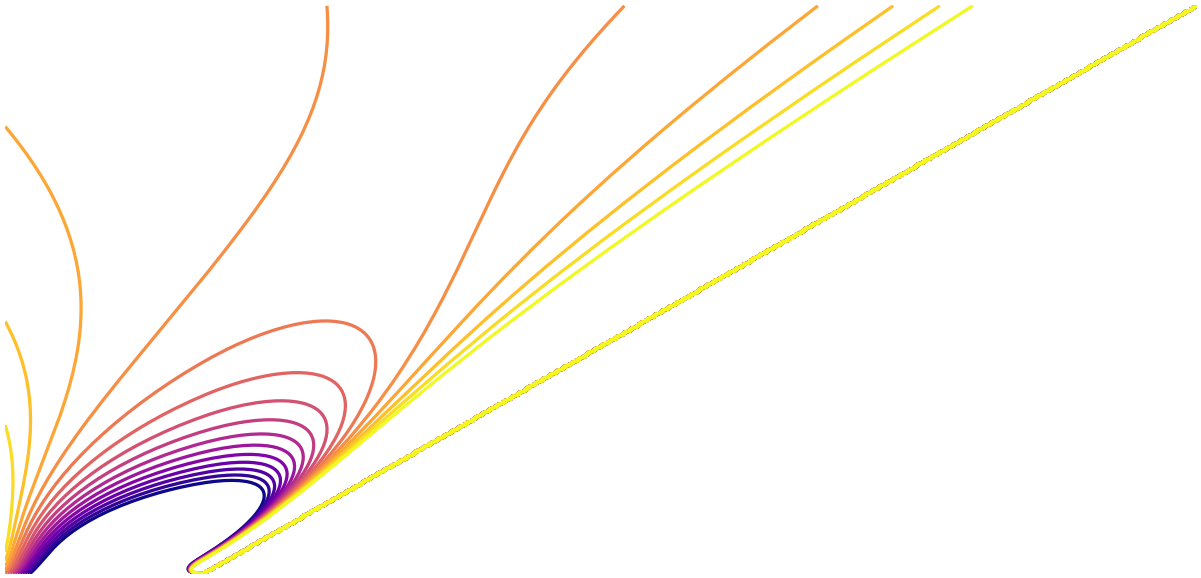


Overstretched Parameters in Lattice-Based Cryptography

Stretching the overstretched regime to new frontiers

Master's thesis in Computer science and engineering

EMIL BABAYEV

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COMPUTING SCIENCE

CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025
www.chalmers.se

MASTER'S THESIS 2025

Overstretched Parameters in Lattice-Based Cryptography

Stretching the overstretched regime to new frontiers

EMIL BABAYEV

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Overstretched Parameters in Lattice-Based Cryptography
Stretching the overstretched regime to new frontiers
EMIL BABAYEV

Supervisor: Joel Frisk Gärtner, Mathematics-KTH
Examiner: Elena Pagnin, CSE

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Level curves of the asymptotic difference in hardness between SKR and DSD, from Section 3.1.

Typeset in LaTeX
Gothenburg, Sweden 2025

Overstretched Parameters in Lattice-Based Cryptography
Stretching the overstretched regime to new frontiers
EMIL BABAYEV
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Lattice-based cryptosystems such as Learning With Errors (LWE) and NTRU have gained popularity as one of the main methods for constructing cryptosystems secure against a quantum attacker. The process of instantiating these involves fixing multiple parameters based on cost estimates for performing lattice reduction attacks to find the secret key. Instantiations of NTRU using large moduli q, known as overstretched NTRU, are known to suffer from an attack that allows discovery of a dense sublattice of the q-ary NTRU lattice, breaking the cryptosystem at a significantly smaller cost than the traditional secret key recovery attack.

The question of whether ring-LWE, module-LWE and NTWE suffer from the dense sublattice attack remains unresolved. We apply the attack model developed by Ducas and van Woerden to these problems to produce asymptotic cost estimates for the dense sublattice attack. We find module-LWE and ring-LWE are not expected to suffer from this attack, while NTWE is. This is further corroborated by experiments performed on NTWE. Additionally, the analysis performed on NTRU by Ducas and van Woerden is extended to larger polynomial coefficients.

Taken together, we show that lattice-based cryptographic problems that have a corresponding problem on a q-ary lattice with a dense sublattice could be vulnerable to the dense sublattice attack and that the attack model can be used to predict such a vulnerability.

Keywords: Cryptography, post-quantum, lattice, NTRU, LWE, NTWE, reduction, attack, overstretched, asymptotic

Acknowledgements

First and foremost I would like to thank my supervisor, Joel Frisk Gärtner, for introducing me to the field of lattice cryptography. Without your patience and suggestions this thesis would not be what it has become. Further, I thank my examiner Elena Pagnin for sparking my interest in cryptography in the first place and for all your valuable feedback during the project. I also would like to extend my thanks to Nils Dunlop who provided me with the computational resources that made all experiments possible and listened to me droning on about lattices for six months. Finally, I would like to say thanks to my mother, for always believing in me.

Emil Babayev, Gothenburg, 2025-06-04

Contents

List of Figures

List of Tables

1

Introduction

Cryptology, which roughly translates to ”the study of secrets”, is the study of concealing and communicating information securely in an adversarial setting. It is one of the cornerstones of digital society and used in every HTTPS connection through the TLS/SSL layer. In particular, public-key cryptography allows anyone to encrypt and transfer information using a public key and send it to another party holding a secret key without previously agreeing on a common key.

Since Peter Shor described a set of quantum algorithms that solve the factoring problem and the discrete logarithm problem in 1995, public-key cryptography has been in a state of crisis. The ubiquitous RSA cryptosystem is based on the factoring problem, while the Diffie-Hellman (DH) key exchange is based on the discrete logarithm problem. Both public-key encryption schemes are widely used on the internet today and can be broken by a quantum computer with sufficiently many qubits. A cryptanalytically relevant computer does not exist yet, but may be available within a decade [1].

Lattices provide a source of computational problems that have no known efficient solution algorithms, be it classical or quantum. These problems have also proved useful as bases for cryptosystems. Lattice-based cryptography is now viewed as a subfield of Post-Quantum Cryptography (PQC), in which the goal is to find cryptosystems resistant against quantum algorithms.

Among lattice-based problems usable for cryptosystems, two have gained particular popularity. These are NTRU, proposed in 1996 by Hoffstein, Pipher and Silverman [2] and Learning With Errors (LWE), introduced by Regev in 2005 [3]. Eventually, the NIST-PQC project was launched by the National Institute of Standards and Technology, which set out to find new public-key cryptosystem standards before the realisation of quantum computers. Variants and further developments of both NTRU and LWE were submitted as candidates. At the time of writing, the module-LWE-based cryptosystem CRYSTALS-Kyber has been selected for standardisation, along with the NTRU-based signature scheme Falcon.

Instantiating NTRU and LWE involves choosing a set of parameters. All choices are not equal in terms of security, as they directly impact the complexity of solving the underlying computational problems on lattices. Thus, there has been a significant effort to estimate the computational hardness of NTRU and LWE in order to determine which parameters are safe. For example, an instantiation of NTRU

involves a polynomial ring $\mathbb{Z}_q[x]/(\Phi(x))$, where $\Phi(x)$ is a monic polynomial and q is a positive integer, a positive real number σ^2 and a distribution χ [4]. The secret keys are polynomials in the ring with coefficients sampled from χ, and the task of finding them can be expressed as a lattice problem. In 2016, Albrecht, Bai and Ducas [5] discovered that the security of the NTRU lattice problem can be significantly reduced if the modulus q is selected too large, an analysis later refined by Kirchner and Fouque [6]. An attack based on lattice reduction can then be performed, which depends on discovery of a dense sublattice containing the secrets. Ducas and van Woerden [7] made this analysis more precise by describing the modulus $\hat{q}$ where the overstretched regime begins, called the fatigue point. Interestingly, the very similar lattices obtained from ring-LWE (RLWE) and Module-LWE (MLWE) also contain a dense sublattice. However, there is no published work on the subject of overstretched instances of RLWE and MLWE.

1.1 Aim, Assignment and Scope

In this thesis, we investigate whether the dense sublattice attack can be performed on ring-LWE [8], module-LWE [9] and the related cryptosystem NTWE [10] in order to begin bridging this knowledge gap about overstretched lattice problems. In addition, we analyse how the fatigue point changes when the size of polynomial coefficients in NTRU increases. These tasks are mainly performed using asymptotic analysis of the attack complexity and substantiated by experiments where applicable.

1.1.1 Research Questions

More specifically, the following six research questions are investigated.

1. The polynomial coefficients in NTRU are usually assumed to be ternary. How does the fatigue point change when this assumption is replaced by a discrete Gaussian distribution with arbitrary variance σ^2?
2. Is it possible to compensate for the reduction in security in an overstretched instance of NTRU by selecting parameters in a suitable way? How does this increase the computational cost of the attack? How does this affect the size of ciphertexts and signatures?
3. Can ring-LWE be overstretched by selecting q too large? Does the dense sublattice attack decrease attack complexity?
4. Is the dense sublattice attack applicable to MLWE? How does this decrease attack complexity?
5. Is the dense sublattice attack applicable to NTWE, a combination of NTRU and LWE? How does this decrease attack complexity?
6. If the answer is no to any of the last three questions, why is it that NTRU can be overstretched yet the other cryptosystem cannot, despite their similarities?

1.1.2 Methodology

The thesis contains two main types of work: theoretical and experimental.

The theoretical work involves finding asymptotic estimates for the complexity of performing the lattice attacks in question. The attacks are modelled as an event that occurs during lattice reduction. The standard 2016 estimate is used as a model for Secret Key Recovery (SKR). Dense Sublattice Discovery is modelled in the same way as in [7]. Standard heuristic estimates are used where applicable. If need be, they are adjusted to suit the problem at hand.

Theoretical predictions are validated using experiments. The code from [7] is used as a base. As such, we only use the lattice library `fpLLL` and run Progressive BKZ, which is the lattice reduction algorithm that works best in practice. The code is adapted to NTWE. Additionally, we use the estimator developed in [7] for some predictions.

1.1.3 Limitations

The following limitations are imposed on the scope of the project.

- While there exist many other reduction algorithms, only classical BKZ reduction is considered. For implementation, we only use the `fpLLL` reduction library.
- There exist other attack models for SKR. We only consider the 2016 estimate, which is the most popular heuristic estimate for the cost of performing an attack. This is also the model considered in [7], which this thesis is based on.
- Lattice basis shape simulators, which incorporate experimental data to accurately predict the structure of a basis after reduction, are not developed as we are mainly interested in asymptotic behaviour.

The author has access to a limited amount of computational resources which also limits the extent of experiments.

1.2 Notation

The following notation is used throughout the text. Scalars and ring elements are denoted by lowercase Latin letters a, vectors of these by bold lowercase Latin letters $\boldsymbol{a}$, while matrices are denoted by uppercase bold Latin letters $\boldsymbol{A}$. For two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathbb{R}^\nu$, we denote $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \sum_{i=1}^{\nu} a_i b_i$ as their inner product. The norm $||\boldsymbol{a}||$ of a vector is the L_2 norm $\sqrt{\langle \boldsymbol{a}, \boldsymbol{a} \rangle}$ unless stated otherwise. In that case, this is indicated by a subscript on the norm. We let // denote integer division, meaning $a \,//\, b = \lfloor a/b \rfloor$. The symbols $\gtrapprox$ and $\lessapprox$ denote greater than approximately and less than approximately, respectively. Concatenation of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is denoted $(\boldsymbol{a}||\boldsymbol{b})$ or $(\boldsymbol{a}|\boldsymbol{b})$. When ring elements are concatenated, this should be interpreted in terms of coefficient embedding. Table 1.1 contains a non-exhaustive list of recurring variables in the text.

1.3 Structure of the Text

Each chapter begins with a statement of the overarching goal and an overview of its contents. Chapter 2 provides an overview of the prerequisite knowledge in lattice cryptography and concludes with a discussion of attack models used in this thesis. The first two research questions are addressed in Chapter 3 and the following two in Chapter 4. The fifth question is attended to in Chapter 5 and builds on the prior chapter. Additional results from the experiments in Chapter 5 are found in Appendix B. Connecting to all prior chapters, the last question is discussed in Chapter 6. Finally, the thesis is concluded in Chapter 7. Appendix A contains additional theory with a focus on lattice theory and reduction algorithms.

Table 1.1: A non-exhaustive list of variables.

Variable	Explanation
ν	An arbitrary dimension.
$\_$	A vector that is reduced modulo q.
d	Rank of a module.
n	Degree of a polynomial.
m	Number of samples.
M	A message.
q	A prime number used as the modulus in $\mathbb{Z}_q$.
e	The error in LWE and variants.
R_q	The ring $R_q = \mathbb{Z}_q[x]/(p(x))$ for a polynomial $p(x)$.
χ, ψ	Arbitrary probability distributions.
t	A positive real number used in lattice embedding, often set to 1.
$\boldsymbol{s}$	The secret in LWE and its variants.
$\boldsymbol{a}$	The random value in LWE and its variants, or an arbitrary vector.
σ^2	Variance of the discrete Gaussian distribution.
f, g, h	Polynomials in NTRU.
$\mathcal{L}(\boldsymbol{B})$	A lattice with basis $\boldsymbol{B}$.
$\boldsymbol{b}_i$	Basis vectors collected in $\boldsymbol{B}$.
$\boldsymbol{b}_i^*$	Gram–Schmidt orthogonalisation of $\boldsymbol{b}_i$.
$\mathbb{K}$	A number field.
$\Phi_n(x)$	The nth cyclotomic polynomial.
$\mathcal{D}$	The discrete Gaussian distribution.
i, j, l, p	An arbitrary index.
k	The number of q-vectors to keep in a q-ary lattice.
μ_{ij}	Gram–Schmidt coefficients between $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$.
β	Block size in BKZ.
α_β	The slope in the Geometric Series Assumption.
r	A random value.
c	A ciphertext, $\lceil \mu_{ij} \rfloor$ or an arbitrary polynomial.
B, S, Q, K	Asymptotic equivalent of β, length of secret vector, q and k, respectively.

2

Theory and Prior Work

The theory of lattice cryptography encompasses multiple fields and contains aspects that vary from deeply abstract to firmly applied. The goal of this chapter is to build just enough understanding to assimilate the prior work that this thesis is based on, found in Section 2.4, and the results in Chapters 3 to 6. In interest of space efficiency, the reader is expected to be familiar with linear algebra, complexity theory, probability theory and introductory abstract algebra.

This chapter provides an overview of the most important subjects. First, in Section 2.1, lattices are introduced and some theory concerning them is discussed. Second, Section 2.2 describes the three primitives, NTRU, LWE and NTWE, that this text concerns. Third, Section 2.3 concerns algorithms used for solving computational problems on lattices and useful heuristics for quantifying their performance on common lattices. Finally, Section 2.4 discusses two lattice attacks, Secret Key Recovery (SKR) and Dense Sublattice Discovery (DSD), on the primitives. Some concepts introduced here are described in more detail in Appendix A.

2.1 Lattice Theory

This section concerns the mathematics of lattices. We begin by providing a definition and discussing geometric properties that are important in computational problems over lattices. The discussion moves on to the introduction of discrete Gaussians, a popular error distribution in lattice cryptography and concludes with computational problems on lattices. The section requires familiarity with linear algebra and complexity theory. A useful reference is [11].

2.1.1 Introduction to Lattices

For the purposes of this text, a lattice $\mathcal{L}$ is the set of all integer linear combinations of lattice vectors,

$$\mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu) = \left\{ \sum_{i=1}^{\nu} x_i \boldsymbol{b}_i \right\}, \quad x_i \in \mathbb{Z}.$$

This is a discrete subgroup of $\mathbb{R}^\nu$ under the addition operation. The set of vectors $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ is called the lattice basis. A convenient way to collect basis elements is to consider them as columns in the matrix $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 & \ldots & \boldsymbol{b}_\nu \end{bmatrix}$. Let $\mathcal{L}(\boldsymbol{B})$ denote the

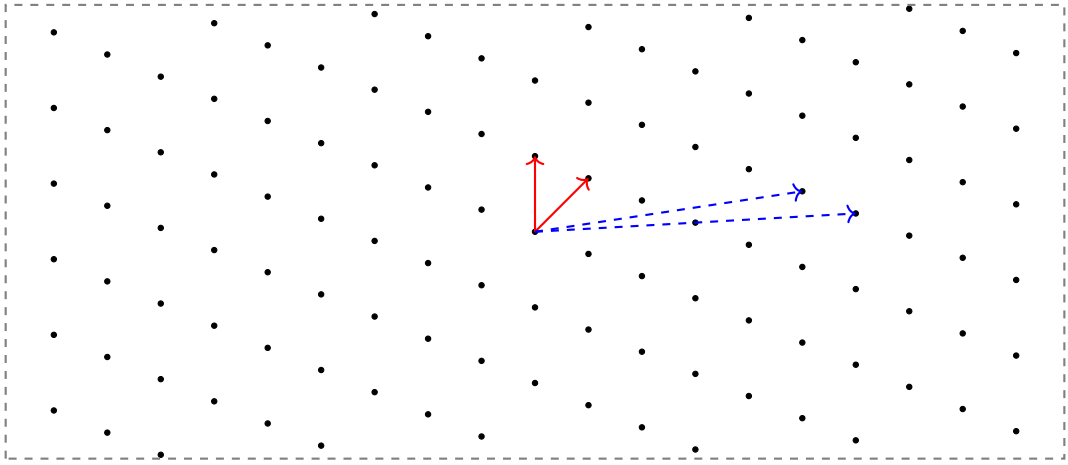


Figure 2.1: A lattice in two dimensions. The two sets of vectors, solid red and dashed blue, are two bases for the lattice.

set $\{\boldsymbol{B}\boldsymbol{x} : \boldsymbol{x} \in \mathbb{Z}^\nu\}$, which is the lattice generated by the columns in $\boldsymbol{B}$. Intuitively, a lattice in $\mathbb{R}^\nu$ is a countably infinite set of regularly spaced points. Figure 2.1 depicts a finite subset of a lattice in two dimensions together with two possible bases. For any given lattice, there exist an infinite number of bases related by a linear transformation (see Theorem A.2.3).

As for bases in linear algebra, a lattice need not cover all dimensions of the vector space it is a subgroup of. This property is captured in the following definition.

Definition 2.1.1. *The rank or dimension of a lattice $\mathcal{L}(\boldsymbol{B})$ is the dimension of $span(\boldsymbol{B})$ as a vector space over $\mathbb{R}^\nu$. If $\dim(span(\boldsymbol{B})) = \nu$, the lattice is called full rank.*

If a subset of lattice points has lattice structure itself, it can be categorised using the following definition.

Definition 2.1.2. *Let $\boldsymbol{B}$ and $\boldsymbol{B}'$ be bases. If $\mathcal{L}(\boldsymbol{B}') \subseteq$ (resp. $\subset$)$\mathcal{L}(\boldsymbol{B})$, we call $\mathcal{L}(\boldsymbol{B}')$ a (resp. proper) sublattice of $\mathcal{L}(\boldsymbol{B})$.*

Consequently, a sublattice is a subgroup of a lattice. A sublattice can also exist in a lower dimension than ν, and will then be spanned by fewer linearly independent vectors.

Although a lattice is an infinite structure, virtually all information about it is encoded in the fundamental parallelepiped $F(\boldsymbol{B}) = \{\boldsymbol{B}\boldsymbol{x} : 0^\nu \leq \boldsymbol{x} < 1^\nu\}$, where 0^ν denotes a ν-vector of zeroes and 1^ν a ν-vector of ones. Copies of $F(\boldsymbol{B})$, shifted by lattice vectors, will fill the entire space $\mathbb{R}^\nu$. The volume of $F(\boldsymbol{B})$ holds many important properties, so we give it a name.

Theorem-Definition 2.1.3. *The determinant of a lattice $\mathcal{L}(\boldsymbol{B})$ is the volume of $F(\boldsymbol{B})$, denoted $\det(\mathcal{L}(\boldsymbol{B}))$. This is also known as the volume of $\mathcal{L}(\boldsymbol{B})$ and denoted $\mathrm{vol}(\mathcal{L}(\boldsymbol{B}))$. It can be computed as $\sqrt{\det(\boldsymbol{B}^T\boldsymbol{B})}$.*

One of the most important properties of a lattice $\mathcal{L}$ is the length of its shortest vector $\min_{\boldsymbol{x}\in\mathcal{L}\setminus\{0\}} ||\boldsymbol{x}||$, denoted by $\lambda_1(\mathcal{L})$. The length of the shortest vector can be

bounded from above.

Theorem 2.1.4 (Minkowski's first theorem, (1.24) in [12]). *Let $\mathcal{L}$ be a lattice of dimension ν. Then*

$$\lambda_1(\mathcal{L}) < \sqrt{\nu}\,\mathrm{vol}(\mathcal{L})^{1/\nu}.$$

Above theorem is also referred to as Minkowski's bound. We conclude by introducing a familiar construction from linear algebra, the Gram–Schmidt orthogonalisation of a basis. This is an indispensable tool in lattice theory.

Definition 2.1.5 (Gram–Schmidt (GS) orthogonalisation). *Given a set of basis vectors $\{\boldsymbol{b}_i\}_{i=1}^{\nu}$, its Gram–Schmidt orthogonalisation $\{\boldsymbol{b}_i^*\}_{i=1}^{\nu}$ is defined as*

$$\boldsymbol{b}_1^* = \boldsymbol{b}_1, \quad \boldsymbol{b}_i^* = \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{ij}\boldsymbol{b}_j^*, \text{ where } \mu_{ij} = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^* \rangle}{\langle \boldsymbol{b}_j^*, \boldsymbol{b}_j^* \rangle}.$$

Throughout the text, μ_{ij} will symbolise Gram–Schmidt coefficients defined as above. The matrix $\boldsymbol{U}$ collects all coefficients, $\boldsymbol{U}_{ij} = \mu_{ij}$.

