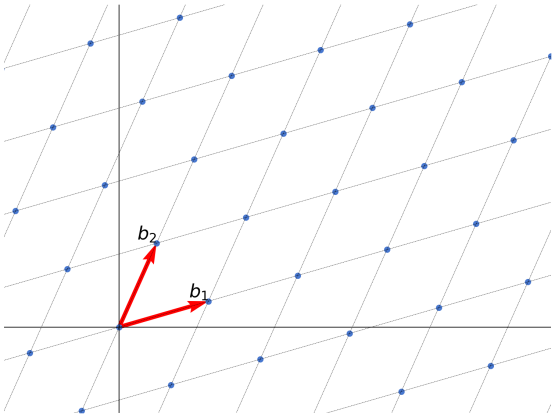


Figure 1.8: Lattice from the starting basis $\{\mathbf{b}_1 = (3.1, 1.2), \mathbf{b}_2 = (1.3, 3.9)\}$.

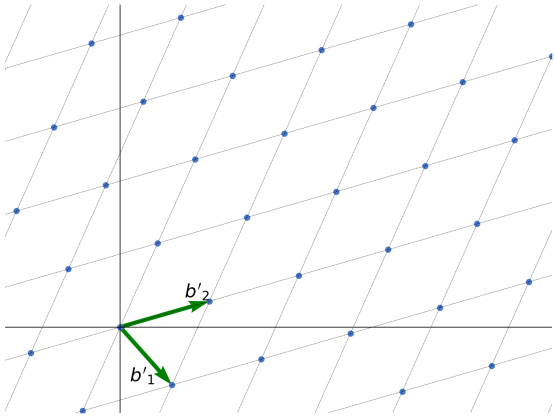


Figure 1.9: Same lattice and the L-basis $\{\mathbf{b}'_1 = (1.8, -2.7), \mathbf{b}'_2 = (3.1, 1.2)\}$.

1.2.2 LLL Algorithm

for n-dimensional lattices with $n > 2$ there are a number of ways that one can perform basis reduction. For instance, one could try to directly generalize the Lagrange algorithm from before. In this section, however, we present an algorithm that follows a slightly different, but definitely successful approach to solving the problem. It is called the *LLL algorithm* (from the names of its creators Lenstra, Lenstra & Lovász) and it is essential for a lot of applications, though we shall only use it as a means to reduce the basis of a lattice. More information on the algorithm and its different uses can be found in [LLL82] and [NV09].

Additionally, as the LLL algorithm exploits the Gram-Schmidt (G-S) orthogonalisation[11], the reader is encouraged to recall it (perhaps through reading Section 17.3 of [Gal18] and Section 14.2 of [$\Delta\rho\alpha$22]). In this thesis we only mention the G-S algorithm (and a related result) as it is used as a subroutine in the LLL algorithm, and do not provide further explanations.

*** Gram-Schmidt Algorithm ***

(Input) A basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \in \mathbb{R}^n$.
(Output) A G-S basis $\{\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*\} \in \mathbb{R}^n$.

Step 1. Set $\mathbf{b}_1^* = \mathbf{b}_1$.
Step 2. For $i = 2$ to n, do:
  (2a) Set $\mathbf{u} \leftarrow \mathbf{b}_i$.
  (2b) For $j = i - 1$ down to 1, do:
    - Compute $\mu_{i,j} \leftarrow \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle$.
    - Set $\mathbf{u} \leftarrow \mathbf{u} - \mu_{i,j} \mathbf{b}_j^*$.
  (2c) Set $\mathbf{b}_i^* = \mathbf{u}$.
Step 3. Return $\{\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*\}$.

The following lemma showcases some interesting properties of G-S orthogonalized vectors, the third of them acting as a small segue to the definition of a LLL-reduced basis right after.

Lemma 1.5. Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be a linearly independent basis in $\mathbb{R}^m$ and let $\{\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*\}$ be its Gram-Schmidt orthogonalisation. Then,

(a) $||\mathbf{b}_i^*|| \leq ||\mathbf{b}_i||$ for $1 \leq i \leq n$

(b) $\langle \mathbf{b}_i, \mathbf{b}_i^* \rangle = \langle \mathbf{b}_i^*, \mathbf{b}_i^* \rangle$ for $1 \leq i \leq n$

[11] Recall that if $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is a set of vectors in $\mathbb{R}^n$, its Gram-Schmidt orthogonalisation is $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$.

(c) Let $j, k \in \mathbb{N}$ be such that $1 < k \leq n$ and $1 \leq j < k$. If $\mathbf{b}'_k = \mathbf{b}_k - \lfloor \mu_{k,j} \rceil \mathbf{b}_j$ and $\mu'_{k,j} = \langle \mathbf{b}'_k, \mathbf{b}^*_j \rangle / \langle \mathbf{b}^*_j, \mathbf{b}^*_j \rangle$, then $|\mu'_{k,j}| \leq 1/2$.

Proof. See Lemma 17.2.2 of [Gal18].

Definition 1.20. Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be an ordered basis for a lattice, denote by $\{\mathbf{b}^*_1, \ldots, \mathbf{b}^*_n\}$ its G-S orthogonalisation, write $B^*_i = ||\mathbf{b}^*_i||^2 = \langle \mathbf{b}^*_i, \mathbf{b}^*_i \rangle$ and let $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}^*_j \rangle / \langle \mathbf{b}^*_j, \mathbf{b}^*_j \rangle$ for $1 \leq j < i \leq n$ be the coefficients from the Gram-Schmidt process. Fix $1/4 < \delta < 1$.[12] Then, the ordered basis is *LLL reduced* (with factor δ) if the following conditions hold:

1. **(Size reduced)** $|\mu_{i,j}| \leq 1/2$ for $1 \leq j < i \leq n$
2. **(Lovász condition)** $B^*_i \geq (\delta - \mu^2_{i,i-1}) B^*_{i-1}$ for $2 \leq i \leq n$

Remark 1.11. We note that traditionally $\delta = 3/4$ is chosen for the Lovász condition and we adhere to this choice throughout this chapter. Nevertheless, one can find lemmas and theorems for a different choice of δ in Chapter 17 of [Gal18].

We continue by showing that an LLL reduced basis has some important "good" properties, which directly connect the LLL algorithm that creates them to solving SVP_γ. Subsequently, we delve deeper into the algorithm itself and its complexity.

Theorem 1.5. Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be an LLL reduced basis with $\delta = 3/4$ for a lattice $\mathcal{L} \subset \mathbb{R}^m$. Then,

(a) $||\mathbf{b}_1|| \leq 2^{(n-1)/2} \lambda_1$

(b) $||\mathbf{b}_j|| \leq 2^{(n-1)/2} \lambda_i$ for $1 \leq j \leq n$ (usually used with i fixed and varying j)

(c) $2^{(1-i)/2} \lambda_i \leq ||\mathbf{b}_i|| \leq 2^{(n-1)/2} \lambda_i$

(d) $\det(\mathcal{L}) \leq \prod_{i=1}^n ||\mathbf{b}_i|| \leq 2^{n(n-1)/4} \det(\mathcal{L})$

(e) $||\mathbf{b}_1|| \leq 2^{(n-1)/4} \det(\mathcal{L})$

Proof. See Theorem 17.2.12 of [Gal18].

Remark 1.12.

(a) The theorem only provides conservative upper bounds on the lengths of $\mathbf{b}_i$ in an LLL-reduced basis. In the majority of practical experiments, LLL-reduced basis vectors were found to be much shorter than these bounds, a topic we will delve more into later.

(b) As a corollary of the theorem, we have that an LLL reduced basis is a solution for SVP_γ with $\gamma = 2^{O(n)}$. Afterwards, we show that this solution can be found in polynomial time. In a similar fashion, one can also show that the set of vectors in an LLL reduced basis are a solution to SIVP_γ for similar exponential values of γ, in polynomial time.

We now present the LLL algorithm, whose goal is to transform a given basis into a LLL-reduced one. We underscore that, although the LLL algorithm can work for any basis in $\mathbb{R}^m$, precise complexity is only provided for $\mathbb{Z}^m$ in the original paper [LLL82], as well as in [Gal18]. Thus, we too chose our input basis to be in $\mathbb{Z}^m$ for the algorithm below.

The LLL algorithm alternates two steps, aimed at achieving the two properties of an LLL reduced basis: After conducting size reduction on the input basis $\mathbf{B}$, the only circumstance in which $\mathbf{B}$ would no longer qualify as LLL reduced is if it violates the Lovász condition. In such a case, the algorithm swaps b_i and b_{i+1}.[13] Following the swap, the basis might no longer be size-reduced, necessitating the repetition of the whole process, starting from the reduction step.

[12] The LLL factor δ should not be confused with the δ for RHF.

[13] Several pairs might violate the second property. Which one is selected for the swapping does not matter.

*** LLL Algorithm ***

(Input) Basis $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^m$

(Output) LLL-reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$

Step 1. Compute the G-S basis $\{\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*\}$ and coefficients $\mu_{i,j}$ for $1 \leq j < i \leq n$

Step 2. Compute $B_i^* = \langle \mathbf{b}_i^*, \mathbf{b}_i^* \rangle = ||\mathbf{b}_i^*||^2$ for $1 \leq i \leq n$

Step 3. Set $k = 2$. While $k \leq n$, do:

(3a) Perform Size Reduction of the basis, i.e. for $j = k-1$ down to 1, do:
- Let $q_j = \lfloor \mu_{k,j} \rceil$ and set $\mathbf{b}_k \leftarrow \mathbf{b}_k - q_j \mathbf{b}_j$
- Update the values $\mu_{k,j}$ for $1 \leq j < k$

(3b) If the Lovász condition is satisfied, i.e. if $B_k^* \geq (\delta - \mu_{k,k-1}^2) B_{k-1}^*$, then set $k \leftarrow k+1$

(3c) If it is not, then
- Swap $\mathbf{b}_k$ with $\mathbf{b}_{k-1}$
- Update the values $\mathbf{b}_k^*, \mathbf{b}_{k-1} j, B_k^*, B_{k-1}^*, \mu_{k-1,j}, \mu_{k,j}$ for $1 \leq j < k$
- Update the values $\mu_{i,k}, \mu_{i,k-1}$ for $k < i \leq n$
- Set $k \leftarrow \max\{2, k-1\}$

Step 4. Return the LLL-reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$, for the new $\mathbf{b}_i$ vectors

It is evident that, upon termination, the basis is LLL-reduced because it is size-reduced and no pairs need to be swapped. Hence, if the algorithm terminates, then it is correct. Moreover, it can be proven that the algorithm terminates[14] and works in polynomial time, but as the details of these are beyond the scope of this thesis, the interested reader is referred to Section 17.5 of [Gal18] and Lecture 3 of [Mic12] for more. For our purposes, we merely state the end result regarding the complexity:

Proposition 1.7. Let $\mathcal{L}$ be a lattice in $\mathbb{Z}^m$ with basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ and let $X \in \mathbb{Z}_{\geq 2}$ be such that $||\mathbf{b}_i||^2 \leq X$ for $1 \leq i \leq n$. Then, the LLL algorithm requires $O(n^3 m \log(X))$ arithmetic operations on integers of size $O(n \log(X))$. Using naive arithmetic gives a running time of $O(n^5 m \log(X)^3)$ bit operations.[15]

No example will be provided directly for this algorithm, as it is quite complex. Nevertheless, for those interested in gaining a deeper understanding, a valuable illustration by Thijs Laarhoven can be found in Tanja Lange's video "Lattice-based cryptography I - Definitions and LLL" [link].

LLL algorithm Vs Lagrange algorithm. It is useful to compare the LLL algorithm with the Lagrange-Gauss reduction algorithm. Although both algorithms share the idea of reducing the length of the basis vectors followed by a swap, they differ significantly in two crucial aspects:

1. **Size Reduction.** The size reduction operation in the Lagrange-Gauss algorithm gives the minimal value for $||\mathbf{b}_2 + q\mathbf{b}_1||$ over $q \in \mathbb{Z}$. In LLL, the coefficient $\mu_{k,j}$ is chosen to depend on $\mathbf{b}_k$ and $\mathbf{b}_j^*$, so it does not necessarily minimize $||\mathbf{b}_k||$. Indeed, $||\mathbf{b}_k||$ can grow during the algorithm's execution. However, it's worth noting that, in the two-dimensional case, the value of $\mu_{2,1}$ coincides with what the Lagrange-Gauss algorithm employs, making the size reduction step identical.
2. **Size Check.** The size check in LLL (the Lovász condition) is on the lengths of the Gram-Schmidt vectors. On the other hand, in the Lagrange-Gauss algorithm, the size check is based on the length of the basis vectors themselves.

[14] Informally, the algorithm terminates in polynomial time because "it makes non-trivial progress at each step".
[15] Since the input size is $O(nm \log(X))$ and $n \leq m$, the running time is cubic in the input size.

At first glance, these features of the LLL algorithm may appear counterintuitive. However, they play a vital role in the algorithm's polynomial-time complexity, which is crucial for its practical applicability and efficiency.

Variants of LLL. Although beyond the scope of this thesis, over the years many refinements of the LLL algorithm were presented, for which the interested reader can read more on Section 17.6 of [Gal18] and its references. Nevertheless, in the next chapter, we shall mention some key facts about one such refinement, namely the block Korkine-Zolotarev (BKZ) algorithm due to Schnorr [Sch87], as it will prove useful for later security assessments of hard lattice problems and thus, the cryptographic schemes that are based on them.

Chapter 2

Algorithms and Complexity of Lattice Problems

In this chapter we delve deeper into the complexity of previously mentioned hard problems, and afterwards examine the capabilities of algorithms attempting to solve them. In the context of cryptography, our primary interest lies in the approximate versions of these problems but not for all values of γ. As demonstrated in the figure below[1], taking SVP_γ as an example (similar results are true for CVP_γ), the closer γ gets to 1, the harder the problem gets, even reaching NP-hard complexity. On the other hand, if we set γ too large, then algorithms like LLL and others manage to solve the problem, in polynomial time.

Cryptography falls somewhere in between, as we unpack in greater detail in the following chapters, where we demonstrate how lattice problems can be a powerful tool for defining security in cryptographic schemes.

Furthermore, we emphasize that the figure below should not be considered "complete" since, with refinements of previous algorithms and better understanding of the problems through research, the situation is updated constantly.

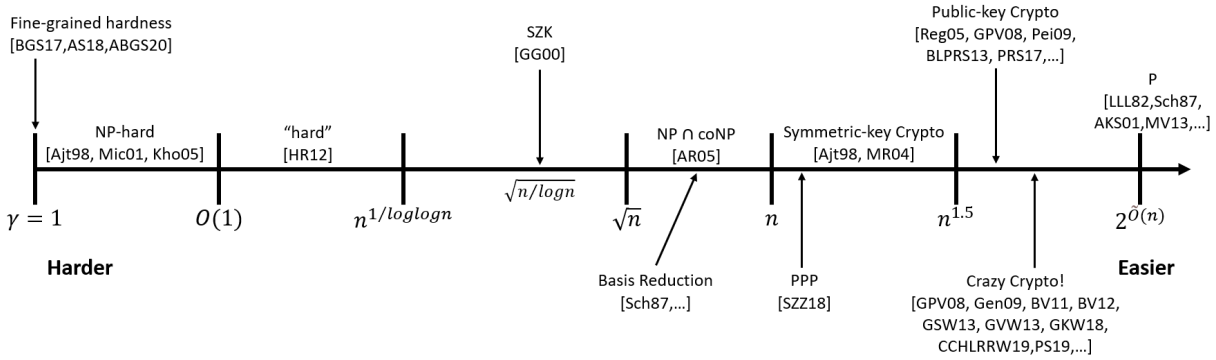


Figure 2.1: Complexity of SVP_γ in regard to γ as a function of n.

Additionally, it is interesting to observe the connections between hard problems in lattices. In fact, most of these problems can be "reduced" to others, sometimes even with the same approximation factor. To be more specific, when we say that a problem P *reduces* to a problem Q, we mean that "if there exists an algorithm for solving Q, then there is also an algorithm for solving P too", indicating that Q is at least as hard as P. This is typically denoted as "$P \leq Q$" and in the figure below we use $P \to Q$ to represent this reduction relationship (also two problems being in the same box means that they are equivalent).

For instance, a reduction can be observed in the case of SVP_γ to CVP_γ, as presented in

[1]The figure was created following Stephens-Davidowitz's presentations in [Ins20].

[GMSS99], where the approximation factor γ remains the same. On the other hand, one can consider how SIVP_γ can be reduced to $\text{SVP}_{\gamma'}$, where the approximation factor changes to $\gamma' = \sqrt{n}\gamma$. These and other reductions are depicted in the figure below, on a diagram that we note is not exhaustive and is fully explained in [Ste]. It actually represents known results until 2016, with some omissions, and significant progress has been made in this field since then.

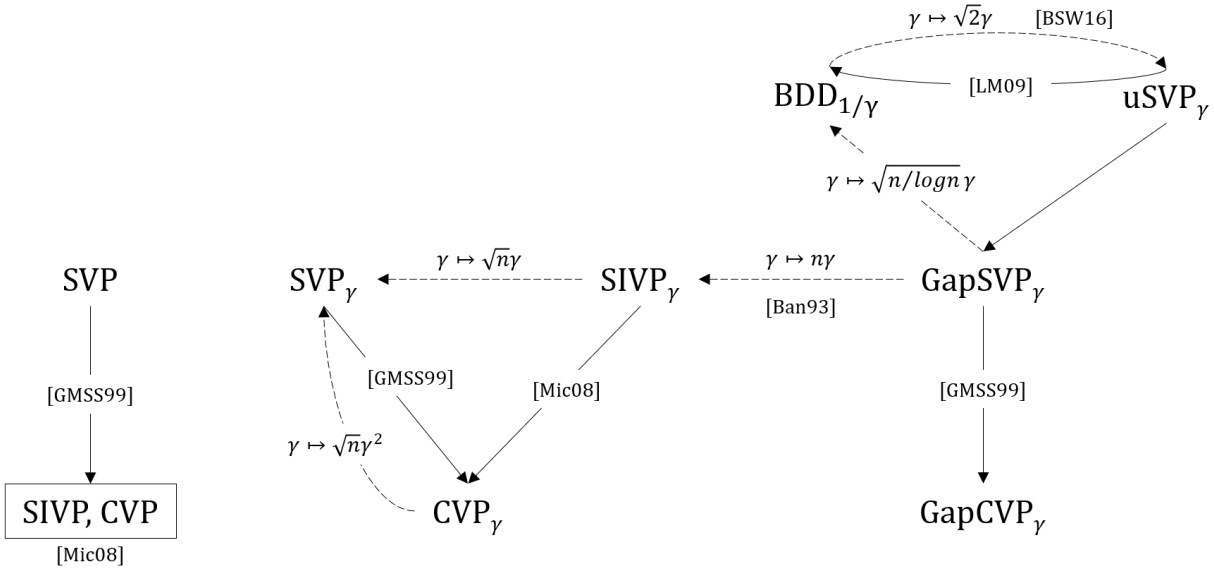


Figure 2.2: Connections between hard lattice problems in regard to γ as a function of n.

With this framework in mind, considering the significance of γ and the connections between hard problems, we now shift our focus to SVP_γ and CVP_γ.

2.1 Solving CVP

We start by presenting two algorithms for finding lattice vectors close to a given point (i.e. solving CVP) . These are *Babai's nearest plane method* and *Babai's rounding technique* (whose ideas will also be useful in latter constructions). Additionally, we remark that there exist alternative methods for solving CVP. As an example, one can read about Kannan's enumeration method, which works in exponential time, in Section 18.3 of [Gal18].

At this point, we remind the definition of CVP_γ (and CVP equals the case where $\gamma = 1$): "Let $\mathcal{L}$ be an n-dimensional lattice and fix a $\gamma(n) = \gamma \geq 1$. Given a basis matrix $\mathbf{B}$ for $\mathcal{L}$ and a point $\mathbf{w} \in \mathbb{R}^n$, compute $\mathbf{u} \in \mathcal{L}$ such that $||\mathbf{w} - \mathbf{u}|| \leq \gamma||\mathbf{w} - \mathbf{y}||$, for all $\mathbf{y} \in \mathcal{L}$."

2.1.1 Babai's Nearest Plane Method

Let $\mathbf{w}$ be a point in $\mathbb{R}^n$ and $\mathcal{L}$ be an n-dimensional lattice with an (ordered) basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ and $\{\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*\}$ its corresponding Gram-Schmidt basis. In [Bab86], Babai presented a method which (attempts) to find a vector of the lattice close to $\mathbf{w}$ (i.e. which (attempts) to solve CVP). However, this method does not guarantee the solution of the problem. For instance, if the lattice basis is LLL-reduced, then it can be proven that the distance of the lattice vector (that Babai's method outputs) from the point $\mathbf{w}$ is within an exponential factor of the minimal value.

Furthermore, in general, Babai's nearest plane algorithm can be expressed in two different forms, while the core of the idea remains the same. The first form is recursive, as demonstrated in Section 14.7 of [Δρα22] and lecture 3 of [Reg09a], and the second form is inductive, as shown in [Gal18]. In this thesis, we also present the second form.

MAIN IDEA. Let $\mathbf{w}$ be our target point in $\mathbb{R}^n$. We begin the process by computing the $l_j \in \mathbb{R}$ such that $\mathbf{w} = \sum_{j=1}^{n} l_j \mathbf{b}_j^*$. Then, we set $U = span(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$ and $\mathcal{L}' = \mathcal{L} \cap U$, which is the sublattice spanned by $\{\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}\}$ (in the figure, the x-axis represents the subspace U (with dimension $n-1$) and the y-axis is perpendicular to U).

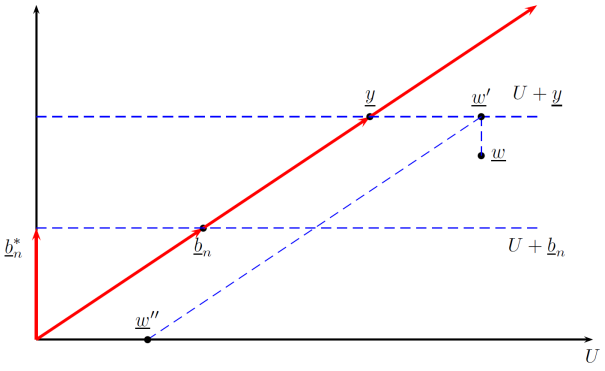


Figure 2.3: Illustration of the Babai nearest plane method (from [Gal18]).

Our goal is to find a vector $\mathbf{y} \in \mathcal{L}$ that minimizes the distance from the point $\mathbf{w}$ to the plane $U + \mathbf{y}$. As proved in Lemma 18.1.1 of [Gal18], this vector is $\mathbf{y} = \lfloor l_n \rceil \mathbf{b}_n$. Then, we set $\mathbf{w}'$ to be the orthogonal projection of $\mathbf{w}$ onto $U + \mathbf{y}$, which again by the lemma is, $\mathbf{w}' = \sum_{j=1}^{n-1} l_j \mathbf{b}_j^* + \lfloor l_n \rceil \mathbf{b}_n^*$. Let now $\mathbf{w}'' = \mathbf{w}' - \mathbf{y} \in U$ and thus

$$\mathbf{w}'' = \sum_{j=1}^{n-1} l_j \mathbf{b}_j^* + \lfloor l_n \rceil \mathbf{b}_n^* - \lfloor l_n \rceil \mathbf{b}_n = \mathbf{w} - (l_n - \lfloor l_n \rceil)\mathbf{b}_n^* - \lfloor l_n \rceil \mathbf{b}_n.$$

One then inductively solves the (lower dimensional) CVP instance of $\mathbf{w}''$ in $\mathcal{L}'$ in order to find a vector $\mathbf{y}' \in \mathcal{L}'$. Then, the solution to the original instance of the CVP is $\mathbf{u} = \mathbf{y} + \mathbf{y}'$.

*** Babai Nearest Plane Algorithm ***

(Input) A basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \in \mathbb{R}^n$ and a point $\mathbf{w} \in \mathbb{R}^n$.
(Output) A vector $\mathbf{u}$.

Step 1. Compute the Gram-Schmidt basis $\{\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*\}$.
Step 2. Set $\mathbf{w}_n = \mathbf{w}$.
Step 3. For $i = n$ down to 1, do:
- Compute $l_i = \langle \mathbf{w}_i, \mathbf{b}_i^* \rangle / \langle \mathbf{b}_i^*, \mathbf{b}_i^* \rangle$.
- Set $\mathbf{y}_i = \lfloor l_i \rceil \mathbf{b}_i$.
- Set $\mathbf{w}_{i-1} = \mathbf{w}_i - (l_i - \lfloor l_i \rceil)\mathbf{b}_i^* - \lfloor l_i \rceil \mathbf{b}_i$.

Step 4. Return $\mathbf{u} = \mathbf{y}_1 + \ldots + \mathbf{y}_n$.

Note that in the algorithm we used the notation $\mathbf{y}_n = \mathbf{y}$, $\mathbf{w}_n = \mathbf{w}$, $\mathbf{w}_{n-1} = \mathbf{w}''$, etc.

Theorem 2.1. Let $\mathbf{w}$ be a point in $\mathbb{R}^n$ and $\mathcal{L}$ be an n-dimensional lattice with a basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ that is LLL-reduced[2], then the output of the Babai nearest plane algorithm on $\mathbf{w}$ is a vector $\mathbf{u} \in \mathcal{L}$ such that $||\mathbf{u} - \mathbf{w}|| < 2^{n/2}||\mathbf{v} - \mathbf{w}||$, for all $\mathbf{v} \in \mathcal{L}$.

Proof. A detailed proof can be found in Theorem 18.1.6 of [Gal18].

Remark 2.1. The vector $\mathbf{u}$ output by the Babai nearest plane method lies in the parallelepiped

$$\left\{ \mathbf{w} + \sum_{j=1}^{n} l_j \mathbf{b}_j^* \ : \ l_j \in \mathbb{R}, \ |l_j| \leq \frac{1}{2} \right\}$$

centered on $\mathbf{w}$. This parallelepiped has a volume equal to the volume of the lattice. Hence, if $\mathbf{w} \in \mathcal{L}$, then there is exactly one lattice point in this parallelepiped.

[2]We remind that, unless stated otherwise, we use only the Euclidean norm and the factor $\delta = 3/4$ for LLL.

2.1.2 Babai's Rounding Technique

This method is also not guaranteed to solve CVP, but we have an approximation result for LLL-reduced bases. There are two ways it can be described, one is through using dual lattices (see Regev's "Cryptanalysis of GGH and NTRU Signatures" of [Uni12] for an informal presentation of this), and the other is the following:

Let $\mathbf{w}$ be a point in $\mathbb{R}^n$ and $\mathcal{L}$ be an n-dimensional lattice with a basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$. We can write $\mathbf{w} = \sum_{i=1}^{n} l_i \mathbf{b}_i$ with $l_i \in \mathbb{R}$, where the coefficients l_i can be computed by solving the system of linear equations (since the lattice is full rank, we can also compute $(l_1, \ldots, l_n)$ as $\mathbf{w}\mathbf{B}^{-1}$).

Then, the rounding technique is simply to set

$$\mathbf{u} = \sum_{i=1}^{n} \lfloor l_i \rceil \mathbf{b}_i.$$

Theorem 2.2. Let $\mathbf{w}$ be a point in $\mathbb{R}^n$ and $\mathcal{L}$ be an n-dimensional lattice with a basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ that is LLL-reduced, then the output of the Babai rounding method on input $\mathbf{w}$ is a vector $\mathbf{u} \in \mathcal{L}$ such that $||\mathbf{u} - \mathbf{w}|| < (1 + 2n(\frac{9}{2})^{n/2})||\mathbf{v} - \mathbf{w}||$, for all $\mathbf{v} \in \mathcal{L}$.

Proof. A detailed proof can be found in Theorem 18.1.6 of [Bab86].

Remark 2.2.

(i) The vector $\mathbf{u}$ output by the Babai rounding technique lies in the parallelepiped

$$\left\{ \mathbf{w} + \sum_{j=1}^{n} l_j \mathbf{b}_j \ : \ l_j \in \mathbb{R}, \ |l_j| \leq \frac{1}{2} \right\}$$

centered on $\mathbf{w}$. This parallelepiped has a volume equal to the volume of the lattice. Hence, if $\mathbf{w} \in \mathcal{L}$, then there is exactly one lattice point in this parallelepiped.

(ii) The quality of the basis significantly influences whether an optimal solution to the CVP is achieved when using rounding, as can be clearly seen in the figure below. This property of the technique is what makes it really important for cryptography too, in different ways. For instance, the Goldreich-Goldwasser-Halev (GGH) cryptosystem employs a "good" basis as a private key for decryption, relying on rounding, while a "bad" basis is made public for encryption. Additional details can be found in Section 19.9 of [Gal18].

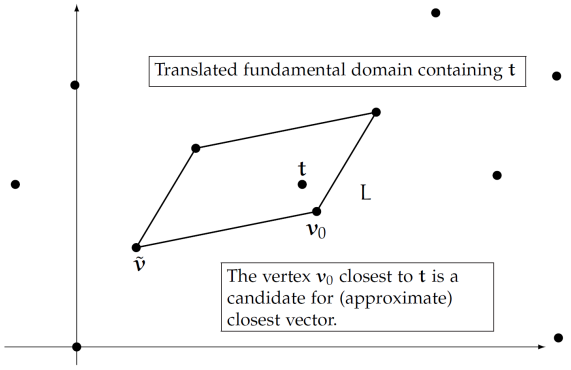


Figure 2.4: Babais's rounding with "good" basis (from [Sil20b]).

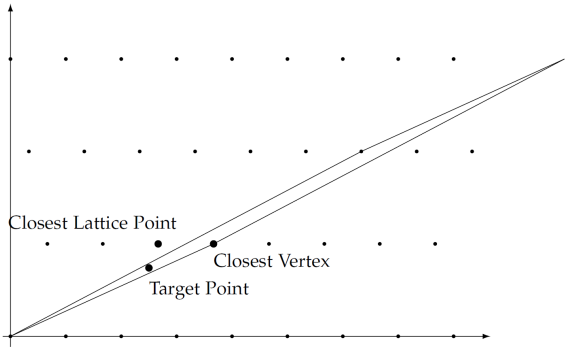


Figure 2.5: Babais's rounding with "bad" basis (from [Sil20b]).

2.2 Solving SVP

In general, algorithms that try to solve, either the exact or the approximate version, of the Shortest Vector Problem (SVP) can be classified in two categories:

- *Exact/Near exact algorithms* (like [AKS01; Kan87]), that provably output a shortest vector.
- *Approximation algorithms* (like [LLL82; Sch87; Gam+06; GN08a]), which output a nonzero lattice vector whose norm is provably not much bigger than that of a shortest vector

Usually exact algorithms are used for lower dimensions as, in higher dimensions ($n > 100$), only approximation algorithms are practical. However, both categories are in fact *complementary* as all known exact algorithms first apply an approximation algorithm (such as LLL) for pre-processing, while all known approximation algorithms make intensive use of an exact algorithm in low dimension. For instance, one of the best known approximation algorithms (as well as its predecessors), that of Gama and Nguyen [GN08a], calls (polynomially many times) an exact algorithm in dimension k. Even the heuristic BKZ algorithm [SE94] (implemented in NTL [Sho] and often used by cryptanalysts) also crucially relies on an exact SVP algorithm in small dimension (typically chosen around 20).

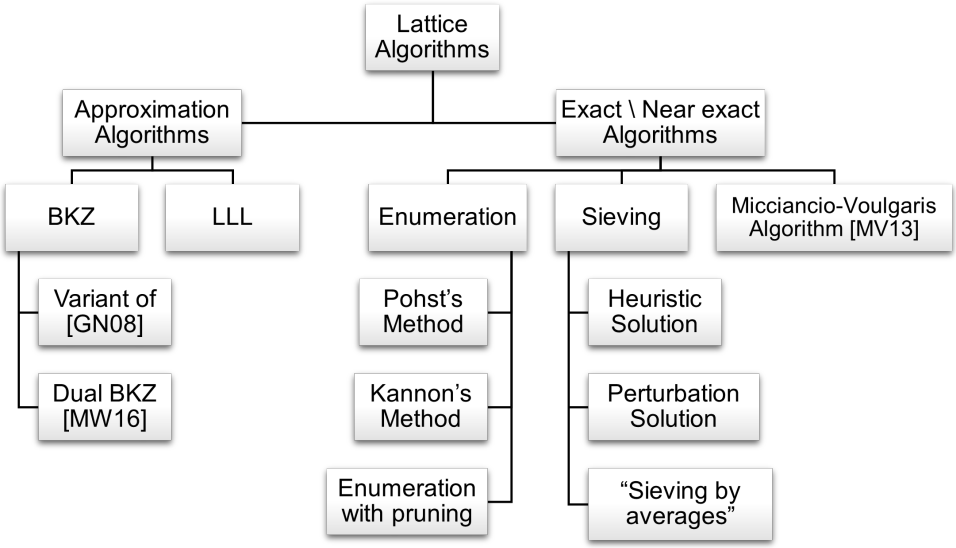


Figure 2.6: (Partial) Classification of lattice algorithms.

As shown in the figure above, in the category of exact algorithms, two main families of algorithms can be found, one based on enumeration and one on sieving, as well as a third standalone algorithm called Micciancio-Voulgaris algorithm [MV13], which uses the geometry of Voronoi cells to deterministically solve SVP. On the other hand, within the domain of approximation algorithms, the landscape is mainly dominated by lattice basis reduction algorithms, like the LLL algorithm and the block Korkine–Zolotarev (BKZ) algorithm.

For our purposes, extensive exploration of the inner workings of these families and algorithms will not be necessary. Instead, we provide a concise overview of the core concepts behind a select few of them. Interested readers can refer to the references from this section for more.

2.2.1 Exact/ Near Exact Algorithms

Enumeration Algorithms. The first enumeration algorithm was presented by Pohst in 1981 [Poh81]. Further refinements and variants were made in the following years with some notable being the Fincke-Pohst algorithm [FP85], Kannan's method [Kan87] and the "Enumeration with pruning" for which more information can be found on Section 14.6 of [Δρα22] and its references.

Lattice enumeration is a standard technique employed to solve SVP (as well as CVP) on arbitrary lattices. It accomplishes this by systematically enumerating all lattice points in a bounded region of space, typically defined as an n-dimensional parallelepiped or ellipsoid. The significance of lattice enumeration methods lies in their minimal memory requirements, which scale linearly with the lattice dimension n, and their exceptional practical performance in moderately low dimensions. However, they do have time complexity exponential in the lattice dimension, as is evident from the table below.

Method	**Time Complexity**	**Space Complexity**
Fincke-Pohst's	$2^{O(n^2)}$	$poly(n)$
Kannan's	$2^{O(n \log n)} = n^{O(n)}$	$poly(n)$

Within the confines of this thesis, we only present the core idea behind enumeration methods and nothing else. The reader is referred to Section 14.6 of [ΔρΑ22], Section 18.4 of [Gal18] and the other references, for more. However, before our presentation, we have included an illustration[3] of how the Fincke-Pohst algorithm works in two dimensions, made by Thijs Laarhoven, and available on his page [link], where one can also find beautiful illustrations for the other algorithms (and other families of this category) too.

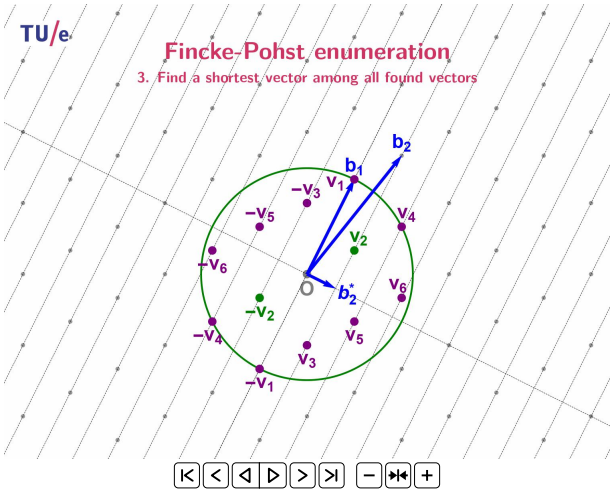


Figure 2.7: Fincke-Pohst algorithm in two dimensions (by Thijs Laarhoven).

Main Idea (Enumeration). Enumeration algorithms, in their core, are algorithms that return all vectors of a lattice $\mathcal{L}$ with length smaller than a positive number R. We note that there is always at least one lattice vector $\leq R$ as $\mathbf{0} \in \mathcal{L}$. Particularly, these vectors are found by the algorithms through the construction of an *enumeration tree* which consists of all the vectors of the lattices

$$\pi_n(\mathcal{L}), \pi_{n-1}(\mathcal{L}), \ldots, \pi_1(\mathcal{L}),$$

that have length $\leq R$, and where the $\pi_i(\mathcal{L})$, for $1 \leq i \leq n$ are the projected lattices with the functions π_i defined as follows:

Definition 2.1. Let $\mathcal{L}$ be a lattice and $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$ be its basis. Then, we call an *orthogonal projection* in the subspace $span(\mathbf{b}_1^*, \ldots, \mathbf{b}_k^*) \subset \mathbb{R}^n$ the function $\pi_i : \mathbb{R}^n \to span(\mathbf{b}_1^*, \ldots, \mathbf{b}_k^*)$, for $1 \leq i \leq k$, such that:

$$\pi_i(\mathbf{x}) = \sum_{j=1}^{k} proj_{\mathbf{b}_j^*}(\mathbf{x}) = \sum_{j=1}^{k} \frac{\langle \mathbf{x}, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle} \mathbf{b}_j^*$$

[3]The illustration starts from the final step of the method in order to provide full view for readers which cannot control the "gif".

Let now $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be a basis of the lattice. Then, it is easy to prove (see page 193 of [Δρα22]) that $\mathbf{b}_i = \mathbf{b}_i^* + \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*$. Therefore, a vector $\mathbf{v} = \sum_{j=1}^{n} x_i\mathbf{b}_i \in \mathcal{L}$ can be written as

$$\mathbf{v} = \sum_{j=1}^{n} \left(x_j + \sum_{i=j+1}^{n} \mu_{i,j}x_i \right) \mathbf{b}_j^*.$$

Hence, we can calculate the length of $\mathbf{v}$ and $\pi_r(\mathbf{v})$ for $1 \leq r \leq n$ as

$$||\mathbf{v}||^2 = \sum_{j=1}^{n} \left(x_j + \sum_{i=j+1}^{n} \mu_{i,j}x_i \right)^2 ||\mathbf{b}_j^*||^2$$

$$||\pi_r(\mathbf{v})||^2 = \sum_{j=r}^{n} \left(x_j + \sum_{i=j+1}^{n} \mu_{i,j}x_i \right)^2 ||\mathbf{b}_j^*||^2.$$

We crucially observe that if $\mathbf{v}$ has length smaller than R then,

$$||\pi_n(\mathbf{v})||^2 \leq ||\pi_{n-1}(\mathbf{v})||^2 \leq \ldots \leq ||\pi_1(\mathbf{v})||^2 = ||\mathbf{v}||^2 \leq R^2.$$

Building this observation, the algorithms try to bound $x_n \in \mathbb{Z}$ (which is $\mathbf{v}$'s n-th coordinate) using the inequality $||\pi_n(\mathbf{v})||^2 \leq R^2$, i.e. they try to compute an interval I_1 in which all possible values for x_n lie. Then, they proceed by calculating an interval I_2 for x_{n-1}, and so on, until I_n is reached. For the exact values of I_i and more details on the matter, see Section 14.6 of [Δρα22].

Therefore, the enumeration tree of height n consists of the possible values taken by x_i ($1 \leq 1 \leq n$), where the nodes of the tree in depth $k = n + 1 - d$ ($d = n, n-1, \ldots, 1$) are the vectors of the lattice $\pi_d(\mathcal{L})$ with length at most R. If the algorithm reaches a leaf, then it outputs that vector (which is a n-dimensional vector with length smaller than R). If it does not, then the algorithm returns FAIL.