This concludes our mathematical discussion of lattices. The interested reader may indulge in Appendix A.2, in which more theorems are presented together with a deeper investigation into the connection between number theory and lattices.

2.1.2 The Discrete Gaussian Distribution

Randomness is introduced into lattice cryptography by sampling certain variables from probability distributions. We introduce the most commonly used error distribution in lattice cryptography called the discrete Gaussian distribution. It is standard in works concerning LWE and has been used in works concerning NTRU. While there are other distributions available, we will only consider discrete Gaussians in this text.

Definition 2.1.6 (Discrete Gaussian [13]). *The discrete Gaussian distribution $\mathcal{D}_{\mathcal{L},\sigma}$ with standard deviation $\sigma > 0$ over a lattice $\mathcal{L}$ is the probability distribution with support on $\mathcal{L}$ that assigns a probability proportional to*

$$\exp\left(\frac{-||\boldsymbol{x}||^2}{2\sigma^2}\right),$$

for each $\boldsymbol{x} \in \mathcal{L}$.

When $\mathcal{L} = \mathbb{Z}^\nu$, it is easy to see that $\mathcal{D}_{\mathbb{Z}^\nu,\sigma}$ is just the product distribution of ν independent random variables distributed as $\mathcal{D}_{\mathbb{Z},\sigma}$.

Remark. *Calling σ a standard deviation is a slight misnomer. A more appropriate (and often used) definition involves a generic parameter $s = \sqrt{2\pi}\sigma$, without referring to standard error or variance. However, this is an accepted misnomer, so it is used here as well.*

When computing bounds for probabilities, the following tail bounds for discrete Gaussians are useful. Below, the distribution is shorthand notation for a random variable distributed as that distribution.

Lemma 2.1.7 (Lemma 2.1 in [13]). *Let $c \geq 1$ and $C = c\exp\left(\frac{1-c^2}{2}\right) < 1$. Then for any $\sigma > 0$ and $\nu \in \mathbb{N}$,*

$$P\left(||\mathcal{D}_{\mathbb{Z}^\nu,\sigma}|| \geq c\sigma\sqrt{\nu}\right) \leq C^\nu.$$

Lemma 2.1.8 (Lemma 2.2 in [13]). *Let $\sigma > 0, T > 0, \boldsymbol{x} \in \mathbb{R}^\nu$, then*

$$P(|\langle \boldsymbol{x}, \mathcal{D}_{\mathbb{Z}^\nu,\sigma}\rangle| \geq \sqrt{2\pi}T\sigma||\boldsymbol{x}||) < 2\exp(-\pi T^2).$$

2.1.3 Computational Problems on Lattices

Thus far we have only discussed the definition and properties of lattices. In this section, we provide three computational problems on lattices: The Shortest Vector Problem (SVP), the Closest Vector Problem (CVP) and the Bounded Distance Decoding Problem (BDD). The common theme between these problems is their geometric nature. There are no known polynomial time algorithms that solve these problems, not even on a quantum computer. Therefore, they are useful as foundations for cryptosystems, similar to the factorisation or discrete logarithm problems in classic public-key cryptography.

The cryptosystem primitives are often algebraic in nature and have no obvious connection to above problems. To provide a connection to them, one finds a way to embed the cryptographic problem into a lattice where solving one of the computational problems also breaks the cryptosystem. We will do this multiple times in Section 2.2. By providing such a method, the hardness of the cryptographic constructions require the hardness of lattice problems.

Minkowski's first theorem provides a bound on the length of the shortest vector, but this bound is seldom tight. Finding the true length of the shortest vector is therefore a computationally nontrivial problem. We begin by providing a slight relaxation of SVP.

Definition 2.1.9 (Approximate Shortest Vector Problem (γ-SVP) [12]). *Given a lattice basis $\boldsymbol{B}$ and $\gamma \geq 1$, find a lattice vector $\boldsymbol{x} \in \mathcal{L}(\boldsymbol{B}) \setminus \{0\}$ such that $||\boldsymbol{x}|| \leq \gamma\lambda_1(\mathcal{L}(\boldsymbol{B}))$.*

When $\gamma = 1$, this problem is called exact SVP or just SVP. For larger γ, the problem becomes easier. We show in Appendix A.3.2 that $2^{(\nu-1)/2}$-SVP is solvable in polynomial time. SVP has a variant that becomes important in this thesis.

Definition 2.1.10 (τ-Unique SVP (uSVP$_\tau$) [14]). *Given a lattice basis $\boldsymbol{B}$ and $\tau \geq 1$ with the property that $\lambda_2(\mathcal{L}(\boldsymbol{B})) > \tau\lambda_1(\mathcal{L}(\boldsymbol{B}))$, find $\boldsymbol{x} \in \mathcal{L}(\boldsymbol{B}) \setminus \{0\}$ such that $||\boldsymbol{x}|| = \lambda_1(\mathcal{L}(\boldsymbol{B}))$.*

This, comparing to the last definition, is exact SVP with the extra requirement that exactly one vector is the shortest. Again, a larger τ yields an easier problem.

Definition 2.1.11 (Approximate Closest Vector Problem (γ-CVP) [12])**.** *Given a lattice basis $\boldsymbol{B}$, a vector $\boldsymbol{u} \in \mathbb{R}^\nu$ and $\gamma \geq 1$, find a lattice vector $\boldsymbol{x} \in \mathcal{L}(\boldsymbol{B})$ such that $||\boldsymbol{x} - \boldsymbol{u}|| \leq \gamma ||\boldsymbol{y} - \boldsymbol{u}||$ for any other $\boldsymbol{y} \in \mathcal{L}(\boldsymbol{B})$.*

Intuitively, γ-CVP asks for a lattice vector $\boldsymbol{x}$ which is at most γ away from $\boldsymbol{u}$. Setting $\gamma = 1$ yields exact CVP or just CVP, meaning we should find the closest lattice vector to $\boldsymbol{u}$. Finally, we formulate a variant of CVP where the maximal distance from the vector $\boldsymbol{u}$ to the lattice is known.

Definition 2.1.12 (α-Bounded Distance Decoding (α-BDD) [14])**.** *Given a lattice basis $\boldsymbol{B}$, a vector $\boldsymbol{u} \in \mathbb{R}^\nu$ such that the distance from $\boldsymbol{u}$ to $\mathcal{L}(\boldsymbol{B})$ is bounded,* $\text{dist}(\boldsymbol{u}, \mathcal{L}(\boldsymbol{B})) \leq \alpha \lambda_1(\mathcal{L}(\boldsymbol{B}))$*, find $\boldsymbol{x} \in \mathcal{L}(\boldsymbol{B})$ closest to $\boldsymbol{u}$.*

The hardness of lattice problems has been extensively studied. Notably, while a larger γ makes SVP easier, a larger α makes α-BDD harder. The decision variant of CVP, i.e. deciding whether a lattice vector exists within a given radius, is known to be NP-complete, while SVP is NP-complete under randomised reductions [12]. α-BDD is NP-hard for $\alpha > 1/\sqrt{2}$ while uSVP_γ is NP-hard for $\gamma < 1 + 2^{-n^c}$ for some constant c [14].

Many more computational problems on lattices exist together with reductions between them as they appear naturally in varying contexts. Most of them relate back to SVP or CVP. The interested reader should refer to [12].

2.2 Lattice-based Cryptosystem Primitives

In this section, we discuss the three cryptosystem primitives that will be studied: NTRU, module-LWE (which also encompasses ring-LWE and unstructured LWE) and NTWE. They are presented in an unified way. For each primitive, we first define the statistical distribution that underlies the problem. We then define a search version for each problem. Finally, reductions to lattice problems are presented. For NTRU, ring-LWE and NTWE, an example cryptosystem is also presented. The examples are used for discussing correctness of decryption.

2.2.1 NTRU

The NTRU problem and cryptosystem was first introduced by Hoffstein, Pipher and Silverman in [2]. The problem is algebraic in nature but can be discussed in terms of lattices using coefficient embedding (see Appendix A.2.2), as the authors do in their seminal paper.

We begin the unified description by defining the NTRU distribution. We let n and q be positive integers and $p(x)$ a polynomial of degree n. Let $\Phi_n(x)$ be the nth cyclotomic polynomial (see Appendix A.2.2). Common choices are n prime and $p(x) = \Phi_1(x)\Phi_{n-1}(x) = x^n - 1$ or n a power of two and $p(x) = \Phi_n(x) = x^n + 1$. Throughout this text, we will work with the former. This is due to ease of calculation and flexibility when running experiments. However, the choice has a slight impact on security, as most proofs are formulated for cyclotomic

polynomials, while $x^n - 1$ is a product of them with one trivial factor. If security is dependent on n, it will in reality be dependent on $n - 1$. For large n this difference is negligible.

Definition 2.2.1 (NTRU distribution)**.** *The NTRU distribution is, given a distribution χ and a ring $R_q = \mathbb{Z}_q[x]/(p(x))$, is the one obtained by sampling small coefficients for two polynomials $f, g \in R_q$ from χ such that f is invertible. The output is $h = g \cdot f^{-1} \mod R_q$.*

The choice of distribution χ affects security. Classic NTRU uses different distributions over ternary coefficients $\{-1, 0, 1\}$. Recent works, such as [7], have also considered $\chi = \mathcal{D}_{\mathbb{Z},\sigma}$. For to consistency with LWE, we will use the discrete Gaussian for NTRU.

Definition 2.2.2 (NTRU problem)**.** *The $NTRU_{R_q,\chi}$ problem is to recover any shift $(x^i f, x^i g)$ of (f, g) from h.*

The original NTRU problem can be generalised when considered in matrix form. When $p(x) = x^n - 1$, elements of R_q may be represented as $n \times n$ matrices, by taking the circulant matrix of the coefficient embedding of the polynomial. This version is therefore called circulant NTRU. Multiplication and inversion of polynomials can then be calculated as multiplication and inversion of matrices, and f and g are replaced by matrices $\boldsymbol{F}$ and $\boldsymbol{G}$. The reduction to a lattice problem is made by embedding the governing equation $\boldsymbol{H} = \boldsymbol{G}\boldsymbol{F}^{-1} \mod q$ into a homogeneous matrix equation

$$\underbrace{\begin{bmatrix} q\boldsymbol{I}_n & \boldsymbol{H} \\ 0 & \boldsymbol{I}_n \end{bmatrix}}_{\boldsymbol{C}_N} \begin{bmatrix} \_ \\ \boldsymbol{F} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{F} \end{bmatrix}, \tag{2.1}$$

where _ denotes the vector that corresponds to reduction modulo q. $\boldsymbol{C}_N$ spans a lattice of dimension $2n$ with volume q^n. It has a sublattice of rank n spanned by $\begin{bmatrix} \boldsymbol{G} & \boldsymbol{F} \end{bmatrix}^T$ as can be seen by the matrix equation. In other words, it contains all rotations of f and g. If the secret keys, viewed as n-vectors, are unusually short in this basis, the sublattice is dense in the sense that it has much smaller volume than q^n.

2.2.1.1 An NTRU Cryptosystem

We illustrate how NTRU may be used in a cryptosystem by presenting NTTRU (not a typo) from [15]. They operate in the ring $R_q = \mathbb{Z}_q[x]/(x^{768} - x^{384} + 1)$ for faster computation. However, the cryptosystem works in our example ring $\mathbb{Z}_q[x]/(x^n - 1)$ as well.

- $\mathsf{KeyGen}(q, R_q)$: Draw $f' \leftarrow \chi^n$ and $g \leftarrow \chi^n$, interpreted as polynomials. Let $f = 3f' + 1$. Return f as secret key and $h = 3gf^{-1}$ as public key.
- $\mathsf{Enc}(M, h)$: Draw $r \leftarrow \chi^n$ and interpret in R_q. Interpret M as a polynomial in R_q with binary coefficients. Compute and return $c = h \cdot r + M \in R_q$.

- $\mathsf{Dec}(c, f)$: Compute

$$c \cdot f = 3(g \cdot r + f' \cdot M) + M. \tag{2.2}$$

  Now, reducing modulo q to the set of representatives $[-q \,//\, 2, q \,//\, 2]$, followed by reduction modulo 3, we are left with M.

It can be shown that decryption is correct if each coefficient of $g \cdot r + f' \cdot M$ is less than $(q-1)/6$ [15, p. 12].

2.2.2 Learning With Errors

Learning With Errors (LWE) is a class of computational problems, first introduced by Regev in his seminal work on lattice cryptography [3]. In this text, we are concerned with three variants: unstructured LWE, Ring-LWE (RLWE) and Module-LWE (MLWE). The last variant generalises the two former. Thus, only MLWE is presented in the main text together with how the other variants may be recovered. For completeness LWE and RLWE are also discussed in Appendix A.4.

The Module Learning With Errors (abbreviated module-LWE or MLWE) problem was introduced in [9]. This allows the addition of algebraic structure over fully unstructured LWE, although without adding as much structure as RLWE. It is the basis for CRYSTALS-Kyber (recall, a NIST-PQC standard).

As before, we begin by introducing the MLWE distribution. We parametrise MLWE by the ring of integers $R = \mathcal{O}_{\mathbb{K}}$ of a number field $\mathbb{K}$ (see Appendix A.2.2). The usual choice is the ring $R_q = \mathbb{Z}_q[x]/(p(x))$ with $p(x) = x^n + 1$ with n a power of two. In this setting, Fourier Transform-related computation techniques provide significant speed-up. We instead use $p = x^n - 1$ where applicable due to the discussion in Section 2.2.1. The probability distribution has variance σ^2 and is usually a discrete Gaussian or a related distribution.

Definition 2.2.3 (MLWE distribution)**.** *The primal discrete MLWE distribution $A_{\boldsymbol{s},\psi}$ is, given a secret $\boldsymbol{s} \in (R_q)^d$ and a probability distribution ψ over R, the one given by sampling $\boldsymbol{a} \in (R_q)^d$ uniformly at random, $e \in R$ from ψ and computing $(\boldsymbol{a}, b = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e \mod qR) \in (R_q)^d \times R_q$. The pair $(\boldsymbol{a}, b)$ is called an MLWE sample.*

Next we introduce the search problem for MLWE.

Definition 2.2.4 (Search MLWE)**.** *The search-MLWE$_{q,\psi}$ problem is to recover $\boldsymbol{s} \in (R_q)^d$ given m independent MLWE samples $(\boldsymbol{a}_i, b_i = \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + e_i \mod qR), 1 \leq i \leq m$.*

Interestingly, MLWE is a generalisation of both LWE and RLWE. We recover RLWE by setting $d = 1$, reducing the module rank to 1. We recover LWE by setting $n = 1$, which removes the algebraic structure.

To convert the MLWE problem into a lattice problem, we first consider it as a BDD problem on the module lattice, which is then turned into SVP using Kannan's embedding (also see Appendix A.4.1). Considering e and b as polynomials, we represent them as $n \times n$ matrices $\boldsymbol{E}$ and $\boldsymbol{B}$. As $\boldsymbol{a}$ and $\boldsymbol{s}$ are vectors of ring elements,

we pack the polynomials into $n \times dn$ and $dn \times n$ matrices $\boldsymbol{A}$ and $\boldsymbol{S}$ respectively. Multiple samples are stacked upon each other. In the end, we end up with the matrix equation

$$\underbrace{\begin{bmatrix} q\boldsymbol{I}_{mn} & -\boldsymbol{A} & \boldsymbol{B} \\ 0 & \boldsymbol{I}_{dn} & 0 \\ 0 & 0 & t\boldsymbol{I}_n \end{bmatrix}}_{\boldsymbol{C}_M} \begin{bmatrix} \_ \\ \boldsymbol{S} \\ \boldsymbol{I}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} \\ \boldsymbol{S} \\ t\boldsymbol{I}_n \end{bmatrix}. \tag{2.3}$$

Here, $\boldsymbol{A}$ has size $mn \times dn$, $\boldsymbol{B}$ and $\boldsymbol{E}$ have size $mn \times n$, and $\boldsymbol{S}$ has size $dn \times n$. The parameter t appears in the embedding process and is usually set to 1. The sum total is that $\boldsymbol{C}_M$ spans a lattice of dimension $n(m+d+1)$ and has a sublattice, generated by $\begin{bmatrix} \boldsymbol{E} & \boldsymbol{S} & t\boldsymbol{I}_n \end{bmatrix}^T$ of dimension n. As MLWE generalises LWE and RLWE, $\boldsymbol{C}_M$ also encapsulates the LWE and RLWE lattices $\boldsymbol{C}$ and $\boldsymbol{C}_M$ respectively if parameters are chosen as above. Specifically, $\boldsymbol{C}_R = \boldsymbol{C}_M | d = 1$ and $\boldsymbol{C} = \boldsymbol{C}_M | n = 1$ (cf. (A.2) and (A.3)).

As $\boldsymbol{C}_M$ is upper triangular, we have

$$\mathrm{vol}(\mathcal{L}(\boldsymbol{C}_M)) = q^{mn} \cdot t^n. \tag{2.4}$$

2.2.2.1 An LWE-based Cryptosystem

We close the section on LWE by presenting a cryptosystem based on RLWE, which means we only work in $d = 1$ copies of R_q. This is mainly to illustrate how the primitives can be used in practice. The cryptosystem will not be further discussed in this text, while the correctness lemma presented below is used in Chapter 4. The system we present here is the ring variant of the Lindner–Peikert cryptosystem, described in [13].

Take $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ for n a power of two and χ_k, χ_e distributions over R that are sufficiently narrow around 0, for example discrete Gaussians with small variance.

- KeyGen: Draw $r_1, r_2 \in R_q$ from χ_k and $a \in R_q$ uniformly. Let $p = r_1 - a \cdot r_2 \in R_q$. Return $pk = (p, a)$ and $sk = r_2$.
- Enc(a, p, M): Draw $e_1, e_2, e_3 \in R_q$ from χ_e. Interpret the message M as an element in R_q (see below), compute and return

$$(c_1, c_2) = (a \cdot e_1 + e_2, p \cdot e_1 + e_3 + M).$$

- Dec$((c_1, c_2), r_2)$: Compute

$$c_1 \cdot r_2 + c_2 = \ldots = e_1 \cdot r_1 + e_2 \cdot r_2 + e_3 + M.$$

For appropriate distributions χ_k and χ_e, the errors $e_1 \cdot r_1 + e_2 \cdot r_2 + e_3$ will be small and the message can be recovered by an error tolerant decoder. In [13], the authors provide an example for interpreting M as ring elements by letting $M \to Mq \,//\, 2$. Decoding is done by letting decrypted coefficients in $[-q \,//\, 4, q \,//\, 4]$ be 0, and

1 otherwise. For a correct decryption in this setting, we need the size of each coefficient in the error sum to be less than $q \,//\, 4$ with a high probability. To this end, we will use a lemma from [13] that gives a condition for correctness.

Lemma 2.2.5 (Correctness in Lindner–Peikert, Lemma 3.1 in [13])**.** *In the cryptosystem above, the error probability per symbol is bounded from above by any desired $\delta > 0$ if*

$$s_k s_e \leq \frac{\pi\tau}{c\sqrt{n\ln(2/\delta)}},$$

for an error tolerance τ and $c \geq 1$ the constant in Lemma 2.1.7.

In above example of message encoding, $\tau = q \,//\, 4$. This lemma will be used for asymptotic analysis in this text, we therefore assume $\tau = O(q)$ if not stated otherwise.

2.2.3 NTWE

NTWE emerges as a natural combination of the NTRU and MLWE problems and is introduced in [10]. We present the distribution, search version of the problem, primal lattice formulation and a cryptosystem based on NTWE.

The main idea behind NTWE is to construct the public key in the same way as NTRU by using two polynomials f and g to construct $h = gf^{-1}$, but to replace g by an MLWE sample. More precisely, the NTWE distribution is given as follows. The definition in [10] uses the ring $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ for n a power of two, which is $\mathcal{O}_{\mathbb{K}}$ if $\mathbb{K}$ is a cyclotomic number field. We will operate in $\mathbb{Z}_q[x]/(x^n - 1)$ due to the discussion in Section 2.2.1. Following the notation for MLWE, d denotes the rank of a module and m denotes the number of samples.

Definition 2.2.6 (NTWE distribution)**.** *The NTWE distribution $\mathcal{W}(\boldsymbol{s}, f, \psi)$ is, given a secret $\boldsymbol{s} \in (R_q)^d$, an invertible polynomial $f \in R_q$ and a distribution ψ, the one obtained by sampling $\boldsymbol{a} \in R_q^d$ uniformly at random, e from ψ and computing*

$$(\boldsymbol{a}, b = (\langle \boldsymbol{a}, \boldsymbol{s} \rangle + e) \cdot f^{-1}) \in R_q^d \times R_q.$$

The pair $(\boldsymbol{a}, b)$ is called an NTWE sample.

Naturally, $\boldsymbol{s}$ and f have to be sampled according to some distribution as well, usually ψ. However, these are constant across multiple samples and therefore kept outside the definition of a sample.

We now provide the search version of the NTWE problem.

Definition 2.2.7 (Search NTWE)**.** *The search-NTWE$_{q,\psi,m}$ problem is to recover any shift $(x^i \boldsymbol{s}, x^i f)$ of the secret key $(\boldsymbol{s}, f)$ given at most m samples from $\mathcal{W}(\boldsymbol{s}, f, \psi)$.*

The decision problem is, informally, to distinguish NTWE samples from uniform given at most m samples.

Collecting m samples, we may construct a lattice to perform lattice attacks. This is done in a similar manner to MLWE. We represent each sample $(\boldsymbol{a}_i, b_i)$ as matrices

$(\boldsymbol{A}_i, \boldsymbol{B}_i) \in \mathbb{R}^{n \times dn} \times \mathbb{R}^{n \times n}$ and stack the samples to get matrices $\boldsymbol{A} \in \mathbb{R}^{mn \times dn}$ and $\boldsymbol{B} \in \mathbb{R}^{mn \times n}$. The secret key $(\boldsymbol{s}, f)$ is represented as matrices $(\boldsymbol{S}, \boldsymbol{F}) \in \mathbb{R}^{dn \times n} \times \mathbb{R}^{n \times n}$ and errors $\boldsymbol{e}$ as $\boldsymbol{E} \in \mathbb{R}^{mn \times n}$. The matrix equation

$$\underbrace{\begin{bmatrix} q\boldsymbol{I}_{mn} & -\boldsymbol{A} & \boldsymbol{B} \\ 0 & \boldsymbol{I}_{dn} & 0 \\ 0 & 0 & t\boldsymbol{I}_n \end{bmatrix}}_{\boldsymbol{C}_W} \begin{bmatrix} \_ \\ \boldsymbol{S} \\ \boldsymbol{F} \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} \\ \boldsymbol{S} \\ t\boldsymbol{F} \end{bmatrix} \tag{2.5}$$

shows that the lattice basis $\boldsymbol{C}_W \in \mathbb{R}^{n(m+d+1) \times n(m+d+1)}$ permits a dense sublattice $[\boldsymbol{E}\ \boldsymbol{S}\ t\boldsymbol{F}]^T \in \mathbb{R}^{n(m+d+1) \times n}$. As usual, we often set $t = 1$.

2.2.3.1 A Cryptosystem from NTWE

We present the cryptosystem suggested by Gärtner in [10] and will use it for correctness analysis. Its security relies on both the NTWE and MLWE problems. We use the same distribution ψ for all secrets, although they may be different. $\lceil \rfloor_R$ is a randomised rounding function, i.e. a rounding function that rounds to one of the two closest integers according to some probability distribution.

- KeyGen: Draw $\boldsymbol{A} \in R_q^{d \times d}$ (note, a matrix of ring elements) uniformly, $\boldsymbol{s} \in R_q^d$, $\boldsymbol{e} \in R_q^d$ and $f \in R_q$ from ψ. Compute f_2^{-1}, the inverse of f in R_2. Compute $\boldsymbol{b} = (\boldsymbol{A} \cdot \boldsymbol{s} + \boldsymbol{e}) \cdot f^{-1} \in R_q^d$. Return $sk = (\boldsymbol{s}, f, f_2^{-1})$ and $pk = (\boldsymbol{A}, \boldsymbol{b})$.
- Enc$(pk, M \in R_2)$: Draw e' from ψ, and $\boldsymbol{s}', \boldsymbol{e}''$ from ψ^d. Compute $c_1 = \langle \boldsymbol{s}', \boldsymbol{b} \rangle + e' + \lceil Mq/2 \rfloor_R$ and $\boldsymbol{c}_2 = \boldsymbol{s}' \cdot \boldsymbol{A} + \boldsymbol{e}''$. Return $(c_1, \boldsymbol{c}_2) \in R_q \times R_q^d$.
- Dec$(sk, (c_1, \boldsymbol{c}_2))$: Compute $v = c_1 \cdot f - \langle \boldsymbol{c}_2, \boldsymbol{s} \rangle \mod q$, interpreting it as an element in R_q. Compute $u = \lceil v2/q \rfloor$ interpreted as an element in R_2. Return $u \cdot f_2^{-1}$.

The cryptosystem is correct if each coefficient of

$$(r + e') \cdot f - \langle \boldsymbol{e}'', \boldsymbol{s} \rangle + s' \cdot e$$

is less than $q/4$ for r sampled uniformly in $\{\frac{-1}{2}, \frac{1}{2}\}$. In [10], the authors provide conditions for correctness for the multiple sample analogue of above cryptosystem. We will use it to analyse correctness, so we present it here.

Lemma 2.2.8 (NTWE correctness, Lemma 7 in [10])**.** *Let ψ_{gen} and ψ_{enc} be discrete Gaussians with standard deviation σ_{gen} and σ_{enc} respectively. The error probability per symbol δ when decrypting the all 0 message is bounded from above given that*

$$\sigma_{gen}\sigma_{enc} \leq \frac{q}{8\sqrt{4(d+1)n\ln(2/\delta)}},$$

except for a probability less than 2^{-n} over the randomness in the ciphertext.

2.3 Lattice Reduction

This section discusses the algorithms behind the most natural and successful way of breaking lattice-based cryptosystems, namely lattice reduction. We present the intuition behind why lattice reduction yields solutions to cryptosystem problems and provide a high-level description of the most used reduction algorithm, BKZ.

Consider solving SVP. A naive way to do this would be to try linear combinations of basis vectors and keeping track of which combination is the shortest. There are, of course, infinitely many linear combinations, so this is an inefficient strategy. It also depends on the choice of basis. A basis with very long vectors that lie close to each other, as the dashed basis in Figure 2.1, will lead to constantly overshooting lattice points close to the origin. This means the SVP solution will be a linear combination with large coefficients. If the lattice basis is close to being orthogonal and short however, as the solid basis in Figure 2.1, we can efficiently test many combinations close to the origin. Thus, for solving SVP a ”good” basis is one with short and close to orthogonal vectors, while a ”bad” one has long and non-orthogonal vectors. The goal of lattice reduction is to turn a bad basis into a good one.

This also has implications on security. If, as we have seen, the secret key is encoded as the shortest vector in a lattice, the security depends on SVP being hard. A reduction algorithm that finds a good basis can therefore be used to break lattice-based cryptosystems.

A basis with randomly selected vectors, such as one constructed from a cryptosystem, is not expected to be good. There will be very long vectors as well as very short vectors. A reduction algorithm attempts to make the long vectors shorter and short vectors longer as to orthogonalise the basis. However, as the volume of a lattice is fixed, this can be thought of as moving basis vector "mass" from the long vectors to the short vectors.

Thus, the quality of a reduced basis can be measured in terms of the Gram–Schmidt norms $||\boldsymbol{b}_i^*||$. Specifically, a common way to represent a basis after reduction is to plot $\ln ||\boldsymbol{b}_i^*||$ in decreasing order, which usually lie on a slope as in Figure 2.2. This is called the shape of the basis. The flatter the slope, the more successful the basis reduction is. In this way, all a reduction algorithm attempts is to flatten this slope.

Naturally, one cannot always select perfectly orthogonal and short basis vectors for all lattices, so instead of short and orthogonal, we call a good basis *reduced*. There are multiple definitions of a reduced basis. If the definition is constructive, there will also be a natural algorithm that follows.

Before embarking on presenting BKZ, we provide a definition of projection, a concept that appears often in this context.

Definition 2.3.1 (Projection, (2.6) in [12]). *Given a vector $\boldsymbol{v}$, a basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$ and*

an index i, the projection of $\boldsymbol{v}$ away from $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$ is defined as

$$\pi_i(\boldsymbol{v}) = \sum_{j=i}^{\nu} \frac{\langle \boldsymbol{x}, \boldsymbol{b}_j^* \rangle}{\langle \boldsymbol{b}_j^*, \boldsymbol{b}_j^* \rangle} \boldsymbol{b}_j^*.$$

Intuitively, $\pi_i(\boldsymbol{v})$ is the component of $\boldsymbol{v}$ that is orthogonal to $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$. Thus, $\boldsymbol{b}_i^* = \pi_i(\boldsymbol{b}_i)$. We also define the local projected block as $\boldsymbol{B}_{[i:j]} = [\pi_i(\boldsymbol{b}_i), \ldots, \pi_i(\boldsymbol{b}_{j-1})]$ and given a lattice $\mathcal{L}$ with basis $\boldsymbol{B}$, a projected lattice as $\mathcal{L}_{[i:j]} = \mathcal{L}(\boldsymbol{B}_{[i:j]})$.

2.3.1 BKZ Reduction

Block Korkine–Zolotarev (BKZ) reduction is the most practically efficient method of reduction as of writing. It builds on the idea of LLL, which in turn builds on Gauss–Lagrange reduction in two dimensions. These are described at length in Appendix A.3 together with a full algorithm description of BKZ. For the extent of this overview, LLL can be seen as a polynomial time algorithm that improves the basis as described above.

The core idea of BKZ is encoded in what is considered a reduced basis.

Definition 2.3.2 (BKZ reduced [16])**.** *A basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$ is Block Korkine-Zolotarev (BKZ) reduced with block size β, β-BKZ reduced for short, if*

1. *It is LLL reduced, and*
2. $||\boldsymbol{b}_i^*|| \leq \lambda_1(\mathcal{L}(\boldsymbol{B}_{[i:\min(i+\beta,\nu)]})) \quad \forall 1 \leq i \leq \nu.$

This definition suggests a reduction algorithm that will satisfy the conditions. The algorithm works by solving SVP on the projected sublattice $\mathcal{L}(\boldsymbol{B}_{[i:\min(i+\beta,\nu)]})$ of dimension β, called a block. The block size is usually a fraction of the lattice dimension ν, thus the active block slides over the lattice basis as to satisfy the second condition at each index i.

There exist multiple algorithms for solving SVP. The common theme among them is their exponential runtime in β, which means BKZ has an exponential runtime in β as well. The performance of BKZ has nevertheless been improved over the years, leading to a variant called Progressive BKZ which is the variant implemented into the reduction library `fpLLL` used in this thesis. A discussion of the improvements is available in Appendix A.3.4.

2.3.2 Heuristics for Lattices and Reduction

For lattice problems and reduction algorithms it is often possible to prove bounds on quality, complexity and length of vectors, among other quantities. In practice however, one finds that lattices and algorithms are much better behaved than the provable bounds. As such, provable bounds are theoretically interesting, but not useful for practical calculations. To make useful predictions that are not too pessimistic, one usually resorts to *heuristics* to predict how lattice problems and algorithms will behave on average. These are usually not proved and are instead

based on empirical observations. In this section, we present heuristics that are common in cryptanalysis based on reduction theory. We present the Gaussian Heuristic that predicts the expected length of a shortest vector and the Geometric Series Assumption that approximates the shape of BKZ reduced bases.

2.3.2.1 Gaussian Heuristic

Heuristic 2.3.3 (Gaussian Heuristic)**.** *Let $\mathcal{L}$ be a lattice in $\mathbb{R}^\nu$. The expected number of lattice points inside a measurable set $\mathcal{S}$ is*

$$|\mathcal{L} \cap \mathcal{S}| = \frac{\mathrm{vol}(\mathcal{S})}{\mathrm{vol}(\mathcal{L})}.$$

Using this heuristic, an expectation for the length of the shortest vector in $\mathcal{L}$ can be derived. To accomplish this, we consider a ball $B_R(0)$ of radius R centred around 0 that only contains the shortest non-zero vectors. For large ν, we have

$$\mathrm{vol}(B_R(0)) \approx \frac{1}{\sqrt{\nu\pi}} \left(\frac{2\pi e}{\nu}\right)^{\nu/2} R^\nu,$$

due to Stirling's approximation. To contain only the shortest vectors, $B_R(0)$ would need to satisfy

$$\frac{\mathrm{vol}(B_R(0))}{\mathrm{vol}(\mathcal{L})} \approx 1 \implies R \approx \mathrm{vol}(\mathcal{L})^{1/\nu} \sqrt{\frac{\nu}{2\pi e}} (\nu\pi)^{1/2\nu} \approx \mathrm{vol}(\mathcal{L})^{1/\nu} \sqrt{\frac{\nu}{2\pi e}}.$$

Thus we expect the following.

Heuristic 2.3.4. *Let $\mathcal{L}$ be a lattice in $\mathbb{R}^\nu$. Then we expect that*

$$\lambda_1(\mathcal{L}) \approx \mathrm{vol}(\mathcal{L})^{1/\nu} \sqrt{\frac{\nu}{2\pi e}}. \tag{2.6}$$

Further, we denote $\mathrm{gh}(\nu) = \sqrt{\frac{\nu}{2\pi e}}$ the expected length of the shortest vector in a ν-dimensional lattice of volume 1.

2.3.2.2 Geometric Series Assumption

Remark. *This heuristic is called the Geometric Series Assumption due to [17], whilst the "series" addressed here is finite. A formally correct name would thus be the Geometric Sum Assumption. Luckily, the abbreviation stays the same.*

Using the definition of a BKZ reduced basis (Definition 2.3.2) in conjunction with the Gaussian Heuristic (Heuristic 2.3.4), it is possible to construct a heuristic for the shape of a BKZ-reduced basis.

Heuristic 2.3.5 (Geometric Series Assumption (GSA), as presented in [16])**.** *If $\boldsymbol{B}$ is a BKZ-β reduced basis of dimension ν such that $\mathrm{vol}(\mathcal{L}(\boldsymbol{B})) = V$, we expect*

$$\ln ||\boldsymbol{b}_j^*|| = \frac{\nu - 1 - 2j}{2} \ln(\alpha_\beta) + \frac{1}{\nu} \ln(V),$$

where $\alpha_\beta = \mathrm{gh}(\beta)^{\frac{2}{\beta-1}}$.

Intuitively, the GSA tells us the shape of the basis will be a decreasing straight line. The slope is determined by $\ln(\alpha_\beta)$, meaning a larger β will lead to a flatter line. The intuition about basis reduction agrees with this, as an orthogonal basis will have a flat shape. The GSA is valid when $\beta \ll \nu$ and the slope is well approximated by α_β when $\beta > 50$. It should be noted that the shape will be incorrect for the last $\nu - \beta$ basis vectors, as they are HKZ reduced (see Definition A.3.4) with a different expected shape. To that end, one can construct the Tail-adapted GSA, which is not discussed further here. Additionally, significant effort has been put into correctly estimating the shape of a basis after reduction using simulators. The interested reader can read more in [16].

We instead turn to describing what happens when lattice reduction is run on a q-ary lattice. The model example here is Kannan's embedding for transforming BDD to uSVP for LWE (A.2) and NTRU (2.1). Inspecting the bases, we see that they contain $(q, \ldots, 0)$ and all its rotations followed by orthogonal vectors with diagonal element 1. As the volume is q^ν for some dimension ν, this will mean the basis will have a shape of ν q-vectors followed by ν ones, depicted as the dashed line in Figure 2.2. Running lattice reduction on this basis will mean the algorithm tries to push the slope as flat as possible. Intuitively, this can be seen as moving area from the q-vectors to the 1-vectors.

LLL guarantees that the first vector will never get longer. As it is a self-dual algorithm in the sense it performs (essentially) the same operations on the basis and its dual (see Definition A.2.7) [18], this also implies the last GS-vector can never get shorter. Further, this applies to all steps of the algorithm, meaning if there are a few GS-vectors in the beginning that are not affected, the last ones will not become longer either. In effect, we may have q-vectors left and by self-duality, 1-vectors as well. Between the q-vectors and the 1-vectors, there will be a joining slope. This produces a characteristic Z-shape for the reduced basis, depicted as the solid line in Figure 2.2.

BKZ is not a self-dual algorithm (although a self-dual variant exists, see [18]). While there is a guarantee that the first vector will never get longer, this does not translate to that the last vector never will get shorter. In experiments, the profile approximately follows a Z-shape with a slight downward kink between the slope and right flat part [16]. Nevertheless, it is possible to restrict BKZ to run only on the middle vectors $\boldsymbol{B}_{[\nu-l:\nu+l]}$ for some l which again will produce a Z-shape. As the shape of q-ary lattices remains an open question, the standard assumption is the following heuristic, again given by the solid line in Figure 2.2.

Heuristic 2.3.6 (Z-shaped GSA (ZGSA) [7])**.** *Let $\mathcal{L}$ be a lattice in dimension 2ν with basis $\boldsymbol{B}$ containing ν q-vectors. After BKZ-β reduction, we expect*

$$||\boldsymbol{b}_j^*|| = \begin{cases} q, & j \leq \nu - l, \\ \sqrt{q}\alpha_\beta^{(2\nu-1-2j)/2}, & \nu - l \leq j \leq \nu + l - 1, \\ 1, & \nu + l \leq j, \end{cases} \tag{2.7}$$

where $\alpha_\beta = \mathrm{gh}(\beta)^{\frac{2}{\beta-1}}$ and $l = \frac{1}{2} + \frac{\ln q}{\ln \alpha_\beta}$.

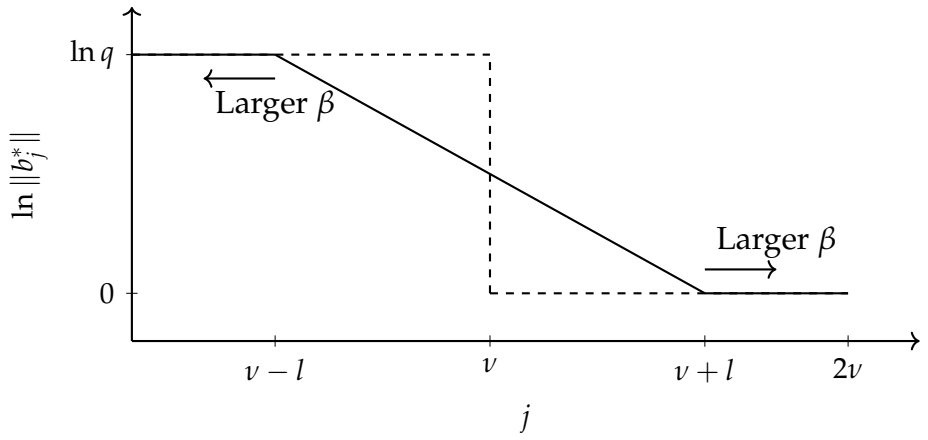


Figure 2.2: A plot of the ZGSA. The dashed line shows the shape before reduction, the solid line after. Larger block sizes β make the slope flatter.

2.4 Reduction Attacks

For LWE and NTRU, the most successful attacks focus on recovering short vectors from lattice versions of the problems. This is done using BKZ or other reduction algorithms by running them until the basis is likely to contain the secret vector. Intuitively, this is similar to factorisation attacks on RSA. Attacks on lattices can target the primal lattices described above, or their duals. In this thesis, we only consider attacks on the primal lattices.

As discussed before, the time complexity of BKZ is mainly dictated by the time complexity of the exact SVP solver that is run on blocks of size β. As such, β effectively serves as a proxy for the cost of an attack and in extension the estimated security of the cryptosystem in question. The time complexity of performing an attack is often given as 2^λ where λ is the number of bits of security. When considering lattice attacks, we will therefore have $\lambda = c\beta + o(1)$, where c and $o(1)$ are constants.

In this section, we describe two reduction attacks, Secret Key Recovery (SKR) and Dense Sublattice Discovery (DSD). The first is the natural attack on the lattice problems and most security estimates for NTRU- and LWE-based cryptosystems are based on the hardness of performing successful SKR. On the other hand, DSD can occur when a q-ary lattice has a dense sublattice. This has only been studied for NTRU as of writing. However, the astute reader will have noticed that all three lattices in (2.1), (2.3) and (2.5) describe such a situation. For certain parameter choices, Albrecht, Bai and Ducas found in [5] that DSD is easier than SKR in NTRU. These instances are called overstretched. The goal of this thesis is to further investigate the relationship between SKR and DSD.

In this treatise, we begin by defining exactly what occurs in these events during a run of BKZ.

Definition 2.4.1 (BKZ events, generalisation of Def. 2.9 in [7])**.** *Consider a BKZ run*

on a q-ary lattice basis with a dense sublattice and define the following events.

1. *Secret Key Recovery at position κ (SKR$_\kappa$). This the first time the secret key or one of its shifts is inserted into the basis, at position κ.*
2. *Dense Sublattice Discovery at position κ (DSD$_\kappa$). This is the first time a vector from the dense sublattice that is strictly longer than the secret keys is inserted, at position κ.*

We include shifts in this definition because the secret key itself can be recovered in polynomial time after a rotation is discovered. An instance of NTRU is considered overstretched when DSD is more likely than SKR. The SKR event is popularly characterised by the 2016 Estimate from [19] (see the next subsection), the DSD event is characterised in [5]–[7] (and described in the second next subsection).

2.4.1 Secret Key Recovery

In the primal lattice formulation, both LWE (A.2) and NTRU (2.1) contain vectors that are much shorter than the Gaussian Heuristic would predict. For cryptanalysis, it is interesting to quantify when such vectors will be found by lattice reduction, as these short vectors contain (rotations of) the secret key. For BKZ, the most used estimate is the following.

Claim 2.4.2 (2016 estimate [19])**.** *Let $\mathcal{L}$ be a lattice of dimension ν with a vector $\boldsymbol{v}$ such that $||\boldsymbol{v}|| \ll \mathrm{gh}(\mathcal{L})$. Then BKZ-$\beta$ finds $\boldsymbol{v}$, assuming the GSA, if*

$$\sqrt{\frac{\beta}{\nu}}||\boldsymbol{v}|| \leq \alpha_\beta^{(2\beta-\nu-1)/2}\mathrm{vol}(\mathcal{L})^{1/\nu}, \tag{2.8}$$

where $\alpha_\beta = \mathrm{gh}(\beta)^{\frac{2}{\beta-1}}$.

Justification. BKZ, solving SVP on β-blocks, will find the shortest vector in each block. Assuming $\boldsymbol{v}$ is evenly distributed over its coordinates, the projection of $\boldsymbol{v}$ on the local block will be short with length $\sqrt{\beta/\nu}||\boldsymbol{v}||$. When BKZ reaches the last full block and operates over $\boldsymbol{B}_{[\nu-\beta:\nu]}$ there will be a chance to recover the last β coordinates of $\boldsymbol{v}$ as we are solving exact SVP. Thus, for a successful insertion into the basis, the projected vector needs to be shorter that what we would usually expect at position $\nu-\beta$, i.e. what the GSA predicts we will find in that position. The success condition is therefore

$$||\pi_{\nu-\beta}(\boldsymbol{v})|| \leq ||\boldsymbol{b}^*_{\nu-\beta}||.$$

To obtain the estimate, we simply insert the GSA's prediction into the right hand side (RHS) and our assumption that the vector is evenly distributed across its coordinates in the left hand side (LHS). △
This estimate has been experimentally validated and is used in [7]. It will be used here to predict the event known as Secret Key Recovery (SKR$_\kappa$).

Using (2.8), it is possible to derive an asymptotic estimate for the hardness of SKR in an NTRU lattice. We follow [7] and show how this is done. One of the main

parts of the theoretical work in this thesis is to produce similar claims for MLWE and NTWE.