Remark 2.3. The number of nodes in a particular depth depends, aside from R, on how "good" the starting basis of the lattice is. This is why most enumeration algorithms use lattice basis reduction algorithms as a *preprocessing phase*, in order to work faster afterwards.

Sieving Algorithms. Due to their complexity, we provide only a very concise overview of this algorithm family. At a high level, these algorithms for finding the shortest vector in a lattice involve selecting numerous lattice points within a large sphere or box in a certain way. For example, in Ajtai-Kumar-Sivakumar (AKS) algorithm [AKS01] this is done by generating random combinations of the basis.

Subsequently, these points undergo a refinement process through an iterative method known as *sieving*. Usually, this means taking pairs of these points, and computing their differences, which will be vectors of the lattice. Short vectors (or potential shortest vectors) are retained, while collisions are discarded. This process is iterated, aiming to converge toward the shortest nonzero vector in the lattice.

Interested readers should also refer to Stephens-Davidowitz's presentations in [Ins20], for better understanding of how sieving generally works and valuable illustrations.

It is evident that a lot of questions arise from the preceding description, and the answers to these questions are the ones that separate the algorithms of this family. These questions include:

1. How are the first vectors selected?
2. How do you find close pairs?
3. Which pairs do you choose?
4. What is the distribution of vectors at each step?
5. How common are collisions?

The diverse algorithms within this family are defined by the unique design decisions made in response to these questions.

Moreover, we provide insights into the computational complexity of established sieving algorithms, as they are the best exact algorithms with practical applications in cryptography, and more precisely in the security analysis of cryptographic schemes. To present this information effectively, we utilize a table[4] taken (partially) from [Mic21], where one can find the corresponding references and a detailed introduction to sieving algorithms. We remark that all the algorithms in this table are heuristic, which means that they will not provably solve the problem. However they are still useful in practical applications, as they are usually faster than provable ones (and work "reasonably well" in practice). Also, regarding the table, we note that c_{space} and c_{time} denote space and time complexity, respectively.

Citation	c_{space}	c_{time}	Notes
NV08	$2^{0.2075n}$	$2^{0.4150n}$	First heuristic sieving algorithm.
MV10	$2^{0.2075n}$	$2^{0.4150n}$	Introduces GaussSieve. Large practical speedup over NV08.
BDGL16	$2^{0.2075n}$	$2^{0.292n}$	Best known heuristic runtime.
HK17	$2^{0.1887n}$	$2^{0.3717n}$	First algorithm to beat GaussSieve in both space and time.
HKL17	$2^{0.1887n}$	$2^{0.3588n}$	Improves runtime of HK17.

Table 2.1: A list of recent heuristic sieving algorithms and their associated heuristic complexity.

2.2.2 Approximation Algorithms

When discussing approximation algorithms, block reduction is essentially the only available choice, i.e. there are, as far as we know, no non-trivial approximation algorithms that cannot be viewed as block reduction (aside from some sub-exponential quantum algorithms for lattices with a certain structure). Moreover, LLL (covered in the previous chapter), is in fact a block reduction algorithm, as is also the widely known BKZ algorithm from [SE94] and its descendants.

BKZ Algorithm. Schnorr introduced in [Sch87] the concept of BKZ reduction as a generalization of LLL and later, along with Euchner, presented the first version of the BKZ algorithm in [SE94]. At its core, this algorithm (and its refinements) use the notion of a Korkine-Zolotarev reduced basis and the following theorem showing the powerful properties of such a basis.

Definition 2.2. Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be an (ordered) basis of a lattice $\mathcal{L}$ in $\mathbb{R}^m$, with rank n. Also, let $\{\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*\}$ be its Gram-Schmidt orthogonalisation with coefficients $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle$ for $1 \leq j < i \leq n$. Then, the basis is called *Korkine-Zolotarev reduced* (or *Hermite-Korkine-Zolotarev reduced*) if it satisfies the following:

1. $\mathbf{b}_1$ is a nonzero vector of minimal length in $\mathcal{L}$.
2. $|\mu_{i,1}| < 1/2$ for $2 \leq i \leq n$.
3. The orthogonal projection of the starting basis onto the orthogonal complement of $\mathbf{b}_1$, i.e. the basis $\{\mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1, \ldots, \mathbf{b}_n - \mu_{n,1}\mathbf{b}_1\}$, is Korkine-Zolotarev reduced.

Theorem 2.3. Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be a Korkine-Zolotarev reduced basis of a lattice $\mathcal{L}$. Then,

(i) For $1 \leq i \leq n$, we have

$$\frac{4}{i+3}\lambda_i^2 \leq ||\mathbf{b}_i||^2 \leq \frac{i+3}{4}\lambda_i^2, \text{ for } 1 \leq i \leq n$$

(ii) It is true that

$$\prod_{i=1}^{n} ||\mathbf{b}_i||^2 \leq \left(\gamma_n^n \prod_{i=1}^{n} \frac{i+3}{4}\right) \det(\mathcal{L})^2.$$

[4]This should not be considered the "complete picture" as the status in the field is ever-changing.

Proof. See Theorem 2.1 and 2.3 of [LLS90].

From what was shown in the previous subsection, it is practical to perform an enumeration of all short vectors if the dimension of the lattice searched is small enough. Thus, one can compute a Korkine-Zolotarev basis for lattices of small dimension.

For larger dimensions, the (heuristic) block Korkine-Zolotarev lattice basis reduction algorithm can be used, which operates by computing Korkine-Zolotarev bases for lower-dimensional projections of the original lattice in combination with the LLL algorithm. The basis which this process outputs can be proved to be "better" than an LLL-reduced one and thus, this algorithm and its variants are the most powerful in solving the approximate SVP.

WHAT IS A "BLOCK"? Using a different notation $\tilde{\mathbf{b}}_i$ instead of $\mathbf{b}_i^*$, it is a well-known result (relating to the Gram-Schmidt orthogonalisation) that the basis matrix $\mathbf{B}$ can transform into the following by a change of coordinates (figures taken by Stephens-Davidowitz's presentations):

$$\begin{pmatrix} \|\widetilde{\mathbf{b}}_1\| & \mu_{1,2}\|\widetilde{\mathbf{b}}_1\| & \mu_{1,3}\|\widetilde{\mathbf{b}}_1\| & \cdots & \mu_{1,n}\|\widetilde{\mathbf{b}}_1\| \\ 0 & \|\widetilde{\mathbf{b}}_2\| & \mu_{2,3}\|\widetilde{\mathbf{b}}_2\| & \cdots & \mu_{2,n}\|\widetilde{\mathbf{b}}_2\| \\ 0 & 0 & \|\widetilde{\mathbf{b}}_3\| & \cdots & \mu_{3,n}\|\widetilde{\mathbf{b}}_3\| \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \|\widetilde{\mathbf{b}}_n\| \end{pmatrix}$$

Figure 2.8: 2×2 LLL block.

$$\begin{pmatrix} \|\widetilde{b}_1\| & \mu_{1,2}\|\widetilde{b}_1\| & \dots & \mu_{1,k}\|\widetilde{b}_1\| & \mu_{1,k+1}\|\widetilde{b}_1\| & \dots & \mu_{1,n}\|\widetilde{b}_1\| \\ 0 & \|\widetilde{b}_2\| & \dots & \dots & \mu_{2,k+1}\|\widetilde{b}_2\| & \dots & \mu_{2,n}\|\widetilde{b}_2\| \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & \|\widetilde{b}_k\| & \mu_{k,k+1}\|\widetilde{b}_k\| & \dots & \mu_{k,n}\|\widetilde{b}_k\| \\ 0 & 0 & \dots & \dots & \|\widetilde{b}_{k+1}\| & \dots & \mu_{k+1,n}\|\widetilde{b}_{k+1}\| \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \dots & \dots & \|\widetilde{b}_n\| \end{pmatrix}$$

Figure 2.9: $k \times k$ BKZ block.

In this modified basis form, the notion of a "block", from which the BKZ algorithm got its name from, becomes apparent. In the case of LLL, the algorithm tries to make the top right vector in each 2×2 block as short as possible (by satisfying the two known conditions). For BKZ, the block will be a $k \times k$ one, which corresponds to the lower-dimensional projection we mentioned earlier, and in which the shortest vector is found using an SVP oracle (in a much smaller dimension k). Moreover, the number of blocks necessitates multiple executions of the SVP oracle, typically a polynomial number of times.

This SVP oracle is exactly how the exact algorithms tie into the approximation algorithms, which are used in practice. Different exact algorithms (enumeration, sieving, etc.) correspond to different instantiations of the BKZ algorithm (and its variations which work similarly with an oracle). This will also be of critical importance in the security analysis of the Kyber cryptographic scheme, which we perform in the final chapter of this thesis.

Due to lack of space, we stop our brief overview of BKZ here, as these key facts are more than enough for the aim of this work but, one can find out more in the sources cited in this subsection.

Part II

Learning with Errors (LWE)

Chapter 3

LWE Theory and Applications

In this chapter, we explore the realm of *Learning With Errors (LWE)*, an essential part of lattice-based cryptography. We commence with an examination of particular variants of SVP_γ, specifically finding short vectors in random q-ary lattices and LWE lattices. The intention is to connect our previous study of lattices and the domain of LWE, as these variants are directly linked to the the *Short Integer Solution (SIS)* problem and the LWE problem, respectively. We note that the SIS problem is a dual problem to LWE and cannot be omitted from any serious study on LWE and LWE cryptosystems.

Afterwards, our discussion extends to Gaussian-like distributions over lattices, as they are essential to LWE. Then, having seen all necessary preliminaries, we progress to an analysis of SIS and LWE, introducing several key concepts and assessing their hardness through worst-case/average-case reductions. Subsequently, the insights gained from this analysis set the stage for an in-depth exploration of LWE cryptographic schemes, focusing on public-key encryption.[1]

3.1 A First Approach Through Lattices

As mentioned earlier, we begin our study with two specialized variants of SVP_γ, which are directly related to cryptography: (i) finding short vectors in random q-ary lattices and (ii) finding short vectors in random LWE lattices. As these instances are directly linked to the average-case SIS and LWE problems that we are interested in, this section acts as a bridge between the lattice foundations we set before, and the realm of LWE.

3.1.1 Finding Short Vectors in Random q-ary Lattices

This and the next subsection are focused in lattices that are useful in cryptography, i.e. the q-ary lattices that were presented in subsection 1.1.8. There, we defined q-ary lattices and proved some useful upper bounds and estimates regarding their shortest nonzero vector.

Expanding on this, we now delve more into the theoretical aspects underlying the search for short nonzero vectors in random q-ary lattices, following Lyubashevsky's tutorial in [Lyu20]. Additionally, we incorporate insights taken from experimental estimates, like the ones made by Gama and Nguyen in [GN08b] and their applications on q-ary lattices, as observed in [MR09].

"Special" Form. In order to simplify our theoretical analysis, we first have to transform the matrix of these lattices into a certain form (which we formally define in the next chapter):

Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. We defined a parity check lattice as $\mathcal{L}_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{x} \equiv \mathbf{0} \bmod q\}$, whereas the lattices under consideration in our analysis take the form of $\mathcal{L}_q^\perp([\mathbf{A}|\mathbf{I}_n])$. Going

[1] While we presume a basic understanding of cryptography from the reader, introductory books on cryptography can be used for a brush-up on concepts like IND-CPA and IND-CCA security, one-way functions, collision resistance, etc.

from the definition to the "special" form requires only the assumption that $\mathbf{A}$ contains n linearly independent columns over $\mathbb{Z}_q$ (which happens with probability exponentially close to 1 when $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$). Without loss of generality, we then suppose that $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2]$, where $\mathbf{A}_2 \in \mathbb{Z}_q^{n\times n}$ is invertible. Thus, we have $\mathbf{A}_2^{-1}\mathbf{A} = [\mathbf{A}_2^{-1}\mathbf{A}_1|\mathbf{I}_n]$ and $\mathcal{L}_q^\perp(\mathbf{A}) = \mathcal{L}_q^\perp(\mathbf{A}_2^{-1}\mathbf{A})$, where the latter is in the desired form.

Remark 3.1. Another use of the above form is that it allows us to easily switch between the "parity check" matrix representation above and the typical ("generator" matrix) representation of lattices with the equality below:

$$\mathcal{L}_q^\perp([\mathbf{A}|\mathbf{I}_n]) = \mathcal{L}\left(\begin{bmatrix} -\mathbf{I}_m & \mathbf{0}_m \\ \mathbf{A} & q\mathbf{I}_n \end{bmatrix}\right).$$

(Non)-existence of short vectors in random lattices. In order to prove some statements related to the (non)-existence of short vectors in random lattices, it is useful first to have the following definitions:

Definition 3.1. Let $\mathcal{L}$ be an m-dimensional lattice and let $\mathbf{r}$ be a point in $\mathbb{Z}^m$. The l_p-norm distance from $\mathbf{r}$ to the lattice is defined as:

$$\Delta_p(\mathbf{r}, \mathcal{L}) = \min_{\mathbf{v}\in\mathcal{L}} ||\mathbf{v} - \mathbf{r}||_p.$$

Remark 3.2. The above notion of distance is well-defined for cosets too, as for any two elements $\mathbf{r}_1, \mathbf{r}_2$ of the same coset of $\mathbb{Z}^m/\mathcal{L}$, we have $\Delta_p(\mathbf{r}_1, \mathcal{L}) = \Delta_p(\mathbf{r}_2, \mathcal{L})$.

Additionally, for the lattice $\mathcal{L}_q^\perp(\mathbf{A})$, it is easy to observe that $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{Z}^m$ are in the same coset if and only if $\mathbf{A}\mathbf{z}_1 \equiv \mathbf{A}\mathbf{z}_2 \,(\mathrm{mod}\, q)$. Therefore, if $\mathbf{t} \equiv \mathbf{A}\mathbf{z} \,(\mathrm{mod}\, q)$ defines a coset $\mathbf{z} + \mathcal{L}$, then we write

$$\Delta_p^C(\mathbf{t}, \mathcal{L}) = \Delta_p(\mathbf{z}, \mathcal{L}),$$

where with Δ^C we denote that $\mathbf{t}$ is the image of the coset under $\mathbf{A}$, rather than using a coset representative.

Next, we present three important results, useful in understanding the landscape of this topic. Prior to that, we remind that $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ denotes that $\mathbf{A}$ is chosen uniformly at random from the set $\mathbb{Z}_q^{n\times m}$.

Proposition 3.1. For any q and any $\mathbf{t} \in \mathbb{Z}_q^n$ that has a coefficient t_i such that $\gcd(t_i, q) = 1$:

$$\Pr_{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m}}\left[\exists\, \mathbf{z} \in [\beta]^{n+m} \text{ s.t. } [\mathbf{A}|\mathbf{I}_n]\mathbf{z} \equiv \mathbf{t} \,(\mathrm{mod}\, q)\right] \leq \frac{(2\beta+1)^{n+m}}{q^n},$$

where $[\beta] = \{-\beta, \ldots, -1, 0, 1, \ldots, \beta\}$.

Proof. See Lemma 1 of [Lyu20].

Corollary 3.1. For the lattice $\mathcal{L}_q^\perp([\mathbf{A}|\mathbf{I}_n])$, we have:

$$\Pr_{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m}, \mathbf{t}\leftarrow\mathbb{Z}_q^n}\left[\Delta_\infty^C(\mathbf{t}, \mathcal{L}_q^\perp([\mathbf{A}|\mathbf{I}_n]) \leq \beta]\right] \leq (1 - |\mathbb{Z}_q^*|/q)^n + \frac{(2\beta+1)^{n+m}}{q^n}$$

In cryptography, usually we have either q prime (giving us $|\mathbb{Z}_q^*| = q - 1$) or q that is a power of two (giving us $|\mathbb{Z}_q^*| = q/2$). Hence, the first term in the bound is negligible[2] in n. Therefore, whenever $\beta^{1+\frac{m}{n}} \ll q$, the above probability will be really small and thus random cosets will be more than distance β away from the lattice.

Proposition 3.2. For any prime q, we have:

$$\Pr_{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m}}\left[\exists\, \mathbf{z} \in [\beta]^{n+m}\backslash\{\mathbf{0}\} \text{ s.t. } [\mathbf{A}|\mathbf{I}_n]\mathbf{z} \equiv \mathbf{0} \,(\mathrm{mod}\, q)\right] \leq \frac{(2\beta+1)^{n+m}}{q^n}$$

[2] A function $\epsilon : \mathbb{N} \to \mathbb{R}$ is *negligible* if for all polynomials p, $\exists N_p$ constant such that $\epsilon(n) \leq 1/p(n)$, $\forall n \geq N_p$.

The proposition above states that the probability, over the choice of $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, that the lattice $\mathcal{L}_q^\perp([\mathbf{A}|\mathbf{I}_n])$ has a "short" nonzero vector is small, as in usual cryptographic applications, the value that bounds the probability is small. Additionally, the proposition was stated only for q prime, as in this case the proof is identical to that of the previous proposition.

Proposition 3.3. For any q and any $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we have:

$$\exists\, \mathbf{z} \in [q^{n/(n+m)}]^{n+m} \backslash \{\mathbf{0}\} \text{ s.t. } [\mathbf{A}|\mathbf{I}_n]\mathbf{z} \equiv \mathbf{0} \,(\mathrm{mod}\, q)$$

Proof. The proof is based on the pigeonhole principle, see also Lemma 3 of [Lyu20].

This last proposition can be considered the converse of the previous one as it gives a lower bound on the length of an existing nonzero vector. Combining this with Proposition 3.2 we have the following:

- If we set $\beta = q^{n/(n+m)}$ then a short nonzero vector always exists in $[\beta]^{n+m}\backslash\{\mathbf{0}\}$.
- If we set $\beta < \frac{1}{4}q^{n/(n+m)}$, then the probability of such a vector existing becomes $\leq 2^{-(n+m)}$.

Finding short vectors in random lattices. From the above, we now know when such a vector (i.e. a nonzero $\mathbf{z} \in [\beta]^{n+m}$ such that $[\mathbf{A}|\mathbf{I}_n]\mathbf{z} \equiv \mathbf{0} \,(\mathrm{mod}\, q)$, with $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$) exists, but we still have not mentioned how it can be found exactly. That is where the algorithms we studied in the previous chapter come in. As we have briefly seen, all known exact algorithms (quantum or not) for finding such vectors (with $\mathbf{A}$ chosen uniformly random from $\mathbb{Z}_q^{n\times m}$) take $2^{\Omega(m+n)}$ time.

Now, let's investigate how the challenge of the problem evolves as we adjust the values of β:

- If $\beta = q/2$, then the problem can be solved trivially:

  Suppose $\mathbf{z}^{(1)} = (z_1, \ldots, z_m)$ and $\mathbf{z}^{(2)} = (z_{m+1}, \ldots, z_{m+n})$ such that $\mathbf{z} = \begin{bmatrix}\mathbf{z}^{(1)}\\ \mathbf{z}^{(2)}\end{bmatrix}$.

  Then,

  $$[\mathbf{A}|\mathbf{I}_n]\mathbf{z} = [\mathbf{A}|\mathbf{I}_n]\begin{bmatrix}\mathbf{z}^{(1)}\\ \mathbf{z}^{(2)}\end{bmatrix} = \mathbf{A}\mathbf{z}^{(1)} + \mathbf{I}_n\mathbf{z}^{(2)}$$

  And thus, setting the coefficients of $\mathbf{z}_2$ to the target coefficients from $[\mathbf{A}|\mathbf{I}_n]\mathbf{z} \equiv \mathbf{0} \,(\mathrm{mod}\, q)$, gives us a solution in $[q/2]^{m+n}$.
- If $\beta \ll \frac{1}{2}q^{n/(n+m)}$, then the problem becomes vacuously hard.
- By setting the value of β "reasonably" in-between the previous values, the difficulty increases while still not making the problem insolvable. In this case, in a reasonable amount of time, one can only hope to find vectors are that some factor larger than the shortest one, using approximation algorithms.

As we have briefly touched upon, all modern polynomial-time approximation algorithms are LLL and its descendants. For instance, assume that we take LLL, which is guaranteed to find a vector of length at most $2^{O(n+m)}$ times larger than λ_1, and combine this with Proposition 3.2. Then, for a random $\mathbf{A}$, LLL will find a vector $\mathbf{z} \in [2^{O(n+m)}q^{n/(n+m)}]^{n+m}$ in $\mathcal{L}_q^\perp([\mathbf{A}|\mathbf{I}_n])$. However, this is a theoretical bound, in practice the estimated length of the vectors found by LLL is much better, as shown by the experiments of [GN08b] and the analysis of [MR09].

Furthermore, the final results we got from this experimental work show that one can find vectors in random lattices of the form $\Lambda = \mathcal{L}_q^\perp([\mathbf{A}|\mathbf{I}_n])$ (with dimension $m+n$) of length approximately[3]

$$\min\left\{q, \det(\Lambda)^{1/(n+m)} \cdot \delta^{n+m}\right\} = \min\left\{q, q^{n/(n+m)} \cdot \delta^{n+m}\right\},$$

where δ is the Root Hermite factor we first mentioned in Subsection 1.1.7. We remind also that δ depends on the efficacy of the algorithm used, with $\delta = 1.010$ considered within reach, while $\delta = 1.005$ may never by achieved by algorithms for lattices with high dimensions (e.g. ≥ 500).

[3] As proved before, a "trivial" vector of length q can always be found, and that's where the q in the estimation originates from.

Another point of interest is the dimension of the lattice, as it can be proved than increasing it does not make the problem harder. This is apparent by the fact that we can always fix some of the coordinates of the solution to 0, reducing the problem to one with smaller dimension. However, this can be done effectively only if the dimension of the lattice is large enough, as having a small dimension makes the lattice too sparse (thus not containing short enough vectors). Moreover, as stated in [MR09], the optimal value for the dimension of the lattice is $\sqrt{n \log q / \log \delta}$, which transforms the above experimental estimation of the size of the vector found to

$$\min \left\{ q, 2^{2\sqrt{n \log q \log \delta}} \right\}.$$

All of this can easily be seen in the figure below (taken from Section 3 of [MR09]), where the value of $q^{n/(n+m)}\delta^{n+m}$ is plotted as a function of the dimension, with $\delta = 1.01$, $q = 44168657$ (with $\log q \approx 22$) and $n = 100$:

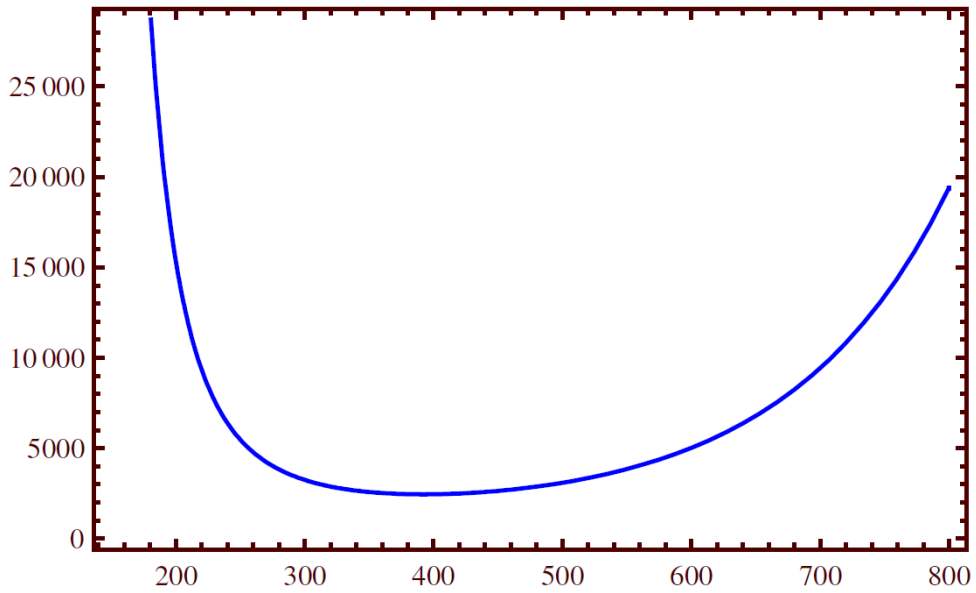


Figure 3.1: Estimated length of vector found as a function of m.

FROM LATTICES TO THE SIS PROBLEM. Having seen everything we needed to know about the difficulty of finding a short nonzero vector in a random lattice of the form $\mathcal{L}_q^{\perp}([\mathbf{A}|\mathbf{I}_n])$, it is a good time to "reveal" that this problem is (a "special form" of) the Short Integer Solution problem, symbolized $\text{SIS}_{n,q,\beta,m}$, which is one of the two main problems in which lattice based cryptography is founded on. However, before delving further into the problem and how it connects to cryptography, it is essential to first touch upon the second key problem, the LWE problem.

3.1.2 Finding Short Vectors in LWE lattices

Here, we introduce the (decision variant of the) Learning with Errors (LWE) problem within the framework of lattices, as stated in [Lyu20], and get a first grasp of its computational complexity.

Assume the two following scenarios:

1. One outputs (i) a lattice $\Lambda = \mathcal{L}_q^{\perp}([\mathbf{A}|\mathbf{I}_n])$, for a random $\mathbf{A}$; and (ii) the image $\mathbf{t}$ of a coset in $\mathbb{Z}_q^m/\Lambda$ such that $\Delta_{\infty}^{C}(\mathbf{t}, \Lambda) \leq \beta$.
2. One outputs (i) a lattice $\Lambda = \mathcal{L}_q^{\perp}([\mathbf{A}|\mathbf{I}_n])$, for a random $\mathbf{A}$; and (ii) the image $\mathbf{t}$ of a random coset in $\mathbb{Z}^m/\Lambda$.

Then, the (decision variant of) LWE, symbolized $\text{LWE}_{n,m,q,\beta}$ can be seen as trying to distinguish between the above two scenarios, i.e. distinguishing cosets that are close to the lattice, and random cosets.

Additionally, it is common for cryptographic schemes to have $m = n$ and $\beta^2 = O(q/\sqrt{m})$ (thus, $\beta \ll \sqrt{q}$). Therefore, from Corollary 3.1, in this case, the (decision variant of) LWE can be seen as distinguishing between cosets that are close to the lattice and cosets that are far away.

Let's now investigate how this problem's difficulty changes when we vary the value of β. At first glance, if $\mathbf{t}$ is according to scenario (1), as the image of such a coset, it will be of the form $\mathbf{t} = \mathbf{A}\mathbf{s} + e \pmod q$ for $\mathbf{s} \leftarrow [\beta]^m$ and $\mathbf{e} \leftarrow [\beta]^n$. This implies that there exists a vector $\mathbf{z} = [\mathbf{z}^{(1)} \,|\, 1 \,|\, \mathbf{z}^{(2)}]^T \in [\beta]^{n+m+1}$, with $\mathbf{z}^{(1)} \in [\beta]^m$ and $\mathbf{z}^{(2)} \in [\beta]^n$, in the lattice $\mathcal{L}_q^{\perp}([\mathbf{A}|\mathbf{t}|\mathbf{I}_n])$. Then, according to [Lyu20], the LLL algorithm is guaranteed to find a vector with coefficients in $[\delta^{n+m+1}\beta]$. If this bound is less than $q^{n/(n+m+1)}$, then by Proposition 3.2 we are able to distinguish the above lattices from those where $\mathbf{t}$ is uniformly random. In light of the above observations, it is evident that the problem gets harder as β grows, becoming vacuously hard when β is large enough (close to $q^{n/(n+m)}$). At this range, the distribution of $\mathbf{t}$ actually becomes uniformly random, thus making us unable to distinguish the two cases.

SIS Vs LWE (for varying β). In this section, we have explored the connection between lattices and the problems SIS and LWE. Also, we delved into the (practical) hardness of these problems and as an extra takeaway we have the impact of β on the hardness of these problems.

When we have an equation like $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$, the challenge of finding a short vector in the lattice $\mathcal{L}_q^{\perp}([\mathbf{A}|\mathbf{t}|\mathbf{I}_n])$ (or distinguishing the lattice from a uniform distribution when $\beta < q^{n/(n+m)}$) becomes harder as β grows within the range of $0 < \beta < q^{n/(n+m)}$. Conversely, it becomes easier as β grows in the range of $q^{n/(n+m)} < \beta < q$. The former range of β values corresponds to the LWE problem, while the latter is associated with the SIS problem.

Therefore, when constructing a cryptographic scheme based on LWE we want to set β as large as possible, while still allowing "functions", like decryption, to work properly. In the table below, taken from [Lyu20], we give some representative parameters for $\mathrm{SIS}_{n,q,\beta}$ and $\mathrm{LWE}_{m,q,\beta}$ (i.e. similar to those used for actual cryptographic schemes), and their security levels expressed in bits of security, where security of n bits means that breaking this problem would take 2^n operations.

$\mathsf{LWE}_{m,q,\beta}$ Parameters			
m	β	q	Security
512	2	2^{13}	110
768	2	2^{13}	160
1024	2	2^{13}	210
768	6	2^{23}	90
1024	5	2^{23}	128
1280	3	2^{23}	160

$\mathsf{SIS}_{n,q,\beta}$ Parameters			
n	β	q	Security
1024	2^{20}	2^{23}	95
1280	2^{20}	2^{23}	125
1536	2^{20}	2^{23}	160

Figure 3.2: Approximate (conservative) hardness of the LWE and SIS problems against quantum algorithms for some representative parameters.

It is evident that the values of β are slightly smaller those that one would expect after reading this section. However, as we progress through Part II, it will be clearer why such β are used, as there are other factors at play too.

3.2 Gaussian-like Distributions over Lattices

In the previous section we tried to approach the SIS and LWE problems from within the lattice framework we build in Part I. However, this deviates from the typical introduction to these

problems. So, in the next sessions we follow through with a proper introduction (and make the connection to the previous one). To do this, we need some more preliminaries, particularly from the world of statistics.

Numerous contemporary studies in the fields of complexity and cryptography heavily depend on probability distributions over lattices that resemble Gaussian distributions, called discrete Gaussians, as well as some notions related to them. Below, we provide a concise overview consisting of key definitions and propositions:

GAUSSIANS. We begin by defining Gaussian functions, continuous Gaussian distributions and then use them as a stepping stone to define discrete Gaussian distributions. For this we followed the works of [Pei16] and [MR04].

Definition 3.2. For any positive integer n, any vectors $\mathbf{c}, \mathbf{x}$ and any real $s > 0$, we define the *Gaussian function* centered in $\mathbf{c}$ and of *parameter* (or *width*) s, to be a function $\rho_{\mathbf{c},s} : \mathbb{R}^n \to \mathbb{R}^+$, where:

$$\rho_{\mathbf{c},s}(\mathbf{x}) \doteqdot e^{-\pi||(\mathbf{x}-\mathbf{c})/2||^2}$$

Remark 3.3.

(a) The total measure associated to $\rho_{\mathbf{c},s}$ is $\int_{\mathbf{x}\in\mathbb{R}^n} \rho_{\mathbf{c},s}(\mathbf{x})dx = s^n$.

(b) When $\mathbf{c}$ or s are omitted, we assume that they are the origin and 1, respectively. Moreover, the function is extended to sets in the usual way, e.g. $\rho_{\mathbf{c},s}(A) = \sum_{\mathbf{x}\in A} \rho_{\mathbf{c},s}(\mathbf{x})$, for any countable set A.

(c) The "normalization factor" π is chosen above so that ρ is its own Fourier transform.

Definition 3.3. For any positive integer n, any vectors $\mathbf{c}, \mathbf{x}$ and any real $s > 0$, the *continuous Gaussian distribution* $D_{\mathbf{c},s}$ around $\mathbf{c}$ with parameter s is defined over $\mathbb{R}^n$ by its probability density function

$$D_{\mathbf{c},\mathbf{s}}(\mathbf{x}) = \rho_{\mathbf{c},s}(\mathbf{x})/s^n$$

Remark 3.4.

(a) The Gaussian distribution is a *product distribution* because for $\mathbf{x} = (x_1, \ldots, x_n)^T$, we have $\rho_{\mathbf{c},s}(\mathbf{x}) = \prod_{i=1}^n \rho_{c_i,s}(x_i)$.

(b) It can be proven that $\rho_{\mathbf{c},s}$ is invariant under rotations of $\mathbb{R}^n$ and therefore the Gaussian distribution is *spherically symmetric*, i.e. the probability of $\mathbf{x}$ only depends on its length and the distribution is "axis-independent".

(c) It can be seen that the expected square distance from $\mathbf{c}$ of a vector chosen from this distribution is $ns^2/(2\pi)$. Therefore, one can think of the distribution as a sphere of radius $s\sqrt{n/(2\pi)}$ centered around $\mathbf{c}$.

(d) In practice, when only finite precision is available, the distribution can be approximated by picking a "reasonably good" grid, and choosing points from this grid with probability approximately proportional to $D_{\mathbf{c},s}$.

Definition 3.4. For vectors $\mathbf{c}$ and $\mathbf{u}$, real $s > 0$ and lattice $\mathcal{L}$, we define

- the *discrete Gaussian distribution* $D_{\mathcal{L},\mathbf{c},s}$ *over* $\mathcal{L}$ by

$$D_{\mathcal{L},\mathbf{c},s}(\mathbf{x}) = \frac{D_{\mathbf{c},s}(\mathbf{x})}{D_{\mathbf{c},s}(\mathcal{L})} = \frac{\rho_{\mathbf{c},s}(\mathbf{x})}{\rho_{\mathbf{c},s}(\mathcal{L})}, \text{ for } \mathbf{x} \in \mathcal{L}$$

- the *discrete Gaussian distribution* $D_{\mathbf{u}+\mathcal{L},\mathbf{c},s}$ *over the coset* $\mathbf{u}+\mathcal{L}$ by

$$D_{\mathbf{u}+\mathcal{L},\mathbf{c},s}(\mathbf{x}) = \frac{D_{\mathbf{c},s}(\mathbf{x})}{D_{\mathbf{c},s}(\mathbf{u}+\mathcal{L})} = \frac{\rho_{\mathbf{c},s}(\mathbf{x})}{\rho_{\mathbf{c},s}(\mathbf{u}+\mathcal{L})}, \text{ for } \mathbf{x} \in \mathbf{u}+\mathcal{L}$$

$D_{\mathbf{c},s}$ and $D_{\mathcal{L},\mathbf{c},s}$ are connected via the following:
If $\mathbf{x}$ is distributed according to $D_{\mathbf{c},s}$ and we condition on $\mathbf{x} \in \mathcal{L}$, the conditional distribution of $\mathbf{x}$ is $D_{\mathcal{L},\mathbf{c},s}$. To understand why this is true, first recall that our vector $\mathbf{x}$ is selected from some "very fine" grid. Then, the probability of getting some grid point $\mathbf{x}$ in a sample from $D_{\mathbf{c},s}$ is

very close to $aD_{\mathbf{c},s}(\mathbf{x})$, where a is the volume of one cell in our grid, whereas the probability of $\mathbf{x} \in \mathcal{L}$ is very close to $aD_{\mathbf{c},s}(\mathcal{L})$.

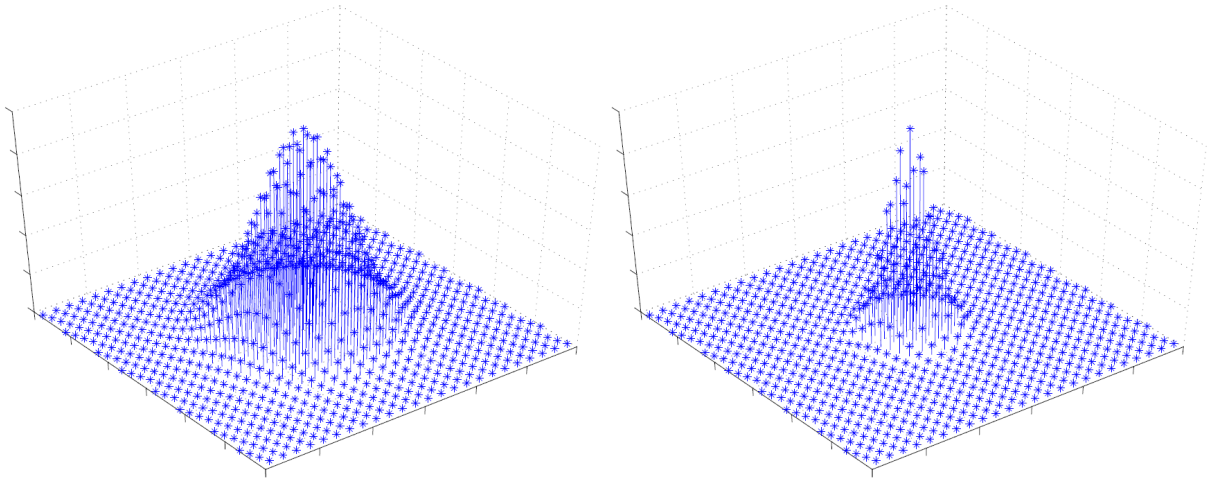


Figure 3.3: $D_{\mathcal{L},2}$ (left) $D_{\mathcal{L},1}$ (right) for a two-dimensional lattice $\mathcal{L}$, where the z-axis represents probability (taken from [Reg10]).

Moreover, in [MR04], it is proven that for a large enough s, $D_{\mathcal{L},\mathbf{c},s}$ behaves in many respects like the continuous Gaussian distribution $D_{\mathbf{c},s}$. Particularly, the average value of vectors distributed according to $D_{\mathcal{L},\mathbf{c},s}$ is very close to $\mathbf{c}$ and the expected squared distance from $\mathbf{c}$ is very close to $s^2n/2\pi$ (for vectors of $D_{\mathbf{c},s}$, these quantities are exactly $\mathbf{c}$ and $s^2n/2\pi$). In fact, in [MR04] a new lattice parameter that tells us how big s has to be in order for this to happen, was defined. This parameter is the smoothing parameter, which we describe briefly in the next paragraph.

Smoothing Parameter. The smoothing parameter is defined for a lattice $\mathcal{L}$ using its dual $\mathcal{L}^*$:

Definition 3.5. For an n-dimensional lattice $\mathcal{L}$ and a positive real $\epsilon > 0$ (the "tolerance"), we define its *smoothing parameter* $\eta_\epsilon(\mathcal{L})$ to be the smallest s such that $\rho_{1/s}(\mathcal{L}^* \backslash \{\mathbf{0}\}) \leq \epsilon$.

Informally, the are two ways to think about this quantity:

- As the amount of Gaussian "blur" necessary to "smooth out" the discrete structure of $\mathcal{L}$.
- As the smallest $s > 0$ such that the Gaussian mass $\rho_{\mathbf{c},s}(\mathbf{u} + \mathcal{L}) = \sum_{\mathbf{x} \in \mathbf{u}+\mathcal{L}} \rho_s(\mathbf{x})$ is nearly the same for every coset $\mathbf{u} + \mathcal{L}$.

More formally, the motivation for the definition originates from the lemma below. In simpler terms, it states that if one starts from a point in $\mathcal{L}$, chosen uniformly at random, and perturbs it by a Gaussian of radius $\eta_\epsilon(\mathcal{L})$, then the resulting distribution is $\epsilon/2$ close to uniform on the entire space.[4] However, before stating the lemma, we need the following definition:

Definition 3.6. The *statistical distance* between two discrete random variables X and Y over a (countable) set A is defined as $\Delta(X, Y) = 1/2 \sum_{a \in A} |\Pr[X = a] - \Pr[Y = a]|$.

Lemma 3.1. For any $s > 0$, $\mathbf{c} \in \mathbb{R}^n$ and lattice $\mathcal{L}(\mathbf{B})$, the statistical distance between $D_{\mathbf{c},s} \bmod \mathcal{P}(\mathbf{B})$ and the uniform distribution over $\mathcal{P}(\mathbf{B})$ is at most $\frac{1}{2}\rho_{1/s}(\mathcal{L}(\mathbf{B})^* \backslash \{\mathbf{0}\})$. Specifically, for any $\epsilon > 0$ and any $s \geq \eta_\epsilon(\mathcal{L}(\mathbf{B}))$, the statistical distance is at most

$$\Delta(D_{\mathbf{c},s} \bmod \mathcal{P}(\mathbf{B}) \,,\, U(\mathcal{P}(\mathbf{B}))) \leq \epsilon/2$$

[4] Indeed, no uniform probability distribution can be defined over a lattice (as it is a countably infinite set) or over the entire space. Formally, in order to define this property, we capture the intuition of "starting from a random lattice point" by working *modulo the lattice*, as shown in [Mic01].