Claim 2.4.3 (NTRU-SKR asymptotics)**.** *The BKZ algorithm with $\beta = Bn$ applied to the NTRU lattice with $q = \Theta(n^Q)$ and $||(g|f)|| = O(n^S)$ recovers (a rotation of) the secret key if*

$$B = \begin{cases} \frac{2Q}{(Q+1-S)^2}, & S < 1, \\ \frac{2}{Q+2-2S}, & S \geq 1, \end{cases} + o(1). \tag{2.9}$$

Justification. We begin by noting the dimension of the problem may be reduced by throwing away q-vectors in the beginning of the lattice basis. Keeping k q-vectors is equivalent to working in the projected lattice $\mathcal{L}_{[n-k:2n]}$. Reducing the dimension yields easier lattice problems in general. On the other hand, recalling $\text{vol}(\mathcal{L}_{[n-k:2n]}) = q^k$, this will reduce the volume of the lattice which means the short vector sticks out less in relation to the expected length $\lambda_1(\mathcal{L})$ and becomes harder to find.

When these two factors are taken into account, this means there is an optimal number k of q-vectors to keep. We find this by minimising the RHS in (2.8). The right hand side is

$$\text{gh}(\beta)^{\frac{2\beta-\nu-1}{\beta-1}}\text{vol}(\mathcal{L})^{1/\nu}.$$

Letting $\gamma = \text{gh}(\beta)^{1/(\beta-1)}$, $\nu = n + k$ and $\text{vol}(\mathcal{L}) = q^k$, we want to minimise

$$\gamma^{2\beta-n-k-1}q^{k/(n+k)}.$$

To do this, we differentiate with respect to k to get

$$\frac{\partial}{\partial k}\left(\gamma^{2\beta-n-k-1}q^{k/(n+k)}\right) = \frac{\gamma^{2\beta-n-k-1}q^{k/(n+k)}(n\ln q - (k+n)^2\ln\gamma)}{(k+n)^2},$$

which is equal to 0 when

$$(k+n)^2\ln\gamma = n\ln q.$$

Now by assumption, $\ln q = Q\ln n$, $\beta = Bn$, and $\ln\gamma = \frac{1}{2(\beta-1)}(\ln\beta - \ln(2\pi e)) = \frac{\ln n}{2(Bn-1)} + o(1)$, so

$$(k+n)^2\frac{\ln n}{2Bn-2} \approx (k+n)^2\frac{\ln n}{2Bn} = nQ\ln n \implies k = n(\sqrt{2BQ}-1).$$

However, k cannot become bigger than n. Consequently, we get

$$k = \min(n, n(\sqrt{2BQ}-1)).$$

Having minimised the right hand side, we seek the smallest β that will yield a successful attack by solving for equality in (2.8). If $k = n$ then $\nu = 2n$ and $\text{vol}(\mathcal{L}) = q^n$. Using the asymptotic assumptions, we have

$$(2.8) \iff \sqrt{\frac{Bn}{2n}}O(n^S) = \text{gh}(\beta)^{\frac{2Bn-2n-1}{Bn-1}}\Theta(n^Q)^{n/2n}.$$

Taking logarithms and cancelling small factors, we seek the solution to

$$S \ln n = \frac{1}{2} \ln n \frac{2Bn - 2n}{Bn} + \frac{Q}{2} \implies B = \frac{2}{Q + 2 - 2S}.$$

A similar computation for $k = n(\sqrt{2BQ} - 1)$ yields $B = \frac{2Q}{(Q+1-S)^2}$.

Finally, we seek the region where the different solutions will be valid. In other words, we seek the region where

$$n \left(\sqrt{2Q \frac{2Q}{(Q + 1 - S)^2}} - 1 \right) < n \implies 0 < S < 1.$$

This must mean the other solution is valid when $S \geq 1$, i.e. $k = n$. Thus, we have justified all parts of the claim. △

Remark. *The last equation has another solution, $Q < \frac{S-1}{2}$. However, there is a statistical limit to how small Q can be selected for a given S. For increasing variance, there will be a point where the distribution is indistinguishable from uniform. At that point, the secret key is indistinguishable from a uniformly sampled vector over $[0, q]$. Under such conditions, a correct cryptosystem can never be constructed. We exclude (S, Q) beyond this limit by the following argument. For uniformly sampled secret keys, we expect $\mathbb{E}[||v||] = \sqrt{n \left(\frac{q}{2}\right)^2} \approx \sqrt{n}q$. As we assume $||\boldsymbol{v}|| = O(n^S)$ and need $||\boldsymbol{v}|| \leq \sqrt{n}q$, this gives*

$$Q > S - \frac{1}{2}. \tag{2.10}$$

Therefore, this solution is not relevant.

2.4.2 Dense Sublattice Discovery

We now provide hardness estimates for the second type of attack, DSD. The term "overstretched" in NTRU refers to selecting the modulus q so large that a vector in the dense sublattice spanned by $[\boldsymbol{G}\ \boldsymbol{F}]^T$ becomes easy to recover. Even though such a vector is not (a rotation of) the secret key itself, the first DSD event is usually followed by cascading DSD events until the secret key is recovered. Therefore, the authors in [7] argue that the first DSD event signifies a successful attack. The term fatigue point $\hat{q}$ is used to refer to the point where DSD becomes easier than SKR.

We illustrate how short the "short vectors" are by an example. Consider the NTRU lattice basis $\boldsymbol{C}_N$, and denote the lattice spanned by it $\mathcal{L}(\boldsymbol{C}_N) = \mathcal{L}^{\boldsymbol{H},q}$. Using the Gaussian heuristic (2.6), we expect the minimal length to be

$$\lambda_1(\mathcal{L}^{\boldsymbol{H},q}) \approx \sqrt{\frac{2n}{2\pi e}} \text{vol}(\mathcal{L}^{\boldsymbol{H},q})^{1/2n} = \sqrt{\frac{nq}{\pi e}}.$$

Inspecting the columns of the dense sublattice, we expect $2n$ polynomial coefficients with a variance σ^2, i.e. each column vector has length $\sqrt{2n\sigma^2}$. For large n,

this will be a good approximation of the length of the secret key. As the secret distribution, we select a discrete Gaussian with variance $\sigma^2 = \frac{2}{3}$. For recommended parameter sets $n = 509$ and $q = 2048$ in the NIST-PQC submission NTRUEncrypt [20], the ratio between secret key and Gaussian heuristic is

$$\frac{\sqrt{2n\sigma^2}}{\sqrt{\frac{nq}{\pi e}}} \approx 0.07,$$

meaning the short vectors are much shorter than what is expected for a uniformly random basis.

This type of attack was first described in [5] as a subfield attack, where the sublattice is massaged for LLL to effectively recover it. In 2017, Kirchner and Fouque showed in [6] that simply running BKZ on an overstretched instance will recover the secret sublattice. They found an asymptotic bound on the fatigue point, setting that $\hat{q} < n^{2.784+o(1)}$. Ducas and van Woerden refine this analysis in [7], reducing the asymptotic fatigue point to $\hat{q} = n^{2.484+o(1)}$. We show how this is done, as a major part of the theoretical work in this thesis is to provide similar estimates for MLWE and NTWE.

Claim 2.4.4 (DSD estimate, Claim 3.5 in [7])**.** *The BKZ algorithm with $\beta = Bn$ applied to NTRU with $q = \Theta(n^Q)$ and $||(g|f)|| = O(n^S)$ triggers the DSD event if*

$$B = \frac{8S}{Q^2 + 1} + o(1).$$

Justification. To justify this claim, we need four ingredients in the form of heuristic arguments. First, the volume of the dense sublattice is $\mathrm{vol}(\mathcal{L}^{\boldsymbol{GF}}) \leq ||(g|f)||^n = (2n\sigma^2)^{n/2}$ by Hadamard's inequality (A.1). Second, as the NTRU lattice is q-ary, the basis is assumed to follow the ZGSA.

Third, the authors in [7] observe empirically that the DSD event often occurs around position $\kappa = n + k - \beta$ for a small k, which is indeed earlier in the BKZ run than the expected $\kappa = 2n - \beta$ for SKR. Thus, a vector $\boldsymbol{v}$ that triggers the DSD event must be short inside the projected sublattice BKZ operates on, $\mathcal{L}^{\boldsymbol{H},q}_{[n+k-\beta:n+k]}$ for some k, but also be in the dense sublattice $\mathcal{L}^{\boldsymbol{GF}}$, in other words that $\boldsymbol{v} \in \mathcal{L}^{\boldsymbol{H},q}_{[n+k-\beta:n+k]} \cap \mathcal{L}^{\boldsymbol{GF}}$. Assuming $\boldsymbol{v}$ is approximately evenly distributed over the Gram–Schmidt vectors $\boldsymbol{b}_i^*$, it is reasonable to assume that it also is a short vector in $\mathcal{L}^{H,q}_{[0:n+k]} \cap \mathcal{L}^{\boldsymbol{GF}}$, a larger lattice. The first assumption is justified by the authors observing that the projected length $\pi_{n+k-\beta}(\boldsymbol{v}) \approx \sqrt{\frac{\beta}{n+k}}||\boldsymbol{v}||$, which would be expected if $\boldsymbol{v}$ was evenly distributed over $\boldsymbol{b}_i^*$. As BKZ inserts $\boldsymbol{v}$ only if it is found by the enumeration procedure on $\mathcal{L}^{\boldsymbol{H},q}_{[n+k-\beta:n+k]}$, it is plausible to conclude the following.

Claim 2.4.5 (Primitive DSD estimate, Claim 3.1 in [7])**.** *A tour of β-BKZ triggers DSD if*

$$\pi_{n+k-\beta}(\boldsymbol{v}) < ||\boldsymbol{b}^*_{n+k-\beta}||,$$

for $\boldsymbol{v} \in \mathcal{L}^{\boldsymbol{H},q}_{[0:n+k]} \cap \mathcal{L}^{\boldsymbol{GF}}$ *a short vector, for some* $0 < k \ll n$.

Having established a condition for DSD, the fourth ingredient is to estimate the length of $\boldsymbol{v}$. The rest of the discussion concerns this. The short vector $\boldsymbol{v}$ is assumed to be $||\boldsymbol{v}|| \approx \lambda_1(\mathcal{L}^{\boldsymbol{H},q}_{[0:n+k]} \cap \mathcal{L}^{\boldsymbol{GF}})$, which is approximated by Minkowski's bound (Theorem 2.1.4). To this end, we need to estimate $\mathrm{vol}(\mathcal{L}^{\boldsymbol{H},q}_{[0:n+k]} \cap \mathcal{L}^{\boldsymbol{GF}})$. In [7], the authors prove a lemma and apply it to NTRU. We present the lemma and its application here.

Lemma 2.4.6 (Generalised Pataki–Tural lemma, Lemma 3.3 in [7])**.** *Consider* $\mathcal{L}$ *a* ν *dimensional lattice spanned by* $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$. *For any* n *dimensional sublattice* $\mathcal{L}' \subset \mathcal{L}$,

$$\mathrm{vol}(\mathcal{L}_{[0:s]} \cap \mathcal{L}') \leq \mathrm{vol}(\mathcal{L}') \cdot \left(\min_J \prod_{j \in J} ||\boldsymbol{b}_j^*||, \right)^{-1}$$

where $k = \dim(\mathcal{L}_{[0:s]} \cap \mathcal{L}')$ *and* J *ranges over all subsets of* $\{s, \ldots, \nu - 1\}$ *of size* $n - k$.

Applying this to the NTRU lattice $\mathcal{L}^{\boldsymbol{H},q}$ with dense sublattice $\mathcal{L}^{\boldsymbol{GF}}$, assuming $\dim(\mathcal{L}_{[0:n+k]} \cap \mathcal{L}') = k$ and the ZGSA, we get

$$\mathrm{vol}(\mathcal{L}_{[0:n+k]} \cap \mathcal{L}') \leq \mathrm{vol}(\mathcal{L}^{\boldsymbol{GF}}) \cdot \left(\prod_{j=n+k}^{2n-1} ||\boldsymbol{b}_j^*|| \right)^{-1}.$$

Applying the Generalised Pataki–Tural lemma involves finding the index set J that yields the smallest product. Due to the ZGSA, this is always given by the $n - k$ last basis vectors, meaning we go from $2n - (n - k) = n + k$ to $2n$.

To put it all together and justify the claim, we have that

$$||(g|f)|| = O(n^S) \implies \ln(\mathrm{vol}(\mathcal{L}^{\boldsymbol{GF}})) \leq Sn \ln n + O(n).$$

Using above lemma, the ZGSA and setting $k = Kn$,

$$\begin{aligned} \ln \mathrm{vol}((\mathcal{L}_{[0:n+k]} \cap \mathcal{L}')) &\leq Sn \ln n - \frac{1}{2} \sum_{j=n+k}^{n+l-1} \left(Q + \frac{2n - 1 - 2j}{Bn} \right) \ln n + O(n) \\ &\leq Sn \ln n - \frac{(BQ - 2K)^2}{8B} n \ln n + O(n). \end{aligned}$$

Now we are ready to apply Minkowski's bound on λ_1,

$$\begin{aligned} \ln(\lambda_1(\mathcal{L}_{[0:n+k]} \cap \mathcal{L}')) &\leq \frac{1}{2} \ln(Kn) + \frac{\ln \mathrm{vol}((\mathcal{L}_{[0:n+k]} \cap \mathcal{L}'))}{Kn} + O(1) \\ &\leq \left(-\frac{(BQ - 2K)^2}{8BK} + \frac{S}{K} + \frac{1}{2} \right) \ln(n) + O(1). \end{aligned}$$

According to above claim, we trigger DSD if (considering that above holds for $\boldsymbol{v}$ in the intersection lattice, and cannot become longer when projected down to the

local block)

$$\ln(\lambda_1(\mathcal{L}_{[0:n+k]} \cap \mathcal{L}')) \leq \ln(||\boldsymbol{b}^*_{n+k-\beta}||) \implies$$
$$\left(-\frac{(BQ-2K)^2}{8BK} + \frac{S}{K} + \frac{1}{2}\right)\ln(n) + O(1) \leq \left(\frac{1}{2}Q + \frac{B-K}{B}\right)\ln(n) + O(1) \implies$$
$$B \geq \frac{2\sqrt{((2S-K)^2 + K^2Q^2} + 2(2S-K)}{Q^2}.$$

We want a lower bound on the block size, so the right hand size should be minimised as a function of K. This is accomplished when $K = \frac{4S}{Q^2+1}$. Substituting this in above expression shows the claim. △

To find the asymptotic fatigue point $\hat{q}$, we find the intersection point between the DSD estimate and the SKR (2016) estimate. Here we set $S = 1/2$ to reflect the small variances used in NTRU. Solving for $\hat{q}$ yields the asymptotic value

$$\hat{q} = n^{2.484 + o(1)}.$$

From experiments it is evident that the fatigue point is significantly smaller than the naive expectation $\hat{q} = n^{2.484}$. However, the estimate can be pinned down even more if the analysis is refined or using experiments. Using experiments, the authors in [7] find the fatigue point as

$$\hat{q} = 0.0038 n^{2.484}. \tag{2.11}$$

They also refine their analysis. To this end, the authors focus on predicting $\mathrm{vol}(\mathcal{L}_{[0:n+k]} \cap \mathcal{L}^{\boldsymbol{GF}})$ and $\mathrm{vol}(\mathcal{L}^{\boldsymbol{GF}})$ heuristically instead of bounding them from above. This is clearly explained in [7] thus the details will not be discussed here, except a short summary in the following claim.

Claim 2.4.7 (Refined estimates for NTRU [7])**.** *For $\mathcal{L}$ a $2n$ dimensional NTRU lattice with $\mathcal{L}^{\boldsymbol{GF}}$ its dense sublattice,* $\dim(\mathcal{L}_{[0:n+k]} \cap \mathcal{L}^{\boldsymbol{GF}}) = k$ *before DSD and*

$$\mathbb{E}[\ln(\mathrm{vol}(\mathcal{L}_{[0:n+k]} \cap \mathcal{L}^{\boldsymbol{GF}}))] = \ln(\mathrm{vol}(\mathcal{L}^{\boldsymbol{GF}})) - \left(\sum_{j=n+k}^{2n-1} \ln ||\boldsymbol{b}^*_j||\right)$$
$$+ \sum_{l=k+1}^{n} \psi(l/2) - \psi((n+l)/2) + \zeta'(l)/\zeta(l),$$

where $\psi = \Gamma'/\Gamma$ is digamma function and ζ the Riemann zeta function.

Further, for Matrix NTRU we have that

$$\mathbb{E}[\ln(\mathrm{vol}(\mathcal{L}^{\boldsymbol{GF}}))] = \frac{1}{2}n(\ln(2\sigma^2) + \psi(n)) + \sum_{j=0}^{n-1}\left(\psi((2n-i)/2) + \psi(n)\right),$$

and for Circulant NTRU we have that

$$\mathbb{E}[\ln(\mathrm{vol}(\mathcal{L}^{GF}))] = \frac{1}{2}n(\ln(2\sigma^2) + \psi(1)) + \frac{1}{2}(1 - \ln 2)(n-1).$$

Computing analytically with these concrete predictions becomes unwieldy in practice, which is why the authors implemented a `sage` script to perform estimations. This script will be used for further analysis of the NTRU problem and is referred to as "the estimator".

3

Polynomial Coefficients in NTRU

This chapter concerns research questions 1 and 2. We investigate how the fatigue point changes when the size of polynomial coefficients, encoded by the variance σ^2 in the error distribution, varies. The primary tools are the NTRU asymptotics described in Section 2.4, however experiments and the estimator are used as well.

This chapter is structured as follows.

1. We begin by investigating the intersection point between the SKR and DSD asymptotics for larger variances to discover the SKR region is asymptotically bounded.
2. For smaller σ^2 and n we propose a linear approximation of the asymptotics and fix the constants using experiments.
3. We investigate the asymptotic block size needed for a successful attack in order to propose a parameter set for a secure cryptosystem under DSD.
4. We show correctness of this parameter set for the NTTRU cryptosystem and investigate key and message sizes.
5. Finally, we investigate the behaviour of the estimator for larger σ^2 and use it to produce estimates of the reduction in security by DSD for an NTRU-based homomorphic cryptosystem YASHE.

The first three points are found in Section 3.1, while the following two are found in Sections 3.2 and 3.3 respectively.

3.1 The Interplay Between Modulus and Size of Coefficients in NTRU

We investigate the relationship between the fatigue point $\hat{q}$ and variance in the coefficient distribution σ^2 in this section. We operate in the framework of [7]. Coefficients are sampled from a discrete Gaussian with mean 0 and variance $\sigma^2 = \frac{2}{3}$. We extend their analysis for $\sigma \in (0, \infty)$, and attempt to model the fatigue point as a function of the variance, i.e. to provide $\hat{q} = f(\sigma^2, n)$.

We begin with the asymptotics. Recall the asymptotic block size B needed to

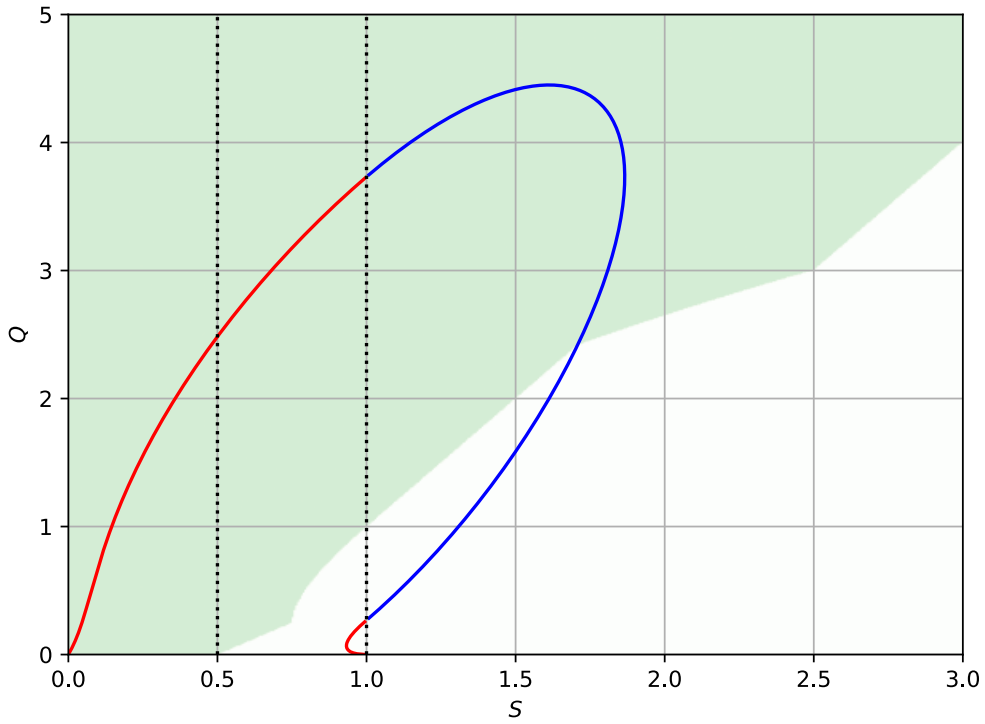


Figure 3.1: The fatigue curve $\hat{Q}(S)$ for $S < 1$ is pictured in red, and for $S \geq 1$ in blue. The green background is the domain of B_{SKR} inside the region bounded by the curve and of B_{DSD} outside the curve.

trigger SKR and DSD respectively was given by

$$B_{\mathrm{DSD}}(S,Q) = \frac{8S}{Q^2+1}, \quad B_{\mathrm{SKR}}(S,Q) = \begin{cases} \frac{2Q}{(Q+1-S)^2}, & S < 1, \\ \frac{2}{Q+2-2S}, & S \geq 1. \end{cases} \tag{3.1}$$

Let $B(S,Q) = \min\{B_{\mathrm{SKR}}(S,Q), B_{\mathrm{DSD}}(S,Q)\}$ be the asymptotic block size needed for successfully performing a primal lattice reduction attack on NTRU. The domain of $B(S,Q)$ is limited by a few factors. First, the uniform indistinguishability limit given by (2.10) comes into play here. Second, we have that $0 \leq \beta \leq 2n$, yielding $0 \leq B(S,Q) \leq 2$. Third, B_{SKR} will be negative when the denominator is negative, so we require $Q > 2S - 2$.