Proof. See Lemma 4.1 of [MR04], as well as [Mic01] and Lyubashevsky's presentation on [Uni12], for better explanation of "mod $\mathcal{P}(\mathbf{B})$", which we cannot explain further here due to lack of space.

For more information on this parameter and Gaussian distributions, the reader is referred to [MR04] and its references, as well as to Lyubashevsky's first presentation on [Uni12], where (as mentioned) he extensively elaborates on the thought process behind the lemma and then continues showing how it can be useful for proofs on the hardness of SIS. Moreover, Micciancio's first presentation in [Ins20] also contains some useful illustrations of the smoothing parameter and the meaning behind the process of "smoothing a lattice".

SUBGAUSSIANS. Finally, we describe the notion of subgaussianity, as defined in [Pei16].

Definition 3.7. A real variable X is *subgaussian* with parameter s if, for every $t \geq 0$, we have

$$\Pr\left[|X| > t\right] \leq 2e^{-\pi t^2/s^2}.$$

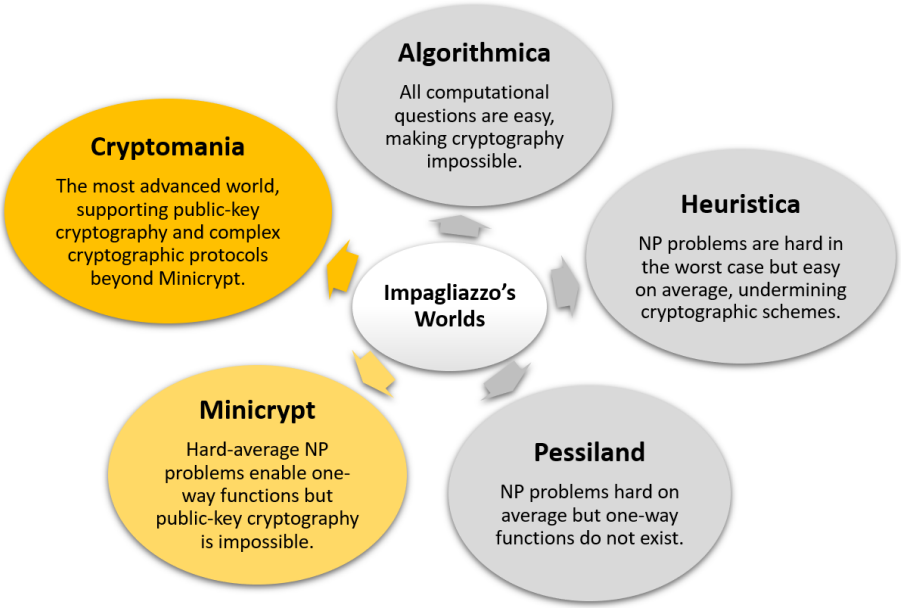
Example 3.1. The continuous Gaussian distribution $D_{\mathbf{c},s}$, and the discrete Gaussian distribution $D_{\mathcal{L},\mathbf{c},s}$ over any lattice $\mathcal{L}$, are notable examples. Also, by slightly relaxing the definition, the discrete Gaussian $D_{\mathbf{u}+\mathcal{L},\mathbf{c},s}$ over any lattice coset $\mathbf{u}+\mathcal{L}$ when $s \geq \eta(\mathcal{L})$, is also subgaussian.

3.3 Short Integer Solution (SIS)

In this section we finally define the Short Integer Solution (SIS) problem properly and to the extent it deserves. Towards this end, we first present its origins and its many uses in cryptography. Then, we define the problem and some of its properties, in order to analyse its hardness in the end.

ORIGINS OF SIS. In his seminal work [Ajt96], Ajtai gave the first *worst-case to average-case reductions*[5] for lattice problems. Additionally, on that same work he presented the first cryptographic object with a proof of security based on the hardness of hard lattice problems. Specifically, Ajtai introduced the SIS problem and its associated one-way function, proving that solving this problem is at least as hard as some lattice approximation problems in the worst case.

SIS & CRYPTOGRAPHY. The SIS problem serves as the basis for one-way and collision-resistant hash functions, identification schemes, digital signatures, and various cryptographic primitives within the *minicrypt* realm.



[5] We remind that, in the start of Chapter 2, we informally defined the meaning of "reduction" between problems.

Regarding "minicrypt", it is one of the five possible worlds of Russel Impagliazzo, as he stated them in [Imp95]. These five worlds are made of ascending levels of hardness and thus, cryptographic possibility. In *Minicrypt* specifically, certain problems within the NP complexity class exhibit average-case hardness, and this level of hardness is useful for creating one-way functions. By having these one-way functions, we gain access to a plethora of essential cryptographic tools, including secret key encryption, digital signatures, and pseudorandom number generators. However, in this world, the hardness is not enough to have public key encryption schemes and other advanced cryptographic primitives. In order to have this advanced constructions we need to be in the world of *Cryptomania*, and as we will see in the next section, SIS might not be enough to achieve the advanced cryptomania primitives, but LWE is.

3.3.1 Principal Definitions for SIS

The SIS problem is parametrized by n, q, β and m, where n, q are positive integers that define the group $\mathbb{Z}_q^n$, β is a positive real, and m is the number of group elements. Of the above, n should be thought of as the main security parameter and $q > \beta$, both of them being a small polynomial in n. Also, we note that most of the following definitions were taken by [Pei16].

Definition 3.8. (Short Integer Solution ($\mathbf{SIS}_{n,q,\beta,m}$))
Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be the matrix whose columns are formed by m vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$, chosen uniformly at random. Find a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^m$ such that

$$||\mathbf{z}|| \leq \beta \ \ \& \ \ \mathbf{Az} = \sum_{i=1}^{m} z_i \, \mathbf{a}_i = \mathbf{0} \bmod q.$$

Remark 3.5.

(i) Evidently, if we remove the condition $||\mathbf{z}|| \leq \beta$, then a solution can be found easily with Gaussian Elimination. In the same way, we want $q > \beta$, as otherwise $\mathbf{z} = (q, 0, \ldots, 0) \in \mathbb{Z}^m$ would always be a valid solution. Of course, as outlined in a previous section, additional constraints must be imposed on β and q for practical applications, but these are not of concern to us at the moment.

(ii) As demonstrated in Section 3.1, if we have a valid solution for $\mathbf{A}$, then it can also work for an extension $[\mathbf{A}|\mathbf{A}']$, by appending the necessary amount of zeroes on the solution. Thus, the SIS problem can only become easier as the number of columns m increases. However, on the other hand, increasing n obviously makes the problem harder.

(iii) If, on the other hand, we try to decrease the number of columns m, the problem might become easier at first, but after a certain value the lattice will become too sparse, narrowing down the number of possible solutions. Thus, m, as well as n, need to be large enough so that a solution is guaranteed to exist. Exact bounds are given in the lemma below.

Lemma 3.2. Let n, q, β, m be as before. Assume also that $\beta \geq \sqrt{\tilde{m}}$ and $m \geq \tilde{m}$, where $\tilde{m}$ is the smallest integer greater than $n \log q$. Then, at least one solution to the SIS problem exists.

Proof. We assume, without loss of generality, that $m = \tilde{m}$. Moreover, as

$$|\{0,1\}^m| = 2^m > 2^{n \log q} = 2^{\log q^n} = q^n,$$

there exist more than q^n vectors $\mathbf{x} \in \{0,1\}^m$. Hence, there must exist $\mathbf{x} \neq \mathbf{x}'$ such that $\mathbf{Ax} = \mathbf{Ax}' \in \mathbb{Z}_q^n$, and thus $\mathbf{z} = \mathbf{x} - \mathbf{x}' \in \{-1, 0, 1\}^m$ is a solution of norm $||\mathbf{z}|| \leq \sqrt{m} \leq \beta$.

□

Lemma 3.3. Let n, q, β, m be as before. Then, the induced function family[6] $\{f_{\mathbf{A}} : \{0,1\}^m \to \mathbb{Z}_q^n \,|\, \mathbf{A} \in \mathbb{Z}_q^{n\times m}\}$ with $f_{\mathbf{A}}(\mathbf{z}) = \mathbf{Az}$, is collision resistant, assuming the hardness of the corresponding SIS problem.

[6]The domain $\{0,1\}^m$ is somewhat random here, and can be replaced by any other large enough set containing sufficiently short vectors.

Proof. Suppose that we have a collision for $f_{\mathbf{A}}$, i.e. distinct vectors $\mathbf{x}, \mathbf{x}' \in \{0,1\}^m$ such that $\mathbf{Ax} = \mathbf{Ax}' \in \mathbb{Z}_q^n$. Then, as in the proof of the previous lemma, there exists a solution to the SIS problem $\mathbf{z} = \mathbf{x} - \mathbf{x}'$. Therefore, under the assumption that the SIS problem is hard, those collisions should also be hard to find, making the function $f_{\mathbf{A}}$ collision resistant.

□

Connection to Lattices. We now describe more formally how the SIS problem relates to lattices. In short, it can be viewed as an *average-case SVP* on a certain family of q-ary m-dimensional integer lattices, the parity check lattices (defined on Subsection 1.1.8), which we remind are:

$$\mathcal{L}_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{Ax} \equiv \mathbf{0} \bmod q\}$$

Therefore, the SIS problem asks to find a sufficiently short nonzero vector in $\mathcal{L}^{\perp}(\mathbf{A})$, where $\mathbf{A}$ is chosen uniformly at random. This is also illustrated for two dimensions in the complementary figure (taken from Peikert's presentation in [Uni12]), where either of the green points inside the circle is a valid solution to the problem.

This differs somewhat from what we informally defined in Section 3.1, where the lattice was presented in a 'special form' as $\mathcal{L}^{\perp}([\mathbf{A}|\mathbf{I}_n)$. To eliminate any confusion, we will clarify this in the following paragraph.

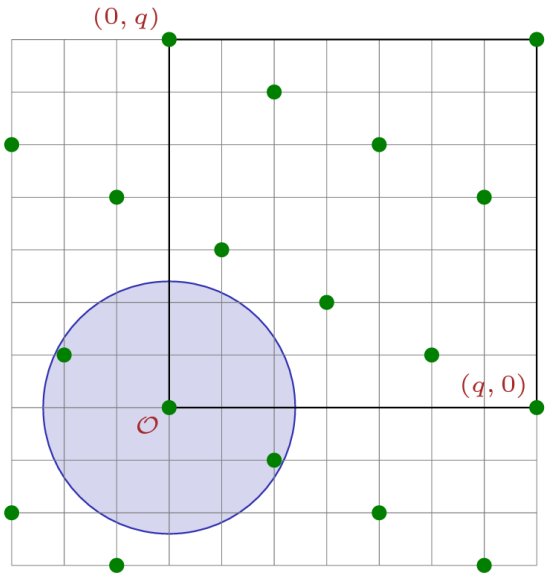


Figure 3.4: Illustration of SIS in 2D.

Normal form. The SIS problem admits a small but important optimization, called the *Hermite normal form (HNF)*[7], which allows for the reduction of instance $\mathbf{A}$ to a size of just n columns, without having an impact in cryptographic functionality or hardness.

There are two approaches to accomplish this task, the first being the one we presented in the "Special Form" paragraph of Section 3.1 (following [Lyu20]), and the other is presented by Peikert in Section 4.1 of [Pei16]. As in the remainder of this thesis we have chosen to predominantly follow Peikert's survey, we also present the second method and define this as the *HNF optimization of SIS* for the continuation of this thesis:

Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and assume that $\mathbf{A}$ contains n linearly independent columns over $\mathbb{Z}_q$ (this happens with probability exponentially close to 1). Without loss of generality, we then suppose that $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2]$, where $A_1 \in \mathbb{Z}_q^{n \times n}$ is invertible. Thus, we have

$$\mathbf{A}_1^{-1}\mathbf{A} = [\mathbf{I}_n|\bar{\mathbf{A}} = \mathbf{A}_1^{-1}\mathbf{A}_2].$$

We note that, due to $\mathbf{A}_2$ being uniform and independent of $\mathbf{A}_1$, $\bar{\mathbf{A}}$ is also uniformly random. Furthermore, it is evident that $\mathbf{A}$ and $[\mathbf{I}_n|\bar{\mathbf{A}}]$ have exactly the same set of (short) SIS solution. Hence, SIS instances of the latter type are at least as hard to solve as those of the former form.

Inhomogeneous SIS. Lastly, it is also useful to present the *inhomogeneous variant* of the SIS problem, which is to find a short integer solution to $\mathbf{Ax} = \mathbf{u} \in \mathbb{Z}_q^n$, where $\mathbf{A}$ and $\mathbf{u}$ are chosen uniformly at random and independently. We remark that for the inhomogeneous version, the solution does not have to be nonzero.

Disregarding the norm constraint, the set of all solutions is the lattice coset $L_{\mathbf{u}}^{\perp}(\mathbf{A}) \doteqdot \mathbf{c} + L^{\perp}(\mathbf{A})$, where $c \in \mathbb{Z}^m$ is an arbitrary (not necessarily short) solution. Moreover, it can be proven that the homogeneous and inhomogeneous problems are essentially equivalent for typical parameters.

[7] The optimization's connection to the Hermite normal form for lattices is explained in Section 5 of [MR09].

3.3.2 Hardness of SIS

As we have already mentioned, the first one to give a worst-case/average-case reduction for SIS was Ajtai in [Ajt96]. Building on this foundation, a series of subsequent studies has achieved increasingly stronger results on the hardness of SIS relative to worst-case lattice problems. However, all such results are instances of the following generalization:

Theorem 3.1. For any $m = poly(n)$, any $\beta > 0$, and any sufficiently large $q \geq \beta \cdot poly(n)$, solving $\text{SIS}_{n,q,\beta,m}$ with non-negligible probability is at least as hard as solving GapSVP_γ and SIVP_γ (among others) on arbitrary n-dimensional lattices (i.e. in the worst case) with overwhelming probability, for some $\gamma = \beta \cdot poly(n)$.

Proof. In order to prove the theorem, one needs to find an efficient (polynomial-time) reduction that uses an oracle for SIS (working on the average, with noticeable probability) to solve GapSVP_γ and SIVP_γ on any n-dimensional lattice.

For a general template of how to obtain such a reduction, the reader is referred to the overview in Subsection 4.1.2 of [Pei16]. However, for the novice reader we first recommend Lyubashevsky's presentation in [Uni12], as it is simpler, with useful visualisations.

Remark 3.6. We remark that the specific values of m and q (with the exception of their lower bounds) have little impact on the ultimate hardness guarantee. However, this differs for the approximation factor γ, which deteriorates as the norm bound β on the SIS solution increases.

As previously stated, there is a constant search of improved results, particularly in attempts to reduce the values of γ and q, as they were quite large in the original paper. However, as these are not in the scope of our work, we merely mention some of these improvements in the following table:

Work	**Approx. Factor (γ)**	**Modulus (q)**	**Notable Techniques**
[Ajt96]	Large, $poly(n)$	Large, $poly(n)$	
[MR04]	$\gamma = \beta \cdot \tilde{O}(\sqrt{n})$	$q = \beta \cdot \tilde{O}(n\sqrt{m})$	Gaussians & use of $\eta_\epsilon(\mathcal{L})$
[GPV08]	$\gamma = \beta \cdot \tilde{O}(\sqrt{n})$	$q = \beta \cdot \tilde{O}(n)$	GPV's sampling algorithm
[MP13]	(subtle)	$q = \beta \cdot n^\epsilon$, $\epsilon > 0$	Use of convolution lemma

Table 3.1: Improvements in approximation factor γ and modulus q throughout the years.

3.4 Learning with errors (LWE)

Origins of LWE. In 2005, Regev[8] first introduced the average-case Learning with Errors (LWE) problem, which has turned out to be an amazingly versatile basis for cryptographic constructions. Also, LWE is a generalization (to a larger moduli and with different noise used) of the well-known Learning Parity with Noise (LPN) problem[9].

LWE & Cryptography. LWE can also be viewed as the "encryption-enabling" analogue of SIS, as these problems are duals of each other, of which one can create various cryptographic primitives within the *minicrypt* realm and the other can do even more, being useful for *cryptomania* primitives.

Particularly, among other things, LWE was used as the basis of IND-CPA and IND-CCA secure public-key encryption schemes, oblivious transfer protocols, identity-based encryption (IBE) schemes, as well as various forms of leakage-resilient encryption (the corresponding references

[8] We refer the reader to the extension [Reg09b] of the original paper, containing details on follow-up works too.
[9] More on this can be found on the survey of Pietrzak in [Pie12].

to these works can be found in Regev's survey [Reg10]). Moreover, of interest are also LWE-homomorphic cryptosystems, like the one created by Brakerski, Gentry and Vaikuntanathan [BGV11], which is sufficiently described in Section 14.8 [Δρα22].

3.4.1 Principal Definitions for LWE

LWE parameters are n, q, χ where n, q are positive integers and χ is an error distribution over $\mathbb{Z}$. Moreover, n and q should be thought of as being similar to those used in SIS. Also, the error distribution χ is commonly chosen to be a discrete Gaussian distribution of width aq for some $a < 1$, which is often called the relative "error rate" (typically taken to be $1/poly(n)$).

Definition 3.9. (LWE Distribution)
Suppose we have a vector $\mathbf{s} \in \mathbb{Z}_q^n$, which we call the *secret*. Then, the *LWE distribution* $A_{\mathbf{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is sampled by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \leftarrow \chi$, and outputting $(\mathbf{a}, b)$, where $b = \langle \mathbf{s}, \mathbf{a} \rangle + e \bmod q$.

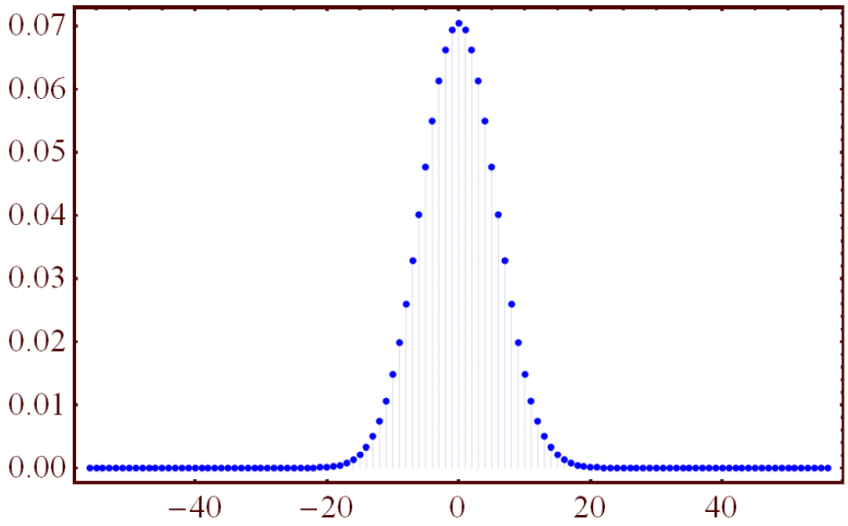


Figure 3.5: The error distribution χ for $q = 113$ and $a = 0.05$ (taken from [Reg10]).

There are two main variations of the LWE problem: search and decision. For those problems, an additional parameter m is needed, which is the number of available samples. Typically, m is taken to be sufficiently large so that the secret is uniquely defined with high probability.

Definition 3.10. (Search-LWE$_{n,q,\chi,m}$)
Given m independent samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ drawn from $A_{\mathbf{s},\chi}$, where $\mathbf{s} \in \mathbb{Z}_q^n$ is chosen uniformly at random (and stays fixed for all samples), find $\mathbf{s}$.

Definition 3.11. (Decision-LWE$_{n,q,\chi,m}$)
Given m independent samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where every sample was either drawn from:

1. $A_{\mathbf{s},\chi}$, for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ (fixed for all samples); or
2. the uniform distribution,

distinguish which is the case (with non-negligible advantage).

Remark 3.7.

(i) Evidently, if we remove the error terms, both problems can be solved trivially. For the search variant, one can recover $\mathbf{s}$ easily through Gaussian elimination, whereas for Decision-LWE, the distributions can be distinguished by observing that, in the uniform case, with high probability no solution $\mathbf{s}$ exists.

(ii) The samples on the above definitions can be combined into a matrix, giving more compact representations of the problems:
- For Search-LWE, the vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$ become the columns of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and the values $b_i \in \mathbb{Z}_q$ combine into a vector $\mathbf{b} \in \mathbb{Z}_q^m$, where[10]
$$\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T \pmod q, \text{ with } \mathbf{e} \leftarrow \chi^m.$$
- For Decision-LWE, we also note that, in the uniform case, $\mathbf{b}$ should be uniformly random and independent of $\mathbf{A}$.

Simple Properties of LWE. We list some properties of LWE that will be useful later. To start with, suppose we are given m LWE samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ $(1 \leq i \leq m)$, then we can

(i) **(Check a candidate solution $\mathbf{s}' \in \mathbb{Z}_q^n$.)** We test whether $b_i - \langle \mathbf{s}', \mathbf{a}_i \rangle$ is "small", $\forall i$. This holds because, if we take a sample $(\mathbf{a}, b)$ and $\mathbf{s}' \neq \mathbf{s}$, then
$$\begin{aligned} b - \langle \mathbf{s}', \mathbf{a} \rangle &= \langle \mathbf{s}, \mathbf{a} \rangle - \langle \mathbf{s}', \mathbf{a} \rangle + e \\ &= \langle \mathbf{s} - \mathbf{s}', \mathbf{a} \rangle + e \end{aligned}$$
will be well-spread in $\mathbb{Z}_q$. Whereas if $\mathbf{s}' = \mathbf{s}$, the value will be small.

(ii) **("Shift" the secret by $\mathbf{t} \in \mathbb{Z}_q^n$.)** Given $(\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a} \rangle + e)$, one can output $\mathbf{a}, b' = b + \langle \mathbf{t}, \mathbf{a} \rangle = \langle \mathbf{s} + \mathbf{t}, \mathbf{a} \rangle + e$.

(iii) **(Have multiple independent secrets.)** If we have r independent secrets $\mathbf{s}_1, \ldots, \mathbf{s}_r$, then $(\mathbf{a}, b_1 = \langle \mathbf{s}_1, \mathbf{a} \rangle, \ldots, b_r = \langle \mathbf{s}_r, \mathbf{a} \rangle)$ is indistinguishable from $(\mathbf{a}, b_1, \ldots, b_r)$ drawn from the uniform distribution. This can be proven via a single hybrid argument, since $\mathbf{a}$ is public.

Connection to Lattices. We now describe how the LWE problem relates to lattices. In short, it can be viewed as an *average-case BDD* problem on a certain family of q-ary m-dimensional integer lattices, namely the row-generated lattices, which we remind are:
$$\begin{aligned} \mathcal{L}_q(\mathbf{A}) &= \{\mathbf{y} \in \mathbb{Z}^m \mid \mathbf{y} = \mathbf{A}^T\mathbf{s} \bmod q, \text{ for } \mathbf{s} \in \mathbb{Z}^n\} \\ &= \mathbf{A}^T\mathbb{Z}^n + q\mathbb{Z}^m. \end{aligned}$$

More precisely, given $\mathbf{A}$ and $\mathbf{b}$ from LWE samples, the vector $\mathbf{b}$ is relatively close to exactly one vector in the "LWE lattice" $\mathcal{L}_q(\mathbf{A}) = \{\mathbf{A}^T\mathbf{s} : \mathbf{s} \in \mathbb{Z}_q^n\} + q\mathbb{Z}^m$, and (in the search variant) we are asked to find that lattice vector. This is also illustrated for two dimensions in the complementary figure (taken from Peikert's presentation in [Uni12]), where the red point denotes the vector $\mathbf{b}$ and the green one in the circle is the vector closer to it.

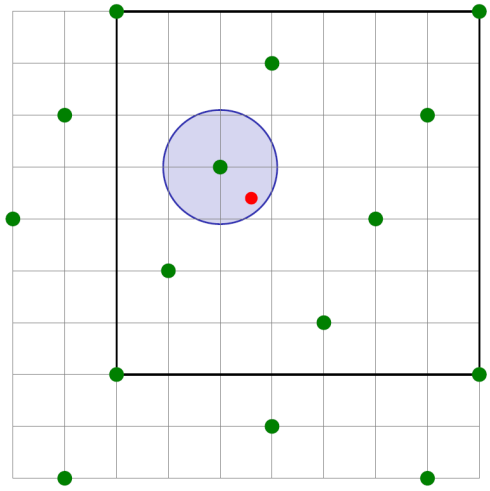


Figure 3.6: The LWE problem in 2D.

For the decision variant, we note that, in the uniform case, $\mathbf{b}$ is far from all points in the lattice $\mathcal{L}_q(\mathbf{A})$ with high probability.

In contrast to the approach described above, it's important to remind that the analysis in Section 3.1 takes a somewhat different route, exploiting the connection between LWE and SIS.

Normal form. In a similar fashion to SIS, the LWE problem also has a Hermite normal form, in which the coordinates of the secret $\mathbf{s}$ are chosen independently, from the error distribution χ (modulo q). This form is particularly useful for cryptographic constructions, as we will see in the next section. Furthermore, a result by [App+09] proves that this new form maintains the hardness of the LWE problem. We present it (in a slightly informal way) in the form of the following lemma:

[10] We follow Peikert's convention to multiply secrets on the right for SIS and on the left for LWE.

Lemma 3.4. The normal form of LWE, in either its search or decision variant, is at least as hard as the same variant for *any* distribution of the secret, up to a small difference in m.

Proof. For a simple but slightly informal proof, we refer the reader to Peikert's LWE presentation in [Uni12] and Section 4.2 of [Pei16]. For the exact lemma and proof, see Lemma 2 of [App+09].

3.4.2 Hardness of LWE

Regev's 2005 paper, in which he introduced LWE, also contained the first worst-case/average-case reductions for LWE. This is summarized in the following theorem (in a bit stronger form).

Theorem 3.2. For any $m = poly(n)$, any modulus $q \leq 2^{poly(n)}$, and any (discretized) Gaussian error distribution χ of parameter $aq \geq 2\sqrt{n}$ (with $0 < a < 1$), solving the Decision-LWE$_{n,q,\chi,m}$ is at least as hard as quantumly solving GapSVP$_\gamma$ and SIVP$_\gamma$ on arbitrary n-dimensional lattices, for some $\gamma = \tilde{O}(n/a)$.

Proof. In order to prove the theorem, one needs to find an efficient quantum reduction that uses an oracle for LWE to solve GapSVP$_\gamma$ and SIVP$_\gamma$ on any n-dimensional lattice.[11] In particular, the theorem is proved in two parts. On the first one, Search-LWE, using a quantum reduction, is proved to be at least as hard as those worst-case lattice problems. While, on the second one Decision-LWE is (classically) proven to be equivalent to Search-LWE (up to some polynomial blowup in the number of samples). This can be seen in the figure below too.

For an overview of the exact proof the reader is referred to Subsection 4.2.2 of [Pei16]. However, for the novice reader we first recommend Regev's (third) presentation in [Uni12], combined with reading [Reg10].

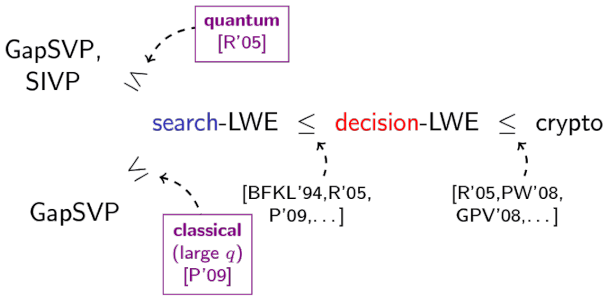


Figure 3.7: Essential results on LWE hardness (from Peikert's first presentation in [Uni12]).

Remark 3.8.

(i) Similarly to SIS, the specific values of both m and q (with the exception of their lower bounds) have little impact on the ultimate hardness guarantee. However, the approximation factor γ deteriorates with the inverse error rate $1/a$ of LWE.

(ii) As no known quantum algorithms for GapSVP$_\gamma$ and SIVP$_\gamma$ perform significantly better than classical ones, the quantum nature of this reduction is meaningful. However, a fully classical reduction would give even further confidence on the hardness of LWE. This was partly given by Peikert, as we discuss below.

Building on the above result, subsequent studies achieved increasingly stronger results on the hardness of LWE relative to worst-case lattice problems. A concise survey of them is presented in [Pei16]. However, we shall only briefly discuss Peikert's classical reduction of [Pei09].

[11] This means that any algorithm (classical or quantum) that solves LWE transforms into a quantum one that solves these hard lattice problems.

More specifically, Peikert managed to classically reduce GapSVP_γ to LWE, for the same $\gamma = \tilde{O}(n/a)$ factor as in the theorem above. However, the reduction only works for GapSVP_γ and not SIVP_γ. Moreover, in order for the reduction to work, an exponentially large modulus $q \geq 2^{n/2}$ is needed, whereas the quantum reduction works for any $q \geq 2\sqrt{n}/a$.

Finally, we also remark that in [Bra+13], a general dimension-modulus tradeoff for LWE was given (influenced by techniques from fully homomorphic encryption), stating that hardness for a particular error rate a is determined almost entirely by $n \log q$ (and not by the n and q chosen), as long as q is bounded from below by some small polynomial. Using this result on Peikert's classical reduction from GapSVP to LWE we have that the same GapSVP problem classically reduces to LWE in dimension n^2 with modulus $q = \text{poly}(n)$.

On Robustness. For our purposes, we simply mention that LWE is considered a very "robust" problem, remaining hard even if an attacker has some extra knowledge on the secret and errors. We refer the reader to the "Robustness" paragraph for LWE in [Pei16] for more.

On Alternative Errors. Lastly, we highlight that there have been works (like [MP13]) where LWE is considered with non-Gaussian and potentially small errors. These distributions are usually simpler to sample and thus more useful for real-world applications. As an example, Kyber does work with errors taken from the binomial distribution.

3.5 LWE Cryptosystems

In this section, we introduce a couple of cryptographic schemes based on LWE. Specifically, we explore some of the first LWE-based public-key encryption schemes, which served as the foundation for subsequent developments. Also, it is important to highlight that the security of these schemes is passive/indistinguishable under chosen-plaintext attack (IND-CPA), a notion which we define formally later. In simple terms, IND-CPA security ensures that an adversary, even with access to the public key and encrypted messages, gains no valuable information. So, the adversary should be able to do no better than if they were guessing randomly.

3.5.1 Regev's LWE Cryptosystem

In Regev's 2005 paper, the first-ever LWE-based public-key cryptosystem was also presented.

*** Regev's LWE Cryptosystem ***

- **Key Generation.**
  - Choose a secret $\mathbf{s} \in \mathbb{Z}_q^n$ uniformly at random.
  - Output m samples $(\bar{\mathbf{a}}_i, b_i) \in \mathbb{Z}_q^{n+1}$, where $b_i = \langle \mathbf{s}, \bar{\mathbf{a}}_i \rangle + e_i$, drawn from the LWE distribution $A_{\mathbf{s},\chi}$.
  - Set $\bar{\mathbf{A}} = [\bar{\mathbf{a}}_1 | \dots | \bar{\mathbf{a}}_m]$ and $\mathbf{b}^T = \mathbf{s}^T \bar{\mathbf{A}} + \mathbf{e}^T \bmod q$, with $\mathbf{e}^T$ a row-vector of errors.
- **Private Key.** The secret $\mathbf{s}$.
- **Public Key.** The matrix $\mathbf{A} = \begin{bmatrix} \bar{\mathbf{A}} \\ \mathbf{b}^T \end{bmatrix} \in \mathbb{Z}_q^{(n+1)\times m}$.
- **Encryption.** To encrypt a bit $\mu \in \mathbb{Z}_2 = \{0, 1\}$, take a random subset-sum of the LWE samples and encode the message bit in the last coordinate. More precisely,
  - Choose a uniformly random $\mathbf{x} \in \{0, 1\}^m$.
  - Output the ciphertext $\mathbf{c} = \mathbf{A}\mathbf{x} + (\mathbf{0}, \mu \cdot \lfloor \frac{q}{2} \rceil) \in \mathbb{Z}_q^{n+1}$.
- **Decryption.** To decrypt the ciphertext $\mathbf{c}$,
  - Compute $(-\mathbf{s}, 1)^T \cdot \mathbf{c} \approx \mu \cdot \lfloor \frac{q}{2} \rceil \pmod q$.
  - Test whether $(-\mathbf{s}, 1)^T \cdot \mathbf{c}$ is closer to 0 or to $\lfloor \frac{q}{2} \rceil \pmod q$.

It is parametrized by an LWE dimension n, a modulus q, an error distribution χ over $\mathbb{Z}$ and the number of samples m. Evidently, the parameters should satisfy some conditions (which we mention later), so as to ensure correct decryption and security.

Remark 3.9.

(a) Note that in the above, the error vector $\mathbf{e}$ is short, and the secret and public key satisfy the relation:
$$(-\mathbf{s}, 1)^T \cdot \mathbf{A} = \mathbf{e}^T \approx \mathbf{0} \,(\mathrm{mod}\, q)$$

(b) The computation in the decryption is done using (a) and the fact that $\mathbf{x}$ is short:
$$\begin{aligned}(-\mathbf{s}, 1)^T \cdot \mathbf{c} &= (-\mathbf{s}, 1)^T \cdot \mathbf{A} \cdot \mathbf{x} + \mu \cdot \left\lfloor \frac{q}{2} \right\rceil \\ &= \mathbf{e}^T \cdot \mathbf{x} + \mu \cdot \left\lfloor \frac{q}{2} \right\rceil \\ &\approx \mu \cdot \left\lfloor \frac{q}{2} \right\rceil \,(\mathrm{mod}\, q) \qquad (\mathbf{e}, \mathbf{x} \in \mathbb{Z}^m \text{ are short})\end{aligned}$$

(c) We remark that, ignoring the $\mu \cdot \lfloor \frac{q}{2} \rceil$ term in the computation of the ciphertext, the encryption is the evaulation of the SIS function $f_{\mathbf{A}}$ on $\mathbf{x}$. However, here the matrix $\mathbf{A}$ is pseudorandom and not uniformly random.

Efficiency. Regev's LWE cryptosystem, on its original variant, can only encrypt one bit at a time, with the public key being $\tilde{O}(n^2)$ bits, and the secret key and ciphertext being $\tilde{O}(n)$ bits.

Correction. In order to be able to decipher a ciphertext, the term $\mathbf{e}^T \cdot \mathbf{x} \in \mathbb{Z}$ should have magnitude less than $q/4$ (if it is larger, the approximation will not hold). Thus, in order to have correctness, q needs to be large enough relative to χ and m.

For instance, if $\chi = D_{\mathbb{Z},r}$ is a discrete Gaussian, which is subgaussian with parameter r, then $\langle \mathbf{e}, \mathbf{x} \rangle$ is subgaussian with parameter at most $r\sqrt{m}$ (note: as $\mathbf{e} \leftarrow D^m_{\mathbb{Z},r}$, it has norm roughly $r\sqrt{m}$, and $x \leftarrow \{0,1\}^m$). Hence, $\langle \mathbf{e}, \mathbf{x} \rangle$ has magnitude less than $r\sqrt{m \ln(1/\epsilon)/\pi}$ with probability at least $1 - 2\epsilon$ (see Definition 3.7, for $X = \langle \mathbf{e}, \mathbf{x} \rangle$, $t = r\sqrt{m \ln(1/\epsilon)/\pi}$ and $s = r\sqrt{m}$).

Therefore, in order to ensure correct decryption with overwhelming probability and security (as we will see next), the values of the parameters can be as small as $r = \Theta(\sqrt{n})$ and $q = \tilde{O}(n)$, which correspond to an LWE problem with error rate $a = r/q = 1/\tilde{O}(\sqrt{n})$ and approximation factor $\gamma = \tilde{O}(n^{3/2})$. Also, m is usually taken to be $(n+1)\log q$.

For a more concrete example, Regev (in [Reg10]) proposes q to be a prime between n and $2n^2$, $m = 1.1 \cdot n \log q$ and $a = 1/(\sqrt{n} \log^2 n)$.

Security. In order to prove the theorem related to the security of the cryptosystem, we first have to describe a useful result for the proof of this and others LWE-cryptosystems:

"With very high probability over the choice of $(\mathbf{a}_i, b_i)$ for $1 \le i \le m$, the distribution (over a set S) of a random subset sum $(\sum_{i \in S} \mathbf{a}_i, \sum_{i \in S} b_i)$ is extremely close to uniform in statistical distance. Alternatively, we have that for uniform and independent $\mathbf{A} \leftarrow \mathbb{Z}_q^{(n+1) \times m}$ and $\mathbf{x} \leftarrow \{0,1\}^m$, $(\mathbf{A}, \mathbf{u} = \mathbf{A}\mathbf{x})$ is statistically indistinguishable from uniformly random, i.e. even a computationally unbounded attacker has only negligible advantage in distinguishing them."

This result follows from an argument based on Fourier analysis (see Section 5 of [NS99]). Alternatively, it follows from a *regularity lemma* (also known as the *Leftover Hash Lemma (LHL)*) for which further details can be found on Section 4 of [IZ89], Section 8.9 of [Sho08] and [Hås+99].

Theorem 3.3. Regev's LWE cryptosystem is IND-CPA secure, assuming that Decision-LWE$_{n,q,\chi,m}$ is hard, which for appropriate parameters is implied by the conjectured worst-case (quantum) hardness of lattice problems.

Due to lack a space, we cannot present a properly detailed proof. However, for an overview of the exact proof, see Subsection 5.2.1 of [Pei16], combined with Piekert's presentation in [Uni12].

Proof. The strategy followed for in this proof is known as "lossiness" argument. Informally, our to main goals is to show that:

1. A public key $\mathbf{A}$ formed properly is indistinguishable from one chosen uniformly at random from $\mathbb{Z}_q^{(n+1)\times m}$.

2. Encrypting under such a uniformly random public key is "lossy", in that it hides the message information-theoretically.

After proving the above using the assumption that LWE is hard and the useful result we presented before, it is trivial to prove the theorem.

1ST VARIANT - NORMAL FORM OPTIMIZATION. As described in [MR09], the cryptosystem can be optimized using the normal forms we defined in the previous sections, and we describe it below. Moreover, the security of this variant can be proven similarly to the prior one, although with two notable changes: (1) we use the assumption that the normal form of LWE is hard; and (2) we use a regularity lemma for matrices of the form $[\mathbf{I}_{n+1}|\mathbf{A}]$, for uniformly random $\mathbf{A}$.

*** Regev's LWE Cryptosystem (Normal form variant) ***

- **Key Generation.**
  - Choose a secret $\mathbf{s} \in \mathbb{Z}^n$, with coordinates chosen from the error distribution χ.
  - Output m samples $(\bar{\mathbf{a}}_i, b_i) \in \mathbb{Z}_q^{n+1}$, where $b_i = \langle \mathbf{s}, \bar{\mathbf{a}}_i \rangle + e_i$, drawn from the LWE distribution $A_{\mathbf{s},\chi}$.
  - We create $\bar{\mathbf{A}}$ and $\bar{\mathbf{b}}^T$ as in the remark below.
- **Private Key.** The secret $\mathbf{s}$.
- **Public Key.** The matrix $\mathbf{A} = \begin{bmatrix} \bar{\mathbf{A}} \\ \bar{\mathbf{b}}^T \end{bmatrix} \in \mathbb{Z}_q^{(n+1)\times(m-n)}$.
- **Encryption.** To encrypt a bit $\mu \in \mathbb{Z}_2 = \{0,1\}$,
  - Choose a uniformly random $\mathbf{x} \in \{0,1\}^{m+1}$.
  - Output the ciphertext $\mathbf{c} = [\mathbf{I}_{n+1}|\mathbf{A}] \cdot \mathbf{x} + (\mathbf{0}, \mu \cdot \lfloor \frac{q}{2} \rceil) \in \mathbb{Z}_q^{n+1}$.
- **Decryption.** To decrypt the ciphertext $\mathbf{c}$,
  - Compute $(-\mathbf{s}, 1)^T \cdot \mathbf{c} \approx \mu \cdot \lfloor \frac{q}{2} \rceil \pmod q$.
  - Test whether $(-\mathbf{s}, 1)^T \cdot \mathbf{c}$ is closer to 0 or to $\lfloor \frac{q}{2} \rceil \pmod q$.