In summary, the domain of $B(S,Q)$ is limited by a maximal block size, a distinguishability criterion and a singularity. The fatigue points are given by the (S,Q) solutions to $B_{\mathrm{DSD}} = B_{\mathrm{SKR}}$, given by the line in Figure 3.1. We denote these by $\hat{Q}(S)$. Note that the curve is only valid within the green shaded region illustrating the domain.

These somewhat surprising asymptotics for the fatigue point permit a few observations.

1. For $S \leq 1$, $\hat{Q}$ monotonously increases with S in an approximately linear fashion.

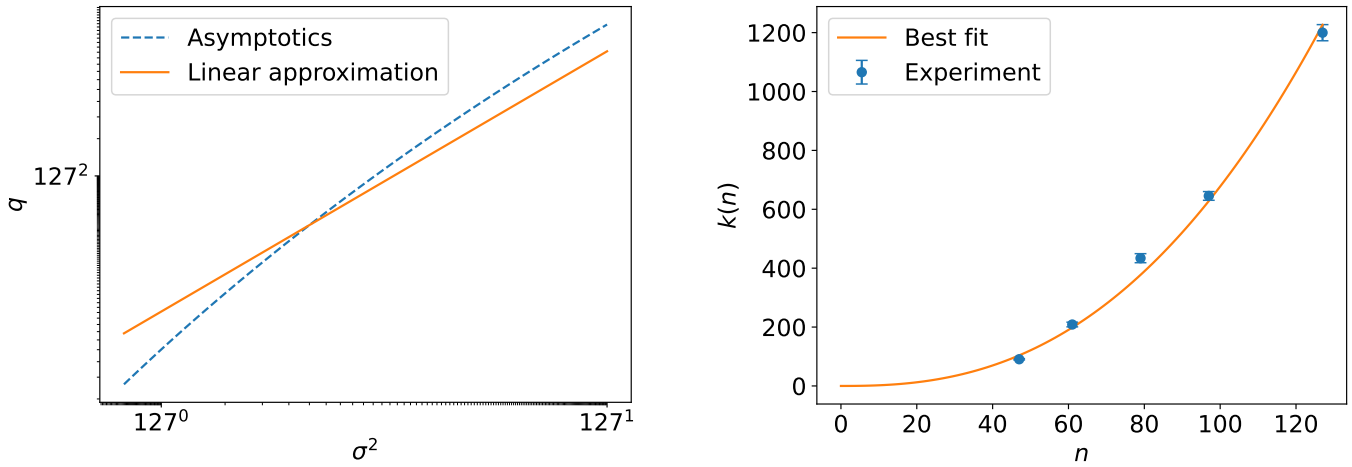
2. For $S > 1$, the slope gradually decreases until there is a plateau at $S \approx 1.6$. At this point, we cannot compensate for selecting a large Q by increasing S and still operate in the SKR regime.
3. For $S \gtrapprox 1.75$, there exist two fatigue points. For some S, this means it is possible to start in the DSD regime, enter the SKR regime and leave it again by increasing Q.

It should be noted that both estimates contain an $o(1)$ constant that is set to 0 in above expressions. This constant may significantly affect the solutions. It is therefore interesting to determine it, although this is not done in this thesis. Nevertheless, it is clear that for $S < 1$ the asymptotic dependence is approximately linear. We therefore make this approximation and investigate the quality of it in the next subsection.

Before continuing, we point out a convenient equation. It is sometimes necessary to translate between σ^2 and S. To this end, recall that $||(g|f)|| = \sqrt{2n\sigma^2}$ for NTRU with discrete Gaussian coefficients. Setting this equal to the asymptotic assumption gives us

$$O(n^S) = ||(g|f)|| = \sqrt{2n\sigma^2} \implies \sigma^2 = O(n^{2S-1}). \tag{3.2}$$

3.1.1 Linear Approximation When $S < 1$



(a) Fatigue point $\hat{q}$ predicted by asymptotics (dashed blue) and by best fit linear approximation (solid orange) on a log-log scale for $n = 127$ and $\frac{2}{3} \leq \sigma^2 \leq n$.

(b) Experimentally determined slopes $k(n)$ for multiple n, together with the line of best fit $k(n) = 0.00729n^{2.484}$ with $R^2 = 0.994$.

Figure 3.2

From (3.2), we get that $0 < S < 1$ is equivalent to $0 < \sigma^2 < n$. From the asymptotic analysis we expect approximately linear dependence between σ^2 and $\hat{q}$. To plot the asymptotic fatigue point for a given n and σ^2, we solve $B_{\text{SKR}}(S, Q) = B_{\text{DSD}}(S, Q)$ numerically for Q with S computed from (3.2) and plot $\hat{q} = 0.0038n^{\hat{Q}}$. This uses the same constant as in (2.11). Such a result for $n = 127$ can be found in Figure 3.2a.

Table 3.1: Experimentally determined slopes $k(n)$ for multiple n, together with coefficients of determination R^2.

n	$k(n)$	Standard error	R^2
47	91	1.16	0.994
61	209	4.04	0.991
79	434	7.84	0.987
97	645	7.55	0.995
127	1200	13.99	0.995

We now proceed with the linear approximation. Working from the hypothesis that

$$\hat{q} = k(n)\sigma^2,$$

for a function $k(n)$, we try to find $k(n)$. The authors in [7] established that $\hat{q} = n^{2.484+o(1)}$. It is therefore reasonable to assume that $k(n) = cn^{2.484}$ for a constant c, meaning that

$$\hat{q} = cn^{2.484}\sigma^2, \quad 0 < \sigma^2 < n.$$

Determining c is a matter finding the fatigue point experimentally, which is only realisable for the author up to $n \approx 128$. To this end, we use the soft binary search program described in [7], that given input n and σ^2 generates NTRU lattices and runs BKZ on them until either SKR or DSD occurs. Multiple q are tested until the fatigue point is found. The author lacks the hardware to perform extensive experiments. Thus, the search is run for $n = 47, 61, 79, 97$ and 127 for $\sigma^2 = \frac{2}{3}, \frac{n}{4}, \frac{n}{2}, \frac{3n}{4}$ and n. For each pair of parameters, the experiment is performed for eight instances of circulant NTRU to find $\hat{q}$.

For each n, a line is fitted for $(\sigma^2, \hat{q})$ pairs. The resulting slopes, together with the coefficients of determination, can be found in Table 3.1. It is evident that the linear hypothesis suits the data well. Using the determined slopes, we can find c in $k(n)$. We find that $c = 0.00729$ with a coefficient of determination of $R^2 = 0.994$. The slopes and best fit line can be found in Figure 3.2b. As an illustrative example, the predicted slope for the linear approximation together with the asymptotic curve for $n = 127$ can be found in Figure 3.2a.

3.1.2 A Deeper Investigation Into Asymptotics

As both Q and S grow, it becomes infeasible to conduct experiments to reveal the behaviour of the lattice problems for larger σ. Instead, we can analyse the asymptotic complexity of performing a successful attack. This analysis is made more informative by considering the asymptotic cost $B(S, Q)$ in addition to the fatigue point itself.

To this end, we plot $B(S, Q) = \min\{B_{\text{DSD}}(S, Q), B_{\text{SKR}}(S, Q)\}$ as the coloured region in Figure 3.3. The general trend is that B increases with S and decreases with Q. Note that the red line is only defined where $B(S, Q)$ is defined. There are multiple interesting features of Figure 3.3.

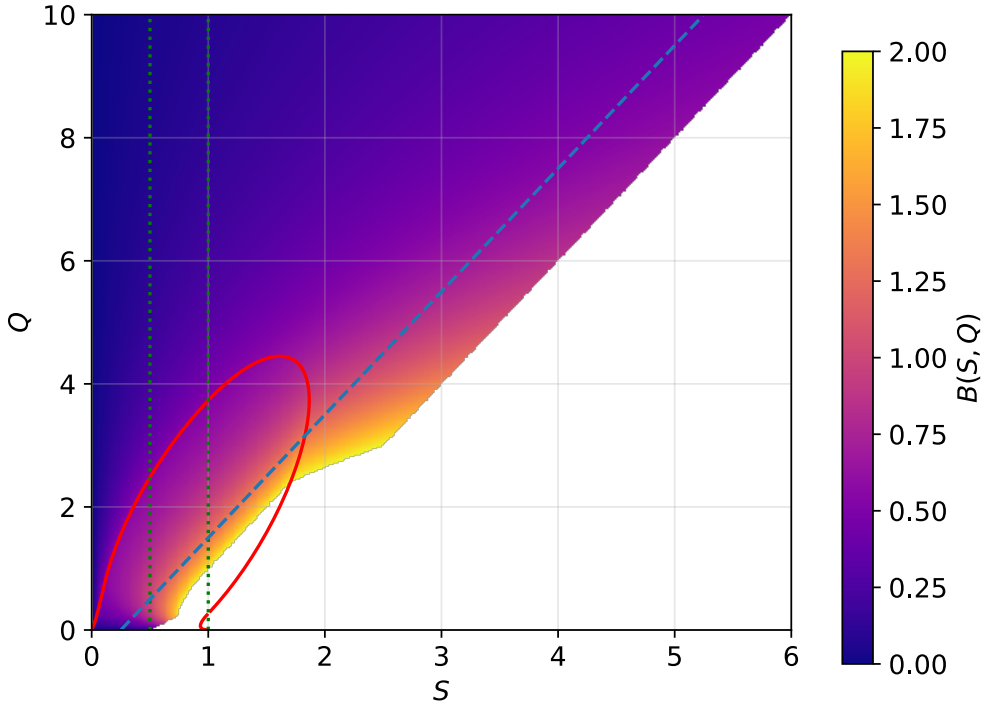


Figure 3.3: $B(S, Q)$ depicted as a heat map with the fatigue points as a red line. The green dotted lines show $S = 1/2$ and $S = 1$ respectively, while the blue dashed line illustrates the correctness limit.

1. First consider the line $S = 1/2$, which is the one for classical NTRU with ternary coefficients. In the SKR regime, $B \approx 1$, while it quickly falls off towards zero for larger Q. What we see here is a prediction of the behaviour observed experimentally, that the problem becomes easier because it suffices to run the SVP solver on smaller blocks to find a vector from the dense sublattice.

2. Next, B is not constant along the fatigue line. For $S \leq 1$, we have $B \approx \frac{1}{2}$ which means we should expect the hardness as S and Q increases to stay the same. From the plateau and onwards however, the required block size to conduct SKR and DSD gradually increases up to 2. We should therefore expect that even at the fatigue point, selecting $S \approx 1.75$ and $Q \approx 4$ should yield a problem with hardness similar to the SKR regime.

3. The SKR regime is described by a bounded domain. This suggests the impact of two effects. Selecting larger q for a given σ makes the sublattice denser, creating an easier lattice problem and higher probability of DSD. Selecting larger σ for a given q makes the short vectors longer and more similar to uniform, which makes SKR less probable. The shape of the domain suggests there is interplay between these two effects.

4. If we accept DSD as a possibility, there still might be a way to construct hard instances of NTRU. Looking outside the SKR regime, we see that $B \approx 2$ along

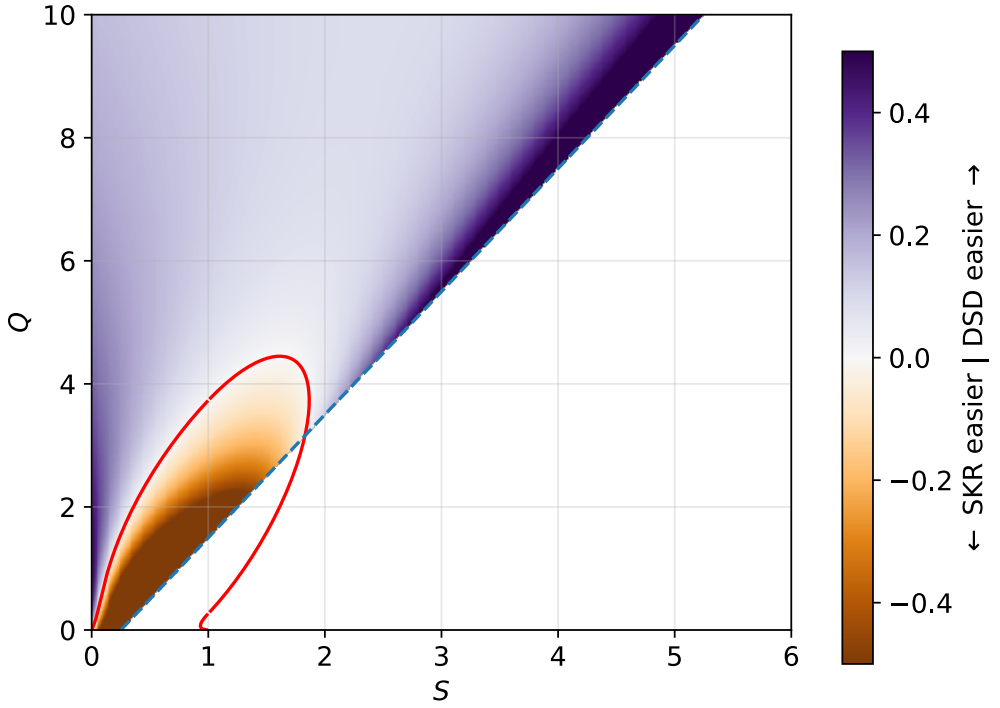


Figure 3.4: The difference in asymptotic block size $B_{\mathrm{SKR}}(S, Q) - B_{\mathrm{DSD}}(S, Q)$. A positive value (blue shading) means DSD is easier, while a negative value (orange) means SKR is easier. Plotted are only correct parameter choices, above the dashed blue correctness line.

the edge of the domain. By selecting σ^2 large ($S \geq 2$), we should be able to construct instances that are at least as hard the SKR regime. In general Q should be selected as small as possible, albeit there is a lower limit due to correctness of the cryptosystem in question. This question will be further investigated in the next section.

We can draw further conclusions by also considering the difference $B_{\mathrm{SKR}}(S, Q) - B_{\mathrm{DSD}}(S, Q)$, which can be found in Figure 3.4. We only plot values above the correctness line, which is derived in the next section. For most choices, the difference in asymptotic complexity is close to zero, meaning DSD is not significantly easier. Two areas in the plot differ from this trend. The first is inside the SKR region, where DSD is significantly harder. The second is just above the correctness line for large S, where SKR seems significantly harder.

3.2 A Secure Instantiation of an Overstretched Cryptosystem

We noted that $S \geq 2$ may yield secure instances even in the overstretched regime, where DSD is easier. In this section, we derive a correctness criterion for NTTRU

(see Section 2.2.1.1) in the style of Lindner–Peikert (see Lemma 2.2.5). Using this criterion and the asymptotics, we propose a set of parameters and compute key and message sizes.

Let us begin by stating the result as a lemma.

Lemma 3.2.1. *In NTTRU with a discrete Gaussian distribution with variance σ^2 for the polynomial coefficients, the error probability per symbol is bounded from above for any desired $\delta > 0$ provided*

$$q \geq 1 + 6\sigma^2 c\sqrt{4n}\sqrt{\ln(2/\delta)}, \tag{3.3}$$

where c is the constant in Lemma 2.1.7.

Proof. Recall that in order to decrypt correctly, we need all coefficients in (2.2) to be smaller than $(q-1)/2$ in absolute value. This corresponds to

$$||g \cdot r + f' \cdot M||_\infty \leq \frac{q-1}{6}, \tag{3.4}$$

when $g \cdot r + f' \cdot M$ is interpreted in the coefficient embedding. In [15], all coefficients in all polynomials are distributed according to the modular binomial distribution β_2. We instead follow [7] and assume $\chi = \mathcal{D}_{\mathbb{Z},\sigma}$.

Having established the ground work, the goal is to find an upper bound for the per-symbol error probability

$$P\left(||g \cdot r + f' \cdot M||_\infty \geq \frac{q-1}{6}\right) =: \delta,$$

where $g, r, f', M \leftarrow \mathcal{D}^n_{\mathbb{Z},\sigma}$ in the coefficient embedding. To this end, we consider two polynomials $p, p' \leftarrow \mathcal{D}^n_{\mathbb{Z}^n,\sigma}$ and discuss their product distribution. If $p = (p_0, \ldots, p_n)$ and $p' = (p'_0, \ldots, p'_n)$, their product will be

$$p \cdot p' = c = \sum_{k=1}^{2n} c_k x^k, \quad c_k = \sum_{i,j>0, i+j=k} p_i p'_j.$$

We will also denote $c_k = (p \cdot p')_k$ as the kth coefficient of a product. Each c_k is a sum of products of random variables distributed as discrete Gaussians, with at most n terms. As we operate in a polynomial ring $\mathbb{Z}_q[x]/(i(x))$, the product $p \cdot p'$ can be of degree at most $n-1$. This is accomplished (see Appendix A.1.2) by dividing by $i(x)$ and letting the remainder be the result. For $i(x) = x^n \pm 1$, this will mean each coefficient of $p \cdot p'$ will be a sum of exactly n terms and we can bound the probability for any coefficient. To this end, we choose to focus on c_n in order to simplify the analysis.

The above reasoning holds for each product $g \cdot r$ and $f' \cdot M$ and we have a sum of them. Nonetheless, the nth coefficient will be

$$(g \cdot r + f' \cdot M)_n = (g \cdot r)_n + (f' \cdot M)_n = \sum_{k=0}^{n} g_k r_{n-k} + \sum_{k=0}^{n} f'_k M_{n-k}$$

We therefore focus on this coefficient and bound $P(|(g \cdot r + f' \cdot M)_n| \geq \frac{q-1}{6})$. To this end, $(g \cdot r + f' \cdot M)_n$ can be seen as a scalar product between two vectors $\boldsymbol{a} = (g||f')$ and $\boldsymbol{b} = (\text{flip}(r)||\text{flip}(M))$ of length $2n$ where each coefficient is sampled from $\mathcal{D}_{\mathbb{Z},\sigma}$. We would therefore like to apply Lemma 2.1.8. We can consider $\boldsymbol{a}$ as a fixed vector of length at most $||\boldsymbol{a}|| = c\sqrt{2n\sigma^2}$ for $c \geq 1$ bounded by Lemma 2.1.7. Recall we can find c such that the probability of $\boldsymbol{a}$ being longer than $c\sqrt{2n\sigma^2}$ is negligible. Further, we view $\boldsymbol{b} \leftarrow \mathcal{D}_{\mathbb{Z}^{2n},\sigma}$. With this reasoning,

$$P\left(|(g \cdot r + f' \cdot M)_n| \geq \frac{q-1}{6}\right) \leq P\left(|\langle \boldsymbol{a}, \boldsymbol{b}\rangle| \geq \sqrt{2\pi}T\sigma||\boldsymbol{a}||\right) \leq 2\exp(-\pi T^2),$$

for some $T > 0$. We fix T by solving

$$\sqrt{2\pi}T\sigma||\boldsymbol{a}|| = T\sqrt{2\pi}c\sigma^2\sqrt{2n} = \frac{q-1}{6} \implies T = \frac{q-1}{6\sigma^2 c\sqrt{4\pi n}}.$$

Then

$$P\left(|(g \cdot r + f' \cdot M)_n| \geq \frac{q-1}{6}\right) =: \delta \leq 2\exp\left(-\left(\frac{q-1}{6\sigma^2 c\sqrt{4n}}\right)^2\right),$$

for the per-symbol error probability δ. Solving for q yields the desired inequality.

□

Example 3.1. Let $n = 127$. The bound in Lemma 2.1.7 should be picked in order to keep the probability of choosing a faulty encryption vector small while simultaneously keeping c small. As a compromise, we pick 2^{-32} and compute $C \approx 0.84$ and $c \approx 1.44$. Now, using $\delta = 2^{-40}$ (as in [13]) and $S = 2$ yields $\sigma^2 = n^{2S-1} = 2\,048\,383$, which means

$$q \geq 1 + 6 \cdot 2\,048\,383 \cdot 1.44 \cdot \sqrt{4 \cdot 127} \cdot \sqrt{-\ln(2^{-41})} \approx 2.12 \times 10^9 \approx 2^{30},$$

which is $Q = \ln(q)/\ln(n) \approx 4.4$. •

Using (3.3), we investigate the asymptotics of the correctness condition. Let $q = O(n^Q)$ and $\sigma^2 = O(n^{2S-1})$, then take logarithms on both sides of the equation. We get

$$Q\ln n \geq \left(2S - 1 + \frac{1}{2} + o(1)\right)\ln n \implies Q \geq 2S - \frac{1}{2}.$$

For $S = 2$ we get $Q \geq 3.5$. The asymptotic block size needed to perform a successful attack along the correctness limit can be found in Figure 3.5. The abrupt change in slope before $S = 2$ marks the change from SKR to DSD. In terms of block size, the most secure instantiations of this cryptosystem are for $1 \leq S \lessapprox 1.75$.