Remark 3.10. Consider an LWE instance $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2]$ and $\mathbf{b}^T = [\mathbf{b}_1^T|\mathbf{b}_2^T]$, where $\mathbf{A}_1 \in \mathbb{Z}_q^{n\times n}$ is invertible and $\mathbf{b}_1 \in \mathbb{Z}_q^n$. Then, in order to transform it to the normal form of LWE, we set $\bar{\mathbf{A}} = -\mathbf{A}_1^{-1} \cdot \mathbf{A}_2$ and $\bar{\mathbf{b}}^T = \mathbf{b}_1^T \bar{\mathbf{A}} + \mathbf{b}_2^T$. It is easy to see that the instance $\bar{\mathbf{A}}, \bar{\mathbf{b}}^T$ comes from the LWE distribution with a secret drawn from the error distribution (see Section 4.2 of [Pei16]).

2ND VARIANT - LONGER MESSAGES. Another useful optimization is one that enables encryption of several bits at a time. A major leap in this direction was made in [PVW08] using an *amortization* technique. With their variant, one can encrypt $l = O(n)$ bits every time, without asymptotically increasing the runtime of the encryption, and the size of the public key and ciphertext (remember also the "simple" property (iii) of LWE). However, there is an increase in the size of the secret key and the runtime of the decryption (from $\tilde{O}(n)$ before, to $\tilde{O}(l \cdot n)$ now).

Regarding the proof of security, it uses a hybrid argument along with the LWE hardness asumption, as well as the lossiness argument described before, which still applies without much effort.

*** Regev's LWE Cryptosystem (Longer messages variant) ***

- **Key Generation.**
  - Choose l independent secrets $\mathbf{s}_j \in \mathbb{Z}_q^n$ $(1 \leq j \leq l)$ uniformly at random, creating a matrix $\mathbf{S}^T \in \mathbb{Z}_q^{l \times n}$ whose rows are independent LWE secrets.
  - For $1 \leq j \leq l$ (independently for each j):
    Output m samples $(\bar{\mathbf{a}}_i, b_{j,i}) \in \mathbb{Z}_q^{n+1}$, where $b_{j,i} = \langle \mathbf{s}_j, \bar{\mathbf{a}}_i \rangle + e_{j,i}$, drawn from the LWE distribution $A_{\mathbf{s}_j,\chi}$.
  - Set $\bar{\mathbf{A}} = [\bar{\mathbf{a}}_1 | \dots | \bar{\mathbf{a}}_m]$ and $\mathbf{B} = \mathbf{S}^T \bar{\mathbf{A}} + \mathbf{E}^T \bmod q$, where $\mathbf{E}^T = \{e_{j,i}\}_{1 \leq j \leq l, 1 \leq i \leq m}$.
- **Private Key.** The secret $\mathbf{s}$.
- **Public Key.** The matrix $\mathbf{A} = \begin{bmatrix} \bar{\mathbf{A}} \\ \mathbf{B} \end{bmatrix} \in \mathbb{Z}_q^{(n+l) \times m}$.
- **Encryption.** To encrypt a message $\mathbf{m} \in \{0,1\}^l$,
  - Choose a uniformly random $\mathbf{x} \in \{0,1\}^m$.
  - Output the ciphertext $\mathbf{c} = \mathbf{A}\mathbf{x} + (\mathbf{0}, \mathbf{m} \cdot \lfloor \frac{q}{2} \rceil) \in \mathbb{Z}_q^{n+l}$.
- **Decryption.** To decrypt the ciphertext $\mathbf{c}$,
  - Compute $[-\mathbf{S}^T | \mathbf{I}_l] \cdot \mathbf{c} \approx \mathbf{m} \cdot \lfloor \frac{q}{2} \rceil \, (\bmod q)$.
  - Test whether $[-\mathbf{S}^T | \mathbf{I}_l] \cdot \mathbf{c}$ is closer to 0 or to $\lfloor \frac{q}{2} \rceil \, (\bmod q)$.

3.5.2 Dual LWE Cryptosystem

In the previous cryptosystem and its variants, public keys were created with a unique secret key and the same ciphertext could be created in many different ways (different errors could give the same ciphertext for the same keys). However, the "dual" of this, i.e. a cryptosystem in which public keys can be created by many possible secret keys and particular ciphertexts are produced by a unique encryption randomness, can also prove useful.

Gentry et al [GPV08] created the first "dual" cryptosystem to Regev's (in this sense),for which we only provide a detailed description of it, its security and briefly discuss on its variants, as its correctness and efficiency can be calculated similarly to Regev's cryptosystem.

*** Dual LWE Cryptosystem ***

- **Key Generation.**
  - Choose a uniformly random $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$ (for a sufficiently large $m \approx n \log q$).
  - Choose an $\mathbf{x} \in \{0,1\}^m$ uniformly at random.
- **Private Key.** The random $\mathbf{x}$.
- **Public Key.** The matrix $\mathbf{A} = [\bar{\mathbf{A}} | \mathbf{u} = \bar{\mathbf{A}}\mathbf{x}] \in \mathbb{Z}_q^{n \times (m+1)}$.
- **Encryption.** To encrypt a bit $\mu \in \{0,1\}$,
  - Choose a LWE secret $\mathbf{s} \in \mathbb{Z}_q^n$.
  - Output the ciphertext $\mathbf{c}^T \approx \mathbf{s}^T \mathbf{A} + (\mathbf{0}, \mu \cdot \lfloor \frac{q}{2} \rceil)^T$, where the approximations hide independent errors drawn from the LWE error distribution χ.
- **Decryption.** To decrypt the ciphertext $\mathbf{c}^T$,
  - Compute $\mathbf{c}^T \cdot (-\mathbf{x}, 1) \approx \mu \cdot \lfloor \frac{q}{2} \rceil \, (\bmod q)$.
  - Test whether $\mathbf{c}^T \cdot (-\mathbf{x}, 1)$ is closer to 0 or to $\lfloor \frac{q}{2} \rceil \, (\bmod q)$.

Remark 3.11.

(a) Note that the secret and public key satisfy the relation: $\mathbf{A} \cdot (-\mathbf{x}, 1) = \mathbf{0} \bmod q$.

(b) Remark that in order for someone to find two distinct valid secret keys that give the same public key (for the same $\bar{\mathbf{A}}$), they would have to solve the SIS problem.

Security. This cryptosystem also is IND-CPA secure, assuming the hardness of LWE:

Theorem 3.4. The Dual LWE cryptosystem is IND-CPA secure, assuming that Decision-LWE$_{n,q,\chi,m+1}$ is hard, which for appropriate parameters is implied by the conjectured worst-case (quantum) hardness of lattice problems.

Proof. The proof is similar to the one for Regev's scheme. For more, see Subsection 5.2.2 of [Pei16], combined with Piekert's presentation in [Uni12].

VARIANTS. We only mention two notable variants:

1. Similarly to Regev's scheme, there is a version of the dual system that can encrypt l bits at a time. For this, the secret key is a matrix $\mathbf{X} \in \{0,1\}^{m \times l}$ uniformly at random, and the public key is $\mathbf{A} = [\bar{\mathbf{A}} | \mathbf{U} = \bar{\mathbf{A}}\mathbf{X}] \in \mathbb{Z}_q^{n \times (m+l)}$. For encryption and decryption, one makes the necessary tweaks as in Regev's cryptosystem.
2. As in Regev's normal form variant, here too the entries of the secret key do not have to be binary nor uniform, and can be chosen from any other distribution for which the public key is statistically indistinguishable from uniform.

3.5.3 Compact LWE Cryptosystem

Finally, we present a cryptosystem which manages to be more compact than the previous ones by a factor of $\log q$, as can be seen in the table below (remember we took $m \approx n \log q$):

Cryptosystem	Matrix $\bar{\mathbf{A}}$ size (# of elements of $\mathbb{Z}_q$)	Secret key size (# of elements)	Ciphertext size (# of elements)
Regev's	$n \times (n \log q)$	n	$n \log q$
Dual	$n \times (n \log q)$	$n \log q$	n
Compact	$n \times n$	n	n

Table 3.2: Comparing the (approximate) memory requirements of LWE cryptosystems.

It was presented in [LP11] and, aside from its smaller keys and ciphertexts, it also has another important (and distinct) property: the secret key and encryption randomness are both unique.

*** Compact LWE Cryptosystem ***

- **Key Generation.**
  - Choose a uniformly random $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times n}$ (square matrix!).
  - Choose an $\mathbf{s} \in \mathbb{Z}^n$, with coordinates chosen independently from the error distribution χ.
- **Private Key.** The secret $\mathbf{s}$.
- **Public Key.** The matrix $\mathbf{A} = [\bar{\mathbf{A}} | \mathbf{u} = \bar{\mathbf{A}}\mathbf{x}] \in \mathbb{Z}_q^{n \times (m+1)}$.
- **Encryption.** To encrypt a bit $\mu \in \{0,1\}$,
  - Choose an $\mathbf{r} \in \mathbb{Z}^n$, with coordinates chosen independently from the error distribution χ.
  - Output the ciphertext $\mathbf{c} \approx \mathbf{A}\mathbf{r} + (\mathbf{0}, \mu \cdot \lfloor \frac{q}{2} \rceil) \in \mathbb{Z}_q^{n+1}$, where the approximations hide independent errors drawn from the LWE error distribution χ.
- **Decryption.** To decrypt the ciphertext $\mathbf{c}^T$,
  - Compute $(-\mathbf{s}, 1)^T \cdot \mathbf{c} \approx \mu \cdot \lfloor \frac{q}{2} \rceil \pmod q$.
  - Test whether $(-\mathbf{s}, 1)^T \cdot \mathbf{c}$ is closer to 0 or to $\lfloor \frac{q}{2} \rceil \pmod q$.

Remark 3.12. Note that the secret and public key satisfy the relation: $(-\mathbf{s}, 1)^T \cdot \mathbf{A} \approx \mathbf{0} \bmod q$.

SECURITY. Regarding security, similarly to other schemes we have:

Theorem 3.5. The Compact LWE cryptosystem is IND-CPA secure, assuming that Decision-LWE$_{n,q,\chi,m}$ is hard, which for appropriate parameters is implied by the conjectured worst-case (quantum) hardness of lattice problems.

Proof. This security proof has some differences compared to the previous ones, mainly that no regularity lemma is required and only the use of the normal form LWE assumption is needed, two times. For a more detailed explanation, see Subsection 5.2.3 of [Pei16], combined with Piekert's presentation in [Uni12].

Remark 3.13. We remark that, as in this scheme the coordinates of both $\mathbf{r}$ and $\mathbf{s}$ are drawn from χ, they need to have magnitude on the order of $\sqrt{n}$ in order to establish proper worst-case guarantees (as described in LWE's hardness theorem). Therefore, this makes the accumulated error larger (by a $\sqrt{n}$ factor) which in turn means that q should be larger and thus a should be smaller too. Hence, the smaller a makes the approximation factor worse ($\gamma = \tilde{O}(n^2)$, instead of $\gamma = \tilde{O}(n^{3/2})$). However, in practice, after adjusting for equivalent estimated hardness against concrete attacks, the compact scheme continues to have smaller key and ciphertext sizes.

VARIANTS. As in the previous cryptosystems, there exists an optimization that can encrypt l bits at a time. For this, the secret key is $\mathbf{S} \in \mathbb{Z}^{n \times l}$, and the public key is $[\bar{\mathbf{A}}|B^T \approx S^T\bar{\mathbf{A}}]^T$.

Chapter 4

Ring-LWE Theory & Applications

As discussed previously, cryptographic schemes based on the SIS and LWE problems have very large key sizes, which is why there has been research towards reducing them. One natural approach towards this goal is to assume that there is some structure in the LWE (or SIS) samples. The first cryptosystem using this idea was the NTRU cryptosystem [HPS98], which is most usefully interpreted in terms of algebraically structured lattices. Inspired by the design of this scheme, Micciancio [Mic07][1] modified SIS one-way/collision resistant function from [Ajt96], improving on its efficiency (from $\tilde{O}(n^2)$ key sizes and runtime to $\tilde{O}(n)$).

Particularly, one kind of "structure" that has proved really useful is the following:
Suppose n is a power of two and that $\mathbf{a}_1 = (a_1, \ldots, a_n)$ is chosen uniformly at random. The $\mathbf{a}$ vectors (related to SIS/LWE) are created in blocks of n samples $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{Z}_q^n$, where the remaining vectors are given by $\mathbf{a}_i = (a_i, \ldots, a_n, -a_1, \ldots, -a_{i-1})$, for $2 \leq i \leq n$. In this way, representing these vectors only requires $O(n)$ elements of $\mathbb{Z}_q$. Moreover, this structure allows the use of the fast Fourier transform (FFT), which leads to considerably faster cryptographic schemes.

In mathematical terms, the above means replacing the group $\mathbb{Z}_q^n$ with the ring $\mathbb{Z}_q[X]/\langle X^n+1\rangle$, where X^n+1 is irreducible (over the rationals), as n is a power of two[2]. This is why, the variants of SIS and LWE that use rings are called *Ring-SIS* and *Ring-LWE*, accordingly. Finally, we remark that it is possible to use other rings too, provided that they satisfy some requirements. For our purposes though, this ring is the most important one as KYBER, which we explore on the last part of this thesis, is "based on" that for $n = 256$ and $q = 3329$.

In this chapter, we delve deeper into these structured lattices, the new problem Ring-LWE (and its relation to them), and present a cryptosystem that is based on this problem, before ending the chapter with the introduction of a more generalized form of the problem, M-LWE.

4.1 Ideal Lattices

Analogically to the way SIS and LWE base their hardness on worst-case problems on lattices, Ring-SIS and Ring-LWE rely on the algebraic structured lattices we mentioned earlier, which are called *ideal lattices*. We discuss more about them and concepts related to them in this section. More background can be found in Milner's notes [Mil20] and the introductory book of [Ste04].

Moreover, we research the underlying rings too, as the algebraic and geometric properties of each chosen ring contribute a lot to the security properties of Ring-SIS and Ring-LWE.

[1] A preliminary version of this paper was presented in FOCS 2002.

[2] It is important that this is satisfied in order for everything to work properly (see Section 3.4 and the "Collision resistance" paragraph in Section 4.3 of [Pei16] for more).

4.1.1 Number Fields

We start presenting the necessary preliminaries with some definitions on number fields:

Definition 4.1. Let $f(X) \in \mathbb{Q}[X]$ be an irreducible, monic (without loss of generality) polynomial of degree n. We define a *number field* as a field extension $K = \mathbb{Q}(\zeta)$, created by adjoining an abstract element $\zeta \notin \mathbb{Q}$ such that $f(\zeta) = 0$, to the field of rationals.

- The *degree of the number field* is the degree of the polynomial, n.
- We call the polynomial f the *minimal polynomial* of ζ.
- As $f(\zeta) = 0$, the number field K can be seen as an n-dimensional vector space over $\mathbb{Q}$ with basis $\{1, \zeta, \ldots, \zeta^{n-1}\}$, which is called the *power basis* of K.

Remark 4.1. Associating ζ with an indeterminate X gives us an isomorphism between K and $\mathbb{Q}[X]/\langle f(X)\rangle$.

Definition 4.2. Let m be a positive integer, we define the m-th *cyclotomic polynomial* $\Phi_m(X)$ as the polynomial of degree $n = \phi(m)$ (ϕ is Euler's phi-function), whose roots are all the primitive m-th roots of unity[3] $\omega_m^i \in \mathbb{C}$, where $\omega_m = e^{2\pi\sqrt{-1}/m}$ and $1 \leq i < m$ with i coprime to m.
In other words, $\Phi_m(X) = \prod_{i \in \mathbb{Z}_m^*}(X - \omega_m^i) \in \mathbb{Z}[X]$.

Example 4.1. Let $m \geq 2$ be a power of 2. Then, $\Phi_m(X) = X^n + 1$ with $n = \phi(m) = m/2$.

Example 4.2. Let m be a positive integer and let $\zeta = \zeta_m$ denote a primitive m-th root of unity. Then, the m-th cyclotomic number field is $K = \mathbb{Q}(\zeta)$, where the minimal polynomial of ζ is the m-th cyclotomic polynomial $\Phi_m(X)$.

4.1.2 Ring of Integers and Ideals

Having concluded our exploration of concepts related to number fields, we now shift our focus to a version of "integers" inside number fields, and several concepts related to this:

Definition 4.3. An element whose minimal polynomial over the rationals has integer coefficients is called an *algebraic integer* (see Lemma 19 of [Orr19]). Moreover, for a number field K, we denote with $\mathcal{O}_K \subset K$ the set of all algebraic integers in K. The ring that is formed by $\mathcal{O}_K$ with the usual addition and multiplication in K is called the *ring of integers* of K.

The set $\mathcal{O}_K$ is a free $\mathbb{Z}$-module of rank n, where n is the degree of the number field K.[4] In other words, $\mathcal{O}_K$ is the set of all $\mathbb{Z}$-linear combinations of some basis $B = \{b_1, \ldots, b_n\} \subset \mathcal{O}_K$. Such a basis is called an *integral basis* and there are infinite such bases when $n > 1$.

For example, for the m-th cyclotomic number field $K = \mathbb{Q}(\zeta_m)$, its power basis $\{1, \zeta_m, \ldots, \zeta_m^{n-1}\}$ is also an integral basis, i.e. $\mathcal{O}_K = \mathbb{Z}[\zeta_m] = \{\sum_{i=0}^{n-1} m_i \zeta_m^i | m_i \in \mathbb{Z}\}$ (proof in Prop. 6.2 of [Mil20]).

We now define classes of ideals related to $\mathcal{O}_K$, getting one step closer to ideal lattices:

Definition 4.4.

1. An additive subgroup $\mathcal{I} \subseteq \mathcal{O}_K$ that is nontrivial ($\neq \{0\}, \emptyset$) and closed under multiplication by $\mathcal{O}_K$ ($r \cdot x \in \mathcal{I}$, for $r \in \mathcal{O}_K$ and $x \in \mathcal{I}$) is called an *(integral) ideal*.
2. Let $\mathcal{I}, \mathcal{J}$ be two ideals. Then, the sum $\mathcal{I} + \mathcal{J}$ is the set $x + y$, for $x \in \mathcal{I}$ and $y \in \mathcal{J}$, and the product $\mathcal{I}\mathcal{J}$ is the set of all finite sums of terms xy, for $x \in \mathcal{I}$ and $y \in \mathcal{J}$.
3. Let $\mathcal{I}, \mathcal{J} \subseteq \mathcal{O}_K$ be two ideals. If $\mathcal{I} + \mathcal{J} = \mathcal{O}_K$, then these ideals are called *coprime* (or *relatively prime*). Moreover, an ideal $\mathfrak{p} \subsetneq \mathcal{O}_K$ is *prime* if, for any $a, b \in \mathcal{O}_K$, we have $ab \in \mathfrak{p} \Rightarrow a \in \mathfrak{p}$ or $b \in \mathfrak{p}$ or both.

[3] An element a being a primitive m-th root of unity means $a^m = 1$ and $a^k \neq 1$, for $k = 1, \ldots, m-1$.

[4] We (informally) remind that a *module* is a generalization of the notion of vector space in which the field of coefficients is replaced by a ring. Moreover, a *free module* is a module that has a basis (there exist non-free modules if the ring of coefficients is not a division ring).

4. A *fractional ideal* $\mathcal{I} \subset K$ is a set such that $d\mathcal{I} \subseteq \mathcal{O}_K$ is an integral ideal for some $d \in \mathcal{O}_K$.

Remark 4.2.

(i) An ideal $\mathcal{I} \subseteq \mathcal{O}_K$ is finitely generated as the set of all $\mathcal{O}_K$-linear combinations of some $g_1, g_2, \ldots \in \mathcal{O}_K$, symbolized by $I = \langle g_1, g_2, \ldots \rangle$. Moreover, it is also a free $\mathbb{Z}$-module of rank n or, in other words, it is generated as the set of all $\mathbb{Z}$-linear combinations of some basis $\{u_1, \ldots, u_n\} \subset \mathcal{O}_K$.

(ii) Every ideal $\mathcal{I} \subseteq \mathcal{O}_K$ can be expressed as a product of powers of prime ideals in a unique way, i.e. the ring of integers has *unique factorization of ideals*.

4.1.3 Embeddings

Having seen some definitions from the algebraic aspect, we now present the concept of *embeddings* of a number field , which induce a geometry on it.

Definition 4.5. For a number field K, an *embedding* of K is a field homomorphism $\sigma : K \to \mathbb{C}$.

Remark 4.3. Every homomorphism of fields is injective and also, for an embedding $\sigma : K \to \mathbb{C}$, we have $\sigma(a) = a$, $\forall a \in \mathbb{Q}$. Thus, there is exactly one embedding for $K = \mathbb{Q}$.

Extending the above notation, if we have a field homomorphism $\sigma : K \to \mathbb{C}$, then this induces an injective ring homomorphism $K[X] \to \mathbb{C}[X]$ (also symbolized by σ), defined by

$$\sigma(a_0 + a_1 X + \cdots + a_n X^n) = \sigma(a_0) + \sigma(a_1)X + \cdots + \sigma(a_n)X^n.$$

Using this, we can state that a number field $K = \mathbb{Q}(\zeta)$ of degree n has exactly n ring embeddings (i.e. injective ring homomorphisms) $\sigma_i : K \to \mathbb{C}$, which map ζ to each of the complex roots of its minimal polynomial f. For a proof, see Proposition 13 of [Orr19].

Definition 4.6. Let $\sigma : K \to \mathbb{C}$ be an embedding of a number field. If its image lies in $\mathbb{R}$, it is called a *real embedding*. Otherwise, i.e. if $\sigma(K) \not\subset \mathbb{R}$, then it is called a *complex embedding*. Those two cases correspond to a real and complex root of f, respectively.

Remark 4.4.

(i) As the complex roots of f come in conjugate pairs, so do too the complex embeddings. In other words, if σ is a complex embedding of K, then $\bar{\sigma}$ such that $\bar{\sigma}(a) = \overline{\sigma(a)}$ is also a complex embedding of K.

(ii) We denote the number of real embeddings with s_1 and that of *pairs* of complex embeddings with s_2. Thus, $n = s_1 + 2s_2$. Moreover, we denote with $\{\sigma_j\}_{j \in \{1,\ldots,s_1\}}$ the real embeddings and order the complex embeddings in such a way that $\sigma_{s_1+s_2+j} = \overline{\sigma_{s_1+j}}$ for $j \in \{1, \ldots, s_2\}$.

Lastly, we define the most useful embedding (for our purposes):

Definition 4.7. Let K be a number field. The *canonical embedding* is a ring homomorphism $\sigma : K \to \mathbb{R}^{s_1} \times \mathbb{R}^{2s_2}$, defined as $\sigma(X) = (\sigma_1(X), \ldots, \sigma_n(X))$, where multiplication and addition in the latter are both component-wise.

The space H. When working with ideal lattices and the canonical embedding, it is convenient to work with the space $H \subseteq \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ for some numbers $s_1 + 2s_2 = n$, defined as:

$$H = \left\{(x_1, \ldots, x_n) \in \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2} \,:\, x_{s_1+s_2+j} = \overline{x_{s_1+j}},\ \forall j \in \{1, \ldots, s_2\}\right\} \subseteq \mathbb{C}^n.$$

It can be proven that H (with the inner product induced on it by $\mathbb{C}^n$) is isomorphic to $\mathbb{R}^n$ as an inner product space. Furthermore, it is evident that, due to the pairing of the complex embeddings, the canonical embedding maps to H, whose usefulness (in combination with the canonical embedding) will become clear in the next paragraphs.

Additionally, we equip the space with the l_p norm induced on it from $\mathbb{C}^n$. For any $p \in [1, \infty]$, this norm is equal within a factor of $\sqrt{2}$ to $(\sum_{i=1}^n |a_i|^p)^{1/p}$, which is the l_p norm induced on H

from the isomorphism with $\mathbb{R}^n$. Moreover, for the l_2 norm, which is the one that we care about in this thesis, we have equality. Thus, this near equivalence between H and $\mathbb{R}^n$ allows us to use known definitions and results for lattices in this setting, as we see in the following.

Canonical Embedding - Norms and Distributions. In order to use the Euclidean (and other) norms on K, we identify elements of K with their canonical embedding in H. Thus, for an element $x \in K$, the l_2 norm of x is $||x|| = ||\sigma(x)|| = (\sum_{i=1}^{n} |\sigma_i(x)|^2)^{1/2}$.

Furthermore, we can use the canonical embedding to think of the Gaussian distribution over H as a distribution over K, a concept we will need for Ring-LWE. One thing to keep in mind is that, in contrast to LWE where we use the (centered) Gaussian for the error distribution, in Ring-LWE the error is an n-dimensional Gaussian. Thus, we would need an $n \times n$ covariance matrix in order to specify such a distribution.

However, fortunately, the exact error distributions for which the Ring-LWE is defined are always diagonal, in the canonical embedding. This means that, when viewed in the canonical embedding, the error distributions are product distributions with each component being a one-dimensional (centered) normal distribution. Thus, it can be defined using only n parameters. When these parameters are equal, we call the distribution *spherical*.

Example 4.3. We again use the m-th cyclotomic field as an example for the above notions, and specifically this time we set $\zeta = \zeta_m$ for $m \geq 3$. Then, we have $2s_2 = n = \phi(m)$ complex embeddings (only complex roots), given by $\sigma_i(\zeta) = \zeta^i$ for $i \in \mathbb{Z}_m^*$ (in this case it is useful to index the embeddings this way). Moreover, for some power $\zeta^j \in K$, we have that all $\sigma_i(\zeta^j) \in \mathbb{C}$ are roots of unity and thus have magnitude 1, giving us $||\zeta^j|| = \sqrt{n}$.

4.1.4 Trace and Norm of a Field

Some extra tools that we need are the *(field) trace* $Tr \doteq Tr_{K/\mathbb{Q}} : K \to \mathbb{Q}$ and the (field) norm $N \doteq N_{K/\mathbb{Q}} : K \to \mathbb{Q}$ for an element $x \in K$, where K is a number field. These functions will be helpful in transforming questions about elements of K into questions about rational numbers.

More precisely, as K is a $\mathbb{Q}$-vector space, we can define for any $a \in K$ the multiplication by a as a $\mathbb{Q}$-linear map $m_{K,a} : K \to K$ such that $m_{K,a}(\beta) = a\beta$. Then, the *trace* of a, $Tr_{K/\mathbb{Q}}(a)$, is the trace of the linear map $m_{K,a}$ and the *norm* of a, $N_{K/\mathbb{Q}}(a)$, is the determinant.

Moreover, it can be shown (see Lemma 18 of [Orr19]) that there is a connection between the trace and norm, and the embeddings of K:

$$Tr(X) \doteq Tr_{K/\mathbb{Q}}(X) = \sum_{i=1}^{n} \sigma_i(X) \text{ and } N(X) \doteq N_{K/\mathbb{Q}}(X) = \prod_{i=1}^{n} \sigma_i(X)$$

Remark 4.5. We can show that the trace and norm are additive and multiplicative, respectively (see Lemma 14 of [Orr19]). Furthermore, we can prove that, for $x, y \in K$:

$$Tr(x \cdot y) = \sum_{i=1}^{n} \sigma_i(x) \cdot \sigma_i(y) = \langle \sigma(x), \overline{\sigma(y)} \rangle.$$

Example 4.4. Let m be the prime number 5, and $\zeta = \zeta_5$ a root of the cyclotomic polynomial $\Phi_5(X) = \prod_{i \in \mathbb{Z}_5^*}(X - \zeta^i)$. We remind that for m prime, we have $\Phi_m(X) = (x^m - 1)/(x - 1) = x^{m-1} + \ldots + 1$ and thus $\Phi_5(X) = x^4 + x^3 + x^2 + x + 1$.

Now consider the element $a = \frac{3}{4} - \zeta \in K \doteq \mathbb{Q}(\zeta)$. We remark that as ζ is a root of Φ_5, we have $1 + \zeta + \zeta^2 + \zeta^3 + \zeta^4 = 0$. Thus,

$$Tr(a) = \sum_{i=1}^{n} \sigma_i(a) = \sum_{i=1}^{n} \left(\frac{3}{4} - \zeta^i \right) = 3 - (\zeta + \zeta^2 + \zeta^3 + \zeta^4) = 3 - (-1) = 4$$

$$N(a) = \prod_{i=1}^{n} \sigma_i(a) = \prod_{i=1}^{n} \left(\frac{3}{4} - \zeta^i \right) = \Phi_5 \left(\frac{3}{4} \right) = \frac{781}{256}.$$

Moreover, the notion of norms can be used for ideals too:

Definition 4.8. The *norm of an ideal* $\mathcal{I}$ is its index as an additive subgroup of $\mathcal{O}_K$. In other words, $N(\mathcal{I}) = |\mathcal{O}_K/\mathcal{I}|$.

We observe that the concept of norm for ideals is extended to the field norm defined earlier, in the sense that $N(\langle x \rangle) = |N(x)|$ for $x \in \mathcal{O}_K$, and $N(\mathcal{I}\mathcal{J}) = N(\mathcal{I})N(\mathcal{J})$.

Moreover, we can define the norm of a fractional ideal $\mathcal{I}$ too as $N(\mathcal{I}) = N(d\mathcal{I})/|N(d)|$. Thus, the set of fractional ideals forms a group under multiplication and the norm is a multiplicative homomorphism on this group.

Finally, with the definition of a norm of an ideal in hand, we can give another definition for the prime ideal. More precisely, an ideal $\mathfrak{p} \subsetneq \mathcal{O}_K$ is prime if and only if it is *maximal*, i.e. if the only proper superideal of $\mathfrak{p}$ is $\mathcal{O}_K$ itself. This implies that the quotient ring $\mathcal{O}_K/\mathfrak{p}$ is the finite field of order $N(\mathfrak{p})$.

4.1.5 Ideal Lattices and Ideal Lattice Problems

Having established the mathematical background related to ideal lattices, we proceed to introduce them, along with some associated results. Additionally, we discuss some interesting characteristics of hard lattice problems in the context of ideal lattices.

Recall that a lattice is an additive subgroup of $\mathbb{Z}^n$. An ideal $I \subseteq R$ is also an additive subgroup (of a ring R) with the additional property of being closed under multiplication by any ring element. Thus, in simple terms, an *ideal lattice* is a lattice corresponding to an ideal $\mathcal{I}$ in a ring R, under some fixed choice of geometric embedding (e.g. the canonical embedding). The multiplicative closure of $\mathcal{I}$ means that ideal lattices have geometric symmetries that lattices do not have generally. This is where the "extra structure" comes from in ideal lattices, and it can be useful (e.g. for efficiency) or dangerous, if it can be utilised by adversaries to solve a hard problem in ideal lattices.

In more precise terms, we first fix an underlying ring R and then consider its ideals. We also fix an additive isomorphism σ mapping the ring R to some lattice $\sigma(R)$ in $\mathbb{R}^n$ (remember also the connection between H and $\mathbb{R}^n$). Then, the *family of ideal lattices* (for the ring R and embedding σ) is the set of all lattices $\sigma(\mathcal{I})$, for ideals $\mathcal{I}$ of the ring R.

We note that although we have only mentioned the canonical embedding until now, prior to works like [LPR13b] it was not widely used in lattice research. In its place was the naive "coefficient embedding" which maps any element of R to the integer vector in $\mathbb{Z}^n$, whose coordinates are exactly the coefficients of that element when viewed as a polynomial residue. However, due to its advantages (more on this in [LPR13b]), the canonical embedding became the preferred choice in many theoretical works.

Example 4.5. Let $K = \mathbb{Q}(\zeta_m)$ be the m-th cyclotomic number field (can also be represented as $\mathbb{Q}[X]/\langle \Phi_m(X) \rangle$). Then, we choose the ring of (algebraic) integers $\mathcal{O}_K$ of this field as our underlying ring, which as we have seen is equal to $\mathbb{Z}[\zeta_m]$ (can also be represented as $\mathbb{Z}[X]/\langle \Phi_m(X) \rangle$).

An ideal for this ring is an additive subgroup with closure under multiplication by X. Moreover, if we assume that m is a power of 2, then $\Phi_m(X) = X^n + 1$ and the ring is $\mathbb{Z}[X]/\langle X^n + 1 \rangle$. When working in this ring together with the coefficient embedding (that maps X^i to the unit vector e_i) we obtain the family of *anti-cyclic integer lattices* which we introduced at the start of the section. Moreover, we remark that the corresponding lattices over $\mathbb{Z}[X]/\langle X^n - 1 \rangle$ are often called *cyclic*.

For this particular ring $\mathbb{Z}[X]/\langle X^n + 1 \rangle$ the coefficient and the canonical embedding yield the same geometry with which we can define norms and inner products over an ideal $\mathcal{I}$. However, as we have mentioned, for more general rings of integers over number fields, this is not true, and the canonical embedding is most commonly used.

Continuing, we also discuss on how (fractional) ideals in a number field K can also yield lattices, under the canonical embedding, and describe some related results.

Let $\mathcal{I}$ be a fractional ideal with a $\mathbb{Z}$-basis $U = \{u_1, \ldots, u_n\}$. Under the canonical embedding σ, this ideal gives us an ideal lattice $\sigma(\mathcal{I})$ of rank n with a basis $\{\sigma(u_1), \ldots, \sigma(u_n)\} \subset H$. It is convenient to identify an ideal with its embedded lattice and this speaking of, for example, the minimum distance $\lambda_1(\mathcal{I})$ of an ideal (instead of $\lambda_1(\sigma(\mathcal{I}))$, etc.

Discriminant of a number field. The last tool we need is the notion of the discriminant of a number field, defined as:

Definition 4.9. Let K be a number field, then the *(absolute) discriminant* Δ_K is defined to be the square of the fundamental volume of $\sigma(\mathcal{O}_K)$, the embedded ring of integers. Equivalently, $\Delta_K = |\det(Tr(b_i \cdot b_j))|$, where $b_1, \ldots, b_n$ is an integral basis of $\mathcal{O}_K$.

Therefore, due to the above, the fundamental volume of an ideal lattice $\sigma(\mathcal{I})$ is $N(\mathcal{I})\sqrt{\Delta_K}$.

Additionally, we present the following important lemma which bounds the minimum distance of an ideal lattice. This will be very important in understanding how the hardness of some problems is affected in the ideal lattices setting.

Lemma 4.1. Let $\mathcal{I}$ be a fractional ideal in a number field K of degree n, then we have

$$n^{1/2}N(\mathcal{I})^{1/n} \leq \lambda_1(\mathcal{I}) \leq n^{1/2}N(\mathcal{I})^{1/n}\sqrt{\Delta_k^{1/n}}.$$

Proof. For a detailed proof (for any l_p norm) see [PR07].

For a detailed example of an ideal lattice, an illustration and detailed quantities, we refer the reader to Section 5.7 of [PR07], as due to lack of space, we cannot include one here.

Hard problems in ideal lattices. Viewing ideals as lattices allows us to extend lattice problems to ideals, thus defining problems Ideal-SVP$_\gamma$, Ideal-SIVP$_\gamma$, Ideal-GapSVP$_\gamma$, etc., as the corresponding problems restricted to ideal lattices. Here we only mention some useful facts about their hardness, and for more, refer the reader to [PR07], [Vai20] and their references.
For typical choices of rings used in cryptography (e.g. $R = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k, k \in \mathbb{N}$):

- The problems Ideal-SVP$_\gamma$ and Ideal-SIVP$_\gamma$ are equivalent, because the symmetries allow one short nonzero vector to be converted into n linearly independent ones of the same length (see Lyubashevsky's third presentation on [Uni12] for an exact example).
- The decision problem Ideal-GapSVP$_\gamma$ for small $\gamma = poly(n)$ (for $\gamma > \sqrt{n}$ in R) is actually easy on ideal lattices (see Lemma 4.1).

Until very recently, our best algorithms for Ideal-SVP$_\gamma$ were essentially no better than the generic ones for SVP$_\gamma$ over general n-dimensional lattices. However, new results gave polynomial-time quantum algorithms for Ideal-SVP$_\gamma$ with very large approximation factor $2^{\tilde{O}(\sqrt{n})}$. Nevertheless, these results do not apply directly to Ring-SIS or Ring-LWE, for reasons one can read on Section 10.3 of [Vai20].

4.1.6 Dual lattice

In order to complete the background required for Ring-LWE, which we present in the next section, we also need to define the notion of a dual lattice in a number field K.

Definition 4.10. Let K be a number filed and $\mathcal{L}$ be a lattice in K (i.e. a $\mathbb{Z}$-span of some $\mathbb{Q}$-basis of K). We define its *dual* as $\mathcal{L}^\vee = \{x \in K : Tr(x\mathcal{L}) \subseteq \mathbb{Z}\}$.

Remark 4.6. Denoting the canonical embedding with σ, we have $\sigma(\mathcal{L}^\vee) = \overline{\sigma(\mathcal{L})^*}$. Moreover, we can prove that $(\mathcal{L}^\vee)^\vee = \mathcal{L}$ and that, if $\mathcal{L}$ is a fractional lattice, then $\mathcal{L}^\vee$ is also one.

It can be proven that, for any fractional ideal $\mathcal{I}$ of a ring $R = \mathcal{O}_K$, its dual ideal is $\mathcal{I}^\vee = \mathcal{I}^{-1}R^\vee$, where $R^\vee$ is a fractional ideal (called the *codifferent ideal*) whose inverse $(R^\vee)^{-1}$ (called the *different ideal*) is integral and of norm $N((R^\vee)^{-1}) = \Delta_K$. Of interest to us is the case when R is the m-th cyclotomic number field of degree n with n a power of 2, for which $R^\vee$ is equivalent to R up to scale.

For an example of an ideal lattice and its dual, as well as the relation between them, one can see Peikert's second presentation in [Ins20].

4.2 Ring-LWE

Due to limited space, we only focus on Ring-LWE. For readers interested in Ring-SIS, we refer to Section 4.3 of [Pei16] for more.

Ring-LWE was presented by Peikert *et al.* in 2010 within the preliminary version of [LPR13b]. There, this ring-based analogue of LWE was introduced, along with proofs on its hardness. In general, Ring-LWE is parametrized by a ring R of degree n over $\mathbb{Z}$ (i.e. a ring of integers for some number field), a positive integer modulus q defining the quotient $R_q = R/qR$, and an error distribution χ over the ring R. The typical choice for R is the ring of integers of a cyclotomic number field, and χ is modelled as some kind of discretized Gaussian in the canonical embedding of R.

Definition 4.11. (Ring-LWE Distribution)
Suppose we have an $s \in R_q$, which we call the *secret*. Then, the Ring-LWE distribution $A_{s,\chi}$ over $R_q \times R_q$ is sampled by choosing $a \in R_q$ uniformly at random, choosing $e \leftarrow \chi$, and outputting $(a, b = s \cdot a + e \bmod q)$.

Remark 4.7. In the original definition of the Ring-LWE distribution from the 2010 paper (see also Section 3 of [LPR13b]), the secret is $s^\vee$ and the noisy product $b^\vee$, as they are taken to be from $R_q^\vee \doteqdot R^\vee/qR^\vee$. This definition is most useful for analysis, especially when using (near)-spherical errors $e^\vee$ in the canonical embedding of R.

However, for typical choices of rings used in cryptography, we have seen that there is a connection between R and $R^\vee$ and that Is what we use to transform the original definition to the one above. Specifically, using a certain "tweak" factor t, where $tR^\vee = R$, we have:

$$\underbrace{t \cdot b^\vee}_{b} = \underbrace{(t \cdot s^\vee)}_{s} \cdot a + \underbrace{(t \cdot e^\vee)}_{e} \in R/qR$$

Moreover, as this tweak is reversible, these forms are equivalent in terms of computation, application, analysis, etc. (see [LPR13a], the accompanying paper of [LPR13b], for more).

For the search and decision variants of the problem we also need the parameter m of the available samples.

Definition 4.12. (Search-Ring-LWE$_{q,\chi,m}$)
Given m independent samples $(a_i, b_i) \in R_q \times R_q$ drawn from $A_{s,\chi}$, where $s \in R_q$ is chosen uniformly at random (and stays fixed for all samples), find s.

Definition 4.13. (Decision-Ring-LWE$_{q,\chi,m}$) Given m independent samples $(a_i, b_i) \in R_q \times R_q$ where every sample was either drawn from:

1. $A_{s,\chi}$, for a uniformly random $s \in R_q$ (fixed for all samples); or
2. the uniform distribution,

distinguish which is the case (with non-negligible advantage).

Remark 4.8.

(i) Evidently, if we remove the error terms, the problem can be solved trivially as the distributions can be distinguished by observing that, in the uniform case, with high probability no solution $\mathbf{s}$ exists (while we can efficiently find s in the first case).

(ii) We remark that the Ring-LWE problem also has a normal form, in which we choose the secret s from the error distribution (modulo q), that can be proven to be at least as hard as the one above.

Advantages of Ring-LWE. As stated in the start of this chapter, the main reason we turned to the ring variants of these problems is that they offer:

- **Compactness.** In place of a single $b_i \in \mathbb{Z}_q$ in LWE, here through each sample (a_i, b_i) we get an n-dimensional ring element $b_i \in R_q$. Therefore, as we will see in the next section, this allows us to send messages of n bits, without using amortization techniques (specifically, $n = 256$ is usually used in practice).
- **Efficiency.** Multiplication in the rings we use can be performed significantly faster (in $\tilde{O}(n)$ time), due to the use of FFT-like techniques (see Subsection 4.4.2 of [Lyu20] for more).

Ring-LWE Vs LWE. We also note some (algebraic) distinctions between Ring-LWE and LWE:

- In place of n LWE samples with random $\mathbf{a}_i \in \mathbb{Z}_q^n$, Ring-LWE uses one Ring-LWE sample with random $a_i \in R_q$. Thus, we can view Ring-LWE as a special case of LWE with "structured" (correlated) samples.
- In LWE, the secret $\mathbf{s}$ and the random $\mathbf{a}_i$ are elements of $\mathbb{Z}_q^n$, which is treated as a $\mathbb{Z}$-module , and they are multiplied using the $\mathbb{Z}$-bilinear inner product $\langle \cdot, \cdot \rangle : \mathbb{Z}_q^n \times \mathbb{Z}_q^n \to \mathbb{Z}_q$. On the other hand, in Ring-LWE, the secret s and the random a_i are elements of R_q, which is treated as a R-module , and they are multiplied using the R-bilinear multiplication in R_q.

4.2.1 Hardness of Ring-LWE

We informally state the worst-case/average-case reduction for Ring-LWE, for the case of cyclotomic rings (for more, and a more general version, see [LPR13b]).

Theorem 4.1. For any $m = poly(n)$, cyclotomic ring R of degree n (over $\mathbb{Z}$), any appropriate modulus q, and error distribution χ of error rate $a < 1$, solving the Decision-Ring-LWE$_{q,\chi,m}$ problem is at least as hard as quantumly solving Ideal-SVP$_\gamma$ on arbitrary ideal lattices in R, for some $\gamma = poly(n)/a$.

Proof. In order to prove it, one needs to find an efficient quantum reduction using an oracle for Ring-LWE to solve Ideal-SVP$_\gamma$ on any ideal lattice in R. Particularly, the theorem is proved in two parts. On the first one, Search-Ring-LWE, using a quantum reduction, is proven to be at least as hard as Ideal-SVP$_\gamma$ (this holds for any ring of integers of a number field and any sufficiently large q). On the second one, Decision-Ring-LWE and Search-Ring-LWE are (classically) proven to be equivalent (relying on algebraic properties of cyclotomics and the form of q).

For the exact proof the reader is referred to [LPR13b], as well as Section 5 of [Reg10]. However, for the novice reader we first recommend Lyubashevsky's third presentation in [Uni12] and Peikert's second presentation in [Ins20].

4.3 Ring-LWE Cryptosystems

The majority of LWE-based cryptographic schemes and applications can be systematically transformed into their more concise and efficient Ring-LWE-based counterparts , using the relationship between LWE and Ring-LWE.

4.3.1 Compact Ring-LWE Cryptosystem

As an example, we present the ring-analogue of the Compact LWE cryptosystem of the previous chapter, which was actually discovered before its LWE counterpart (which was "backported" from the Ring-LWE one). As shown in [LPR13a], it can be instantiated for any cyclotomic ring (although the ring $\mathbb{Z}[X]/(X^n+1)$, for n a power of two, was used when it was first introduced in the 2010 preliminary version of [LPR13b]).

*** Compact Ring-LWE cryptosystem ***

- **Key Generation.**
  - Choose a uniformly random $a \in R_q$ (square matrix!).
  - Choose $s \in R$, with coordinates chosen independently from the Ring-LWE error distribution χ.
- **Private Key.** The s.
- **Public Key.** The normal-form Ring-LWE sample $(a, b) \in R_q \times R_q$, where $b = s \cdot a + e$, with $e \leftarrow \chi$.
- **Encryption.** To encrypt a message $\mu \in R_2$ (corresponding to an n-string),
  - Choose an $r \in R$, with coordinates chosen independently from the error distribution χ.
  - Output the ciphertext $(u \approx a \cdot r, v \approx b \cdot r + \mu\lfloor \frac{q}{2} \rfloor) \in R_q \times R_q$, where the approximations hide independent errors drawn from χ.
- **Decryption.** To decrypt the ciphertext (u, v),
  - Compute $u - s \cdot v \approx \mu \cdot \lfloor \frac{q}{2} \rceil \pmod q$.
  - Test whether $u - s \cdot v$ is closer to 0 or to $\lfloor \frac{q}{2} \rceil \pmod q$.

Security. Regarding security, similar to the LWE schemes, we have:

Theorem 4.2. The Compact Ring-LWE cryptosystem is IND-CPA secure, assuming that Decision-Ring-LWE$_{q,\chi,m}$ is hard, which for appropriate parameters is implied by the conjectured worst-case (quantum) hardness of Ideal-SVP$_\gamma$ on arbitrary ideal lattices in the ring R.

Proof. The main change in this security proof is that it needs a regulatory lemma for rings, which is substantially harder to prove (for more information on this see Subsection 5.2.4 of [LPR13b]). For the complete proof of security, see Section 8.2 of [LPR13a].

4.4 From Ring-LWE to Module-LWE

In this final section, we introduce (in a slightly informal and concise way) the *Module-LWE* problem (similarly, the *Module-SIS* problem) which acts as a bridge between LWE and Ring-LWE (similarly, SIS and Ring-SIS).[5] These average-case problems were proven to be at least as hard as challenging lattice problems restricted to module lattices (which themselves connect arbitrary and ideal lattices), in [LS14]. Moreover, from now on we will denote them with M-LWE and M-SIS, where M stands for module (in the same way, some authors denote Ring-SIS and Ring-LWE as R-SIS and R-LWE, respectively).

The first introduction of this generalized concept was made in [BGV12], with the introduction of M-LWE (called *Generalized Ring-LWE (R-GLWE) problem* there), which can be (informally) described in the following way:

- For a secret, it uses a vector $\mathbf{s} \in R_q^k$ of ring elements

[5] Informally, we note that a module is an algebraic structure generalizing rings and vector spaces, whereas module lattices (corresponding to finitely generated modules over the ring of integers of a number field) generalize both arbitrary lattices and ideal lattices.

- Its samples are of the form $(\mathbf{a}, b) \in R_q^k \times R_q$, where either $b = \langle \mathbf{s}, \mathbf{a} \rangle + e \bmod q$, for $e \leftarrow \chi$ or $b \in R_q$ is uniformly random.

For $R = \mathbb{Z}$, this is equal to the k-dimensional $\text{LWE}_{k,q,\chi}$ problem.
For $k = 1$, this is equal to $\text{R-LWE}_{q,\chi}$.

Additionally, we remark that the exact worst-case/average-case reductions related to M-SIS and M-LWE are (i) a classical reduction from Mod-SIVP (i.e. SIVP restricted to module lattices) to M-SIS; and (ii) a quantum reduction from Mod-SIVP to M-LWE in both its search and decision versions. For more information about the exact reductions, the reader is referred to [LS14]. In addition, we suggest [PP19] for a unified survey on the different variants of LWE.

From the cryptographic construction viewpoint, most constructions based on R-SIS and R-LWE can be adapted to M-SIS and M-LWE, with an efficiency slowdown within a constant factor of those based on R-SIS/R-LWE (in terms of memory requirements, communication costs, and algorithm run-times).This is most times justified as constructing a scheme on M-LWE is considered somewhat safer than R-LWE (similarly for SIS), relying on Mod-SIVP instead of Ideal-SIVP. We will discuss more about how this choice affects schemes in the last part of this thesis, using KYBER (which is based on M-LWE) as an example.

Remark 4.9. We remark that there is no definite proof that Module-LWE is strictly harder than Ring-LWE, assuming same modulus (and other parameters). In fact, for different modulus parameters there has been a reduction in the opposite direction [AD17]. Thus, it has been argued by some that the use of the more efficient Ring-LWE should be preferred over Module-LWE, as for them the security difference is not that large. For more opinions on the subject, one can look into the "ROUND 2 OFFICIAL COMMENT: NewHope" discussion in the pqc-forum [link].

Part III

CRYSTALS-Kyber: LWE-based Post-Quantum Standard

Chapter 5

Essential Concepts from Cryptography

Shifting gears, we prepare for our exploration of Crystals-Kyber by delving into essential preliminaries. These include fundamental cryptographic concepts such as transformations from CPA to CCA schemes, optimizations for LWE-based schemes, and the use of the Number Theoretic Transform in order to perform fast multiplication in the rings we are interested in.

5.1 Security Notions and Transformations

The notion of INDistinguishability against Chosen-Ciphertext Attacks (IND-CCA) is widely accepted as the prevailing security criterion for asymmetric encryption schemes. However, even though it is the desired notion of security, it is (usually) much more difficult to prove than INDistinguishability against Chosen-Plaintext Attacks (IND-CPA). Hence, multiple transformations have been proposed that transform a public-key encryption (PKE) scheme possessing weaker security properties, into an IND-CCA one.

In similar fashion, several transformations have been proposed that turn a PKE with weaker security into an IND-CCA secure Key-Encapsulation Mechanism (KEM). Then, this IND-CCA secure KEM can be combined with any (one-time) chosen-ciphertext secure symmetric encryption scheme to obtain an IND-CCA secure PKE scheme [CS03]. Such hybrid encryption schemes are frequently employed in practical applications, due to their efficiency and versatility.

The team behind CRYSTALS-Kyber also followed this route in their original paper [Bos+17]. In that work, they instantiate an IND-CPA secure PKE scheme called Kyber.CPA, and subsequently apply a variant of the Fujisaki-Okamoto transform [FO99] to establish an IND-CCA secure KEM, Kyber. Following this, they also construct IND-CCA secure encryption (Kyber.Hybrid), key exchange (Kyber.KE) and authenticated-key exchange (Kyber.AKE) schemes.

Therefore, in this section, we provide precise security definitions and embark on an analysis of the precise workings of such transformations and their resultant effects on cryptographic constructions, as they will prove useful when analysing both Kyber's security and Kyber itself.

5.1.1 Fundamental Security Concepts for Schemes

For the definitions below we mainly follow [HHK17], which uses code-based games for both its definitions and proofs.[1] Moreover, for novice readers we first suggest reading [FO99] and [Den03], as their definitions and transformations are simpler and can constitute a good introduction to the subject area.

Random Oracle Model. In general, we do not know how to construct efficient schemes which are provably secure based on standard cryptographic assumptions. To combat this, the *Random*

[1] For those not familiar with code-based games in cryptography, we refer to [Sho04].

Oracle Model (ROM) was introduced and now most famous cryptographic schemes used are proven secure in this model, which can be described as follows (more details on [KM15]):

"Assume that there is public *random oracle* that everyone (including all honest parties as well as the adversary) has access to. This oracle implements a truly random function in the following manner: the first time the oracle is asked a query x, it selects a value y, uniformly at random from its output domain, and returns this value. Afterwards, if the query x is posed again, the initial answer y is returned." We remark that this setup is analogous to treating the oracle as a "black box" housing a completely random function.

Moreover, one can observe that the random oracle acts similar to a hash function, such that we know nothing about the output we could get for a given input message x, until x is quarried. Thus, as in the real world such random oracles cannot exist (and truly random function cannot be implemented efficiently), efficient cryptographic constructions are created as follows:

"A scheme is designed in the random oracle model and its security is proved within that model (i.e. all parties in this scheme can make use of the public random oracle). Then, in the real world, a cryptographic hash function takes the place of the random oracle." However, due to this necessary change, a proof in this model does not actually guarantee security in the real world, but it provides a useful check that the construction under evaluation is not inherently flawed.

NOTATION. Following the notation from [HHK17], for this section, we denote by:

(i) $x \xleftarrow{\$} S$, the sampling of a uniform element x from a finite set S.
(ii) $x \leftarrow \mathfrak{D}$, the sampling according to some distribution $\mathfrak{D}$.
(iii) $[\![B]\!]$, the bit that is 1 if the Boolean Statement B is true, and 0 otherwise.
(iv) $y := A(x)$, a deterministic computation of an algorithm A on input x.
(v) $A^{\mathcal{O}}$, an algorithm with access to an oracle $\mathcal{O}$.
(vi) $y \leftarrow A(x)$, the computation of a probabilistic algorithm A on input x.

PUBLIC-KEY ENCRYPTION. We now formally define some important concepts on PKE:

Definition 5.1. A *(probabilistic) public-key encryption scheme* PKE=(Gen, Enc, Dec) consists of three algorithms and a finite message space $\mathcal{M}$, where:

- Gen, is a key generation algorithm outputting a key pair (pk, sk), where pk also defines a randomness space $\mathcal{R} = \mathcal{R}(pk)$.
- Enc, is an encryption algorithm that takes the public key pk and a message $m \in \mathcal{M}$, and outputs an encryption $c \leftarrow \mathrm{Enc}(pk, m)$ of m under the public key pk. This notation uses randomness which can be made explicit by writing $c := \mathrm{Enc}(pk, m; r)$ with $r \xleftarrow{\$} \mathcal{R}$.
- Dec, is a decryption algorithm that takes a pair of strings sk and c, and outputs either a message $m = \mathrm{Dec}(sk, c) \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$, if c is an invalid ciphertext.

Additionally, the notion of correctness is important for public-key encryption schemes, as previously stated (when calculating it in LWE cryptosystems). We remark also that, giving a definition of correctness in PKE that are defined relative to a random oracle $\mathcal{G}$ is a bit more complex, as the correctness bound might depend on the number of queries $q_{\mathcal{G}}$ to $\mathcal{G}$.

Definition 5.2.

(i) A PKE scheme is called δ-*correct* if $\mathbf{E}\left[\max_{m \in \mathcal{M}} \Pr\left[\mathrm{Dec}(sk, c) \neq m \,|\, c \leftarrow \mathrm{Enc}(pk, m)\right]\right] \leq \delta$, where the expectation is taken over $(pk, sk) \leftarrow \mathrm{Gen}$. Alternatively, we say that a PKE is δ-*correct* if for all (possibly unbounded) adversaries A, we have $\Pr[\mathrm{COR}^{\mathrm{A}}_{\mathrm{PKE}} \Rightarrow 1] \leq \delta$, where the correctness game COR is defined in the figure below (left). In simple terms, it is a game in which an (unbounded) adversary learns (pk, sk) and wins if he can find a message m inducing a correctness error.

(ii) A PKE scheme *in the random oracle model* is called $\delta(q_{\mathcal{G}})$-*correct* if for all (possibly unbounded) adversaries A making at most $q_{\mathcal{G}}$ queries to a random oracle $\mathcal{G}$, $\Pr[\text{COR-RO}^{\text{A}}_{\text{PKE}} \Rightarrow 1] \leq \delta(q_{\mathcal{G}})$, where the correctness game COR-RO is defined in the figure below (right). Also, the definition extends naturally to a case where PKE is defined relative to several oracles.

GAME COR:	**GAME** COR-RO:
01 $(pk, sk) \leftarrow \mathsf{Gen}$	01 $(pk, sk) \leftarrow \mathsf{Gen}$
02 $m \leftarrow \mathsf{A}(sk, pk)$	02 $m \leftarrow \mathsf{A}^{\mathsf{G}}(sk, pk)$
03 $c \leftarrow \mathsf{Enc}(pk, m)$	03 $c \leftarrow \mathsf{Enc}(pk, m)$
04 **return** $[\![\mathsf{Dec}(sk, c) = m]\!]$	04 **return** $[\![\mathsf{Dec}(sk, c) = m]\!]$

Figure 5.1: Correctness games COR and COR-RO (from [HHK17]).

Remark 5.1. We highlight that definition (i), which is in the standard model, is merely a special case of (ii) in ROM where the number of queries is zero ($q_{\mathcal{G}} = 0$) and thus $\delta(q_{\mathcal{G}})$ is constant.

SECURITY OF PKE. Before, defining IND-CPA properly, we provide the definitions of two more important security notions: One-Wayness under Chosen Plaintext Attacks (OW-CPA) and One-Wayness under Plaintext Checking Attacks (OW-PCA).

Definition 5.3. Let PKE=(Gen,Enc,Dec) be a public-key encyption scheme with message space $\mathcal{M}$. For ATK $\in$ {CPA,PCA}, we define OW-ATK *games* in the figure below, where the $\mathcal{O}_{\text{ATK}}$ oracle of the adversary is defined as

$$\mathcal{O}_{\text{ATK}} := \begin{cases} - & \text{ATK} = \text{CPA} \\ \textsc{Pco}(\cdot,\cdot) & \text{ATK} = \text{PCA} \end{cases}$$

Furthermore, we define the OW-ATK *advantage function of an adversary* A *against* PKE as

$$\text{Adv}^{\text{OW-ATK}}_{\text{PKE}}(\text{A}) := \Pr[\text{OW-ATK}^{\text{A}}_{\text{PKE}} \Rightarrow 1].$$

GAME OW-ATK:	$\textsc{Pco}(m \in \mathcal{M}, c)$
01 $(pk, sk) \leftarrow \mathsf{Gen}$	01 **return** $[\![\mathsf{Dec}(sk, c) = m]\!]$
02 $m^* \xleftarrow{\$} \mathcal{M}$	
03 $c^* \leftarrow \mathsf{Enc}(pk, m^*)$	
04 $m' \leftarrow \mathsf{A}^{\mathcal{O}_{\text{ATK}}}(pk, c)$	
05 **return** $\textsc{Pco}(m', c^*)$	

Figure 5.2: Games OW-ATK, ATK $\in$ {CPA,PCA} and game-definition of $\textsc{Pco}$ (from [HHK17]).

Remark 5.2.

(i) Given a pair $(m, c) \in \mathcal{M} \times \mathcal{C}$, the *plaintext checking oracle* $\textsc{Pco}(m, c)$ correctly determines whether c is an encryption of m or not. However, we remark that $\textsc{Pco}(m, c)$ implicitly disallows queries for $m \notin \mathcal{M}$ (with the convention that $\textsc{Pco}(m \notin \mathcal{M}, c)$ yields $\perp$).

(ii) Observant readers might notice that in the end of the OW-ATK, the $\textsc{Pco}$ is used. This is done in order to check the correctness of m', returning 1 if and only if $\text{Dec}(sk, c^*) = m'$.

We continue by defining Indistinguishability under Chosen Plaintext Attacks (IND-CPA) properly. We also remark that OW-CPA and IND-CPA security are connected: for a large enough message space $\mathcal{M}$, an IND-CPA secure PKE is also OW-CPA (see Lemma 2.3 of [HHK17]).

Regarding the definitions, we note again that, novice readers should first look at the detailed descriptions in [FO99] (for IND-CCA KEM, see [Den03])) before trying to understand these ones (that use code-based games). Moreover, we note that there are some subtleties when defining these concepts in the ROM and refer to [HHK17] for more.

Definition 5.4. Let PKE=(Gen,Enc,Dec) be a public-key encyption scheme with message space $\mathcal{M}$. We define the IND-CPA *game* as in the figure below, and the IND-CPA *advantage function of an adversary* $\mathrm{A} = (\mathrm{A}_1, \mathrm{A}_2)$ *against* PKE (with A_2 having a binary output) as

$$\mathrm{Adv}^{\text{IND-CPA}}_{\text{PKE}} := |\Pr[\text{IND-CPA}^{\mathrm{A}} \Rightarrow 1] - 1/2|.$$

GAME IND-CPA	**GAME** IND-CCA	DECAPS($c \neq c^*$)
01 $(pk, sk) \leftarrow \mathsf{Gen}$	01 $(pk, sk) \leftarrow \mathsf{Gen}$	01 $K := \mathsf{Decaps}(sk, c)$
02 $b \xleftarrow{\$} \{0,1\}$	02 $b \xleftarrow{\$} \{0,1\}$	02 **return** K
03 $(m_0^*, m_1^*, st) \leftarrow \mathsf{A}_1(pk)$	03 $(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$	
04 $c^* \leftarrow \mathsf{Enc}(pk, m_b^*)$	04 $K_1^* \xleftarrow{\$} \mathcal{K}$	
05 $b' \leftarrow \mathsf{A}_2(pk, c^*, st)$	05 $b' \leftarrow \mathsf{A}^{\text{DECAPS}}(c^*, K_b^*)$	
06 **return** $[\![b' = b]\!]$	06 **return** $[\![b' = b]\!]$	

Figure 5.3: Games IND-CPA for PKE and IND-CCA for KEM (from [HHK17]).

KEY ENCAPSULATION MECHANISM. We now similarly define KEMs and surrounding properties like δ-correctness:

Definition 5.5. A *Key Encapsulation Mechanism* KEM=(Gen, Encaps, Decaps) consists of three algorithms:

- Gen, is a key generation algorithm outputting a key pair (pk, sk), where pk also defines a finite key space $\mathcal{K}$.
- Encaps, is an encapsulation algorithm that takes a string pk, and outputs a tuple (K, c), where c is an encapsulation of a key $K \in \mathcal{K}$.
- Decaps, is a deterministic decapsulation algorithm that takes a pair of strings sk and c, and outputs either a key $K := \mathrm{Decaps}(sk, c) \in \mathcal{K}$ or a special symbol $\perp \notin \mathcal{K}$, if c is not a valid encapsulation.

Definition 5.6. A key encapsulation mechanism KEM is called δ-*correct* if

$$\Pr\left[\mathrm{Decaps}(sk, c) \neq K \mid (pk, sk) \leftarrow \mathrm{Gen}; (K, c) \leftarrow \mathrm{Encaps}(pk)\right] \leq \delta$$

We note that the above definition stays the same in ROM, as KEM encapsulations do not depend on messages.

SECURITY OF KEM. In PKE schemes we only defined IND-CPA and not IND-CCA, as it was not of interest for the latter transformations. Similarly, we only care about (and define) the notion of IND-CCA in KEMs:

Definition 5.7. Let KEM=(Gen, Encaps, Decaps) be a key encapsulation mechanism. Then, we define IND-CCA *game* as in the figure above, and the IND-CCA *advantage function of an adversary* A against KEM (having a binary output) as

$$\mathrm{Adv}^{\text{IND-CCA}}_{\text{KEM}} := |\Pr[\text{IND-CCA}^{\mathrm{A}} \Rightarrow 1] - 1/2|.$$

5.1.2 Modular FO transformations

The reader should keep in mind that the goal of this section is to define some transformations similar to the one used for CRYSTALS-Kyber in order to go from Kyber.CPA to Kyber. Thus, our discussion is limited to only results related to these transformations.

In general, the need for new transformations came from three main drawbacks of previous transformations (e.g. FO, React/GEM and "more modern" transformations from Dent, for more see [Den03]), making them not useful for a post-quantum scheme like Kyber:

(i) **Correctness Error:** Most efficient lattice-based encryption schemes (Kyber.CPA too) have at least some decryption errors, making them incompatible to previous constructions like FO and REACT/GEM that required the underlying PKE to have no decryption errors.

(ii) **Post-quantum Security:** There has been an interest in finding IND-CCA secure schemes that are also secure against quantum adversaries, as this technology may allow attackers to execute all "offline primitives" (like hash functions) in a way that necessitates the use of a different security model for security proofs, the *Quantum (accessible) Random Oracle Model (QROM)* [Bon+11]. Thus, transformations that are secure in in the QROM are needed ([TU16] marked the first move towards security in this model with a modified FO transformation.)

(iii) **Tightness:** When proving the security of a scheme P under the hardness of a problem S, we often construct a reduction algorithm R that utilises an adversary A as a subroutine against the security of the scheme, and solves the problem S. We say that a reduction is *tight* if we have $T \approx T'$ and $\epsilon \approx \epsilon'$, where (T, ϵ) and (T', ϵ') are the running times and success probabilities of A and R, respectively.

Therefore, tight security ensures that breaking P is as hard as solving S, whereas non-tight reductions make it unclear whether breaking P is difficult, even if S is hard, thus requiring to adapt system parameters accordingly, resulting in considerably less efficient schemes. While tight transformations exist, they are inadequate for most lattice-based primitives, necessitating better alternatives.

These problems were addressed in [HHK17] with a modular treatment (i.e. in several steps) of FO-like transformations which are robust against PKE with correctness errors (i.e. with the correctness error of the original scheme bounding the correctness error of the resulting one)

Particularly, of importance to us are the transformations T and $\text{U}^{\not\perp}$ included in the table:

Transformation	**Security Implication**	**QROM ?**	**ROM Tightness ?**
$\text{PKE}_1 = \text{T}[\text{PKE}, \mathcal{G}]$	IND-CPA $\Rightarrow$ OW-PCA	✓	✓
$\text{KEM}^{\not\perp} = \text{U}^{\not\perp}[\text{PKE}_1, \mathcal{H}]$	OW-PCA $\Rightarrow$ IND-CCA		✓

T Transformation. We start with T which can transform an IND-CPA secure scheme into an OW-PCA one, and works as follows:

Let PKE=(Gen, Enc, Dec) be a public-key encryption scheme with message space $\mathcal{M}$ and randomness space $\mathcal{R}$, and $\mathcal{G} : \mathcal{M} \to \mathcal{R}$ be a random oracle. From PKE and $\mathcal{G}$, through T, we get the scheme PKE_1=T[PKE,$\mathcal{G}$], where $\text{PKE}_1 = (\text{Gen}, \text{Enc}_1, \text{Dec}_1)$ with $\text{Enc}_1, \text{Dec}_1$ defined below:

$\underline{\mathsf{Enc}_1(pk, m)}$	$\underline{\mathsf{Dec}_1(sk, c)}$
01 $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$	01 $m' := \mathsf{Dec}(sk, c)$.
02 **return** c	02 **if** $m' = \perp$ **or** $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$
	03 **return** $\perp$
	04 **else return** m'

Figure 5.4: Encryption and decryption algorithms for PKE_1=T[PKE,$\mathcal{G}$] (from [HHK17]).

Furthermore, the theorem below establishes that OW-PCA security of PKE_1 tightly reduces to IND-CPA security of PKE, in the random oracle:

Theorem 5.1. (PKE IND-CPA $\overset{ROM}{\Rightarrow}$ PKE_1 OW-PCA)
Suppose PKE is δ-correct. Then, for any OW-PCA adversary B that issues at most $q_{\mathcal{G}}$ queries to the random oracle $\mathcal{G}$ and q_{P} queries to a plaintext checking oracle Pco, there exists an IND-CPA adversary A, with running time approximate equal to that of B, such that:

$$\text{Adv}^{\text{OW-PCA}}_{\text{PKE}_1}(\text{B}) \leq (q_{\mathcal{G}} + q_{\text{P}}) \cdot \delta + \frac{2q_{\mathcal{G}} + 1}{|\mathcal{M}|} + 3 \cdot \text{Adv}^{\text{IND-CPA}}_{\text{PKE}}(\text{A})$$

Proof. See the game-based proof in Theorem 3.2 of [HHK17] and the remark after it.

$\mathrm{U}^{\not\perp}$ TRANSFORMATION. We now introduce the transformation $\mathrm{U}^{\not\perp}$, which converts an OW-PCA secure PKE_1 into an IND-CCA secure KEM, with "implicit rejection" of invalid ciphertexts (returning a pseudorandom key K in that case).

Let $\mathrm{PKE}_1 = (\mathrm{Gen}, \mathrm{Enc}_1, \mathrm{Dec}_1)$ be a public-key encryption scheme with message space $\mathcal{M}$, and $\mathcal{H} : \{0,1\}^* \to \mathcal{M}$ be a random oracle. From PKE_1 and $\mathcal{H}$, through $\mathrm{U}^{\not\perp}$, we get the scheme $\mathrm{KEM}^{\not\perp} = \mathrm{U}^{\not\perp}[\mathrm{PKE}_1, \mathcal{H}]$, where $\mathrm{KEM}^{\not\perp} = (\mathrm{Gen}^{\not\perp}, \mathrm{Encaps}^{\not\perp}, \mathrm{Decaps}^{\not\perp})$ with the algorithms defined below:

$\underline{\mathrm{Gen}^{\not\perp}}$	$\underline{\mathsf{Encaps}(pk)}$	$\underline{\mathsf{Decaps}^{\not\perp}(sk, c)}$
01 $(pk', sk') \leftarrow \mathsf{Gen}_1$	01 $m \xleftarrow{\$} \mathcal{M}$	01 Parse $sk = (sk', s)$
02 $s \xleftarrow{\$} \mathcal{M}$	02 $c \leftarrow \mathsf{Enc}_1(pk, m)$	02 $m' := \mathsf{Dec}_1(sk', c)$
03 $sk := (sk', s)$	03 $K := \mathsf{H}(m, c)$	03 **if** $m' \neq \perp$
04 **return** (pk', sk)	04 **return** (K, c)	04 **return** $K := \mathsf{H}(m', c)$
		05 **else return** $K := \mathsf{H}(s, c)$

Figure 5.5: Definitions of algorithms from $\mathrm{KEM}^{\not\perp} = \mathrm{U}^{\not\perp}[\mathrm{PKE}_1, \mathcal{H}]$ (from [HHK17]).

Furthermore, the theorem below establishes that IND-CCA security of $\mathrm{KEM}^{\not\perp}$ tightly reduces to OW-CPA security of PKE_1, in the random oracle:

Theorem 5.2. (PKE_1 OW-CPA $\overset{ROM}{\Rightarrow}$ $\mathrm{KEM}^{\not\perp}$ IND-CCA)
If PKE_1 is δ_1-correct, then $\mathrm{KEM}^{\not\perp}$ is also δ_1-correct. Additionally, for any IND-CCA adversary B against $\mathrm{KEM}^{\not\perp}$ that issues at most $q_{\mathcal{H}}$ queries to the random oracle $\mathcal{H}$ and q_{D} queries to the decapsulation oracle $\mathrm{Decaps}^{\not\perp}$, there exists an OW-PCA adversary A against PKE_1 making at most $q_{\mathcal{H}}$ queries to the PCO oracle, with running time approximate equal to that of B, such that:

$$\mathrm{Adv}^{\text{IND-CCA}}_{\mathrm{KEM}^{\not\perp}}(\mathrm{B}) \leq \frac{q_{\mathcal{H}}}{|\mathcal{M}|} + \mathrm{Adv}^{\text{OW-PCA}}_{\mathrm{PKE}_1}(\mathrm{A})$$

Proof. See the game-based proof in Theorem 3.4 of [HHK17].

Therefore, combining this and the previous theorem we get the concrete bound below (where adversary B makes at most q_{RO} queries to random oracles $\mathcal{H}$ and $\mathcal{G}$). As a demonstration, we refer the reader to Theorem 3 of [Bos+17] (i.e. the theorem regarding the classical security of Kyber's transformation), which has $|\mathcal{M}| = 2^{256}$.

KEM	**Bound on** $\mathbf{Adv}^{\text{IND-CCA}}_{\mathbf{KEM}^{\not\perp}}(\mathbf{B}) \leq$
$\mathrm{KEM}^{\not\perp}$	$q_{\mathrm{RO}} \cdot \delta + \frac{3q_{\mathrm{RO}}}{\lvert\mathcal{M}\rvert} + 3 \cdot \mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{PKE}}(\mathrm{A})$

FROM THEORY TO PRACTICE. After seeing the analysis above, one might wonder on how this translates to security (and parameter optimization) in actual schemes. In general, if we want a scheme to have "κ bits of security" we require that, for all adversaries B with advantage Adv(B) and running time Time(B), it holds that $\mathrm{Time(B)}/\mathrm{Adv(B)} \geq 2^{\kappa}$.

Example 5.1.

(a) If a security bound contains the term $q_{\mathrm{RO}} \cdot \delta$, then, in order to have κ bits of security, we need the underlying scheme to be δ-correct with $\delta \leq 2^{-\kappa}$, due to

$$\frac{\mathrm{Time(B)}}{\mathrm{Adv(B)}} \geq \frac{q_{\mathrm{RO}}}{q_{\mathrm{RO}} \cdot \delta} = \frac{1}{\delta} \geq 2^{\kappa}$$

(b) Similarly for the term $q_{\mathrm{RO}}/|\mathcal{M}|$, we need $|\mathcal{M}| \geq 2^{\kappa}$, due to

$$\frac{\mathrm{Time(B)}}{\mathrm{Adv(B)}} \geq \frac{q_{\mathrm{RO}}}{q_{\mathrm{RO}}/|\mathcal{M}|} = \mathcal{M} \geq 2^{\kappa}$$

Remark 5.3. Due to its complexity, we refrain from discussing directly about QROM and related transformations at this point and will continue this directly when analysing Kyber's security.

5.2 Optimizations for Schemes based on LWE

We continue our dive into concepts that are used in CRYSTALS-Kyber with two concepts coming from the realm of LWE: (i) reducing the ciphertext size in an LWE scheme by removing the "low-order part"; and (ii) Learning with Rounding (LWR).

5.2.1 Compression and Decompression

In order to show the benefits of this optimization, we need a cryptosystem on which to apply it, serving as an example. For this, we use a slightly simplified form of the Compact LWE cryptosystem we analysed in Subsection 3.5.3. Moreover, for our analysis we suppose that proper values for the parameters (and the distribution χ) have been selected so as to decrypt with overwhelming probability (for more details see Chapter 2 of [Lyu20]). Thus, we have:

- **Key Generation:** The secret key is $\mathbf{s} \leftarrow \chi^n$, and, selecting $\mathbf{e}_1 \leftarrow \chi^n$, we get the public key $(\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times n}, \mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}_1)$.
- **Encryption:** To encrypt a bit $\mu \in \{0,1\}$, we first choose $\mathbf{r}, \mathbf{e}_2 \leftarrow \chi^n$ and $e_3 \leftarrow \chi$, and then output $(\mathbf{u} = \mathbf{A}^T\mathbf{r} + \mathbf{e}_2, v = \mathbf{t}^T\mathbf{r} + e_3 + \lfloor \frac{q}{2} \rceil \cdot \mu)$.
- **Decryption:** To decrypt, we compute $v - \mathbf{s}^T\mathbf{u} = \mathbf{e}_1^T\mathbf{r} + e_3 + \mu \cdot \lfloor \frac{q}{2} \rceil - \mathbf{s}^T\mathbf{e}_2 \approx \mu \cdot \lfloor \frac{q}{2} \rceil \pmod q$ and test whether $v - \mathbf{s}^T\mathbf{u}$ is closer to 0 or to $\lfloor \frac{q}{2} \rceil \pmod q$.

The Optimization. On the above cryptosystem, it is evident that the ciphetext part v contributes $\log q$ bits to the overall ciphertext size (this repeats N times for N bits of information, thus contributing $N \log q$). Our objective is to devise a method so that, rather than transmitting $\log q$ bits for v, we transmit only a fix number κ of bits. This can be achieved, albeit with the small caveat of introducing an additional error to the decryption equation.

Put differently, considering $v \in \mathbb{Z}_q$, our objective is to identify a set $S \subset \mathbb{Z}_q$ of size 2^κ such that the maximum distance between adjacent elements in S is minimal. Also, if we imagine the additive group $\mathbb{Z}_q$ as points arranged on a circle, we can measure the distance between two points of S as the number of $\mathbb{Z}_q$ points between them (for a visualization see Figure 1 in [Lyu20]). Thus, as the smallest attainable value for the maximum distance can be $q/2^\kappa$, we would like to get as close to this as possible. One potential set S, where any $v \in \mathbb{Z}_q$ is within a distance of $\lceil q/2^\kappa \rceil$ from an element of S, is

$$S = \{\lceil i \cdot q/2^\kappa \rfloor : 0 \le i < 2^\kappa\}.$$

Example 5.2. For $q = 3329$ and $k = 10$, which are the values used in a specific instantiation of Kyber, we have $\mathbb{Z}_{3329}$ and $S = \{\lceil i \cdot 3329/2^{10} \rfloor : 0 \le i < 2^{10}\} = \{0, 3, \ldots, 3326\}$. Hence, S can be represented by $\kappa = 10$ bits and every point in $\mathbb{Z}_{3329}$ is within a distance of $\lceil 3329/2^{10} \rceil = 4$ from an elements of S.

On such a set S, for any $v \in \mathbb{Z}_q$ we define

- $\texttt{HIGH}_S(v)$ to be the element in S that is closest to v; and
- $\texttt{LOW}_S(v)$ to be $v - \texttt{HIGH}_S(v)$.

Then, instead of publishing the v, we transmit $v' = \texttt{HIGH}_S(v) \in S$, and use the equality $v = v' + e'$ with $e' = \texttt{LOW}_S(v) \in [q/2^{\kappa+1}]$, for decryption. More specifically, for the decryption of a ciphertext $(\mathbf{u}, v')$, we compute

$$v' - \mathbf{s}^T\mathbf{u} = \mathbf{e}_1^T\mathbf{r} + e_3 - e' + \mu \cdot \left\lfloor \frac{q}{2} \right\rceil - \mathbf{s}^T\mathbf{e}_2 \approx \mu \cdot \left\lfloor \frac{q}{2} \right\rceil \pmod q.$$

Hence, the only difference is the error e', which does not affect the decryption error too much for small values of κ.

Remark 5.4. At times, reducing bits in the ciphertext part $\mathbf{u}$ using a similar bit reduction procedure may be useful too. However, doing so noticeably increases the decryption error because any error added to U gets multiplied by S, introducing an additional $\mathbf{s}^T\mathbf{e}''$ term in the

decryption equation (where $\mathbf{e}''$ is defined similarly to e'). Nevertheless, considering how $\mathbf{u}$ affects the ciphertext size, even a slight reduction in bits can be significant. Hence, the challenge lies in finding the right balance between decryption error and ciphertext size through trial-and-error.

Taking again Kyber as an example, we see that it does use this procedure in both $\mathbf{u}$ and v, through functions called `Compress` and `Decompress`, which we will see in the next chapter. However, one should keep in mind that Kyber bases its security in Module-LWE, and thus the exact elements and error calculations will be different.

5.2.2 Learning with Rounding (LWR) Problem

Suppose that we have $(\mathbf{A}, \mathbf{b}^T = \mathbf{s}^T\mathbf{A}+\mathbf{e}^T \pmod q) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, where the coefficients of $\mathbf{s}, \mathbf{e}$ are chosen from the error distribution χ. Assuming that the LWE problem is hard, the distribution of the above samples is indistinguishable from uniform. Moreover, if we round each coefficient of $\mathbf{b} \in \mathbb{Z}_q^m$ to the closest point in a set $S \subset \mathbb{Z}_q$ (similarly to the previous subsection), the distribution of $(\mathbf{A}, \texttt{HIGH}_S(\mathbf{b}))$ is still indistinguishable from $(\mathbf{A}, \texttt{HIGH}_S(\mathbf{u}))$, where $\mathbf{u}$ is selected uniformly at random.