With this parameter set and the asymptotics from Figure 3.3, we make an estimate of the security of a cryptosystem in terms of the security parameter λ. Using $(S, Q) = (2, 4)$, we get $B = 0.94$ (with the DSD attack). We make use of the

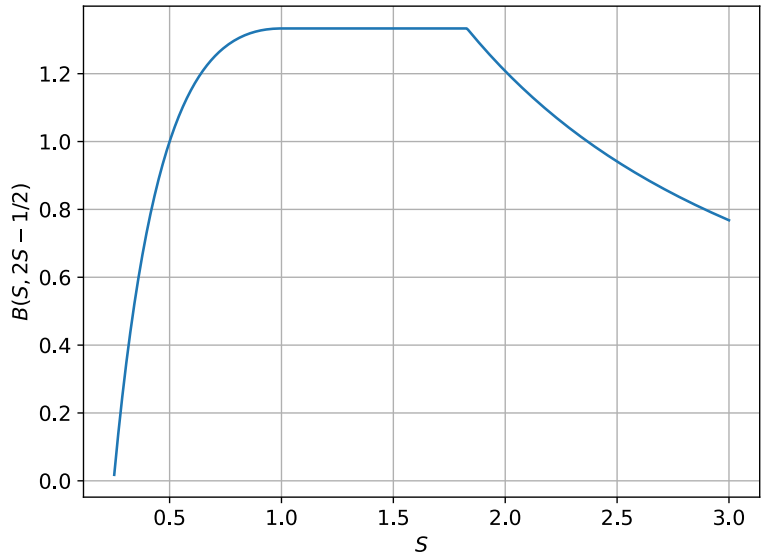


Figure 3.5: The asymptotic block size $B(S, 2S - 1/2)$ along the asymptotic correctness line $Q = 2S - 1/2$ for NTTRU.

classical cost estimate $\lambda = 0.292\beta$ [21], which means $\lambda = 0.292 \cdot 0.94n$. For $n = 127$, we get 34 bits of security. For $\lambda = 128$ bits, we need $n \approx 500$. This seems comparable to other NTRU cryptosystems, for example NTRUEncrypt [20]. If we were to use SKR instead, this would yield $B = 1$, meaning one would believe we have 37 bits and 146 bits of security respectively. This is not a large difference, which stems from the observation that the asymptotic complexity does not differ significantly between DSD and SKR for these parameters (see Figure 3.4).

For practical applications, performance and storage requirements are important. We show that this instantiation does not incur too large keys and ciphertexts. According to the description in Section 2.2.1.1, the private key is (f', g), the public key h and ciphertext c. Storing a polynomial in coefficient representation means storing n integers in $[0, q]$ with a size of

$$\frac{n \log_2(q)}{8} \text{ bytes.}$$

With our instantiation for 128 bits of security, each polynomial takes up 1850 bytes. Thus, public keys and ciphertexts are 1850 bytes each. This is the same order of magnitude as Kyber [22].

3.3 Predictions Using the Estimator

In this section, we use the estimator from [7]. First, we predict the reduction in security for the homomorphic encryption system YASHE by the DSD attack. Second, we compare the methods for predicting the fatigue point for large σ^2 with each other and experiments.

3.3.1 Reduction in Security by DSD in YASHE

In [23], the authors propose a homomorphic encryption scheme based on NTRU called YASHE. The recommendation is to use a very large q, which we today know will yield an overstretched system. We analyse how much security is lost due to DSD by using the estimator described above.

The authors propose multiple parameter sets with a security of $\lambda = 80$ bits. The two smallest instantiations can be found in Table 3.2. They use ternary coefficients, which we model as a discrete Gaussian with $\sigma^2 = \frac{2}{3}$. We compute the block sizes required for a successful attack based on the 2016 estimate [19] and DSD estimator [7], both of which are implemented into `https://github.com/malb/lattice-estimator`. For the two smallest instantiations of YASHE, we expect a large reduction in block size.

Table 3.2: Parameter sets and estimated security when DSD is taken into account for YASHE.

n	q	β_{SKR}	β_{DSD}
1641	2^{64}	175	54
3329	2^{128}	180	43

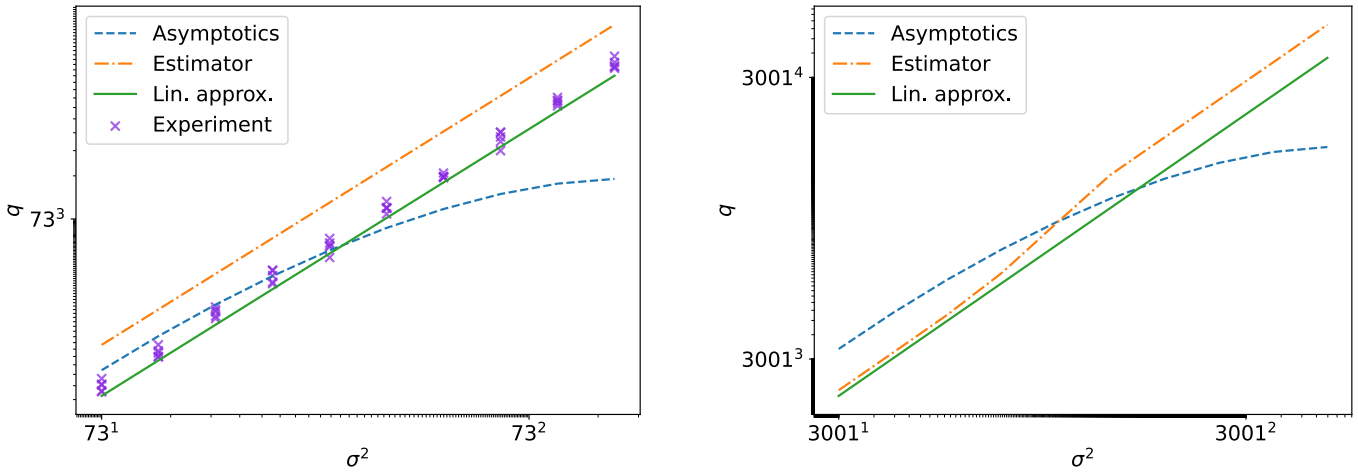
A natural question to ask is whether security can be increased again by increasing n. Experiments with the estimator show that a doubling of n will restore the security level, even though the system is still overstretched.

In comparison, the asymptotics predict $\beta_{\mathrm{SKR}} = 466$ and $\beta_{\mathrm{DSD}} = 177$, significantly overestimating security. This can be plausibly explained by considering the validity of the heuristics used in the asymptotics. For example, the GSA is known to be valid only for $\beta > 50$ and the error of using the GSA for smaller β can be significant as pointed out in [16]. The estimators incorporate experimental slopes for smaller β which lower the estimated block size.

3.3.2 Fatigue Point for Large σ^2 Using the Estimator

At this point, we have four tools available to compute the fatigue point: experiments, asymptotics, the estimator and our linear approximation. In this subsection, we compare these four methods of characterising the fatigue point for larger σ^2. As noted above, there is a plateau in the asymptotics at $S \approx 1.6$. We therefore focus $1 \leq S \leq 1.6$ to see if this plateau also appears for the other methods. The results are found in Figure 3.6.

First, consider Figure 3.6a, which shows all four methods for $n = 73$. As we have seen in the asymptotics, complexity increases along the fatigue line. This n was selected to allow for performing experiments for large σ^2. Ten logarithmically distributed values of σ^2 between $1 \leq S \leq 1.6$ were selected, and the soft binary search program was run for five instances each. The results are depicted as purple crosses in the figure. Comparing to the predictions, we see best agreement between



(a) $n = 73$.

(b) $n = 3001$.

Figure 3.6: Fatigue points computed in different ways. The dashed blue line shows asymptotics, the dash-dotted orange line the estimator, the solid green the linear approximation and purple crosses experiments.

the linear approximation and experiments, even though the linear approximation was developed for $S \leq 1$. The estimator seems to increase linearly with the same slope and a constant offset from experiments. We clearly see the plateau in the asymptotics, and the experiments do not agree with this.

These results may be plausibly explained by the observation, as mentioned before, that the heuristics are not valid for small β. The asymptotics are therefore not expected to be valid here, and the existence of a plateau for larger β cannot be excluded.

Second, to see how the estimator scales when n is increased, the asymptotics, estimator and linear approximation are compared in Figure 3.6b for $n = 3001$. No experiment is performed due the complexity of the problem. We see a clear agreement between the linear approximation and estimator for smaller σ^2, however they diverge slightly when σ^2 increases. The asymptotics initially agree in slope with the other methods and diverges to the plateau for larger σ^2.

As n is large, we expect β to increase as well. The estimator predicts $\beta \approx 1600$ for $n = 3001$. The heuristics used hold for $2n \gg \beta$ [16], which is not the case here. Again, we therefore cannot expect the asymptotics to hold and a plateau cannot be excluded.

3.4 Summary and Discussion

In summary, the goal of this chapter was to investigate the dependence between the variance σ^2 of the coefficient distribution and the fatigue point $\hat{q}$. We found that there definitely is a dependence as $\hat{q}$ increases with σ^2.

For relatively small σ^2, the dependence is approximated as linear and a model is proposed. After fitting a constant, the model seems to match data well. It is also used for larger σ^2 for $n = 73$ and 3001. The model matches experiments and predictions from the estimator.

The asymptotics described in theory predict the SKR region is bounded, which is not observed in the estimator and experiments. This is plausibly explained by the conditions for validity of asymptotics not being valid for these parameter choices.

We investigate the relation between SKR and DSD further by deriving a correctness condition and proposing an asymptotically secure instantiation. The asymptotics are used to compare the security from SKR and DSD.

Taken together, it is without doubt that the choice of σ^2 affects $\hat{q}$. Nevertheless, the question of boundedness of the SKR region remains unresolved. The performed experiments show linear dependence, while we have shown the asymptotics are not expected to hold for these instantiations. Simultaneously, there is nothing to suggest the asymptotics will not hold once these conditions are satisfied. A final conclusion about the asymptotics can be drawn if experiments are performed for an instantiation where the heuristics are expected to hold. To find such parametrisations, the estimator might be useful.

It should be pointed out that the set of parameters affected by this question may not be cryptographically relevant anyway. If the linear model were true, meaning

$$\hat{q} = c\sigma^2 n^{2.484},$$

we would asymptotically have $\hat{Q} = \frac{\ln c}{\ln n} + 2S - 1 + 2.484 = 2S + O(1)$. In the limit $n \to \infty$, the $O(1)$ constant is 1.484. As such, the cryptographically relevant SKR region would be everything below this line but above the correctness line. This is a small sliver where practical instantiations of NTRU cryptosystems do not exist today. In conclusion, while the question of boundedness of the SKR region is theoretically interesting, it does not affect most security estimates.

4

Overstretched Parameters in LWE

This chapter concerns research questions 3 and 4. The main goal is to derive asymptotics for SKR and DSD for the MLWE lattice problem. RLWE is a special case of MLWE and is therefore implicitly included. We assume both attacks can be modelled in the same way as for NTRU, hence the calculations are very similar to Section 2.4. As we find DSD is not expected for MLWE, no experiments are performed.

The chapter has the following structure.

1. In Section 4.1, we derive an estimate for the asymptotic cost of SKR in MLWE, assuming the last $n-1$ dimensions in the lattice can be ignored.
2. Next, in Section 4.2, we propose a ZGSA-type heuristic for the MLWE basis after reduction. It is then used to find an estimate for DSD in Section 4.3.
3. Finally we analyse the cost estimates in search of a fatigue point in Section 4.4.

4.1 SKR Asymptotics for Module-LWE

In this subsection, we use the 2016 estimate (2.8) to derive an estimate for the asymptotic block size needed to perform successful SKR in MLWE, in a similar vein to (2.9). We only provide an exposition for MLWE as RLWE can be retrieved by setting $d = 1$. The analysis is performed in $O(nd)$ asymptotics to keep the security parameter, which is dependent on nd, constant when d changes.

Claim 4.1.1 (MLWE-SKR asymptotics)**.** *The BKZ algorithm with $\beta = Bnd$ applied to an MLWE instantiation with $q = \Theta((nd)^Q)$ and $||(e,s,t)|| = O((nd)^S)$ recovers (a shift of) the secret key if*

$$B = \frac{2Q}{(Q-S+1)^2} + o(1). \tag{4.1}$$

Justification. We begin justifying this claim by noting that the lattice basis (2.3) contains redundant information. The bottom block in $\boldsymbol{C}_M$, the secret $\begin{bmatrix}\_ & \boldsymbol{S} & \boldsymbol{I}_n\end{bmatrix}^T$ and the dense sublattice $\begin{bmatrix}\boldsymbol{E} & \boldsymbol{S} & t\boldsymbol{I}_n\end{bmatrix}^T$ all contain an identity matrix that contains no extra information about the secret key. Thus, the last $n-1$ columns may be removed to reduce the lattice dimension to $mn + dn + 1$. Consequently the last

$n-1$ rows in the basis matrix will be zeroes and may be removed as well. This will still keep a copy of the secret vector, as

$$\underbrace{\begin{bmatrix} q\boldsymbol{I}_{mn} & -\boldsymbol{A} & \boldsymbol{b} \\ 0 & \boldsymbol{I}_{dn} & 0 \\ 0 & 0 & t \end{bmatrix}}_{\boldsymbol{C}'_M} \begin{bmatrix} \_ \\ \boldsymbol{S} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{s} \\ t \end{bmatrix},$$

where the right hand side should be viewed as keeping one row of the resulting matrices from polynomial multiplication in $-\boldsymbol{AS}$. Using $\boldsymbol{C}'_M$ as lattice basis, we again apply the minimisation procedure that was used to justify Claim 2.4.3 to decide the number k of q-vectors to keep of the mn available. The right hand side in the 2016 estimate (2.8) is

$$\gamma^{2\beta-\nu-1}\mathrm{vol}(\mathcal{L})^{1/\nu},$$

where $\gamma = \mathrm{gh}(\beta)^{1/(\beta-1)}$. We set $\nu = dn+k+1$ and $\mathrm{vol}(\mathcal{L}) = q^k t$ with $t=1$. The task is therefore to minimise

$$\gamma^{2\beta-dn-k-2}q^{k/(dn+k+1)}.$$

We obtain the minimum by differentiating with respect to k and solving the equation, substituting our asymptotic assumptions in the process. The solution is

$$k = \sqrt{2BQdn(dn+1)} - dn - 1 \approx dn(\sqrt{2BQ}-1).$$

We note this expression can always be used as no upper limit exists for k. If it existed, it would be mn. However, LWE allows us to draw more samples if needed, so m can be as large as needed to ensure $k \leq mn$.

We now solve for equality in (2.8) using the projected lattice of dimension $dn+k+1 \approx dn\sqrt{2BQ}$. This means solving

$$\sqrt{\frac{Bnd}{nd\sqrt{2BQ}}}\Theta((nd)^S) = \mathrm{gh}(\beta)^{\frac{2Bnd-nd\sqrt{2BQ}-1}{Bnd-1}}O((nd)^Q)^{\frac{dn(\sqrt{2BQ}-1)}{dn\sqrt{2BQ}}}.$$

We take logarithms, use $\ln(\mathrm{gh}(\beta)) = \frac{1}{2}\ln(nd) + o(1)$ and cancel small terms. Solving for B gives the desired result. △

We note here this is the same expression as in (2.9) for $S \leq 1$. This seems reasonable as the structure of the NTRU lattice is similar to the reduced MLWE lattice, apart from their dimensions. Intuitively, one would therefore expect a similar expression.

Another remark is that the analysis may equivalently be performed assuming the parameters are $O(n)$ instead of $O(nd)$. This will lead to the expression

$$B = \frac{2Qd}{(Q-S+1)^2} + o(1), \tag{4.2}$$

as the complexity scales with module rank d.

4.2 An Extended ZGSA for Module-LWE

This section concerns the basis profile after reduction for the MLWE lattice (2.3), for which the ZGSA cannot be applied directly due to dimensional mismatch. Thus it is modified to match the structure of the basis. We call it the extended ZGSA.

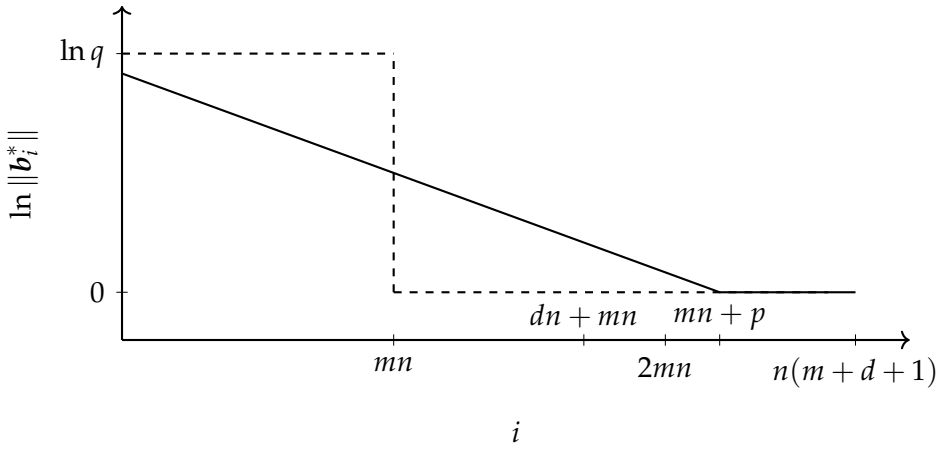


Figure 4.1: An illustration of the extended ZGSA in (4.3). The dashed line shows the basis profile before reduction, the solid line after. This illustrates a situation where $d < m$ and $p > mn$.

Heuristic 4.2.1 (ZGSA for MLWE)**.** *Let $\boldsymbol{C}_M$ be a basis of an $n(m+d+1)$-dimensional q-ary MLWE lattice $\mathcal{L}$ with mn q-vectors. After BKZ-β reduction the basis shape is*

$$||\boldsymbol{b}_i^*|| = \begin{cases} q, & i \leq mn - p, \\ \sqrt{q}\alpha_\beta^{(2mn-1-2i)/2}, & mn - p < i < mn + p - 1, \\ 1, & i \geq mn + p - 1, \end{cases} \tag{4.3}$$

where $p = \frac{\ln q}{2\ln(\alpha_\beta)} + \frac{1}{2}$.

Justification. The starting basis has shape

$$||\boldsymbol{b}_i^*|| = \begin{cases} q, & i \leq mn, \\ 1, & i \geq mn + 1, \end{cases}$$

illustrated by the dashed line in Figure 4.1. The basis is $n(m+d+1)$-dimensional, so we should expect an asymmetric shape for the basis after reduction. Intuitively, the reduction slope will pivot around the point mn. We model this as a symmetric shape around mn, from 0 to $2mn$. Note that $2mn$ may be greater than $dn + mn$ (we could even have $2mn > n(d + m + 1)$) as the solid line in Figure 4.1. Furthermore, as there are n extra vectors beyond $dn + mn$, we may also let $p > mn$. These behaviours are captured in (4.3). △

4.3 DSD Asymptotics for Module-LWE

Using the extended ZGSA, we model DSD for MLWE in the same way as [7] and find an asymptotic estimate.

Claim 4.3.1. *The BKZ algorithm with $\beta = Bnd$ applied to MLWE with module rank d with $q = \Theta((nd)^Q)$ and $||(\boldsymbol{e}, \boldsymbol{s}, t)|| = O((nd)^S)$ results in DSD if*

$$B = 2\frac{\sqrt{4S^2 + 4QSd - d^2} + dQ + 2S}{d(Q^2+1)},$$

assuming the extended ZGSA (4.3).

Justification. Let $\mathcal{L}^{\boldsymbol{S}}$ be the secret dense sublattice spanned by the basis $\begin{bmatrix}\boldsymbol{E} & \boldsymbol{S} & t\boldsymbol{I}_n\end{bmatrix}^T$ and $\mathcal{L} = \mathcal{L}(\boldsymbol{C}_M)$ the full MLWE basis.

The DSD condition translated to MLWE is

$$\pi_{mn+dn+k-\beta}(\boldsymbol{v}) \leq ||\boldsymbol{b}^*_{mn+dn+k-\beta}||,$$

where $\boldsymbol{v}$ is a short vector in $\mathcal{L}_{[0:mn+dn+k]}$. Denote $\mathcal{L}_{[0:mn+dn+k]} \cap \mathcal{L}^{\boldsymbol{S}} = \mathcal{L}^{\cap}$. The choice of index is to ensure the intersection lattice $\mathcal{L}^{\cap}$ has non-zero dimension k when using the generalised Pataki-Tural lemma (Lemma 2.4.6) with $\dim(\mathcal{L}^{\cap}) = k$. Applying this lemma, we have

$$\mathrm{vol}(\mathcal{L}^{\cap}) \leq \mathrm{vol}(\mathcal{L}^{\boldsymbol{S}}) \left(\prod_{i=mn+dn+k}^{mn+dn+n} ||\boldsymbol{b}^*_i||\right)^{-1}.$$

As an index set, we can always pick the last $n-k$ vectors as the basis shape is monotonously decreasing in all cases of the extended ZGSA. Assuming $||(\boldsymbol{e}, \boldsymbol{s}, t)|| = O((nd)^S)$, the Hadamard bound (A.1) gives $\ln \mathrm{vol}(\mathcal{L}^{\boldsymbol{S}}) \leq Sn\ln(nd)$. Applying (4.3) to the intersected lattice, we get

$$\ln \mathrm{vol}(\mathcal{L}^{\cap}) \leq Sn\ln(nd) - \sum_{i=mn+dn+k}^{mn+p-1} \ln ||\boldsymbol{b}^*_i||. \tag{4.4}$$

The only interesting case will be when some part of the sum is covered by the sloped part of the ZGSA, meaning that $dn + k \leq p - 1$. Assuming that $\beta = Bnd$, $q = O((nd)^Q)$ and thus $\ln \alpha_\beta = \frac{\ln(nd)}{Bnd} + O((nd)^{-1})$, the Gram–Schmidt vectors will have length

$$\ln ||\boldsymbol{b}^*_i|| = \frac{Q}{2}\ln(nd) + \frac{2mn - 2i - 1}{2Bnd}\ln(nd).$$

Therefore, assuming $k = Knd$ and getting $p = \frac{1}{2}QBnd$ the volume is

$$\begin{aligned}\ln \mathrm{vol}(\mathcal{L}^{\cap}) &\leq Sn\ln(nd) - \frac{1}{2}\ln(nd)\sum_{i=n(m+d+Kd)}^{n(m+\frac{1}{2}BQd)-1}\left(Q + \frac{2mn-2i-1}{Bnd}\right)\\ &\leq Sn\ln(nd) - \frac{1}{2}\ln(nd)\left(\frac{dn(BQ-2K-2)^2}{4B}\right)\\ &\leq n\ln(nd)\left(S - d\frac{(BQ-2K-2)^2}{8B}\right).\end{aligned}$$

Here, we compute the arithmetic sum and simplify.