Suppose now that the distribution χ is simply the uniform one over a set $[\beta] = \{-\beta, \dots, \beta\}$ for some β. Consider how the small error $\mathbf{e}$ affects the value of $\texttt{HIGH}_S(\mathbf{b}) = \texttt{HIGH}_S(\mathbf{s}^T\mathbf{A}+\mathbf{e}^T)$, for a set S as the one defined previously. According to [Lyu20], it can be proven that the probability (over the randomness of $\mathbf{A}, \mathbf{s}, \mathbf{e}$) of $\texttt{HIGH}_S(\mathbf{s}^T\mathbf{A}+\mathbf{e}^T) = \texttt{HIGH}_S(\mathbf{s}^T\mathbf{A})$ is approximately $(1-\frac{\beta|S|}{2q})^n$. Thus, whenever q is sufficiently large relative to β and $|S|$, the presence of $\mathbf{e}$ is irrelevant. Distinguishing the distribution $\texttt{HIGH}_S(\mathbf{s}^T\mathbf{A}+\mathbf{e}^T)$ from uniform is called *Learning with Rounding (LWR) problem*, and it can be proven that it is at least as hard as LWE whenever adding $\mathbf{e}$ does not affect the rounded output (and thus $\texttt{HIGH}_S(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T) = \texttt{HIGH}_S(\mathbf{s}^T\mathbf{A})$). For more on this see Subsection 2.5.2 of [Lyu20] and its references.

The benefits of using this optimization are evident: eliminating errors allows for enhanced parameters, with Kyber (later) serving as a notable example for its effectiveness.

5.3 Fast Multiplication in Rings

Lastly, we introduce one of the most important components of Kyber (and Ring-LWE/M-LWE schemes in general), the polynomial multiplication via the Number Theoretic Transform (NTT), which is a special case of the Discrete Fourier Transform (DFT) over the field $\mathbb{Z}_q^*$ (instead of complex numbers). This technique was also vaguely mentioned when discussing the efficiency of Ring-LWE with the term "FFT-like techniques", so now its the time to further delve on this concept.

For our analysis we assume to be working in a ring $R_{f,q} = \mathbb{Z}_q[X]/\langle f(X)\rangle$, for f and q chosen appropriately (we will see what this means later). In practice, our interest lies in the case where $f(X) = X^n + 1$ with $n = 256$, and either $q = 7681$ (as in the first Kyber paper [Bos+17]) or $q = 3329$ (as in the "Round 3" documentation for Kyber in [Ava+21] [2]). On such a ring, we are interested in the multiplication of two polynomials of degree $n-1$, and state that there is a method which makes it more efficient, from $O(n^2)$ operations (with basic multiplication), to $O(n \log n)$ operations over $\mathbb{Z}_q$. Moreover, there are two more benefits to using NTT:

- **Parallelization:** The NTT multiplication procedure is favorable for parallel implementation, and can thus be performed extremely fast on architecture supporting the AVX2 vector-instruction set [Sei18].
- **Can be done "in place":** In other words, the operation does not require extra temporary storage space while it is being computed.

[2] In the duration of NIST's PQC project, changes were made to the original Kyber in order to optimize it. In our later analysis, we follow the last official paper for Kyber [Ava+21].

5.3.1 Chinese Remainder Theorem and Multiplication

Before presenting the multiplication procedure, it is essential to first provide a reminder of how the Chinese Remainder Theorem (CRT) and the Chinese Remainder Representation (CRR) can be useful in performing faster multiplication in $\mathbb{Z}$.

Theorem 5.3. (Chinese Remainder Theorem)
Let $n_1, \ldots, n_k$ be integers ≥ 2 that are pairwise coprime (i.e. $\gcd(n_i, n_j) = 1, \forall\, i \neq j$). Let $a_1, \ldots, a_k$ be integers with $0 \leq a_i < n_i$ $(i = 1, \ldots, k)$ and set $n = n_1 \ldots n_k$. Then, the system of congruences

$$X \equiv a_i \,(\mathrm{mod}\, n_i)\,(i = 1, \ldots, k)$$

has a unique solution $X \equiv \sum_{i=1}^{k} a_i y_i N_i \,(\mathrm{mod}\, n)$, where $N_i = n/n_i$ and $y_i = N_i^{-1} \bmod n_i$.

As a corollary, we have $\mathbb{Z}_n \cong \mathbb{Z}_{n_1} \times \ldots \times \mathbb{Z}_{n_k}$.

Definition 5.8. If a is any number less than n, the *Chinese Remainder Representation* of a is the sequence $a_1, \ldots, a_k$ where $a \equiv a_i \bmod n_i$. According to CRT, every number from 0 to $n-1$ is uniquely represented by such a sequence.

We show how the above can be used for multiplication with the example below:

Example 5.3. Let $n_1 = 5, n_2 = 7, n_3 = 11$ and thus $n = 385$. Suppose that we want to multiply the numbers 368 and 347 in $\mathbb{Z}_n$. To do that, we first compute the CRR of 368 and 347, and get $(368 \bmod 5,\ 368 \bmod 7,\ 368 \bmod 11) = (3, 4, 5)$ and $(2, 4, 6)$ respectively. Then, multiply pairwise their representations and get $(3 \cdot 2, 4 \cdot 4, 5 \cdot 6) = (6, 16, 30) = (1, 2, 8)$. Using the CRT to calculate the number in $\mathbb{Z}_n$ with CRR $(1, 2, 8)$, we find that it is 261 (which is the same as $368 \cdot 347 \bmod 385$).

While this example does not explicitly explain why this method is faster than basic multiplication (due to the size of the numbers chosen) or its relevance to our objectives, the advantages become apparent when dealing with large numbers in $\mathbb{Z}_n$ where n is large (e.g. 4096-bit size). RSA illustrates this application well, as detailed in [Gro00]. Additionally, novice readers are referred to [Sho08] for some more background on CRT and its uses.

Moving on to how CRT and CRR can be useful for multiplication in a ring like $R_{q,f}$, we first present a more generalized CRT theorem for rings (see Theorem 7.17 in [DF03]) and then continue with the NTT multiplication. For beginners, we also refer to the blog post "The Number Theoretic Transform in Kyber and Dilithium" of PhD candidate Amber Sprenkels [link].

Theorem 5.4. (Chinese Remainder Theorem for Rings)
Let R be a commutative ring with ideals $\mathcal{I}_1, \ldots, \mathcal{I}_n$. Then, the map $\psi : R \to (R/\mathcal{I}_1) \times \ldots \times (R/\mathcal{I}_n)$ defined by $\psi(r) = (r + \mathcal{I}_1, \ldots, r + \mathcal{I}_n)$ is a ring homomorphism with kernel $\mathcal{I}_1 \cap \ldots \cap \mathcal{I}_n$. If all of the ideals $\mathcal{I}_1, \ldots, \mathcal{I}_n$ are pairwise comaximal[3] (i.e. $\mathcal{I}_1 + \ldots, +\mathcal{I}_n = R$), then ψ is surjective and $\mathcal{I}_1 \cap \ldots \cap \mathcal{I}_n = \mathcal{I}_1 \ldots \mathcal{I}_n$ and thus $R/(\mathcal{I}_1 \ldots \mathcal{I}_n) \cong (R/\mathcal{I}_1) \times \ldots \times (R/\mathcal{I}_n)$.

Suppose, for example, that we are in a ring $R_{f,q} = \mathbb{Z}_q[X]/\langle f(X)\rangle$ where $f(X) = (X - r_1) \cdot \ldots \cdot (X - r_n)$ for distinct $r_i \in \mathbb{Z}_q$. For an element $\mathbf{a} \in R_{f,q}$, its CRR $\hat{\mathbf{a}}$ is defined as

$$\hat{\mathbf{a}} = (\mathbf{a} \bmod (X - r_1), \ldots, \mathbf{a} \bmod (X - r_n)) \in \mathbb{Z}_q^n.$$

Moreover, if we want two multiply two elements $a, b \in R_{q,f}$, we follow a process akin to the one described earlier:

1. We compute the CRR of a and b as above, getting $\hat{\mathbf{a}} = (a_1, \ldots, a_n)$ and $\hat{\mathbf{b}} = (b_1, \ldots, b_n)$.
2. Then, we perform point-wise multiplication, getting $\hat{\mathbf{ab}} = \hat{\mathbf{a}} \odot \hat{\mathbf{b}} = (a_1 b_1, \ldots, a_n b_n)$.[4]
3. Lastly, we compute $\mathbf{c}$ such that $\hat{\mathbf{c}} = \hat{\mathbf{ab}}$.

[3] For rings that we are interested in (i.e. rings of integers), pairwise comaximal ideals are also pairwise coprime.
[4] The symbol "$\odot$" denotes the point-wise multiplication, i.e. multiplication of corresponding components.

Remark 5.5. As step 2 only needs n multiplications over $\mathbb{Z}_q$, the other steps are the primary factors behind the running time of the multiplication procedure. The efficiency of these two steps relies on the factorization of the polynomial f over $\mathbb{Z}_q[X]$, with the optimal polynomials being of the form $X^n \pm 1$ when n is a power of 2.

In the ring $R_{q,f}$ for $f(X) = X^n + 1$ with n a power of 2 especially, by using the NTT on steps 1 and 3, we can achieve the goals we set earlier for fast multiplication on rings.

5.3.2 Fast Multiplication via the Number Theoretic Transform (NTT)

We first give some important definitions (taken from [LZ22]) and then move on to NTT:

Definition 5.9. Without loss of generality, we consider two polynomials $a(X), b(X)$ of degree $n-1$ and represent them as vectors $\mathbf{a} = (a_0, \ldots, a_{n-1})$, $\mathbf{b} = (b_0, \ldots, b_{n-1})$. If their lengths are less than $n-1$ we pad them with zero. For such polynomials we define the following operations:

(i) **Linear Convolution.**
- The multiplication of such polynomials, $\mathbf{c} = \mathbf{a} \cdot \mathbf{b} \in \mathbb{Z}_q[X]$, can be computed as $\mathbf{c} = \sum_{k=0}^{2n-2} c_k X^k \in \mathbb{Z}_q[X]$, where $c_k = \sum_{i+j=k} a_i b_j \bmod q$, for $k = 0, \ldots, 2n-2$.
- Suppose now $\mathbf{c} = \mathbf{a} \cdot \mathbf{b} \in \mathbb{Z}_q[X]/(\phi(X))$. In this case, one can first compute $\mathbf{c}' = \mathbf{a} \cdot \mathbf{b} \in \mathbb{Z}_q[X]$ and then $\mathbf{c} = \mathbf{c}' \bmod \phi(X)$.

In these cases, we call $\mathbf{c}$ the *linear convolution* of $\mathbf{a}$ and $\mathbf{b}$.

(ii) **Negative Wrapped Convolution.** Consider $\mathbf{c} = \mathbf{a} \cdot \mathbf{b} \in \mathbb{Z}_q[X]/(X^n + 1)$. Then, we can also compute $\mathbf{c}$ as $\mathbf{c} = \sum_{k=0}^{n-1} c_k X^k$, where $c_k = \sum_{i=0}^{k} a_i b_{k-i} - \sum_{i=k+1}^{n-1} a_i b_{k+n-i} \bmod q$, for $k = 0, \ldots, n-1$. Here, $\mathbf{c}$ is called the *negative wrapped convolution* (NWC) of $\mathbf{a}$ and $\mathbf{b}$.

As previously noted, NTT is a specific instance of DFT over a finite field, first introduced by Pollard in [Pol71]. Consequently, many FFT techniques (designed for fast DFT computation) can be extended to NTT, yielding analogous fast algorithms for it. Our focus lies on NTT multiplication, which, due to certain restrictions, cannot be uniformly implemented across all rings. Specifically, we are interested in the ring $\mathbb{Z}_q[X]/\langle X^n + 1\rangle$, where n is a power of two and the (prime) modulus adheres to either $q \equiv 1 \pmod{2n}$ (as used in original Kyber [Bos+17]) or $q \equiv 1 \pmod{n}$ (as employed in latter variations and especially in the last official one [Ava+21]).

• **NTT in $\mathbb{Z}_q[X]/\langle X^n + 1\rangle$ when $q \equiv 1 \pmod{2n}$.**

In the ring $R_{q,f}$ for $f(X) = X^n + 1$, we require n to be a power of 2 and q to be a prime such that $q \equiv 1 \pmod{2n}$. In that case, for multiplication we can use the (full) *negative wrapped convolution-based NTT* (NWC-based NTT), which is described in the next paragraph.

Negative Wrapped Convolution-based NTT. The reason q was chosen in this way is so that the primitive $2n$-th root of unity ψ_{2n} in $\mathbb{Z}_q$ exist. Denote $\psi = (1, \psi_{2n}, \psi_{2n}^2, \ldots, \psi_{2n}^{n-1})$, $\psi^{-1} = (1, \psi_{2n}^{-1}, \psi_{2n}^{-2}, \ldots, \psi_{2n}^{-(n-1)})$ and set $\omega_n = \psi_{2n}^2 \bmod q$. Then, for a polynomial $\mathbf{a}$ of degree $n-1$ (without loss of generality), its "forward" transform $\hat{\mathbf{a}} = \mathrm{NTT}(\mathbf{a})$ can be written as

$$\hat{a}_j = \sum_{i=0}^{n-1} a_i \psi_{2n}^i \omega_n^{ij} \bmod q \ (j = 0, \ldots, n-1)$$

Moreover, its inverse transform $\mathbf{a} = \mathrm{INTT}(\hat{\mathbf{a}})$ can be written as

$$a_i = n^{-1} \psi_{2n}^{-1} \sum_{j=0}^{n-1} \hat{a}_j \omega_n^{-ij} \bmod q \ (i = 0, \ldots, n-1)$$

Remark 5.6. In the notation of [LZ22], this specific NTT is labeled as NTT^{ψ}. However, for simplicity, we refer to it as NTT here, as we exclusively consider this particular case. Moreover, we note that the precise transformations described above are put into practice in the "NTT domain" paragraph of the original Kyber paper [Bos+17].

Proposition 5.1.
(i) For a polynomial $\mathbf{a}$ we have $\mathbf{a} = \text{INTT}(\text{NTT}(\mathbf{a}))$.
(ii) If $\mathbf{c}$ is the negative wrapped convolution of $\mathbf{a}$ and $\mathbf{b}$, then $\text{NTT}(\mathbf{c}) = \text{NTT}(\mathbf{a}) \odot \text{NTT}(\mathbf{b})$.

Negative wrapped convolution-based polynomial multiplication. From the above, it becomes obvious how multiplication of polynomials is performed using NTT (similarly to what we saw for CRT). Particularly, for this ring, NTT is used to compute the negative wrapped convolution (which is equivalent to polynomial multiplication), where the NWC $\mathbf{c} = \mathbf{a} \cdot \mathbf{b} \in \mathbb{Z}_q[X]/\langle X^n + 1\rangle$ can be computed as:

$$\mathbf{c} = \text{INTT}\left(\text{NTT}(\mathbf{a}) \odot \text{NTT}(\mathbf{b})\right).$$

Remark 5.7. We note that the complexity of directly computing the above NTT-based multiplication is $O(n^2)$, and not $O(n \log n)$. This is where fast "FFT-like" techniques (e.g. Cooley-Tukey NTT algorithm) come into play, achieving $O(n \log n)$. Due to lack of space, we cannot present them here and refer to Sections 4 and 5 of [LZ22] for more.

• **NTT in $\mathbb{Z}_q[X]/\langle X^n + 1\rangle$ when $q \equiv 1 \pmod n$.**

We mentioned earlier that the (full) NWC-based NTT required n to be a power of two and q a prime such that $q \equiv 1 \pmod{2n}$, which is not the case if we have q chosen such that $q \equiv 1 \pmod n$. In this case, variants of the above method have to be used, like the "Method based on incomplete FFT trick" (used in "Round 2" Kyber) and the "Method based on splitting polynomial ring" (utilized in "Round 3" Kyber). Again, due to lack of space, we refer to [LZ22] (specifically Section 6) as well as Subsection 4.4.2 of [Lyu20] for the exact details.

However, it is important to mention that they differ from what we saw for CRT and NTT before, in one major aspect. Mainly, for $q \equiv 1 \pmod n$, the base field $\mathbb{Z}_q$ contains n-th primitive roots of unity, but not $2n$-th roots (see remark below). Thus, instead of splitting the polynomial $X^n + 1$ into n polynomials of degree 1, we split it into $n/2$ polynomials of degree 2 modulo q, i.e.

$$X^n + 1 = (X^2 - r_1) \cdot \ldots \cdot (X^2 - r_{n/2}),$$

where $r_1, \ldots, r_{n/2}$ are all the n-th primitive roots of unity.

In that case, the CRR of a polynomial $\mathbf{a}$ (similarly for NTT) is defined as

$$\begin{aligned}\hat{\mathbf{a}} &= \left(\mathbf{a} \bmod (X^2 - r_1), \ldots, \mathbf{a} \bmod (X^2 - r_{n/2})\right) \\ &= (\hat{a}_0 + \hat{a}_1 X, \ldots, \hat{a}_{n-2} + \hat{a}_{n-1} X)\end{aligned}$$

and the only difference in the multiplication procedure is that now, in step 2, the multiplication is performed over the field $R_{q,f}$ with $f = X^2 - r_i$, for $i = 1, \ldots, n/2$ (instead of the point-wise multiplication we had before). For more details on how this works exactly, again we refer to [LZ22] and [Lyu20].

Remark 5.8. We define the sets $\rho^{(j)} = \{r \in \mathbb{Z}_q^* \mid r^j = -1\}$, for j power of 2. The elements of these sets are the j-th roots of -1 (also the $2j$-th primitive roots of $\mathbb{Z}_q$). Moreover, we have $|\rho^{(j)}| \leq j$, as $X^j + 1$ has at most j solutions over the field $\mathbb{Z}_q^*$ (for q prime).

If we set q to be a prime such that $q \equiv 1 \pmod{2j}$, then $|\rho^{(j)}| = j$ (this is a direct result of Proposition 5.20 in [Ηου15]). Thus, by setting q prime and $q \equiv 1 \pmod{2n}$, $X^n + 1$ can be factored into n polynomials of degree 1 modulo q (as it has n roots in the field $\mathbb{Z}_q$).

Working similarly in the case of $q \equiv 1 \pmod n$ (with n even), we have that $|\rho^{(n/2)}| = n/2$ (i.e. $\mathbb{Z}_q$ has n-th roots of unity and not $2n$-th) and $X^n + 1$ can be factored into $n/2$ polynomials of degree 2 modulo q (as it has $n/2$ roots in the field $\mathbb{Z}_q$). For instance, in Kyber [Ava+21] with $n = 256$ and $q = 3329 = 256 * 13 + 1$, we end up with a field that doesn't have 512-th roots of unity, but does have 256-th ones. Moreover, using the CRT for rings we have the following isomorphism (for $r_i \in \rho^{(n/2)}$, i.e. r_i are all the 256-th primitive roots of unity):

$$\mathbb{Z}_{3329}[X]/\langle X^{256} + 1\rangle \cong \prod_{i=0}^{127} \mathbb{Z}_{3329}[X]/\langle X^2 - r_i^2\rangle$$

Chapter 6

CRYSTALS-Kyber

In the concluding chapter of this thesis, it is time for CRYSTALS-Kyber to take center stage. First introduced in 2017 [Bos+17], CRYSTALS-Kyber, a part of the CRYptographic SuiTe for ALgebraic LatticeS (CRYSTALS)[1], was submitted to NIST's post-quantum cryptography (pqc) standardization competition as a portfolio of pq cryptographic primitives, whose security is based on the hardness of MLWE (see Section 4.4).

Particularly, in CRYSTALS-Kyber (Kyber for short), an IND-CCA2 secure key-encapsulation mechanism (Kyber.KEM) is constructed, starting from a IND-CPA secure public-key encryption scheme (Kyber.CPA) using a modified FO transform. Following the creation of Kyber.KEM, again with the use of transformations, IND-CCA2 secure encryption (Kyber.Hybrid), key exchange (Kyber.KE) and authenticated-key exchange (Kyber.AKE) schemes were constructed. In our examination, we emphasize on Kyber.CPA and Kyber.KEM (sometimes denoted just as Kyber, due to its central position in CRYSTALS-Kyber), the core components of the portfolio.

NIST's post-quantum cryptography competition was initiated in 2016, featuring over 80 submissions (encryption/KEM schemes and signature schemes). By 2022, the winners were announced (including CRYSTALS-Kyber) after three rounds of rigorous evaluation, feedback, and refinement. Notably, CRYSTALS-Kyber evolved through these stages, reaching its final version in [Ava+21]. Thus, our analysis in this chapter concentrates on this refined version.[2]

Remark 6.1. In reality there are two IND-CCA distinct notions, IND-CCA1 and IND-CCA2. For the IND-CCA1 game, the attacker is given access to the decapsulation oracle only until the challenge ciphertext c^* arrives (i.e. he is only capable of a non-adaptive attack) whereas for the IND-CCA2 game, he can access the oracle using information from the message c^* (i.e. he is capable of an adaptive attack).

From this distinction it is clear that the "IND-CCA" notion defined in [Bos+17; HHK17; SXY18] is actually referring to IND-CCA2, which is used explicitly in [Ava+21]. From a personal standpoint, we decided to stick with the "IND-CCA" term in Chapter 5 and move on to the "IND-CCA2" term in Chapter 6, so as not to deviate from our sources and confuse the reader.

• Notation

Before we commence, it is imperative to define the notation that we will use for the chapter:

From Bits to Bytes and Back Again. The functions within Kyber operate with byte arrays, i.e. data structures that store a sequence of bytes. Thus, we use relative notation:

$\mathcal{B}$ is the set $\{0, \ldots, 255\}$, i.e. the set of 8-bit unsigned integers (bytes).

[1] CRYSTALS encompasses two portfolios, CRYSTALS-Kyber and CRYSTALS-Dilithium (this containts the signature scheme Dilithium).

[2] In reality, its "final" version should be NIST's corresponding standard (ML-KEM) [Nat23]. However, as it is still just a draft undergoing peer review, we (mostly) consider it non-existent in our analysis.

$\mathcal{B}^k$ is the set of byte arrays of length k (i.e. containing k bytes).

$\mathcal{B}^*$ is the set of byte arrays of arbitrary length (byte streams).

$(a||b)$ denotes the concatenation of two byte arrays a and b.

$a + k$ denotes the byte array starting at byte k of a byte array a (indexing starts at zero). For instance, if we concatenate a byte array b after $a = (01000001, 11000001)$ (i.e. have $(a||b)$), then $b = a + 2$.

Moreover, it is useful to have functions that can convert an array of bytes into an array of bits (and vice versa). This is performed by `BytesToBits` and `BitsToBytes`, as follows:

(i) For a byte array $b = (b_0, \ldots, b_{l-1})$ of length l, `BytesToBits` produces an array of $8l$ bits (with each segment of 8 bits representing a byte in little-endian order[3]), where bit β_i at position i of the output bit array is obtained from the byte $b_{\lfloor i/8 \rfloor}$, by computing

$$\beta_i = \left\lfloor \frac{b_{\lfloor i/8 \rfloor}}{2^{(i \bmod 8)}} \right\rfloor \bmod 2.$$

(ii) For a bit array $\beta = (\beta_0, \ldots, \beta_{l-1})$ of length l (a multiple of 8), `BitsToBytes` produces an array of $l/8$ bytes, in the following manner

$$\texttt{BitsToBytes}(\beta) = \sum_{i=0}^{l-1} \beta_i \cdot 2^{(i \bmod 8)}.$$

Global Constants and Variables. All algorithms and functions have access to two global integer constants: $n = 256$, $q = 3329$. In addition, five global integer variables are used: $k, \eta_1, \eta_2, d_u, d_v$; specified values are set when selecting a parameter set (more in Subsection 6.2.3).

Vectors and Matrices in the Ring. We denote by R the ring $\mathbb{Z}[X]/\langle X^n + 1\rangle$ for $n = 2^{n'-1}$ (thus $X^n + 1 = \Phi_{2^{n'}}(X)$ and $n' = 9$), and by R_q the ring $R/qR = \mathbb{Z}_q[X]/\langle X^n + 1\rangle$.

Moreover, we denote by $\mathbf{v}[i]$ the i-th entry of a vector and with $\mathbf{A}[i][j]$ the entry in row i, column j of a matrix (with index starting at zero).

Modular Reductions.

- For an even (resp. odd) positive integer a, we set $r' = r \bmod^{\pm} a$ as the unique element r' in the range $-\frac{a}{2} < r' \leq \frac{a}{2}$ (resp. $-\frac{a-1}{2} < r' \leq \frac{a-1}{2}$).
- For any positive integer a, we set $r' \bmod^{+} a$ as the unique element r' in the range $0 \leq r' < a$ such that $r' \bmod a$.

When the exact form is not important, the notation $r \bmod a$ is used.

Norms.

- Let w be an element in $\mathbb{Z}_q$. Then, we set $||w||_\infty = |w \bmod^{\pm} q|$.
- Let $w = w_0 + w_1 X + \cdots + w_{n-1} X^{n-1}$ be a polynomial in R. Then, we set

$$||w||_\infty = \max_i ||w_i||_\infty \qquad ||w|| = \sqrt{||w_0||_\infty^2 + \cdots + ||w_{n-1}||_\infty^2}.$$

- Let $\mathbf{w} = (w_1, \ldots, w_k)$ be a vector of polynomials in R^k. Then, we set

$$||\mathbf{w}||_\infty = \max_i ||w_i||_\infty \qquad ||\mathbf{w}|| = \sqrt{||w_1||^2 + \cdots + ||w_k||^2}.$$

Bit-reversal. Let $r = r_0 + r_1 \cdot 2^1 + r_2 \cdot 2^2 + \cdots + r_{i-1} \cdot 2^{i-1}$ ($r_j \in \{0, 1\}$) be an (unsigned) i-th bit integer. Then, we define its bit-reversal as $\texttt{br}_i(r) = r_{i-1} + r_{i-2} \cdot 2^1 + \cdots + r_0 \cdot 2^{i-1}$.

[3]For example, the number 147 is written as (10010011) in big-endian and (11001001) in little-endian format.

ROUNDING. In the preceding chapters, we set $\lceil x \rfloor \doteqdot \lceil x - \frac{1}{2} \rceil$ (where ties are broken downwards) and $\lfloor x \rceil \doteqdot \lfloor x + \frac{1}{2} \rfloor$ (where ties are broken upwards), following the notation of works like [ΔρΑ22; Gal18; Pei16]. However, there exists no worldwide agreement on the usage of these symbols and some employ them in a manner opposite to ours. From a personal standpoint, the above mode of symbolization was chosen, partly because it can be linked to the mnemonic trick depicted in the figure below.

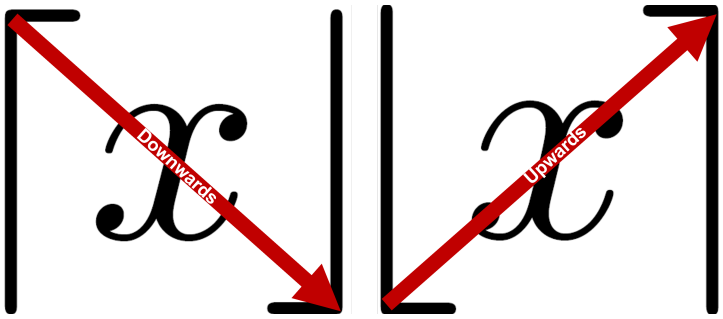


Figure 6.1: Mnemonic trick for notation.

Nevertheless, as Kyber's documentation [Bos+17; Ava+21; Nat23] utilises the opposite notation, we also adopt it throughout this chapter. Particularly, they state that "For an element $x \in \mathbb{Q}$ we denote by $\lceil x \rfloor$ rounding of x to the closest integer with ties being rounded up".

6.1 Auxiliary Algorithms

6.1.1 Cryptographic Functions

Kyber mainly utilizes three symmetric primitives:

- A PseudoRandom Function `PRF`: $\mathcal{B}^{32} \times \mathcal{B} \to \mathcal{B}^*$
- An eXtendable Output Function `XOF`: $\mathcal{B}^* \times \mathcal{B} \times \mathcal{B} \to \mathcal{B}^*$.
- Two hash functions, `H`: $\mathcal{B}^* \to\!\!\to \mathcal{B}^{32}$ and `G`: $\mathcal{B}^* \to \mathcal{B}^{32} \times \mathcal{B}^{32}$.
- A Key Derivation Function `KDF`: $\mathcal{B}^* \to \mathcal{B}^*$.

These primitives are instantiated with certain cryptographic functions accordingly (more on Subsection 6.2.3).

6.1.2 NTT and multiplication

As we are in the ring $R_q = \mathbb{Z}_q[X]/\langle X^n + 1\rangle$ with n a power of two and $q = 3329 = 2^8 \cdot 13 + 1$ adhering to $q \equiv 1 \,(\mathrm{mod}\, n)$, the base field $\mathbb{Z}_q$ contains n-th primitive roots of unity, but no $2n$-th roots. Moreover, $X^n + 1$ of R factors into $n/2$ polynomials of degree 2 modulo q and the NTT of a polynomial $f \in R_q$ is a vector of $n/2$ polynomials of degree one. All of this is evident after our analysis in Subsection 5.3.2 and the Remark 5.8.

However, even though the core concept remains the same, due to the fast "FFT-like" techniques that come into place, the form of NTT will be a bit different than what we saw in that previous analysis. In addition, as the usual "in-place" implementations of NTT (without reordering) output the polynomials of NTT in bit-reversed order, we have to define NTT in the same manner too here.

Particularly, if we set $\zeta = 17$ as the first n-th primitive root of unity modulo q, then the set of all n-th roots of unity can be proven to be $\{\zeta, \zeta^3, \zeta^5, \ldots, \zeta^{255}\}$ (this can be seen easily if one factors the polynomial $X^n + 1$, see also the blog post "The Number Theoretic Transform in Kyber and Dilithium" [link]), and thus we can write (reminder: $n = 256$):

$$X^{256}+1=\prod_{i=0}^{127}(X^2-\zeta^{2i+1})=\prod_{i=0}^{127}(X^2-\zeta^{2\texttt{br}_7(i)+1})$$

NTT AND $\texttt{NTT}$,$\texttt{NTT}^{-1}$ FUNCTIONS. Thus, the NTT of a polynomial $f\in R_q$ is written as:

$$\left(f \bmod (X^2-\zeta^{2\texttt{br}_7(0)+1}),\ldots,f \bmod (X^2-\zeta^{2\texttt{br}_7(127)+1})\right),$$

which is then serialized (canonically) into a vector in $\mathbb{Z}_q^{256}$. Furthermore, it is useful (for technical reasons) to define a function $\texttt{NTT}:R_q\to R_q$ as the bijection mapping $f\in R_q$ to the polynomial[4] that corresponds to the previously mentioned coefficient vector (similarly, its inverse function $\texttt{NTT}^{-1}$).[5] Therefore,

$$\texttt{NTT}(f)=\hat{f}=\hat{f}_0+\hat{f}_1X+\ldots+\hat{f}_{255}X^{255},$$

where $\hat{f}_i$ can be computed by the formulas (see Chapter 6 and Section 8.1 of [LZ22] for more):

$$\hat{f}_{2i}=\sum_{j=0}^{127}f_{2j}\zeta^{(2\texttt{br}_7(i)+1)j},$$

$$\hat{f}_{2i+1}=\sum_{j=0}^{127}f_{2j+1}\zeta^{(2\texttt{br}_7(i)+1)j}.$$

MULTIPLYING ELEMENTS OF R_q. Analogously to what we saw in our analysis, the multiplication of two polynomials $f,g\in R_q$ can be computed as

$$\texttt{NTT}^{-1}(\texttt{NTT}(f)\circ\texttt{NTT}(g)),$$

where $\texttt{NTT}(f)\circ\texttt{NTT}(g)=\hat{f}\circ\hat{g}=\hat{h}$ denotes the basecase multiplication (over the ring $\mathbb{Z}_q[X]/\langle X^2-\zeta^{2\texttt{br}_7(i)+1}\rangle$, for $i=0,\ldots,127$) consisting of the 128 products

$$\hat{h}_{2i}+\hat{h}_{2i+1}X=(\hat{f}_{2i}+\hat{f}_{2i+1}X)(\hat{g}_{2i}+\hat{g}_{2i+1}X) \bmod (X^2-\zeta^{2\texttt{br}_7(i)+1})$$

Remark 6.2. When applying $\texttt{NTT}$, $\texttt{NTT}^{-1}$ to a vector or matrix of elements of R_q, the respective operation is employed to each entry individually. When applying $\circ$ to vectors or matrices, the usual matrix multiplication is being computed, with the individual products of entries being the basecase multiplications above.

6.1.3 Encoding and Decoding

It is obvious from the above that we also need functions that can serialize polynomials to byte arrays, and vice-versa. That is respectively what the functions $\texttt{Encode}_l$ and $\texttt{Decode}_l$ do, for $l<12$ (exact algorithms for them below). In particular, $\texttt{Decode}_l$ deserializes an array of $32l$ bytes into a polynomial $f=f_0+f_1X+\ldots+f_{255}X^{255}$ where f_i are l-bit integers (i.e. $f_i\in\{0,\ldots,2^l-1\}$), and $\texttt{Encode}_l$ reverses this.

$\texttt{Decode}_l:\mathcal{B}^{32l}\to R_q$

(Input) Byte array $B\in\mathcal{B}^{32l}$.
(Output) Polynomial $f\in R_q$.

Step 1. Compute $(\beta_0,\ldots,\beta_{256l-1}):=\texttt{BytesToBits}(B)$.
Step 2. Set $i=0$. While $i\leq 255$, do:
- Compute $f_i:=\sum_{j=0}^{l-1}\beta_{il+j}2^j$ and then set $i:=i+1$.

Step 3. Return $f_0+f_1X+\ldots+f_{255}X^{255}$.

[4] $\hat{f}$ might be written as a polynomial but it has no algerbaic meaning as such.
[5] We remark that with $\texttt{NTT}$ we denote the previous function, whereas with NTT we denote the transform itself.

$\texttt{Encode}_l : R_q \to \mathcal{B}^{32l}$

(Input) Polynomial $f = f_0 + f_1 X + \ldots + f_{255} X^{255} \in R_q$.
(Output) Byte array $B \in \mathcal{B}^{32l}$.

Step 1. Initialize a bit array $\beta = (\beta_0, \ldots, \beta_{256l-1})$.
Step 2. Set $i = 0$. While $i \leq 255$, do:
- Set $j = 0$. While $j \leq l - 1$, do:
  - Compute $\beta_{il+j} = f_i \bmod 2$.
  - Set $f_i \leftarrow (f_i - \beta_{il+j})/2$ and then $j := j + 1$.
- Set $i := i + 1$.

Step 3. Set $B \leftarrow \texttt{BitsToBytes}(\beta)$.
Step 4. Return the byte array B.

We note that, when $\texttt{Encode}_l$ is applied to a vector of polynomials, each polynomial is encoded individually and then the output byte arrays are concatenated.

Example 6.1. The polynomial $f(X) = 2 + 3X + 2X^2 + 3X^3 + 2X^4 + \ldots + 3X^{255}$ (f_i are 2-bit integers) can be encoded into a byte array with $\texttt{Encode}_2$: following the algorithm, we get the bit array[6] $\beta = (0, 1, 1, 1, 0, 1, 1, 1, \ldots, \beta_{511} = 1)$, which we use to attain a byte array of 64 bytes, $B = \texttt{BitsToBytes}(\beta) = (238, \ldots, 238) \in \mathcal{B}^{64}$.

6.1.4 Compression and Decompression

Following our analysis in Subsection 5.2.1, we now define the exact functions `Compress` and `Decompress` used on Kyber. The function $\texttt{Compress}_q(x, d)$ takes an element $x \in \mathbb{Z}_q$ and outputs an integer in $\{0, \ldots, 2^d - 1\}$, with $d < \lceil \log_2(q) \rceil$ (i.e. the input is an integer of $\log_2(q)$ bits and the output is d bits long). On the other hand, $\texttt{Decompress}_q$ is defined as the "inverse" of `Compress`, with $x' = \texttt{Decompress}_q(\texttt{Compress}_q(x, d), d)$ giving us an element close to x, and more specifically, such that

$$|x' - x \bmod^{\pm} q| \leq B_q = \left\lceil \frac{q}{2^{d+1}} \right\rfloor .$$

Particularly, the functions are defined as[7]:

$$\texttt{Compress}_q(x, d) = \lceil (2^d/q) \cdot x \rfloor \bmod^{+} 2^d,$$
$$\texttt{Decompress}_q(x, d) = \lceil (q/2^d) \cdot x \rfloor.$$

Remark 6.3. Aside reducing the ciphertext size by discarding low-order bits, these functions are also used to perform the LWE error correction during encryption and decryption:

- In Kyber.CPAPKE's encryption[8], the computation of the ciphertext part v uses the $\texttt{Decompress}_q$ function ($v := \mathbf{t}^T \mathbf{r} + e_2 + \texttt{Decompress}_q(m, 1)$) to create error tolerance gaps by sending the message bit 0 to 0 and 1 to $\lceil q/2 \rfloor$.
- In Kyber.CPAPKE's decryption, $\texttt{Compress}_q$ is used to decrypt to a 1 if $v - \mathbf{s}^T \mathbf{u}$ is closer to $\lceil q/2 \rfloor$ than to 0, and decrypt to 0 otherwise ($m := \texttt{Compress}_q(v - \mathbf{s}^T \mathbf{u}, 1)$).

6.1.5 Sampling Algorithms

The algorithms of Kyber require two sampling subroutines that can convert a stream of uniformly random bytes into a sample from either the uniform or the binomial distribution.

[6] Remember that the little-endian order is used.
[7] For $x \in R_q$ or $\mathbf{x} \in R_q^k$, the function is executed for each coefficient individually.
[8] The name Kyber.CPA is from the original paper. The PKE scheme is called Kyber.CPAPKE in [Ava+21].

Uniform sampling of NTT representations. The $\texttt{Parse} : \mathcal{B}^* \to R_q$ converts a byte stream $B = b_0, b_1, b_2, \ldots$ into the NTT-representation $\hat{a} = \hat{a}_0 + \hat{a}_1 X + \ldots + \hat{a}_{n-1} X^{n-1} \in R_q$ of a polynomial $a \in R_q$. This is a *deterministic* way to sample elements in R_q that are statistically close to a uniformly random distribution.

When the input stream is statistically close to one that consists of uniformly random bytes, the output is also statistically close to a uniformly random element of R_q (and thus can represent a uniformly random polynomial in R_q as NTT is bijective).

$\texttt{Parse} : \mathcal{B}^* \to R_q$

(Input) Byte stream $B = b_0, b_1, \ldots \in \mathcal{B}*$.
(Output) NTT-representation $\hat{a} \in R_q$ of a polynomial $a \in R_q$.

Step 1. Set $i := 0$ and $j := 0$. While $j < n$, do:
- **(1a)** Set $d_1 := b_i + 256 \cdot (b_{i+1} \bmod^+ 16)$ and $d_2 := \lfloor b_{i+1}/16 \rfloor + 16 \cdot b_{i+2}$.
- **(1b)** If $d_1 < q$, then do:
  - Set $\hat{a}_j := d_1$, and then $j := j + 1$.
- **(1c)** If $d_2 < q$ and $j < n$, then do:
  - Set $\hat{a}_j := d_2$, and then $j := j + 1$.
- **(1d)** Set $i := i + 3$.

Step 2. Return $\hat{a} = \hat{a}_0 + \hat{a}_1 X + \ldots + \hat{a}_{n-1} X^{n-1}$.

Sampling from the Centered Binomial Distribution. Kyber makes use of a different distribution for errors than what we have seen in previous LWE/RLWE schemes. It uses the centered binomial distribution defined as follows:

Definition 6.1. Let η be a positive integer (which will be either 2 or 3 when used in Kyber). Then, the *centered binomial distribution* B_η is defined as:

$$\text{Sample}(a_1, \ldots, a_\eta, b_1, \ldots, b_\eta) \leftarrow \{0,1\}^{2\eta} \text{ and output } \sum_{i=1}^{\eta} (a_i - b_i).$$

Similarly to previous definitions, when we say that a polynomial $f \in R_q$ (or a vector of polynomials) is sampled from B_η, we mean that each coefficient is individually sampled from B_η.[9]

Additionally, for our purposes in Kyber, we need a function that samples a polynomial $f \in R_q$ according to B_η, *deterministically* from 64η bytes of input (the input byte array will always be the output of the pseudorandom function PRF). This function is $\texttt{CBD}_\eta$ and is defined as follows:

$\texttt{CBD}_\eta : \mathcal{B}^{64\eta} \to R_q$

(Input) Byte array $B = (b_0, b_1, \ldots, b_{64\eta-1}) \in \mathcal{B}^{64\eta}$.
(Output) Polynomial $f \in R_q$.

Step 1. Compute $(\beta_0, \ldots, \beta_{512\eta-1}) := \texttt{BytesToBits}(B)$.
Step 2. Set $i = 0$. While $i < n$, do:
- Compute $a := \sum_{j=0}^{\eta-1} \beta_{2i\eta+j}$ and $b := \sum_{j=0}^{\eta-1} \beta_{2i\eta+\eta+j}$.
- Set $f_i := a - b$ and then $i := i + 1$.

Step 3. Return $f_0 + f_1 X + f_2 X^2 + \ldots + f_{n-1} X^{n-1}$.

[9] In [Bos+17] the notation for the binomial distribution had some differences, $u \leftarrow \beta_\eta$ meant that every coefficient of $u \in R$ was generated according to B_η and similarly β_η^k for k-dimensional vectors $\mathbf{v} \in R_q^k$.

Remark 6.4. Although most theoretic results regarding LWE encryption use LWE with Gaussian-like distributions (e.g. Discrete Gaussian), in reality, sampling from such distributions is not efficient (and vulnerable to timing attacks, see [Bru+16; Esp+17; PGY17]). Moreover, as we have seen, one can use a different distribution to sample from without affecting the hardness of the problem (see "Normal Form Optimization" in Subsection 3.5.1). Thus, modern schemes make use of noise distributions that one can easily, efficiently and securely sample from.

In Kyber, we have two such examples: (i) the centered binomial distribution; and (ii) the deterministic uniform noise that is being added by dropping bits when using the $\texttt{Compress}_q$ function (i.e. using the LWR problem we saw in Subsection 5.2.2).

6.2 The Kyber Scheme

6.2.1 Kyber.CPAPKE

As we briefly mentioned in Chapter 5, Kyber.CPAPKE is similar to the Compact Ring-LWE that was introduced in [LPR13b], principally differing in the use of Module-LWE instead of Ring-LWE, along with a different approach in the generation of the matrix and the shortening of ciphertexts by rounding off the low bits.

We note that Kyber.CPAPKE is parametrized by n, q and $k, \eta_1, \eta_2, d_u, d_v$, which will be inherited from Kyber.CCAKEM, when implementing Kyber (as the PKE will be instantiated as a part of the KEM).

For a simpler explanation of the scheme's key elements, we suggest first looking at the original Kyber.CPA scheme from the first paper [Bos+17]. However, here we follow [Ava+21] and formally define Kyber.CPAPKE as the tuple of the following three algorithms (key generation (takes no input), encryption, decryption):

Kyber.CPAPKE.KeyGen()

(Output) A key pair $(pk, sk) \in \mathcal{B}^{12 \cdot k \cdot n/8+32} \times \mathcal{B}^{12 \cdot k \cdot n/8}$.

Step 1. Uniformly sample $d \leftarrow \mathcal{B}^{32}$.
Step 2. Expand d to two pseudorandom seeds $(\rho, \sigma) \leftarrow \texttt{G}(d)$ and set $N := 0$.
Step 2. (*Generate a matrix* $\hat{\mathbf{A}} \in R_q^{k \times k}$ *in* NTT *domain.*)
Set $i, j := 0$ and while $i < k$ and $j < k$, do:
Calculate $\hat{\mathbf{A}}[i][j] := \texttt{Parse}(\texttt{XOF}(\rho, j, i))$.
Step 3. (*Sample* $\mathbf{s} \in R_q^k$ *from* B_{η_1}.)
Set $i := 0$ and while $i < k$, do:
Calculate $\mathbf{s}[i] := \texttt{CBD}_{\eta_1}(\texttt{PRF}(\sigma, N))$ and then $N := N + 1$.
Step 4. (*Sample* $\mathbf{e} \in R_q^k$ *from* B_{η_1}.)
Set $i := 0$ and while $i < k$, do:
Calculate $\mathbf{e}[i] := \texttt{CBD}_{\eta_1}(\texttt{PRF}(\sigma, N))$ and then $N := N + 1$.
Step 5. Compute $\hat{\mathbf{s}} \leftarrow \texttt{NTT}(\mathbf{s})$ and $\hat{\mathbf{e}} \leftarrow \texttt{NTT}(\mathbf{e})$.
Step 6. Compute $\hat{\mathbf{t}} := \hat{\mathbf{A}} \circ \hat{\mathbf{s}} + \hat{\mathbf{e}}$.
Step 7. Compute $pk := (\texttt{Encode}_{12}(\hat{\mathbf{t}} \bmod^+ q) || \rho)$.
Step 8. Compute $sk := \texttt{Encode}_{12}(\hat{\mathbf{s}} \bmod^+ q)$.
Step 9. Return the key pair (pk, sk).

Although we make most comments on the design after presenting the whole scheme, it is important to note that the matrix generation seed ρ was appended into the public key pk,

instead of having the whole matrix $\hat{\mathbf{A}}$, so as to reduce the public key size. However, this also incurs the cost of producing the matrix in every encryption, which increases its running time.

Kyber.CPAPKE.Enc(pk, m, r)

(Input) Tuple $(pk, m, r) \in \mathcal{B}^{12\cdot k\cdot n/8+32} \times \mathcal{B}^{32} \times \mathcal{B}^{32}$ of public key, message and random coins r.
(Output) Ciphertext $c \in \mathcal{B}^{d_u\cdot k\cdot n/8+d_v\cdot n/8}$.

Step 1. Set $N := 0$, compute $\hat{\mathbf{t}} := \mathtt{Decode}_{12}(pk)$ and $\rho := pk + 12 \cdot k \cdot n/8$.
Step 2. (*Generate the matrix* $\hat{\mathbf{A}} \in R_q^{k\times k}$ *in* NTT *domain.*)
Set $i, j := 0$ and while $i < k$ and $j < k$, do:
Calculate $\hat{\mathbf{A}}^T[i][j] := \mathtt{Parse}(\mathtt{XOF}(\rho, i, j))$.
Step 3. (*Sample* $\mathbf{r} \in R_q^k$ *from* B_{η_1}.)
Set $i := 0$ and while $i < k$, do:
Calculate $\mathbf{r}[i] := \mathtt{CBD}_{\eta_1}(\mathtt{PRF}(r, N))$ and then $N := N + 1$.
Step 4. (*Sample* $\mathbf{e}_1 \in R_q^k$ *from* B_{η_2}.)
Set $i := 0$ and while $i < k$, do:
Calculate $\mathbf{e}_1[i] := \mathtt{CBD}_{\eta_2}(\mathtt{PRF}(r, N))$ and then $N := N + 1$.
Step 5. Calculate $e_2 := \mathtt{CBD}_{\eta_2}(\mathtt{PRF}(r, N))$.
Step 6. Compute $\hat{\mathbf{r}} := \mathtt{NTT}(\mathbf{r})$.
Step 7. Compute $\mathbf{u} := \mathtt{NTT}^{-1}(\hat{\mathbf{A}}^T \circ \hat{\mathbf{r}}) + \mathbf{e}_1$.
Step 8. Compute $v := \mathtt{NTT}^{-1}(\hat{\mathbf{t}}^T \circ \hat{\mathbf{r}}) + e_2 + \mathtt{Decompress}_q(\mathtt{Decode}_1(m), 1)$.
Step 9. Calculate $c_1 := \mathtt{Encode}_{d_u}(\mathtt{Compress}_q(\mathbf{u}, d_u))$.
Step 10. Calculate $c_2 := \mathtt{Encode}_{d_v}(\mathtt{Compress}_q(v, d_v))$.
Step 11. Return the ciphertext $c = (c_1, c_2)$.

Kyber.CPAPKE.Dec(sk, c)

(Input) Tuple $(sk, c) \in \mathcal{B}^{12\cdot k\cdot n/8} \times \mathcal{B}^{d_u\cdot k\cdot n/8+d_v\cdot n/8}$.
(Output) Message $m \in \mathcal{B}^{32}$.

Step 1. Calculate $\mathbf{u} := \mathtt{Decompress}_q(\mathtt{Decode}_{d_u}(c), d_u)$.
Step 2. Calculate $v := \mathtt{Decompress}_q(\mathtt{Decode}_{d_v}(c + d_u \cdot k \cdot n/8), d_v)$.
Step 3. Calculate $\hat{\mathbf{s}} := \mathtt{Decode}_{12}(sk)$.
Step 4. Compute $m := \mathtt{Encode}_1(\mathtt{Compress}_q(v - \mathtt{NTT}^{-1}(\hat{\mathbf{s}}^T \circ \mathtt{NTT}(\mathbf{u})), 1))$.
Step 5. Return the message m.

• Comments on the scheme

"Against All Authority" Approach for Matrix Generation. The *against-all-authority* approach of NewHope [Alk+16] was chosen for the generation of the public matrix $\mathbf{A}$. Briefly, this entails having the matrix not be a system parameter, generating it every time using the public key. As we mentioned earlier, this adds the cost of having to produce the matrix during encryption. However, it has the following advantages too:

(i) It avoids discussions about how the uniformly random system parameter was generated, i.e. if it was generated honestly or not, as a fixed parameter could be backdoored.

(ii) It protects against the *all-for-the-price-of-one attack* where an entity discovers an algorithm making the required lattice reduction possible in e.g. a year of computation. The entity then utilises the algorithm to find a short basis of the lattice spanned by $\mathbf{A}$ *once* and uses this to attack all users.

NTT Integration into Kyber. NTT was chosen to be an integral part of Kyber, following the example of NewHope and others, which resulted in the following design choices:

(i) The public matrix is sampled in NTT domain directly (this is why we use $\hat{\mathbf{A}}$, and not $\mathbf{A}$).

(ii) As all multiplications by $\hat{\mathbf{A}}$ have to use NTT, $\hat{\mathbf{t}}$ can also be computed directly in NTT (and all multiplication using $\hat{\mathbf{t}}$ now have to be in NTT form too).

(iii) The secret key sk is stored in NTT domain.

(iv) The ciphertext was not chosen to be also sent in NTT domain, as this form would create a problem when using the $\texttt{Compress}_q$ function.

Hence, the choice of directly integrating NTT to Kyber made the scheme's description a bit more complex, but saved k^2 additional NTT operations in both the key generation and encryption algorithm, increasing efficiency.

6.2.2 Kyber.CCAKEM

As discussed before, Kyber.CCAKEM is an IND-CCA2 secure KEM constructed from the IND-CPA secure Kyber.CPAPKE using a slightly tweaked Fujisaki-Okamoto transform (with "implicit rejection"). It is formally defined as the tuple of the algorithms of key generation (takes no input), encapsulation and decapsulation, described in the figures below.

• Comments on the scheme

KEM Construction. In Chapter 5, we explored scheme transformations. Now, before delving into Kyber.CCAKEM, it is imperative to discuss a bit more about key exchange mechanisms. Generally, a KEM's goal is the establishment of a shared secret key between two parties (e.g. Alice and Bob) in three phases:

- **KeyGen($\cdot$)**: Alice generates a secret *decapsulation key* sk and a shared (with Bob) *encapsulation key* pk.
- **Encaps($\cdot$)**: Bob uses pk to generate a copy of the shared secret K_{B}, along with a ciphertext c, which is sent to Alice.
- **Decaps($\cdot$)**: Alice uses sk and c to compute another copy of the shared secret, K_{A}.

Moreover, it is crucial for Alice to verify that $K_{\mathrm{A}} = K_{\mathrm{B}}$ and this is where sk comes into play. Specifically, sk is used by Alice to decrypt c (created with pk, a message m and random coins r) in order to obtain a message m' and random coins r'. Then, using them, she constructs a ciphertext c' (through encryption) and checks if $c' = c$. If this holds, then it acts as a validation that $K_{\mathrm{A}} = K_{\mathrm{B}}$, given that they will have originated from the same input.

"Implicit rejection" approach. While Chapter 5 introduced implicit rejection, its purpose was not explicitly discussed. The primary motivation is its impact on implementations of Kyber, which are made safe to use even if higher level protocols neglect to verify the return value's correctness.

Kyber.CCAKEM.KeyGen()

(Output) A key pair $(pk, sk) \in \mathcal{B}^{12 \cdot k \cdot n/8+32} \times \mathcal{B}^{24 \cdot k \cdot n/8+96}$.

Step 1. Select $z \leftarrow \mathcal{B}^{32}$ (will be used for implicit rejection).
Step 2. Construct $(pk, sk') :=$ Kyber.CPAPKE.KeyGen().
Step 3. Set $sk := (sk'||pk||H(pk||z))$.
Step 4. Return the key pair (pk, sk).

Kyber.CCAKEM.Enc(pk)

(Input) Public key $pk \in \mathcal{B}^{12 \cdot k \cdot n/8+32}$.
(Output) Tuple $(c, K) \in \mathcal{B}^{d_u \cdot k \cdot n/8+d_v \cdot n/8}$ of ciphertext and shared key.

Step 1. Select $m \leftarrow \mathcal{B}^{32}$.
Step 2. Set $m \leftarrow H(m)$.
Step 3. Compute $(\overline{K}, r) := G(m||H(pk))$.
Step 4. Construct Kyber.CPAPKE.Enc(pk, m, r).
Step 5. Compute $K :=$ KDF$(\overline{K}||H(c))$
Step 6. Return the ciphertext and shared key (c, K).

Kyber.CCAKEM.Dec(c, sk)

(Input) Tuple $(c, sk) \in \mathcal{B}^{d_u \cdot k \cdot n/8+d_v \cdot n/8} \times \mathcal{B}^{24 \cdot k \cdot n/8+96}$ of ciphertext and secret key.
(Output) Shared key $K \in \mathcal{B}^*$.

Step 1. Calculate $pk := sk + 12 \cdot k \cdot n/8$.
Step 2. Calculate $h := sk + 24 \cdot k \cdot n/8 + 32 \in \mathcal{B}^{32}$.
Step 3. Calculate $z := sk + 24 \cdot k \cdot n/8 + 64$.
Step 4. Construct $m' :=$ Kyber.CPAPKE.Dec(sk, c).
Step 5. Compute $(\overline{K}', r') := G(m'||h)$.
Step 6. Construct $c' :=$ Kyber.CPAPKE.Enc(pk, m', r')
Step 7. (*Check if ciphertexts match. If not, "implicitly reject".*)
If $c = c'$, then:
Return $K :=$ KDF$(\overline{K}'||H(c))$.
else:
Return $K :=$ KDF$(z||H(c))$.
Step 8. Return the shared key K.

Remark 6.5. We note that, even though the KEM is IND-CCA2, this does not mean it prevents neither Man-in-the-Middle attacks (MitM) nor Impersonation attacks. For these, more measures need to be taken. For instance, in order to prevent MitM, the authenticated key exchange protocol Kyber.AKE is constructed from Kyber (for more see Section 5 in [Bos+17]).

6.2.3 Kyber Parameter Sets

Kyber is equipped with three parameter sets called Kyber512, Kyber768 and Kyber1024 which are described in the following table, containing also δ (from δ-correctness, see Subsection 5.1.1) and the security level achieved. We provide explanations on δ and security levels later on.

Parameter Set	n	k	q	η_1	η_2	(d_u, d_v)	δ	NIST Security level
Kyber512	256	2	3329	3	2	(10, 4)	2^{-139}	1
Kyber768	256	3	3329	2	2	(10, 4)	2^{-164}	3
Kyber1024	256	4	3329	2	2	(11, 5)	2^{-174}	5

We now provide brief explanations on the above choices:

- n is selected to be 256 in order to encapsulate keys with 256 bits of entropy (getting a message space $\mathcal{M} = \{0,1\}^{256}$ for Kyber.CPAPKE).
- q is set to 3329 as it is the smallest prime satisfying $q \equiv 1 \,(\mathrm{mod}\, n)$ such that the failure probability is negligible.
- k is the main mechanism with which security is scaled in Kyber, as it determines the dimensions of $\mathbf{s}, \mathbf{e}, \hat{\mathbf{A}}, \mathbf{r}, \mathbf{e}_1, \mathbf{e}_2$ by fixing the lattice dimension as a multiple of n.
- η_1, η_2, d_u, d_v were chosen afterwards with trial and error in order to have a balance between security, ciphertext size and failure probability.

Remark 6.6. In Kyber.CPAPKE.Enc, the function $\mathtt{Compress}_q$ adds implicit noise, which can also be interpreted as increasing the noise of $\mathbf{e}_1$ and e_2. Thus, one can increase the noise of the other secret terms $\mathbf{s}, \mathbf{e}, \mathbf{r}$ to reach the level of $\mathbf{e}_1$ plus the deterministic noise, while keeping the decryption error probability in check. This approach is utilized only in Kyber512, which is why $\eta_1 > \eta_2$ on it specifically. Moreover, as discussed in Chapter 5, this trick is similar to the LWR assumption and has the effect of slightly increasing the hardness of the underlying problem (and thus its security in bits). Further details are presented in the following section.

Instantiating Symmetric Primitives. [Ava+21] includes two variants of Kyber with the selection of symmetric primitives determining the variant. One has newer but slower (due to lack of hardware support) standardized functions, taken from FIPS-202 [Dwo15], and the other has older standardized functions (that are supported by most systems today), called *90's variant*. We decided to describe here only the first variant (see table below) as [Nat23] also seems to follow it (with some differences), and refer to [Ava+21] for more information on the other, as well as the reasons those particular functions were chosen.

XOF	H	G	PRF(s, b)	KDF
SHAKE-128	SHA3-256	SHA3-512	SHAKE-256$(s\|\|b)$	SHAKE-128

6.2.4 Correctness and Efficiency of Kyber

Regarding Correctness. The decryption error calculation that resulted in the above values of the failure probability δ is based on an analogue of Theorem 1 of [Bos+17], the proposition below and computations done with the Kyber.py script of [Ava+21]. We refer to Section 4.7 of [Lyu20] for a theoretical example that can aid in understanding how all these are combined.

Proposition 6.1. If Kyber.CPAPKE is $(1-\delta)$-correct and G is a random oracle, then Kyber.CCAKEM is also $(1-\delta)$-correct.

Remark 6.7. We saw an analogue of the above proposition on Theorem 5.2 of [HHK17].

Regarding Efficiency. We refer to Section 2 of [Ava+21] for a detailed performance analysis of Kyber. However, as most of these values are relatively old and are highly depended on the system tested, we choose to only discuss more regarding the sizes (and not speeds). Particularly, we decided to simply include a figure that pits Kyber against other Round-3 schemes in an effort to show where the scheme stands in the PQC realm. For the exact values see also Appendix D of [Ala+22].

We do note however that even those results can be considered old (e.g. SIKE, an isogeny-based candidate, was badly broken in 2022 by the preliminary version of [CD23]).

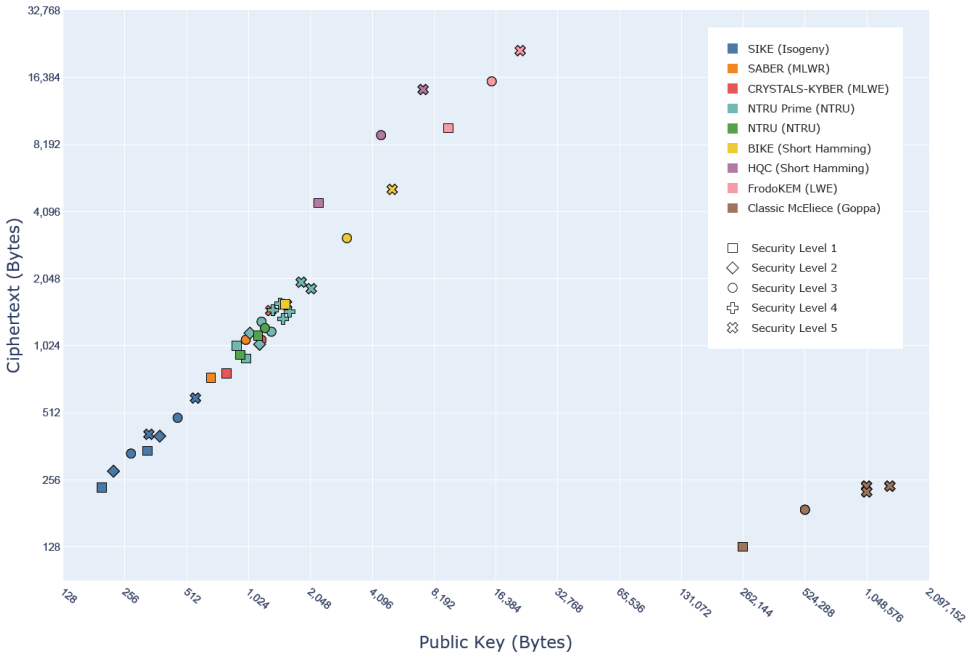


Figure 6.2: Ciphertext and public key sizes for Round 3 PQC KEMs (taken from [DMG21]).

Advantages of Kyber. We outline that Kyber has two qualities that differentiate it from other post-quantum schemes: its scalability (simply by changing the parameters a bit) and its simple implementation. For more advantages and a comparison to other schemes, we refer to Section 6 of [Ava+21].

6.3 Security Analysis

We now continue with a brief presentation of NIST's security levels in order to then delve into an examination of Kyber's security, following Sections 4 and 5 of [Ava+21], as well as some newer results.

6.3.1 NIST security Levels

NIST recognizes significant uncertainties in gauging the security of post-quantum schemes, stemming from the possibility of new quantum algorithms and the difficulty to predict the future performance of quantum computers . Thus, new broader security categories had to be defined instead of using precise estimates in "bits of security". To this end, each category below is defined by a "reference primitive" that is expected to offer sufficient resistance to quantum cryptanalysis, and whose security sets a floor for various metrics deemed relevant by NIST for practical security.

For instance, in order for a pq scheme to belong in security category 1, its relevant security definition must be susceptible only to attacks that require computation resources comparable to

Security Category	Corresponding Attack Type	Example
1	Key search on block cipher with 128-bit key	AES-128
2	Collision search on 256-bit hash function	SHA3-256
3	Key search on block cipher with 192-bit key	AES-192
4	Collision search on 384-bit hash function	SHA3-384
5	Key search on block cipher with 256-bit key	AES-256

or greater than those required for "Key search on block cipher with 128-bit key", exemplified by AES-128.

Moreover, on the above definition, computational resources may be measured using various metrics, such as the number of classical elementary operations or quantum circuit size. For a cryptosystem to meet the specified security requirements, any attack must demand computational resources equal to or surpass the stated threshold across all metrics considered relevant to practical security by NIST. More details on these metrics can be found in the initial call for proposals [ST16], as well as [Ala+22].

6.3.2 Expected Security

Before we begin our analysis, we properly define the (decisional) MLWE problem with the error distribution that was chosen for Kyber:

Definition 6.2. (Module-LWE problem in Kyber)
Given m independent samples $(\mathbf{a}_i, b_i) \leftarrow R_q^k \times R_q$ where every sample was either drawn from:

1. a distribution where $\mathbf{a}_i \leftarrow R_q^k$ is uniform, $b_i = \mathbf{a}_i^T\mathbf{s} + e_i$ with $\mathbf{s} \leftarrow B_\eta^k$ common to all samples and $e_i \leftarrow B_\eta$ fresh for every sample.
2. the uniform distribution

distinguish which is the case (with non-negligible advantage).

Suppose now that we are trying to prove that a scheme P is secure (e.g. is IND-CPA secure) under the assumption that a problem S is hard. Then, we have to show that the probability of an adversary A breaking that scheme (e.g. for IND-CPA security, the probability of an adversary winning the IND-CPA game) is smaller than the probability of another adversary B solving the problem S. Then, due to the conjecture that S is hard, the probability of B solving it is small, and thus the possibility of A breaking the (e.g. IND-CPA) security of the scheme is too.

Moreover, if hash functions are also included in the scheme, then the above becomes a bit more complex. One way to "bypass" this, is to model hash functions as random oracles and prove its security in the random oracle model (ROM) (similarly, QROM).

In Kyber, we first want to prove that Kyber.CPAPKE is IND-CPA secure (and then use the corresponding FO transformation to get an IND-CCA secure KEM). Therefore, following the remarks above, this translates to showing that the advantage of an adversary A in winning the IND-CPA game (defined in Section 5.1), $\mathrm{Adv}_{\mathrm{PKE}}^{\text{IND-CPA}}(\mathrm{A})$, is smaller than the advantage of an adversary B solving the MLWE problem (i.e. distinguishing MLWE samples from uniform), $\mathrm{Adv}_{m,k,\eta}^{\mathrm{mlwe}}(\mathrm{B}) \doteqdot$

$$\left|\Pr\left[b' = 1 : \begin{array}{c}\mathbf{A} \leftarrow R_q^{m\times k};\ (\mathbf{s},\mathbf{e}) \leftarrow \beta_\eta^k \times \beta_\eta^m; \\ \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e};\ b' \leftarrow \mathrm{B}(\mathbf{A},\mathbf{b})\end{array}\right] - \Pr\left[b' = 1 : \begin{array}{c}\mathbf{A} \leftarrow R_q^{m\times k}; \\ \mathbf{b} \leftarrow R_q^m;\ b' \leftarrow \mathrm{B}(\mathbf{A},\mathbf{b})\end{array}\right]\right|.$$

Therefore, in this subsection, we mention results from such reductions (i.e. that reduce Kyber's security to solving MLWE) in the ROM and QROM (non-tight result). Furthermore, with those reductions at hand, potential avenues for compromising the security of Kyber (i.e. theoretical model-based threats) are limited to the following:

1. Attacking MLWE

2. Attacking the symmetric primitives used
3. Exploit the non-tightness (of the QROM) in the decryption failure probability of Kyber.CPAPKE with quantum attacks.
4. Exploit the non-tightness (of the QROM) between quantum attacks against MLWE and quantum attacks against Kyber.

Hence, with only the above attacks in mind, we present Kyber's security estimates in the end of the subsection, along with some additional properties and real-world vulnerabilities when implementing it.

• Security assumption - Reductions

Reduction in the ROM. We now provide the proper (tight) reductions from MLWE in the ROM, for Kyber.CPAPKE and Kyber.CCAKEM:

Theorem 6.1. (Theorem 1 of [Ava+21])
Suppose `XOF` and `G` are random oracles. For any adversary A, there exist adversaries B and C, of roughly the same running time as that of A, such that

$$\mathrm{Adv}^{\mathrm{cpa}}_{\mathsf{Kyber.CPAPKE}}(\mathrm{A}) \leq 2 \cdot \mathrm{Adv}^{\mathrm{mlwe}}_{k+1,k,\eta}(\mathrm{B}) + \mathrm{Adv}^{\mathrm{prf}}_{\mathtt{PRF}}(\mathrm{C}).$$

Proof. Briefly, this holds because, under the MLWE assumption, public-key and ciphertext are pseudorandom. We also recommend looking at Theorem 2 of [Bos+17], which proves a similar result for a modified (and less complex) PKE, Kyber.CPA'.

Theorem 6.2. (Theorem 2 of [Ava+21])
Suppose `XOF`, `H` and `G` are random oracles. For any classical adversary A that makes at most q_{RO} many queries to the random oracles `XOF`, `H` and `G`, there exist adversaries B and C, with roughly the same running time as that of A, such that

$$\mathrm{Adv}^{\mathrm{cca}}_{\mathsf{Kyber.CCAKEM}}(\mathrm{A}) \leq 2 \cdot \mathrm{Adv}^{\mathrm{mlwe}}_{k+1,k,\eta}(\mathrm{B}) + \mathrm{Adv}^{\mathrm{prf}}_{\mathtt{PRF}}(\mathrm{C}) + 4q_{\mathrm{RO}}\delta.$$

Proof. The result is achieved by combining results from [HHK17] and the previous theorem (and optimizing constants appearing in the bound). We recommend also to look again at the end of Subsection 5.1.2 for some reminders on how such transformations affect the bounds.

Remark 6.8. The above security bound is tight because the term $4q_{\mathrm{RO}}\delta$ is negligible due to the effect of Kyber.CPAPKE's "very small" decryption failure probability δ. Particularly, the parameter sets of level 3 and level 5 have $\delta \approx 2^{-164}$ and $\delta \approx 2^{-174}$, respectively. These are values smaller than 2^{-160}, which means that, according to [Ava+21]: "if 2^{80} instances of Kyber were run every second from now until our sun becomes a white dwarf, the odds still heavily favor there never being a decryption failure".

Reduction in the QROM. For QROM, things are more fluid. Specifically, the security bound one can derive from the transformation (IND-CPA PKE to IND-CCA KEM) is non-tight and thus can only serve as an asymptotic indication of Kyber.CCAKEM's IND-CCA security in the QROM. However, under some non-standard assumptions (namely "pseudorandomness" and "statistical disjointness"), a tight reduction can be achieved (see Subsection 4.3.2 of [Ava+21] for more).

Theorem 6.3. (Theorem 3 of [Ava+21])
Suppose `XOF`, `H` and `G` are random oracles. For any quantum adversary A that makes at most q_{RO} many queries to the random oracles `XOF`, `H` and `G`, there exist quantum adversaries B and C, with roughly the same running time as that of A, such that

$$\mathrm{Adv}^{\mathrm{cca}}_{\mathsf{Kyber.CCAKEM}}(\mathrm{A}) \leq 4q_{\mathrm{RO}} \cdot \sqrt{\mathrm{Adv}^{\mathrm{mlwe}}_{k+1,k,\eta}(\mathrm{B})} + \mathrm{Adv}^{\mathrm{prf}}_{\mathtt{PRF}}(\mathrm{C}) + 8q^2_{\mathrm{RO}}\delta.$$

Proof. Again, we refer the reader to [HHK17] and [Sei18] to see how these bounds are calculated for such transformations.

REDUCTION FOR KYBER512. For Kyber512, more analysis needs to be done, as it is being affected by the deterministic noise added by $\texttt{Compress}_q$. However, due to lack of space, we only mention that its impact is an additional 6 bits of security with respect to the currently-best attacks (for more see Subsection 4.4 of [Ava+21]).

• **Estimated security strength**

When calculating the estimated security strength, one considers only the theoretical avenues of attack. From these avenues,

- The 2nd one (attacking the symmetric primitives) need not be considered, as the scheme employs standardized and previously analyzed secure functions (which can be replaced in the event of a major breakthrough or vulnerability).
- The 3rd one (exploiting the non-tightness in δ) can also be excluded, as the decryption-failure probability is so small.
- The 4th one (exploiting the non-tightness between quantum attacks against MLWE and quantum attacks against Kyber) is unlikely to matter in practice, as this non-tightness can be eliminated with a (reasonable, but non-standard) assumption.

Thus, only the 1st avenue remains (i.e. attacking MLWE) and the security assessment below is derived by the cost estimates of the best known attacks against the Module-LWE problem. We discuss more on how these cost estimates were computed in the next subsection.

	Kyber512	Kyber768	Kyber1024
NIST Security Level	1	3	5
Core-SVP Methodology (Primal Attack Only, older and unrefined)			
Lattice Attack Dim. d	999	1419	1885
BKZ-Blocksize b	406	626	878
Core-SVP Classical Hardness	118	183	256
Core-SVP Quantum Hardness	107	166	232
Refined Estimate for Classical Attacks (see Section 5.2 in [Ava+21])			
Lattice attack dim. d	1025	1467	1918
BKZ-Blocksize b	413	637	894
Sieving dim. $b' = b - d_{4f}$	375	586	829
$\log_2$**(gates)**	**151.5**	**215.1**	**287.3**
$\log_2$(memory in bits)	93.8	138.5	189.7

Table 6.1: Classical and quantum hardness of the different proposed parameter sets of Kyber, along with the corresponding NIST security level (taken from [Ava+21]).

For now, from the table above, we choose to only focus on the "$\log_2$(gates)" row. Taking for example the Kyber512 column, the value signifies that recovering a key from this scheme is estimated to require $2^{151.5}$ classical gates. Similarly, the "classical gate counts" estimate for the optimal key recovery on AES-128 is 2^{143}. Thus, as Kyber512's count is quite larger, it belongs in NIST's security category 1. The same comparison can be performed for Kyber768 and Kyber1024 by noting that key recovery for AES-192 needs 2^{207} classical gates and AES-256, 2^{272}. We also note that a similar count of quantum gates is not necessary for Kyber's security claims (as the quantum speed-ups are tenuous for the attacks that these costs are estimated from).

• **Additional security properties and concerns**

Forward Secrecy. We simply note that Kyber can be useful for applications that utilize frequent key generations to achieve forward secrecy (as Kyber.CCAKEM.KeyGen$(\cdot)$ is very efficient).

Extra Vulnerabilities. As stated before, aside from the four potential "theoretical attack avenues", Kyber can also be susceptible to *side-channel attacks* (i.e. attacks that exploit information leaked from a physical system) if it is implemented thoughtlessly (like almost all schemes). For example, it is susceptible to differential attacks if it does not have dedicated protection.

Moreover, *multi-target attacks* (i.e. attacks targeting multiple users) is another point of interest when implementing Kyber, as well as *resilience to misuse*. In accordance with NIST's request for comments on these (from every submission to the competition), more information on these attacks (as well as the side-channel ones) can be found in Section 4.5 of [Ava+21].

6.3.3 Attacks against MLWE

Overall, Module-LWE can be attacked through two primary strategies:

(i) Exploiting the structure of module lattices (or ideal lattices).

(ii) Attacking MLWE as an LWE problem.

At the moment, the best known attacks do not make use of the extra structure in the lattice, so we abstain from providing detailed explanations on them. However the interested reader is referred to Subsection 5.3.1 in [Ava+21], as well as to Léo Ducas' presentation in [Ins20] where he explores the realm of classical and quantum algorithms for algebraic lattices.

• **Attacks on MLWE as an LWE problem**

In general, there are many algorithms that manage to solve LWE. For a reference, we highlight several useful sources: the presentation of Martin R. Albrecht in [Ins20], the survey of [APS15] which accompanies the "lwe-estimator" (useful tool for estimating the cost of algorithms that attack LWE), as well as [Alb+18] and the accompanying repository [link] (this is focused on LWE and NTRU schemes proposed in NIST's competition).

From these algorithms, the only ones that hold significance are two attacks that utilize the BKZ algorithm (and its refinements), called *primal attack* and *dual attack*, respectively. The exact form of these attacks is not important for our explanation, but it is important for the actual estimation of the costs in Table 6.1 (for more, see Section 5.1 of [Ava+21]).

Remark 6.9. At this point, it is crucial to recall certain key details regarding BKZ algorithms (see also Subsection 2.2.2 for a brief reminder). Primarily, the algorithms work by reducing a lattice basis using an SVP oracle in a smaller dimension b. In fact, the number of calls to that oracle has been proven to be polynomial, but an exact value is difficult to be computed.

Moreover, we remark that the efficiency of BKZ algorithms is also highly dependant on the type of SVP oracle used (enumeration or a sieving algorithm). As we discussed in Section 2.2, these algorithms have different performance characteristics and thus it is difficult to compare them (and choose the best one to be used as an oracle). Importantly, we remark that:

(i) Enumeration algorithms have super-exponential running times and sieving algorithms exponential. Hence, sieving is asymptotically faster.

(ii) Sieving algorithms need exponential memory, whereas enumeration algorithms require only small amounts of memory (albeit this difference will not play a role in our latter analysis).

(iii) In earlier experiments of implementing BKZ, enumeration algorithms were faster in "small" dimensions and thus it was important to know at what dimension sieving gets ahead in practice. At the time Kyber was introduced, sieving was slower for dimensions up to $b \approx 130$. However, latter techniques managed to reach $b \approx 80$, outperforming enumeration

in practice. Specifically, one of the most important optimizations was the *dimensions-for-free technique*, allowing to solve SVP in dimension b by sieving in a smaller dimension $b' = b - d_{4f}$.

CORE-SVP HARDNESS. For analysing the security of Kyber under this attack, even from the first paper, the approach of [Alk+16] was used. Specifically, this entails ignoring the polynomial factor of calls to the SVP oracle and evaluating only the *core SVP hardness*, considering only the cost of a single call in dimension b. Moreover, again following [Alk+16], we imagine that we work in the *RAM Model*, i.e. assuming that access into even exponentially large memory is free (similarly QRAM when utilizing algorithms with quantum speed ups).

For the analysis, we choose the sieving algorithms to predict the core hardness and argue that, for the targeted dimension, enumeration algorithms are expected to be greatly slower than sieving. Specifically, we use $2^{0.292b}$ as the classical and $2^{0.265b}$ as the quantum cost estimate of both the primal and dual attacks with block size (dimension) b.[10] Following the above analysis and remarks, we get the first values of Table 6.1.

However, this first methodology, although useful at the beginning of NIST's competition (and informative even now), is too rough to produce accurate security estimates now (considering the optimizations on sieving we mentioned in the remark above). Thus, more refined estimates had to be given.

BEYOND CORE-SVP HARDNESS. Although we abstain from further expounding on the matter ourselves, these more refined estimates are computed in Section 5.2 of [Ava+21], where detailed explanations are given on how the values in the "Refined estimates" part of Table 6.1 came to be.

Moreover, the authors of [Ava+21] gave in the subsequent section, Sextion 5.3, a "research direction list" with open problems that could have an effect on the above refined estimates. There, they concluded that the refined estimates could move by a factor somewhere between 2^{-16} and 2^{14}.

• Newer Results and the uncertainty of PQC schemes

This last remark could give one a sense of uncertainty for Kyber, as if the estimates move by a factor of 2^{-16} in the following years, then e.g. "$\log_2$(gates)" for Kyber512 would become 135.5, falling below NIST's threshold.

Additionally, aside from this, there have been improved attacks in recent years that have challenged Kyber's security. Particularly, we refer to an "improved dual attack" by the MATZOV unit of the IDF [MAT22] which stated that the security of Kyber512 could be down to 137.5 (and similarly for other LWE schemes and parameter sets). However, an even more recent paper refutes the central heuristics of [MAT22], and concludes that its effectiveness is significantly overestimated [DP23]. Thus, KYBER is "saved", but even so, this attack (and others like it) happening months before the standardization of KYBER does further amplify this sense of uncertainty in its security.

However, this is uncertainty holds for all PQC schemes, owing to their recent emergence in cryptography without ample time for comprehensive testing. This is why, even though introducing post-quantum defenses as soon as possible is important for long term security, these post-quantum schemes will not be implemented alone. Organizations like the french ANSSI (i.e. the french National Agency for the Security of Information Systems) and the german BSI (i.e. the german Federal Office for Information Security) advocate *hybridization*, which is the use of "hybrid schemes" combining a "pre-quantum" and a post-quantum scheme [ANS22].

[10] The attacks cost estimates are taken form [Ava+21]'s analysis. Moreover, we note that the complexity goes down a bit in quantum attacks due to Grover's quantum search algorithm. However, more precise analysis shows that the actual quantum speed-up is tenuous.