Next, using Minkowski's bound (2.1.4), we have that

$$\begin{aligned}\ln(\lambda_1(\mathcal{L}^{\cap})) &\leq \frac{1}{2}\ln(Knd) + \frac{1}{Knd}\left(S - d\frac{(BQ-2K-2)^2}{8B}\right)n\ln(nd)\\ &\leq \ln(nd)\left(\frac{1}{2} + \frac{S}{Kd} - \frac{(BQ-2K-2)^2}{8BK}\right).\end{aligned}$$

Provided an estimate for the right hand side of the DSD condition, we estimate the left hand side using the extended ZGSA to get

$$\begin{aligned}||\boldsymbol{b}^*_{mn+dn+k-\beta}|| &\leq \frac{Q}{2}\ln(nd) + \frac{2mn-1-2(mn+dn+k-\beta)}{2}\frac{\ln(nd)}{Bnd-1}\\ &\leq \ln(nd)\left(\frac{Q}{2} + \frac{2Bnd-2nd-2Knd-1}{2Bnd-2}\right)\\ &\leq \ln(nd)\left(\frac{Q}{2} + \frac{B-K-1}{B}\right).\end{aligned}$$

Now, solving for the inequality for B yields

$$B \geq 2\frac{\sqrt{d^2K^2Q^2 + d^2K^2 - 2d^2KQ - 4dKS + 4dQS + 4S^2} - dK + dQ + 2S}{dQ^2}.$$

This is minimised when

$$K = \frac{\sqrt{4S^2 + 4QSd - d^2} + dQ + 2S}{d(Q^2+1)}.$$

Substituting this into the expression for B yields the desired result. △
We note the expression will be multiplied by d if the analysis is made in $O(n)$.

Remark. *It is also conceivable that* $||\boldsymbol{b}^*_{mn+dn+k-\beta}|| = 1$*, but this would imply the sum in* (4.4) *is* 0 *as all vectors in a strictly larger index must have length 1 as well. This yields the equation* $\frac{1}{2} + \frac{S}{K} \leq 0$ *which has no meaningful solutions. As such, the above expression for B is the only solution to a DSD attack modelled this way.*

4.4 Do Ring-LWE and Module-LWE Suffer from Overstretching?

We reap the rewards from the above analysis to determine whether RLWE and MLWE suffer from overstretching. First, we discuss the case of standard RLWE with $S = 1/2$ and then move on to general MLWE with arbitrary S.

For ring-LWE, we set $d = 1$ in all expressions for MLWE. Thus, for the common choice of $\sigma^2 \in O(1)$, we have $S = 1/2$ and solve

$$\frac{2Q}{(Q - S + 1)^2} = \frac{2(\sqrt{4S^2 + 4QS - 1} + Q + 2S)}{Q^2 + 1}, \quad S = 1/2.$$

This has no real solutions. Another way to see this is the lack of an equality point at $S = 1/2$ in Figure 4.2, in the upper left plot. In other words, the SKR attack is always easier than DSD, and no fatigue point is expected for RLWE.

Using the full expressions for B for DSD and SKR, we define $B(S, Q, d)$ as $\min\{B_{\mathrm{SKR}}(S, Q, d), B_{\mathrm{DSD}}(S, Q, d)\}$ and compute B for $0 \leq S, Q \leq 10$ and $d = 1, 2, 5$ and 10. The chosen module ranks reflect those used in standardised cryptosystems. The result can be found in Figure 4.2. Interestingly, the DSD attack is easier inside a cone, meaning selecting Q inside the cone for a given S yields DSD.

However, correctness should also be considered. Having made the asymptotic analysis in terms of nd allows us to consider the MLWE samples as vectors in $\mathbb{Z}^{2nd}$ as done in [22]. This means the Lindner–Peikert correctness theorem (Lemma 2.2.5) can be applied directly to these vectors. We therefore consider $O((nd)^S) = ||(\boldsymbol{e}, \boldsymbol{s}, 1)|| \approx \sqrt{2nd\sigma^2}$ giving us $\sigma = O((nd)^{S-1/2})$. Using this in Lindner–Peikert and cancelling small terms gives us that an MLWE instantiation is correct if

$$Q \ln(nd) \geq (2S - 1/2) \ln(nd).$$

This is plotted as the blue lines in Figure 4.2. The region where DSD is allowed lies below the correctness line for $d = 1$, though it grows to allow for DSD for parameter choices inside the cone and above the correctness line.

Taking correctness into account means the entire DSD region is excluded for RLWE, irrespective of σ^2. Thus, correct cryptosystems using RLWE only suffer from SKR, even when S increases. Considering the increase in module rank d, we can observe a few phenomena. First, the opening angle of the "cone" seems to increase with d while the overall required block size stays the same or decreases slightly. In other words, the decrease in n compensates for the increase in d to keep the attack complexity constant.

The possibility of DSD is a somewhat surprising result. For a fixed nd, an increase in d will lead to a decrease in dimension of the dense sublattice (which has dimension n). Consequently, discovering a vector in $\mathcal{L}^S$ becomes more and more similar to discovering the secret key itself. We should therefore expect more SKR and less DSD. An equivalent statement is that the SKR and DSD attack will become more

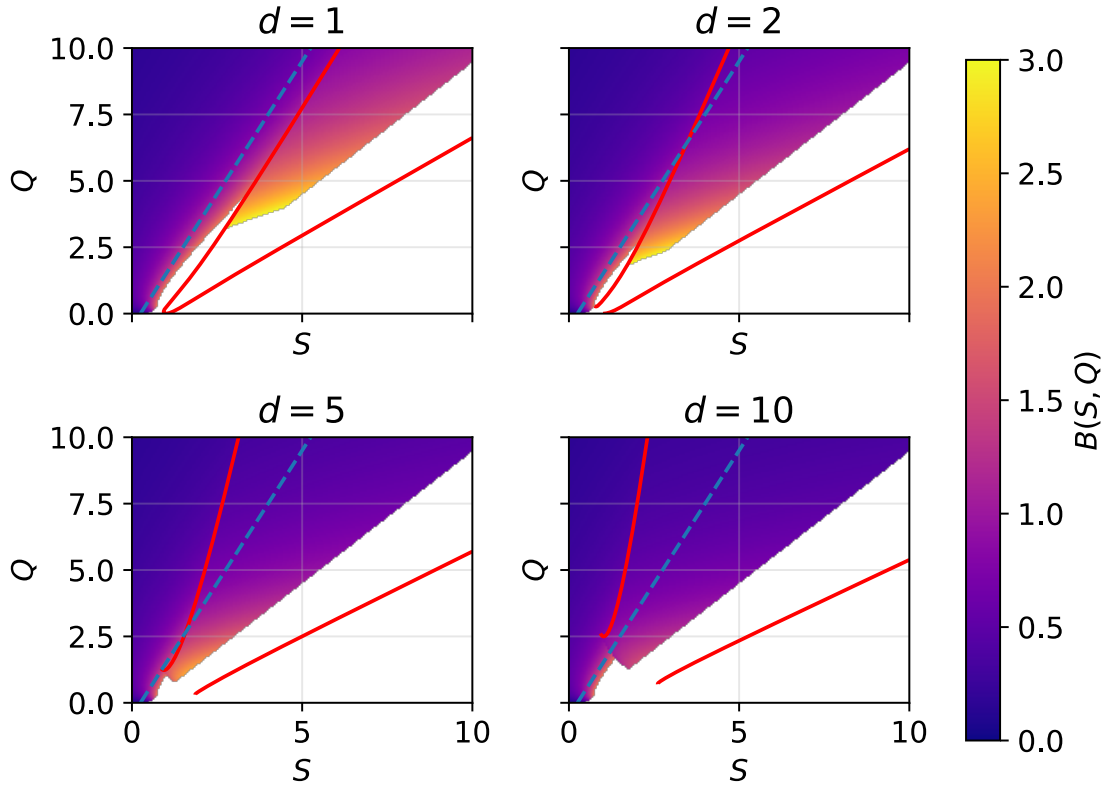


Figure 4.2: The shaded regions show $B(S, Q)$ for MLWE. The red lines are the solutions $B_{\text{SKR}}(S, Q, d) = B_{\text{DSD}}(S, Q, d)$ and the blue dashed lines the correctness limit $Q = 2S - 1/2$. Note the maximal allowed B is 3 for DSD and 2 for SKR. The uniform limit $Q \geq S - 1/2$ also limits the plot, together with the singularity line $Q - S + 1 = 0$.

similar and should therefore have equal complexity. In Figure 4.3, the difference between the asymptotic hardness of SKR and DSD are plotted where applicable. A negative difference means SKR is easier, and a positive difference means DSD is easier. Evidently, as d increases, the difference becomes smaller. We can conclude that while DSD is asymptotically easier for appropriate parameter choices, the difference between SKR and DSD will be small in those regions. For those choices we should therefore expect BKZ recovers vectors very similar to the secret key, that are only slightly larger in norm, as the attacks become more similar.

4.5 Summary and Discussion

Summarising the above results, we first proceeded by observing that SKR can be successful in a lattice containing only one copy of the secret vector and derived an estimate for the complexity. The ZGSA was extended to the module-LWE lattice. After establishing a model for a DSD attack on MLWE, an estimate was derived for DSD under the extended ZGSA.

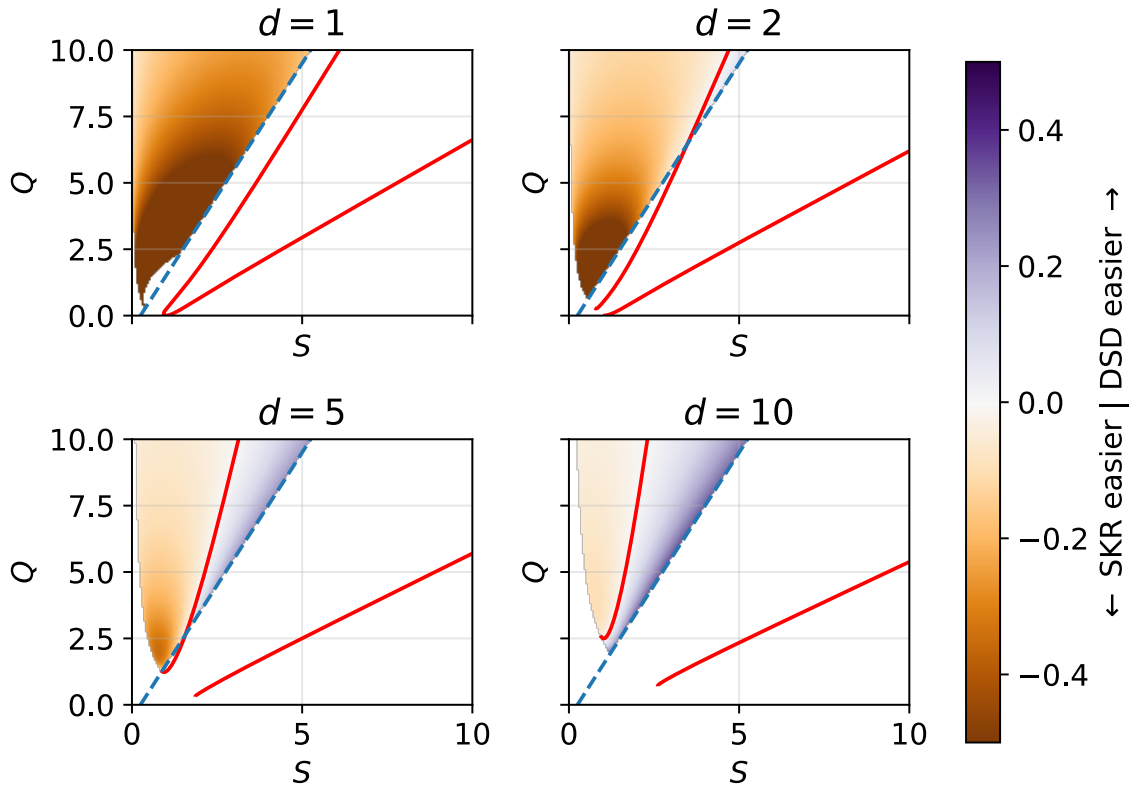


Figure 4.3: $B_{SKR}(S, Q) - B_{DSD}(S, Q)$ for MLWE above the correctness line where nd is kept constant for increasing d. The coloured regions are where the asymptotics for both attacks are defined and a correct cryptosystem can be obtained.

Searching for a fatigue point for common sizes of error by setting $S = 1/2$ yields no results in any module rank d. When d increases however, the asymptotics point to a region where DSD is easier. Investigating the difference in complexity between SKR and DSD shows that it is not significant in the DSD region. This is plausibly explained by the observation what SKR and DSD become more similar as d increases.

These results indicate that while a fatigue point may exist in a technical sense, it is not expected to affect security in the same way as for NTRU. It is unexpected that the fatigue point may appear when selecting Q *small* for a given S, contrary to the situation for NTRU. The lack of a clear fatigue point for common choices of S agrees with the literature. In fact, the author cannot find any works concerning overstretched LWE, indicating that only negative results have been found, if any. It is nevertheless satisfactory that the DSD model produces results in agreement with (the lack of) literature.

5

Overstretched Parameters in NTWE

This chapter concerns research question 5. The goal is to derive asymptotics for SKR and DSD for the NTWE problem. It has a structure similar to MLWE which means many components of the analysis can be reused. For uniformity, we roughly follow the same structure as for MLWE.

The chapter is structured as follows.

1. We derive the asymptotics for NTWE in Section 5.1. The DSD attack, by way of the ZGSA being identical, has the same asymptotic complexity as MLWE, while SKR is re-derived.
2. The asymptotics are investigated in Section 5.2 where an asymptotic fatigue point is found.
3. In Section 5.3 above predictions are experimentally verified.
4. The chapter concludes with a discussion in Section 5.4.

5.1 Asymptotic Analysis of NTWE

Similar to MLWE, we discuss each component of the asymptotic analysis: the SKR asymptotics, basis profile and DSD asymptotics.

5.1.1 SKR Asymptotics for NTWE

Claim 5.1.1. *The BKZ algorithm with $\beta = Bnd$ applied to an NTWE instantiation with $q = \Theta((nd)^Q)$ and $||(\boldsymbol{e}, \boldsymbol{s}, tf)|| = O((nd)^S)$ recovers (a shift of) the secret key if*

$$B = \frac{2Q(d+1)}{d(Q-S+1)^2} + o(1). \tag{5.1}$$

Justification. To justify this claim, we begin by noting that the trick of removing the last $n-1$ vectors applied to MLWE does not work here, as we have a full multiplication $t\boldsymbol{BF}$ instead of an identity matrix in the last position. SKR is therefore performed in the full lattice in (2.5). As usual, we may exclude q-vectors and need to perform optimisation over the right hand side in the 2016 estimate (2.8).

Similar to MLWE, there is no limit on the number of samples m. We follow the same procedure used to justify Claim 2.4.3. Recall the right hand side

$$\gamma^{2\beta-\nu-1}\mathrm{vol}(\mathcal{L})^{1/\nu},$$

where $\gamma = \mathrm{gh}(\beta)^{1/(\beta-1)}$. Keeping k q-vectors, we have $\mathrm{vol}(\mathcal{L}) = q^k t^n$, $\nu = n + dn + k$ and set $t = 1$. The right hand side becomes

$$\gamma^{2\beta-n-dn-k-1} q^{k/(n+dn+k)}.$$

Differentiating with respect to k, taking logarithms and rearranging, the right hand side is minimised when

$$(d+1)n\ln(q) - \ln(\gamma)(dn+n+k)^2 = 0.$$

Substituting in the asymptotics (using $\ln(\gamma) = \frac{\ln(nd)}{2(Bnd-1)}$) and solving for k gives

$$k = n\sqrt{2BQd(d+1)} - n(d+1).$$

We can now solve the inequality (2.8) for B. We use our minimised right hand side, the asymptotics and take logarithms. Cancelling small terms, we are left with

$$S \leq \frac{2Bd - \sqrt{2BQd(d+1)}}{2Bd} + Q\frac{\sqrt{2BQd(d+1)} - (d+1)}{\sqrt{2BQd(d+1)}}.$$

Solving for B yields the desired result. △

5.1.2 NTWE Follows the Extended ZGSA

Comparing the lattice bases for MLWE (2.3) and NTWE (2.5), the structure is the same. Before reduction, there are mn GS-vectors with norm q followed by $dn + n$ GS-vectors with norm 1. We therefore expect the extended ZGSA (4.3) to hold for the NTWE lattice.

5.1.3 DSD Asymptotics for NTWE

Claim 5.1.2. *The BKZ algorithm with $\beta = Bnd$ applied to NTWE with module rank d with $q = \Theta((nd)^Q)$ and $||(\boldsymbol{e}, \boldsymbol{s}, tf)|| = O((nd)^S)$ results in DSD if*

$$B = 2\frac{\sqrt{4S^2 + 4QSd - d^2} + dQ + 2S}{d(Q^2+1)}.$$

Justification. This is the same expression as for MLWE. The structure of the lattice bases are the same, and we established they will follow the same extended ZGSA after reduction. The dimension of the lattices are also the same, which means the DSD condition will be the same. Assuming DSD can be modelled the same way, the rest of the calculations only involve using the asymptotics and estimating the length of the projection of the short vector. Thus, the end result must be the same. △

5.2 Does NTWE Suffer from Overstretching?

We use above estimates to predict that NTWE will suffer from DSD and compute an asymptotic fatigue point. Before proceeding with this, we discuss how NTWE yields both unstructured LWE and NTRU problems, given the right parameter choices.

Both (unstructured) LWE and NTRU can be recovered from NTWE. LWE is recovered by setting $n = 1$ which removes the algebraic structure. We therefore model LWE-like NTWE instances by letting d increase while keeping dn constant. NTRU is recovered by removing the module structure altogether, i.e. setting $\boldsymbol{s} = 0$ or $\boldsymbol{a} = 0$. Comparing definitions, e in NTWE then plays the role of g in NTRU. We model this by setting $d = 0$ for NTRU-like NTWE instances.

Another observation is that in similarity to MLWE, we can transform between an analysis in terms of nd and n by multiplying both expressions by d. This is useful when setting $d = 0$ to avoid a zero in the denominator. Reassuringly, if we do this and set $d = 0$, we get the equation

$$\frac{2Q}{(Q - S + 1)^2} = \frac{8S}{Q^2 + 1}.$$

These are the asymptotic expressions for NTRU and the equation has one real solution $(S, Q) = (0.5, 2.484)$. This sanity check indicates the analysis is correct.

Now, setting $S = 0.5$, which would be the case when all distributions in NTWE have variance $O(1)$, we can solve $B_{\text{DSD}} = B_{\text{SKR}}$ for the same d as in Chapter 4. The solutions are available in Table 5.1. We see that even though a fatigue point exists for these d it starts at around three orders of magnitude larger than the fatigue point for NTRU and increases with d. Thus an effective countermeasure against DSD is to increase d.

Table 5.1: Fatigue points $\hat{Q}$ for multiple d.

d	$\hat{Q}$
1	5.923
2	9.352
5	19.60
10	36.68

To continue, we find an asymptotic lower bound for correctness using Lemma 2.2.8. As a simplification, we assume that $\psi_{\text{enc}} = \psi_{\text{gen}}$ with standard deviation σ. We also assume f is sampled from the same distribution and that $m = d + 1$ as in [10]. Then $O((nd)^S) = ||(\boldsymbol{e}, \boldsymbol{s}, f)|| \approx \sigma\sqrt{(2d + 2)n}$ which roughly implies $\sigma = O((nd)^{S-1/2})$.

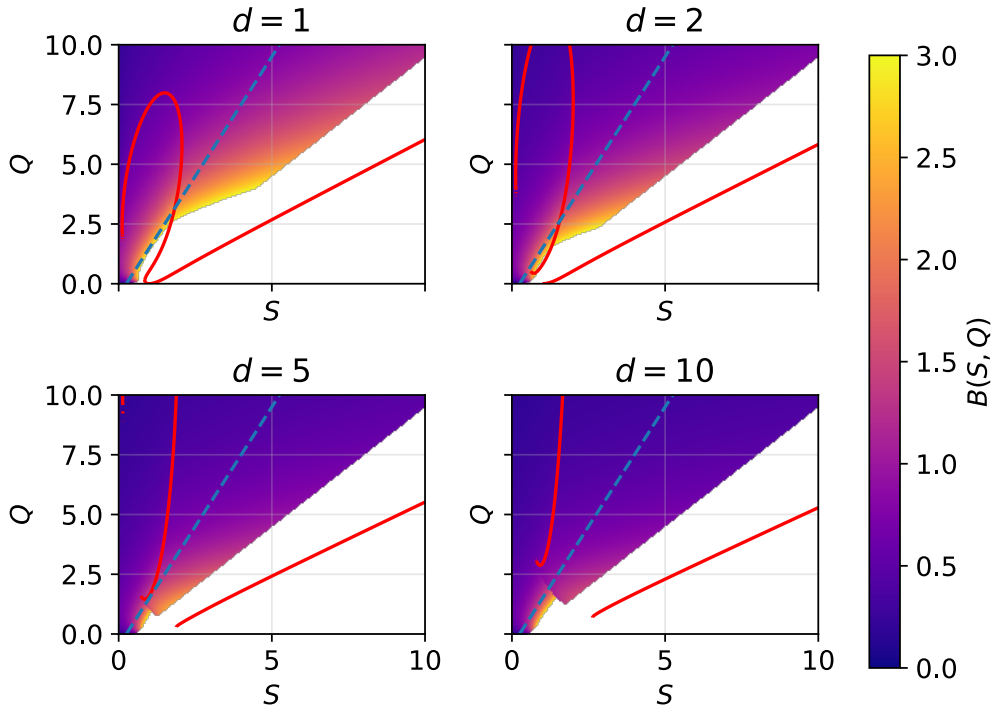


Figure 5.1: $B(S, Q, d)$ for S and Q between 0 and 10, for $0 < B < 3$ and multiple d. Pictured are also the correctness limit $Q = 2S - 1/2$ in dashed blue and solution curves to $B_{\mathrm{SKR}}(S, Q, d) = B_{\mathrm{DSD}}(S, Q, d)$ in red.