6.4 Kyber in the Real World: X25519Kyber728

In this context of hybridization, we now present a real-world application of Kyber, combined with the pre-quantum X25519 (an Elliptic Curve Diffie-Hellman (ECDH) key exchange using "Curve25519"). To do that, we first perform a brief introduction into the fundamentals of elliptic curves and elliptic curve cryptography, followed by a concise overview of ECDH and X25519. Finally, we conclude this section (and thesis) with a presentation of the hybrid (or "composite") scheme X25519Kyber728, which is already used in Google Chrome's security.

6.4.1 Essential Background on Elliptic Curves

We begin with a definition of curves over a field K, and then define the well-known Weierstrass equation of an elliptic curve:

Definition 6.3. Let K be a field with $\bar{K}$ its algebraic closure[11] and $\mathbb{P}^n_K$ denote the projective space of dimension n over K. Moreover, let also $f(x,y) \in K[x,y]\backslash K$ and $F(x,y,z) \in K[x,y,z]$ be a homogeneous polynomial of degree ≥ 1. Then :

(a_1) An *algebraic (plane) curve* over K, defined by $f(x,y)$, is the set
$$V_f = \{(x,y) \in \bar{K}^2 / f(x,y) = 0\}$$

(a_2) If L is a field such that $K \subseteq L \subseteq \bar{K}$, the *set of the L-rational points of* V_f is the set
$$V_f(L) = \{(x,y) \in L^2 / f(x,y) = 0\}$$

(b_1) A *projective algebraic (plane) curve* over K, defined by $F(x,y,z)$, is the set
$$V_F = \{P \in \mathbb{P}^2_K / f(P) = 0\}$$

(b_2) If L is a field such that $K \subseteq L \subseteq \bar{K}$, the *set of the L-rational points of* V_F is the set
$$V_F(L) = V_F \cap \mathbb{P}^2_L$$

Definition 6.4. A *Weierstrass equation* of an elliptic curve E over a field K is
$$E: \ y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$
where $a_1, a_2, a_3, a_4, a_6 \in K$ and $\Delta \neq 0$ where Δ denotes the discriminant of E.

Remarks.

(i) Let $f(x,y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6$, with $a_1, a_2, a_3, a_4, a_6 \in K$ and $\Delta \neq 0$. Then, we also say that V_f is an *elliptic curve* over K *in Weierstrass form*.

(ii) The condition $\Delta \neq 0$ ensures that V_E has no singular point. If $char(K) \neq 2, 3$ it can be proven that $\Delta \neq 0$ if-f $4a_4^3 + 27a_6^2 = 0$. For a detailed proof we refer the reader to [Was08].

With the above equation we have not given the definition of an elliptic curve[12] but only given an equation of this object. It should be known that an elliptic curve is an abstract object with many models, one of which is the Weierstrass equation (others include the Edwards model, the Montgomery model etc).

Moreover, we shall often "confuse" an elliptic curve and its equation but one has to keep in mind that abstract curve $\neq$ a model of a curve $\neq$ an equation of the curve [Rit12].

Next, we mention a more complete version of the curve (in terms of Bézout Theorem[13]), as described by the projective Weierstrass equation:

Definition 6.5. A *(projective) Weierstrass equation* of an elliptic curve E over a field K is
$$\tilde{E}: \ y^2z + a_1xyz + a_3yz^2 = x^3 + a_2x^2z + a_4xz^2 + a_6z^3,$$
where $a_1, a_2, a_3, a_4, a_6 \in K$ and $\Delta \neq 0$ where Δ denotes the discriminant of E.

[11] An algebraic closure is an algebraic extension of K that is algebraically closed. [Was08; Sil09]

[12] (Abstract definition) An *elliptic curve* over a field K is a projective non-singular curve of genus 1 with a K-rational point O. [Was08; Sil09]

[13] For more information, the reader is referred to [Was08; Sil09]

Remark 6.10. By defining the homogeneous polynomial $F(x, y, z)$ associated with f (i.e. such that $F(x, y, 1) = f(x, y)$), we can obtain the projective version of the affine curve in Definition 6.4. We can also say that V_F is an elliptic curve over K in Weierstrass form.

Additionally, we observe that this version is more complete by distinguishing the points of V_F. Namely, we have the affine points of V_F (i.e. the ones with $z \neq 0$) and the points at infinity of V_F (i.e. the ones with $z = 0$). More precisely, by finding a representative $z = 1$, we can easily see that the affine points of V_F are the points of V_f. Also, letting $z = 0$ in the equation, we get $x^3 = 0$ and we see that the curve has a unique point at infinity, which we will denote $O = (0 : 1 : 0)$. Lastly, one can easily prove that the point O is non singular and therefore V_F is non-singular.

Finally, when $char(K) \neq 2, 3$, by using morphisms one can derive a simplified Weierstrass model of the form $y^2 = x^3 + ax + b$ (or $y^2z = x^3 + axz^2 + bz^3$ for the projective version), where $a, b \in K$ such that $4a^3 + 27b^2 \neq 0$. For more information the reader is referred to [Was08; Sil09].

• Operations on points of elliptic curves

We can define[14] an addition law for points of elliptic curves over K of simplified Weierstrass form (with $char(K) \neq 2, 3$) as described below:

Let $P, Q \in V_F$ be two points and E_{PQ} be the line connecting them (tangent to V_F if $P = Q$). Let R be the third point of intersection of L with V_F by Bézout and L' be the line connecting R and O. Then, $P + Q$ is the residual point of the intersection of L' and V_F.

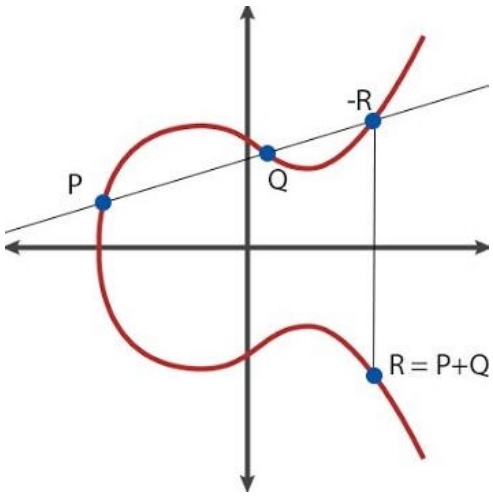


Figure 6.3: Example - Addition law for Weierstrass elliptic curves. [link]

It can be proven that the points of an elliptic curve form a commutative group under this addition law, with the point at infinity O as the identity element and the inverse of a point $P = (a : b : 1)$ being its symmetric about the x-axis, $-P = (a : -b : 1)$. Moreover, the proposition below can give us the coordinates of the point $P + Q$ when adding some P and Q:

Proposition 6.2. Let $P_i = (x_i : y_i : 1)$ $(i = 1, 2, 3)$ be points of an elliptic curve V_F in simplified Weierstrass form such that $P_1 + P_2 = P_3$. Then, $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = -\lambda x_3 - \mu$, where

· If $P_1 \neq P_2$, $\lambda = \dfrac{y_1 - y_2}{x_1 - x_2}$ and $\mu = y_1 - \lambda x_1$

· If $P_1 = P_2$, $\lambda = \dfrac{3x_1^2 + a}{2y_1}$ and $\mu = y_1 - \lambda x_1$

[14]For a more detailed description of the concept, [Was08; Sil09] are suggested.

Finally, we can also define another operation with these points, the scalar multiplication:

$$nP = \underbrace{P + \ldots + P}_{\text{n times}}$$

Remark 6.11. Elliptic curve cryptography (ECC) relies on two more important families of curves, Edwards curves and Montgomery curves, with each of them having their own corresponding operation like the above. *Edwards curves*, introduced by Harold M. Edwards, offer efficiency and security advantages. *Montgomery curves*, developed by Peter L. Montgomery, provide faster computations and enhanced resistance against side-channel attacks.

Moreover, Montgomery curves can utilise the *Montgomery ladder algorithm* which efficiently computes scalar multiples on them. This algorithm is not only fast but is also resistant to side-channel attacks, which makes it useful in real-world applications. We also note that it can work with only using the x-coordinate of a point and calculating the y-coordinate once in the end of the algorithm. For more information, we refer again to [BL17] and [CS17].

6.4.2 Elliptic Curve Diffie-Hellman and X25519

After the previous brief introduction to the fundamentals of ECC, we present the analogue of Diffie-Hellman Key Exchange, the *Elliptic Curve Diffie-Hellman (ECDH)*:

*** Elliptic Curve Diffie-Hellman ***

Suppose that Alice and Bob want to construct a shared key (for use in a symmetric scheme), communicating through an unsafe channel: First, they agree to use a specific finite field $\mathbb{F}_q$ and an elliptic curve V_E over $\mathbb{F}_q$. Then, they choose a point $P \in V_E(\mathbb{F}_q)$ which generates a sufficiently large subgroup of $V_E(\mathbb{F}_q)$, and adhere to the following:

Step 1. Alice chooses a random integer a and calculates $Q_{\mathrm{A}} = aP$.
She keeps a secret and sends Q_{A} to Bob.

Step 2. Bob chooses a random integer b and calculates $Q_{\mathrm{B}} = bP$.
He keeps b secret and sends Q_{B} to Alice.

Step 3. Alice calculates $aQ_{\mathrm{B}} = abP$ and Bob computes $bQ_{\mathrm{A}} = baP$.

Thus, Alice and Bob have calculated the same point $K = abP = baP$ which acts as their shared secret with which they can compute a shared secret key (e.g. using a key-derivation function).

Analogously to the standard Diffie-Hellman, the secrecy of K relies on the difficulty of computing abP knowing only aP and bP (this is the Diffie-Hellman problem on Elliptic Curves), which can be proven to be hard if the Elliptic Curve Discrete Logarithm Problem (ECDLP) is hard. For more information, we refer to [Rit12] and Section XI.5 of [Sil09].

• `X25519` function

Of particular interest to us is Curve25519 (i.e. the Montgomery curve $y^2 = x^3 + 486662x^2 + x$ defined in [Ber06]), which is recommended by the Internet Research Task Force (IRTF) [LHT16] for use in cryptographic applications.[15] As we mentioned before, these types of curves exhibit desirable properties, including efficient constant-time implementation and resilience against various side-channel attacks like timing and cache attacks. Thus, it offers a high level of practical security in cryptographic applications, including Transport Layer Security (TLS).

More importantly, we are interested in combining Diffie–Hellman and this curve in order to get an efficient, secure protocol (also called X25519). This is where the function `X25519` comes

[15] Its name originates from the prime number it uses for the underlying field, $p = 2^{255} - 19$.

in, which basically performs scalar multiplication on the Montgomery form of Curve25519 (this is used when implementing Diffie-Hellman). Specifically, it takes a scalar and an x-coordinate as inputs and produces an x-coordinate as output (inputs and outputs are 32-byte strings, though internally it works with integers). In an ECDH protocol (where $\mathbb{F}_p$ and $P \in V_E(\mathbb{F}_p)$ are already selected, along with Curve25519) its functionality is described in the process below (fFor more, see [LHT16]):

1. Alice generates 32 random bytes in $a[0], \ldots, a[31]$ and transmits $K_A = \texttt{X25519}(a, 9)$ to Bob, where 9 is the u-coordinate of the point P chosen for ECDH and is encoded as a byte with value 9, followed by 31 zero bytes.
2. Bob generates 32 random bytes in $b[0], \ldots, b[31]$ and calculates $K_B = \texttt{X25519}(b, 9)$, which is then transmitted to Alice.
3. Alice computes $\texttt{X25519}(a, K_B)$ and Bob computes $\texttt{X25519}(b, K_A)$.

Hence, they now both have the shared secret $K = \texttt{X25519}(a, X25519(b, 9)) = \texttt{X25519}(b, X25519(a, 9))$, which they can use to derive a shared secret key for use in a symmetric scheme.

6.4.3 X25519Kyber728: Hybrid Post-Quantum Key Agreement

We conclude this thesis with a brief presentation of X25519Kyber728, a hybrid post-quantum key exchange designed for use in TLS 1.3. Transport Layer Security (or TLS) is a cryptographic protocol allowing secure communication between client/server applications on the Internet, and TLS 1.3 is simply its latest version as specified in [Res18].

Our focus lies in the key exchange part of TLS 1.3, and we provide a concise overview of how X25519Kyber728 handles it. Particularly, we assume that the client begins the exchange (as is most common in TLS) and the process below is followed, as presented in the draft [WS23]. Moreover, the Kyber algorithms below are instantiated with the Kyber768 parameter set.

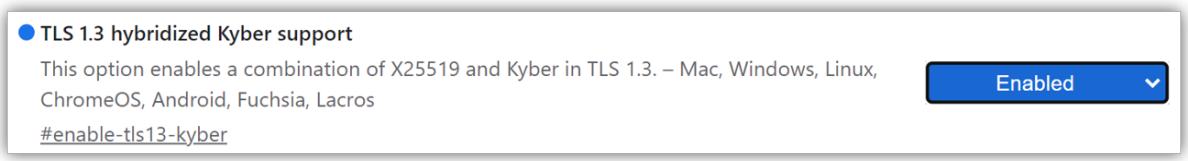
1. The client generates 32 random bytes with which he creates his X25519 ephemeral share K_C of 32 bytes. Independently, he runs Kyber.CCAKEM.KeyGen(), creating a key-pair (pk, sk). Then, he sends the concatenation of K_C and pk (1184 bytes) to the server, which is a value of 1216 bytes in length.
2. The server generates 32 random bytes with which it creates its X25519 ephemeral share K_S of 32 bytes. Independently, it runs Kyber.CCAKEM.Enc(pk), creating a pair (c, K). It holds the Kyber768 shared secret K and then sends a concatenation of K_S and c (1088 bytes) to the client, which is a value of 1120 bytes in length.
3. Client and server both compute the X25519 shared secret (32 bytes). Moreover, the client runs Kyber.CCAKEM.Dec(c, sk), which returns Kyber768's shared secret K. Then, the shared secret is calculated by both, as the concatenation of the X25519 shared secret and the Kyber768 shared secret, getting a value of 64 bytes in length.

Remark 6.12.

(i) We remark that Kyber is not the first LWE scheme used in TLS as in [Bos+15], a post-quantum key exchange from Ring-LWE suitable for TLS was introduced. This paper is particularly useful for assessing the performance of such schemes in applications like TLS.

(ii) Regarding the security of the above scheme, we refer the reader to the draft [SFG23] and its previous editions, and note that the scheme is secure if *either* component is secure.

Finally, we note again that this scheme is actively used now in Google Chrome (started in Chrome 116) and anyone can enable it in "chrome://flags":



Bibliography

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems". In: *Electron. Colloquium Comput. Complex.* TR96 (1996). URL: https://api.semanticscholar.org/CorpusID:1480070.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. "A Sieve Algorithm for the Shortest Lattice Vector Problem". In: *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*. STOC '01. Hersonissos, Greece: Association for Computing Machinery, 2001, pp. 601–610. ISBN: 1581133499. DOI: 10.1145/380752.380857. URL: https://doi.org/10.1145/380752.380857.

[Ala+22] Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*. Technical Report NIST Interagency or Internal Report (IR) 8413. Gaithersburg, MD: National Institute of Standards and Technology, July 2022.

[AD21] Martin Albrecht and Léo Ducas. *Lattice Attacks on NTRU and LWE: A History of Refinements*. Cryptology ePrint Archive, Paper 2021/799. 2021. URL: https://eprint.iacr.org/2021/799.

[Alb+18] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alexander Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. *Estimate All the LWE, NTRU Schemes!* 2018.

[AD17] Martin R. Albrecht and Amit Deo. "Large Modulus Ring-LWE $\geq$ Module-LWE". In: *Advances in Cryptology – ASIACRYPT 2017*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Cham: Springer International Publishing, 2017, pp. 267–296. ISBN: 978-3-319-70694-8.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. "On the Concrete Hardness of Learning with Errors". In: *Journal of Mathematical Cryptology* 9.3 (Oct. 2015), pp. 169–203. ISSN: 1862-2984. DOI: 10.1515/jmc-2015-0016.

[Alk+16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. "Post-quantum Key Exchange—A New Hope". In: *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 327–343. ISBN: 978-1-931971-32-4. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim.

[ANS22] ANSSI. *Views on the Post-Quantum Cryptography Transition*. 2022. URL: https://www.ssi.gouv.fr/publication/anssi-views-on-the-post-quantum-cryptography-transition/.

[App+09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". In: *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference*. Vol. 5677. Lecture Notes in Computer Science. Springer, 2009, pp. 595–618. DOI: 10.1007/978-3-642-03356-8_35. URL: https://www.iacr.org/archive/crypto2009/56770585/56770585.pdf.

[Ava+21] Robert Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. *CRYSTALS-Kyber Algorithm Specifications and Supporting Documentation (Version 3.02)*. Third-round submission to the NIST's post-quantum cryptography standardization process. Aug. 2021. URL: https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[Bab86] László Babai. "On Lovász' lattice reduction and the nearest lattice point problem". In: *Combinatorica* 6 (1986), pp. 1–13. URL: https://api.semanticscholar.org/CorpusID:7914792.

[Ber06] Daniel J. Bernstein. "Curve25519: New Diffie-Hellman Speed Records". In: *International Conference on Theory and Practice of Public Key Cryptography*. 2006.

[BL17] Daniel J. Bernstein and Tanja Lange. "Montgomery curves and the Montgomery ladder". In: *IACR Cryptol. ePrint Arch.* 2017 (2017), p. 293.

[Bli29] H.F. Blichfeldt. "The Minimum Value of Quadratic Forms, and the Closest Packing of Spheres". In: *Mathematische Annalen* 101 (1929), pp. 605–607. URL: http://eudml.org/doc/159357.

[Bon+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random oracles in a quantum world". In: *ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. LNCS. Heidelberg: Springer, Dec. 2011, pp. 41–69.

[Bos+15] Joppe Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. "Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem". In: vol. 2015. Apr. 2015. DOI: 10.1109/SP.2015.40.

[Bos+17] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, and Damien Stehlé. "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM". In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)* (2017), pp. 353–367. URL: https://api.semanticscholar.org/CorpusID:20449721.

[BGV11] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. *Fully Homomorphic Encryption without Bootstrapping*. Cryptology ePrint Archive, Paper 2011/277. 2011. URL: https://eprint.iacr.org/2011/277.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping". In: *Information Technology Convergence and Services*. 2012. URL: https://api.semanticscholar.org/CorpusID:2602543.

[Bra+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. *Classical Hardness of Learning with Errors*. 2013. arXiv: 1306.0281 [cs.CC].

[Bru+16] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. "Flush, Gauss, and Reload - A Cache Attack on the BLISS Lattice-Based Signature Scheme". In: *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*. Ed. by Benedikt Gierlichs and Axel Y. Poschmann. Vol. 9813. Lecture Notes in Computer

Science. Springer, 2016, pp. 323–345. ISBN: 978-3-662-53139-6. DOI: 10.1007/978-3-662-53140-2_16. URL: http://dx.doi.org/10.1007/978-3-662-53140-2_16.

[CD23] Wouter Castryck and Thomas Decru. "An Efficient Key Recovery Attack on SIDH". In: *Advances in Cryptology – EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Lyon, France: Springer-Verlag, 2023, pp. 423–447. ISBN: 978-3-031-30588-7. DOI: 10.1007/978-3-031-30589-4_15. URL: https://doi.org/10.1007/978-3-031-30589-4_15.

[CS17] Craig Costello and Benjamin Smith. "Montgomery curves and their arithmetic". In: *CoRR* abs/1703.01863 (2017). arXiv: 1703.01863. URL: http://arxiv.org/abs/1703.01863.

[CS03] Ronald Cramer and Victor Shoup. "Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack". In: *IACR Cryptol. ePrint Arch.* 2001 (2003), p. 108. URL: https://api.semanticscholar.org/CorpusID:1486897.

[DD18] D. Dadush and L. Ducas. *Intro to Lattice Algorithms and Cryptography*. Mastermath, Spring 2018. 2018. URL: https://homepages.cwi.nl/~dadush/teaching/lattices-2018/ (visited on 12/01/2023).

[DMG21] Viet Ba Dang, Kamyar Mohajerani, and Kris Gaj. *High-Speed Hardware Architectures and FPGA Benchmarking of CRYSTALS-Kyber, NTRU, and Saber*. Cryptology ePrint Archive, Paper 2021/1508. 2021. URL: https://eprint.iacr.org/2021/1508.

[Den03] Alexander W. Dent. "A Designer's Guide to KEMs". In: *Cryptography and Coding*. Ed. by Kenneth G. Paterson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 133–151. ISBN: 978-3-540-40974-8.

[DP23] Léo Ducas and Ludo N. Pulles. "Does the Dual-Sieve Attack on Learning with Errors Even Work?" In: *Advances in Cryptology – CRYPTO 2023*. Ed. by Helena Handschuh and Anna Lysyanskaya. Cham: Springer Nature Switzerland, 2023, pp. 37–69. ISBN: 978-3-031-38548-3.

[DF03] D.S. Dummit and R.M. Foote. *Abstract Algebra*. Wiley, 2003. ISBN: 9780471433347. URL: https://books.google.gr/books?id=KJDBQgAACAAJ.

[Dwo15] Morris Dworkin. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. en. National Institute of Standards and Technology. Aug. 2015. URL: https://doi.org/10.6028/NIST.FIPS.202.

[Esp+17] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. "Side-Channel Attacks on BLISS Lattice-Based Signatures: Exploiting Branch Tracing against strongSwan and Electromagnetic Emanations in Microcontrollers". In: Oct. 2017, pp. 1857–1874. DOI: 10.1145/3133956.3134028.

[FP85] Ulrich Fincke and Michael E. Pohst. "Improved methods for calculating vectors of short length in a lattice". In: *Mathematics of Computation* (1985). URL: https://api.semanticscholar.org/CorpusID:15519722.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. "Secure Integration of Asymmetric and Symmetric Encryption Schemes". In: *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*. CRYPTO '99. Berlin, Heidelberg: Springer-Verlag, 1999, pp. 537–554. ISBN: 3540663479.

[Gal18] Steven D. Galbraith. *Mathematics of Public Key Cryptography (Version 2.0*. 2018. URL: https://www.math.auckland.ac.nz/~sgal018/crypto-book/main.pdf (visited on 12/01/2023).

[Gam+06] Nicolas Gama, Nick Howgrave-Graham, Henrik Koy, and Phong Nguyen. "Rankin's Constant and Blockwise Lattice Reduction". In: Aug. 2006, pp. 112–130. ISBN: 978-3-540-37432-9. DOI: 10.1007/11818175_7.

[GN08a] Nicolas Gama and Phong Nguyen. "Finding short lattice vectors within mordell's inequality". In: May 2008, pp. 207–216. DOI: 10.1145/1374376.1374408.

[GN08b] Nicolas Gama and Phong Nguyen. "Predicting Lattice Reduction". In: vol. 4965. Apr. 2008, pp. 31–51. ISBN: 978-3-540-78966-6. DOI: 10.1007/978-3-540-78967-3_3.

[Gau01] Carl Friedrich Gauss. *Disquisitiones Arithmeticæ*. Leipzig: Typis et Impensis G. Fleischer, 1801. URL: https://library.si.edu/digital-library/book/disquisitionesa00gaus (visited on 12/01/2023).

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: *Proceedings of the fortieth annual ACM symposium on Theory of computing* (2008). URL: https://api.semanticscholar.org/CorpusID:1474892.

[GMSS99] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. "Approximating Shortest Lattice Vectors is Not Harder Than Approximating Closest Lattice Vectors". In: *Inf. Process. Lett.* 71 (1999), pp. 55–61. URL: https://api.semanticscholar.org/CorpusID:7897107.

[Gro00] Johann Großschädl. "The Chinese Remainder Theorem and its Application in a High-Speed RSA Crypto Chip". In: *Asia-Pacific Computer Systems Architecture Conference*. 2000. URL: https://api.semanticscholar.org/CorpusID:5831162.

[Hås+99] Johan Håstad, Russell Impagliazzo, Leonid Levin, and Michael Luby. "A Pseudorandom Generator from any One-way Function". In: *SIAM Journal on Computing* 28 (Feb. 1999). DOI: 10.1137/S0097539793244708.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. "NTRU: A ring-based public key cryptosystem". In: *Algorithmic Number Theory*. Ed. by Joe P. Buhler. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 267–288. ISBN: 978-3-540-69113-6.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. "A Modular Analysis of the Fujisaki-Okamoto Transformation". In: Nov. 2017, pp. 341–371. ISBN: 978-3-319-70499-9. DOI: 10.1007/978-3-319-70500-2_12.

[IZ89] R. Impagliazzo and D. Zuckerman. "How to Recycle Random Bits". In: *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 1989, pp. 248–253.

[Imp95] Russell Impagliazzo. "Personal view of average-case complexity". In: July 1995, pp. 134–147. ISBN: 0-8186-7052-5. DOI: 10.1109/SCT.1995.514853.

[Ins20] Simons Institute. *Lattices: Algorithms, Complexity, and Cryptography Boot Camp*. Program, Location: Calvin Lab Auditorium, Simons Institute. Jan. 2020. URL: https://simons.berkeley.edu/workshops/lattices-algorithms-complexity-cryptography-boot-camp.

[Kan87] Ravi Kannan. "Minkowski's Convex Body Theorem and Integer Programming". In: *Mathematics of Operations Research* 12.3 (1987), pp. 415–440. DOI: 10.1287/moor.12.3.415.

[KM15] Neal Koblitz and Alfred Menezes. "The random oracle model: a twenty-year retrospective". In: *Designs, Codes and Cryptography* 77 (2015), pp. 587–610. URL: https://api.semanticscholar.org/CorpusID:2363461.

[Laa15] Thijs Laarhoven. *Lattice Cryptography and Lattice Cryptanalysis (Lecture for Cryptography I)*. Online. Part of the MasterMath course "Selected Areas in Cryptology," Lecture for Cryptography I. 2015. URL: https://thijs.com/docs/lec1.pdf (visited on 12/01/2023).

[LLS90] Jeffrey Lagarias, Hendrik Lenstra, and Claus Schnorr. "Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice". In: *Combinatorica* 10 (Dec. 1990), pp. 333–348. DOI: 10.1007/BF02128669.

[Lag73] Joseph-Louis Lagrange. "Recherches d'arithmétique". In: *Nouv. Mem. Acad.* (1773). URL: https://gallica.bnf.fr/ark:/12148/bpt6k229222d/f696 (visited on 12/01/2023).

[LHT16] Adam Langley, Mike Hamburg, and Sean Turner. *Elliptic Curves for Security*. RFC 7748. Jan. 2016. DOI: 10.17487/RFC7748. URL: https://www.rfc-editor.org/info/rfc7748.

[LS14] Adeline Langlois and Damien Stehlé. "Worst-Case to Average-Case Reductions for Module Lattices". In: *Designs, Codes and Cryptography* 75 (June 2014). DOI: 10.1007/s10623-014-9938-4.

[LLL82] Arjen Lenstra, Hendrik Lenstra, and Lovász László. "Factoring Polynomials with Rational Coefficients". In: *Mathematische Annalen* 261 (Dec. 1982). DOI: 10.1007/BF01457454. URL: https://www.researchgate.net/publication/50863306_Factoring_Polynomials_with_Rational_Coefficients.

[LZ22] Zhichuang Liang and Yunlei Zhao. *Number Theoretic Transform and Its Applications in Lattice-based Cryptosystems: A Survey*. 2022. arXiv: 2211.13546 [cs.CR].

[LP11] Richard Lindner and Chris Peikert. "Better Key Sizes (and Attacks) for LWE-Based Encryption". In: Feb. 2011, pp. 319–339. ISBN: 978-3-642-19073-5. DOI: 10.1007/978-3-642-19074-2_21.

[Lyu20] Vadim Lyubashevsky. "Basic Lattice Cryptography: Encryption and Fiat-Shamir Signatures". In: (2020). URL: https://tjerandsilde.no/lattices/ (visited on 12/01/2023).

[LPR13a] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "A Toolkit for Ring-LWE Cryptography". In: *IACR Cryptology ePrint Archive*. 2013. URL: https://api.semanticscholar.org/CorpusID:488653.

[LPR13b] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *J. ACM* 60.6 (Nov. 2013). ISSN: 0004-5411. DOI: 10.1145/2535925. URL: https://doi.org/10.1145/2535925.

[MAT22] MATZOV. "Report on the Security of LWE: Improved Dual Lattice Attack". In: 2022. URL: https://api.semanticscholar.org/CorpusID:251600824.

[MR04] D. Micciancio and O. Regev. "Worst-case to average-case reductions based on Gaussian measures". In: *45th Annual IEEE Symposium on Foundations of Computer Science*. 2004, pp. 372–381. DOI: 10.1109/FOCS.2004.72.

[Mic01] Daniele Micciancio. "Improving Lattice-based Cryptosystems using the Hermite Normal Form". In: *Cryptography and Lattices Conference — CaLC 2001*. Ed. by Joseph Silverman. Vol. 2146. Lecture Notes in Computer Science. Providence, Rhode Island: Springer-Verlag, Mar. 2001, pp. 126–145.

[Mic07] Daniele Micciancio. "Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions". In: *computational complexity* 16 (Dec. 2007), pp. 365–411. DOI: 10.1007/s00037-007-0234-9.

[Mic12] Daniele Micciancio. *CSE206A: Lattices Algorithms and Applications (2012)*. University of California, San Diego. 2012. URL: https://cseweb.ucsd.edu/classes/wi12/cse206A-a/ (visited on 12/01/2023).

[Mic14] Daniele Micciancio. *CSE206A: Lattice Algorithms and Applications (2014)*. University of California, San Diego. 2014. URL: https://cseweb.ucsd.edu/classes/sp14/cse206A-a/index.html (visited on 12/01/2023).

[Mic21] Daniele Micciancio. *CSE206A: Lattice Algorithms and Applications (Fall 2021)*. University of California, San Diego. 2021. URL: https://cseweb.ucsd.edu/classes/fa21/cse206A-a/ (visited on 12/01/2023).

[MP13] Daniele Micciancio and Chris Peikert. *Hardness of SIS and LWE with Small Parameters*. Cryptology ePrint Archive, Paper 2013/069. 2013. DOI: 10.1007/978-3-642-40041-4_2. URL: https://eprint.iacr.org/2013/069.

[MR09] Daniele Micciancio and Oded Regev. "Lattice-based Cryptography". In: Jan. 2009, pp. 147–191. ISBN: 978-3-540-88701-0. DOI: 10.1007/978-3-540-88702-7_5.

[MV13] Daniele Micciancio and Panagiotis Voulgaris. "A Deterministic Single Exponential Time Algorithm for Most Lattice Problems Based on Voronoi Cell Computations". In: *SIAM Journal on Computing* 42.3 (2013), pp. 1364–1391. DOI: 10.1137/100811970.

[Mil20] James S. Milne. *Algebraic Number Theory (v3.08)*. 2020. URL: www.jmilne.org/math/.

[Nat23] National Institute of Standards and Technology. *Module-Lattice-based Key-Encapsulation Mechanism Standard*. Federal Information Processing Standards Publication (FIPS) NIST FIPS 203 ipd. Department of Commerce, Washington, D.C. 2023. URL: https://doi.org/10.6028/NIST.FIPS.203.ipd.

[NV09] P.Q. Nguyen and B. Vallée. *The LLL Algorithm: Survey and Applications*. Information Security and Cryptography. Springer Berlin Heidelberg, 2009. ISBN: 9783642022951. URL: https://books.google.gr/books?id=IXlbRAAACAAJ.

[Ngu10] Phong Q. Nguyen. "Hermite's Constant and Lattice Algorithms". In: *The LLL Algorithm*. 2010. URL: https://api.semanticscholar.org/CorpusID:2312457.

[NS99] Phong Q. Nguyen and Jacques Stern. "The Hardness of the Hidden Subset Sum Problem and Its Cryptographic Implications". In: *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*. Vol. 1666. Lecture Notes in Computer Science. Springer, 1999, pp. 31–46. DOI: 10.1007/3-540-48405-1_3.

[Orr19] Martin Orr. *Algebraic Number Theory (lecture notes)*. Lecture notes (complete). Manchester, United Kingdom, 2019.

[PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. "A Framework for Efficient and Composable Oblivious Transfer". In: *CRYPTO*. 2008, pp. 554–571.

[Pei09] Chris Peikert. "Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem". In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. STOC '09. Bethesda, MD, USA: Association for Computing Machinery, 2009, pp. 333–342. ISBN: 9781605585062. DOI: 10.1145/1536414.1536461. URL: https://doi.org/10.1145/1536414.1536461.

[Pei16] Chris Peikert. "A Decade of Lattice Cryptography". In: *Foundations and Trends® in Theoretical Computer Science* 10 (Mar. 2016), pp. 283–424. DOI: 10.1561/0400000074.

[PP19] Chris Peikert and Zachary Pepin. "Algebraically Structured LWE, Revisited". In: Nov. 2019, pp. 1–23. ISBN: 978-3-030-36029-0. DOI: 10.1007/978-3-030-36030-6_1.

[PR07] Chris Peikert and Alon Rosen. "Lattices that admit logarithmic worst-case to average-case connection factors". In: Jan. 2007, pp. 478–487.

[PGY17] Peter Pessl, Leon Groot Bruinderink, and Yuval Yarom. "To BLISS-B or not to be: Attacking strongSwan's Implementation of Post-Quantum Signatures". In: Oct. 2017, pp. 1843–1855. DOI: 10.1145/3133956.3134023.

[Pie12] Krzysztof Pietrzak. "Cryptography from Learning Parity with Noise". In: *SOFSEM 2012*. Ed. by M. Bieliková et al. Vol. 7147. LNCS. Springer-Verlag Berlin Heidelberg, 2012, pp. 99–114.

[Poh81] Michael Pohst. "On the Computation of Lattice Vectors of Minimal Length, Successive Minima and Reduced Bases with Applications". In: *SIGSAM Bull.* 15.1 (Feb. 1981), pp. 37–44. ISSN: 0163-5824. DOI: 10.1145/1089242.1089247. URL: https://doi.org/10.1145/1089242.1089247.

[Pol71] J. M. Pollard. "The fast Fourier transform in a finite field". In: *Mathematics of Computation* 25 (1971), pp. 365–374. URL: https://api.semanticscholar.org/CorpusID:123174851.

[Reg09a] Oded Regev. *Lattices in Computer Science*. Course Syllabus, Lectures, and Materials. School of Computer Science, Tel Aviv University. 2009. URL: https://cims.nyu.edu/~regev/teaching/lattices_fall_2009/.

[Reg09b] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *J. ACM* 56 (2009), 34:1–34:40. URL: https://api.semanticscholar.org/CorpusID:207156623.

[Reg10] Oded Regev. "The Learning with Errors Problem". In: (2010).

[Res18] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018. DOI: 10.17487/RFC8446. URL: https://www.rfc-editor.org/info/rfc8446.

[Rit12] Christophe Ritzenthaler. *Elliptic curves and applications to cryptography (notes)*. 2011-2012.

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. "Tightly-Secure Key-Encapsulation Mechanism in the Quantum Random Oracle Model". In: *IACR Cryptology ePrint Archive*. 2018. URL: https://api.semanticscholar.org/CorpusID:3580803.

[SE94] C.P. Schnorr and M. Euchner. "Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems". In: *Mathematical Programming* 66 (1994), pp. 181–199. DOI: 10.1007/BF01581144.

[Sch87] Claus Schnorr. "A hierarchy of polynomial time basis reduction algorithms". In: *Theoretical Computer Science* 53 (Dec. 1987). DOI: 10.1016/0304-3975(87)90064-8.

[Sch03] Claus-Peter Schnorr. "Lattice Reduction by Random Sampling and Birthday Methods". In: *Symposium on Theoretical Aspects of Computer Science*. 2003. URL: https://api.semanticscholar.org/CorpusID:10720935.

[Sei18] Gregor Seiler. "Faster AVX2 optimized NTT multiplication for Ring-LWE lattice cryptography". In: *IACR Cryptol. ePrint Arch.* 2018 (2018), p. 39. URL: https://api.semanticscholar.org/CorpusID:4037516.

[Sho04] Victor Shoup. "Sequences of Games: A Tool for Taming Complexity in Security Proofs". In: *IACR Cryptology ePrint Archive* 2004 (Jan. 2004), p. 332.

[Sho08] Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. 2nd ed. Version 2 [pdf] (6/16/2008, corresponds to the second print edition). Cambridge University Press, 2008. URL: https://shoup.net/ntb/.

[Sho] Victor Shoup. *Number Theory C++ Library (NTL) version 5.4*. URL: http://www.shoup.net/ntl/.

[Sil09] Joseph Silverman. *The Arithmetic of Elliptic Curves*. Vol. 106. Jan. 2009. ISBN: 978-0-387-09493-9. DOI: 10.1007/978-0-387-09494-6.

[Sil20a] Joseph H. Silverman. *An Introduction to Lattices, Lattice Reduction, and Lattice-Based Cryptography*. Brown University, Institute for Advanced Study (IAS), Park City Mathematics Institute (PCMI). 2020. URL: https://www.youtube.com/playlist?list=PLldN_DpkXL3Zp0fZEUKpMq3H0JBGBFXvO (visited on 12/01/2023).

[Sil20b] Joseph H. Silverman. "An Introduction to Lattices, Lattice Reduction, and Lattice-Based Cryptography (notes)". In: (2020). URL: https://www.ias.edu/sites/default/files%20Silverman_PCMI_Note_DistributionVersion_220705.pdf (visited on 12/01/2023).

[ST16] National Institute of Standards and Technology. *Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process*. 2016. URL: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-fnal-dec-2016.pdf.

[SFG23] Douglas Stebila, Scott Fluhrer, and Shay Gueron. *Hybrid key exchange in TLS 1.3*. Internet-Draft draft-ietf-tls-hybrid-design-09. Work in Progress. Internet Engineering Task Force, Sept. 2023. 23 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/09/.

[Ste04] William A. Stein. "A Brief Introduction to Classical and Adelic Algebraic Number Theory". In: 2004. URL: https://www.williamstein.org/papers/ant/.

[Ste] Noah Stephens-Davidowitz. *Dimension-Preserving Reductions Between Lattice Problems*. Last updated September 6, 2016.

[ST01] Ian Stewart and David Tall. *Algebraic Number Theory and Fermat's Last Theorem: Third Edition*. Dec. 2001. ISBN: 9780429107504. DOI: 10.1201/9781439864081.

[TU16] Ehsan Targhi and Dominique Unruh. "Post-Quantum Security of the Fujisaki-Okamoto and OAEP Transforms". In: Nov. 2016, pp. 192–216. ISBN: 978-3-662-53643-8. DOI: 10.1007/978-3-662-53644-5_8.

[Uni12] Bar-Ilan University. *Winter School on Lattice-Based Cryptography and Applications*. May 31, 2012. Event: Winter School on Cryptography 2012, held at Bar-Ilan University from February 19 - 22. 2012.

[Vai20] Vinod Vaikuntanathan. *CS294: Lattices, Learning with Errors and Post-Quantum Cryptography (lecture notes)*. UC Berkeley. 2020.

[Was08] L.C. Washington. *Elliptic Curves: Number Theory and Cryptography, Second Edition*. Discrete Mathematics and Its Applications. CRC Press, 2008. ISBN: 9781420071474. URL: https://books.google.gr/books?id=nBfCEqpYKW0C.

[WS23] Bas Westerbaan and Douglas Stebila. *X25519Kyber768Draft00 hybrid post-quantum key agreement*. Internet-Draft draft-tls-westerbaan-xyber768d00-03. Work in Progress. Internet Engineering Task Force, Sept. 2023. 6 pp. URL: https://datatracker.ietf.org/doc/draft-tls-westerbaan-xyber768d00/03/.

[Δρα22] Κωνσταντίνος Δραζιώτης. *Εισαγωγή στην Κρυπτογραφία [Προπτυχιακό εγχειρίδιο]*. Κάλλιπος Ανοικτές Ακαδημαϊκές Εκδοσεις, 2022. DOI: https://dx.doi.org/10.57713/kallipos-17.

[Που15] Δημήτρης Πουλάκης. *Υπολογιστική θεωρία αριθμών [Προπτυχιακό εγχειρίδιο]*. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδοσεις, 2015. URL: https://dx.doi.org/10.57713/kallipos-323.