Incorporating all assumptions, the correctness condition becomes

$$\begin{aligned} \sigma^2 &\leq \frac{q}{8\sqrt{4(d+1)n\ln(2/\delta)}} \\ &\implies 2(S - 1/2)\ln(nd) \leq (Q - 1/2 + o(1))\ln(nd) \\ &\implies Q \geq 2S - 1/2. \end{aligned}$$

We plot this bound together with $B(S, Q, d) = \min\{B_{\mathrm{SKR}}(S, Q, d), B_{\mathrm{DSD}}(S, Q, d)\}$ for $0 \leq S, Q \leq 10$ and $d = 1, 2, 5$ and 10 in Figure 5.1. The fatigue curve validates above calculations, showing that the fatigue point for $S = 1/2$ lies on the top part on an ellipse which "opens up" as d increases. It appears unclear if this region is a bounded region as in NTRU due to numerical instability in the solutions. Unlike MLWE, there exist correct choices of (S, Q) that are expected to yield DSD.

To deepen the analysis, we also plot the difference $B_{\mathrm{SKR}}(S, Q, d) - B_{\mathrm{DSD}}(S, Q, d)$ in Figure 5.2. We find that for small d, the SKR attack is considerably easier inside the SKR region with a smaller difference as d increases. Outside SKR, the difference is approximately constant and DSD seems significantly easier close to the correctness line. A choice of parameters that yields DSD with significant advantage would therefore be a small Q to the right of the ellipse with S picked accordingly. At the usual choice $S = 1/2$ DSD does not seem to have a significant advantage.

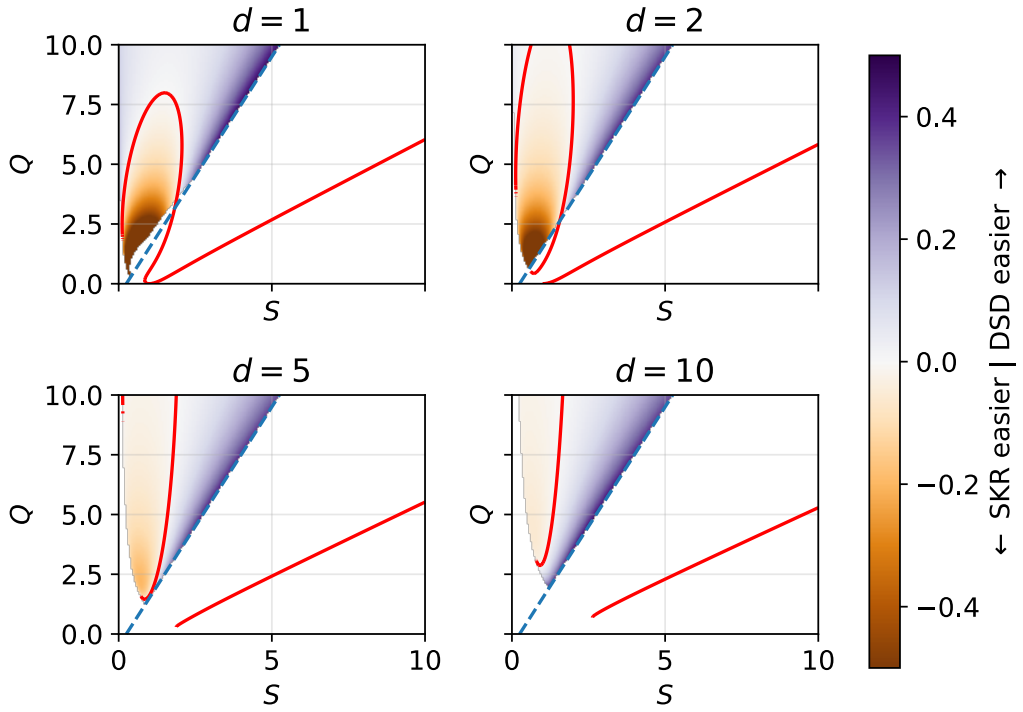


Figure 5.2: $B_{\mathrm{SKR}}(S, Q, d) - B_{\mathrm{DSD}}(S, Q, d)$ for S and Q between 0 and 10, for $0 < B < 3$ and multiple d. Pictured are also the correctness line $Q \geq 2S - 1/2$ in dashed blue and solution curves to $B_{\mathrm{SKR}}(S, Q, d) = B_{\mathrm{DSD}}(S, Q, d)$ in red.

5.3 Finding the Fatigue Point Experimentally

To validate the predicted existence of a fatigue point, experiments were performed. We begin by describing the experimental setup and proceed by presenting results.

Experiments were performed on NTWE instantiations with the parameters in Table 5.2. The ring was selected to allow for more choices of n in the range of computationally feasible block sizes. Module rank, number of samples and variance were selected to yield the easiest possible instantiations. The moduli q were selected based on preliminary experiments to show the transition from the SKR to DSD regimes.

The programs used in [7] was adapted to the NTWE lattice and is available at `https://github.com/zentabit/NTWEFatigue`. The experiments are performed by running progressive BKZ in `fpLLL`. After each call to the SVP subroutine, the projected SVP solution is checked for SKR and DSD using a callback function. If the solution lies in the dense sublattice, BKZ terminates. The relative length of the solution to the dense basis is checked. Any solution that is shorter than or as long as the longest secret vector is classified as SKR, while any other solution is classified as DSD. Each run therefore terminates with SKR, DSD or nothing as output. For each choice of parameters, eight trials were performed.

Table 5.2: Parameter choices for NTWE experiments.

Parameter	Value(s)
R_q	$\mathbb{Z}_q[x]/(x^n - 1)$
n	$97, 101, 103, 107, 109, 113$
d	1
m	1
σ^2	$2/3$
q	Around 100 evenly spaced primes from $0.1 \times 2.74 \times 10^{-6} \times n^{5.92}$ to $1.2 \times 2.74 \times 10^{-6} \times n^{5.92}$

For brevity, we present results for $n = 101$ and 113. The remaining results are available in Appendix B. Consider Figures 5.3 and 5.4. They show the outcome of each trial, together with the average successful β for each q. The colour of each average shows the DSD ratio, the proportion of trials that end with DSD. The general trend is clear: NTWE suffers from DSD as predicted by the asymptotics. The DSD ratio increases with q, clearly showing a transition from an SKR only regime to a DSD only regime. To verify the effect is indeed DSD a histogram over insertion positions κ can be found in Figure 5.6a. The characteristic change in insert position from DSD (c.f. Figure 1 in [7]) can clearly be seen, even though more data would be needed to build a complete picture.

The transition appears continuous and no obvious fatigue point exists. In [7], the fatigue point is roughly defined as the point where SKR and DSD occurs in equal proportion. To this end, a way of computing a fatigue point is devised. Figures 5.5a and 5.5b show a moving average of the DSD ratio computed with a window size of 9. We define the fatigue point as the first occurrence of an average over 0.5, shown as the green circles in the plots.

Using this definition, fatigue points are computed for all n and shown in Figure 5.6b. We expect $\hat{q} = cn^{5.923}$ (see Table 5.1) for some constant c, thus a loglog-linear regression is performed to find the constant. The result is $c = 1.2 \times 10^{-6}$ with the resulting line and confidence region shown in the figure. We see good agreement between the model and data.

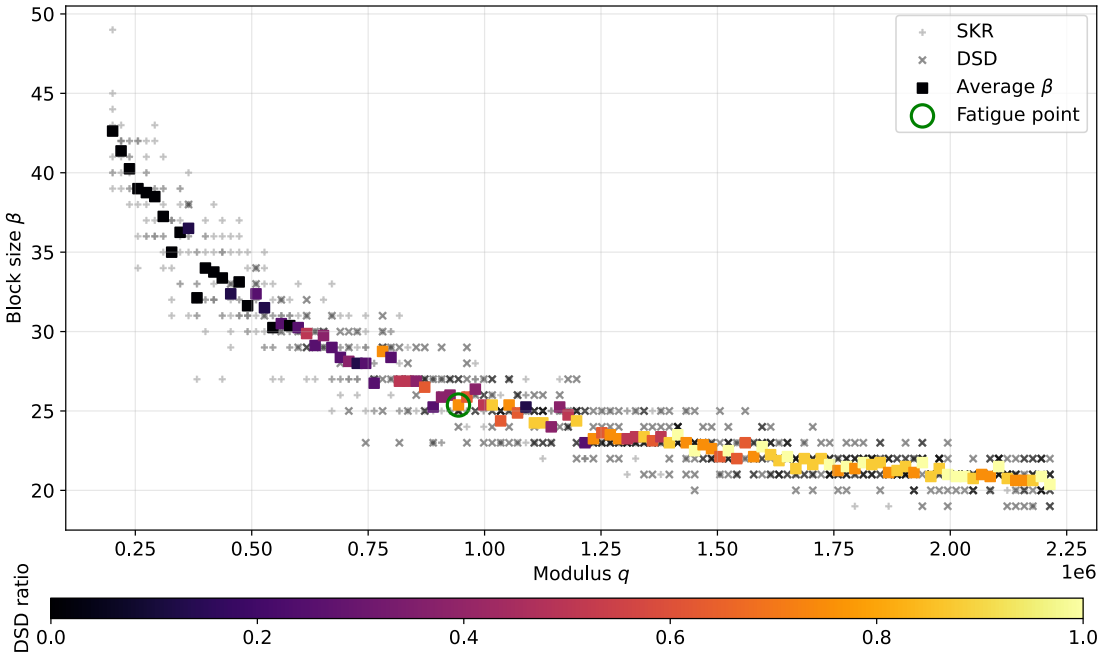


Figure 5.3: Successful block sizes for each given q for $n = 101$. Note the horizontal scaling. Plotted are the outcome of each experiment (grey plus signs and crosses), the average β (squares) and the ratio of DSD (colour of the squares). The fatigue point is shown as a green circle.

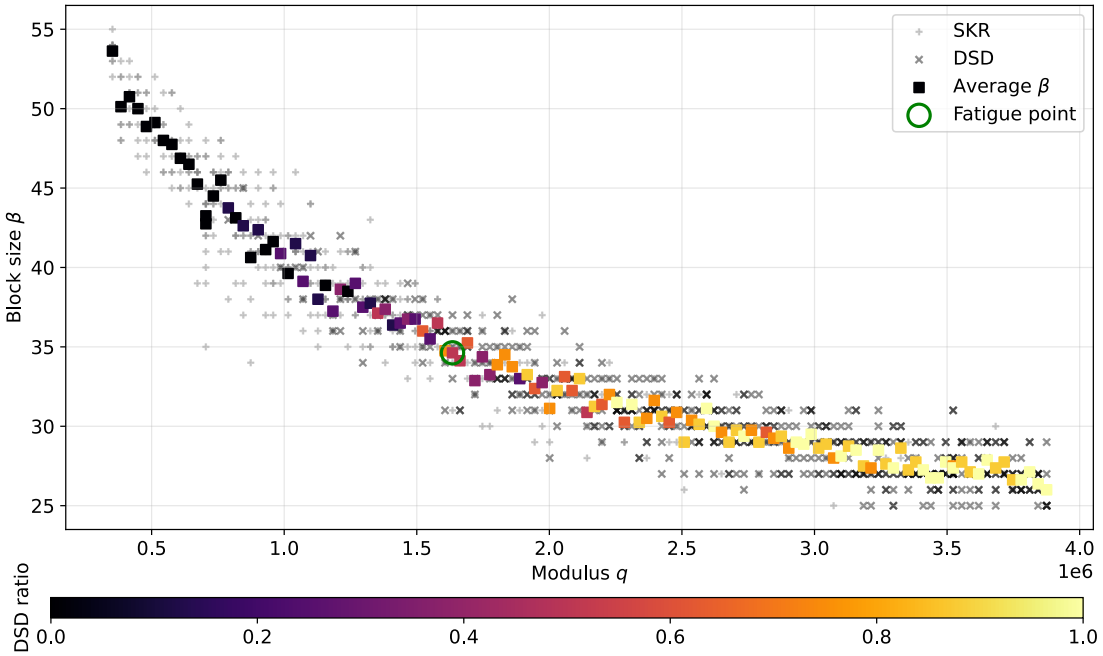
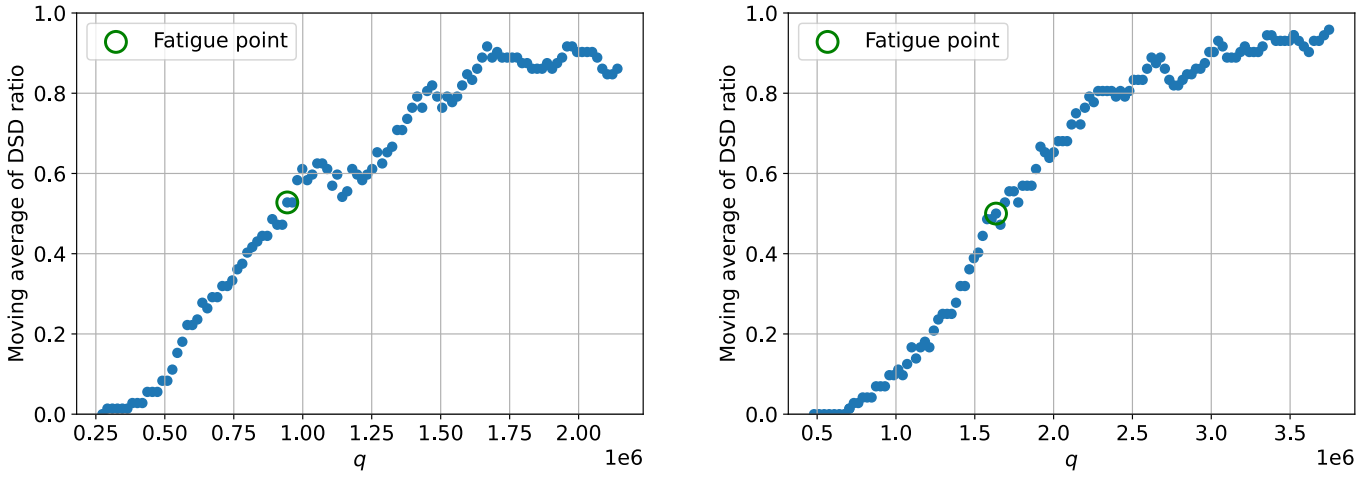


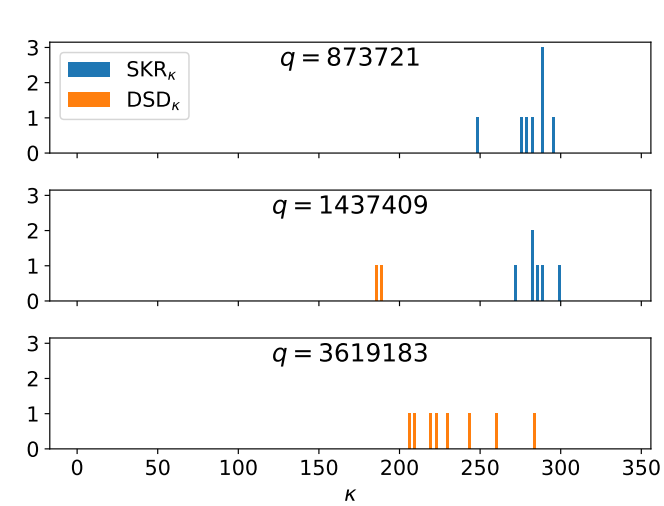
Figure 5.4: Successful block sizes for each given q for $n = 113$. Note the horizontal scaling. Plotted are the outcome of each experiment (grey plus signs and crosses), the average β (squares) and the ratio of DSD (colour of the squares). The fatigue point is shown as a green circle.



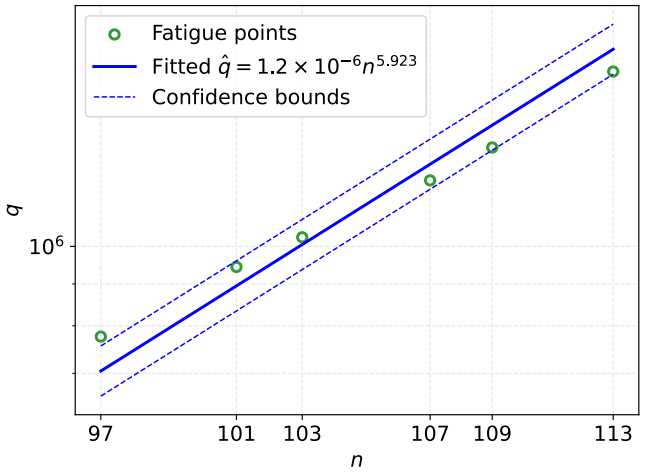
(a) $n = 101$.

(b) $n = 113$.

Figure 5.5: Moving average of DSD ratio for each given q. The fatigue point is shown as a green circle.



(a) Histogram over insertion position κ and type of attack for $n = 113$ over eight trials, for different q.



(b) Fatigue points (green circles) for multiple n together with line of best fit (solid blue) and confidence bounds (dashed blue).

Figure 5.6

5.4 Summary and Discussion

We establish that the DSD asymptotics are the same as for MLWE and derive SKR asymptotics using the full $n(m+d+1)$-dimensional lattice. In search for a theoretical fatigue point, we find it exists for all d. It increases with d which means for common choices of q and d, overstretching is not expected to be a problem.

The theoretical predictions are validated using BKZ reduction on NTWE lattices for multiple n between 97 and 113. Indeed, DSD occurs for the selected parameters and a clear transition from SKR can be observed. Fatigue points are computed as the first occurrence of a ratio over 0.5 in a moving average. Using a loglog-linear fit to the data, the fatigue point is fixed to

$$\hat{q} = 1.2 \times 10^{-6} \times n^{5.923}.$$

The agreement between theory and experiment is satisfactory and validates that Dense Sublattice Discovery is not isolated to NTRU. Interestingly, the transition point between SKR and DSD in Figures 5.3 and 5.4 is not as clearly defined compared to experiments on NTRU such as Figure 1 in [7]. This may stem from selecting q that are too "zoomed in" around the fatigue point. However, we should still expect a change in slope in the β-q line when the transition is complete. Such a change may be discerned in Figure 5.4 around the fatigue point. However, no significant advantage is gained from the DSD attack for any n tested.

A lack of change in slope may plausibly explained by observing that the predicted advantage of DSD in Figure 5.2 is very small above the fatigue point. Therefore, while a change in slope should occur, we should not expect it to be sharp. Furthermore, a clear transition might appear for larger n, where the behaviour should be closer to the asymptotics. This is substantiated by the observation that the moving averages in Figure 5.5 tend towards a monotonously increasing function as n increases. Finally, the lack of a clearly discernable fatigue point and change in slope may simply stem from a lack of data.

6

A Common Framework

This chapter concerns research question 6. We discuss why MLWE was not found to have a fatigue point. By a modification to the MLWE asymptotics, we show it is possible to predict a fatigue point that is very large.

Thus far, we have discussed the NTRU, MLWE and NTWE problems and their corresponding lattices. While NTRU and NTWE are found to have fatigue points, it appears no such point exists for MLWE. The lattices for MLWE and NTWE are very similar, with the only significant difference being that the last $n-1$ columns may be ignored due to the structure of the MLWE lattice. If this was not the case, the SKR analysis for NTWE and MLWE would be identical, with a fatigue point as a result.

It is therefore conceivable that a fatigue point appears when more of the last columns are included. To this end, we can consider keeping j columns in the MLWE basis (2.3). This will not add any extra information, as this embeds extra copies of the secret key that is already available. We then repeat the analysis of SKR for MLWE, this time in $O(n)$ in order to compare to NTRU. We therefore have $\nu = dn + k + j$, where k is the number of q-vectors, and assume $q = \Theta(n^Q)$, $||(\boldsymbol{e}, \boldsymbol{s}, t)|| = O(n^S)$ and $\ln(\gamma) = \frac{1}{2Bn} \ln n$. Minimising the right hand side for k gives

$$k = \sqrt{2BQ(dn+j)} - dn - j.$$

Still, the number of samples is unlimited meaning we can use this optimal k without obstruction. Using the assumptions, setting $j = Jn$ and taking logarithms in the 2016 estimate, the equation to be solved is

$$S = \frac{2Bn - n\sqrt{2BQ(d+J)} - 1}{2Bn - 2} + \frac{n\sqrt{2BQ(d+J)} - (J+d)}{n\sqrt{2BQ(d+J)}} Q.$$

Cancelling small terms and solving for B gives

$$B_{\text{SKR}} = \frac{2Q(d+J)}{(Q-S+1)^2}.$$

The asymptotic block size for successful DSD is again given by

$$B_{\text{DSD}} = 2\frac{\sqrt{4S^2 + 4QSd - d^2} + dQ + 2S}{Q^2+1},$$

which is the same expression for DSD in MLWE and NTWE scaled by a factor d due to operating in $O(n)$. Setting $S = 1/2$, the equation $B_{\text{SKR}} = B_{\text{DSD}}$ has a real solution for any $d \geq 0$ and $0 < J \leq 1$. The asymptotic fatigue point increases with d and decreases with J. In the analysis of MLWE we implicitly set $J = 0$. A more accurate setting would be $J = 1/n$ which means that asymptotically, MLWE is also expected to have a fatigue point. However, it will be very large. Consider Kyber [22] as an example, with $J = 1/256, d = 2$ and $S = 1/2$. Solving the equation $B_{\text{SKR}}(S, Q) = B_{\text{DSD}}(S, Q)$ using the expressions above yields an asymptotic fatigue point $\hat{Q}$ of size 2^{17}, which for real scenarios can be considered to be a point at infinity. Even with this analysis, MLWE realistically does not suffer from DSD.

Theoretically however, this is a satisfying conclusion. The SKR asymptotics are valid for any lattice with same the structure as MLWE: q-ary such that it follows the ZGSA and containing j copies of the secret key of length $O(n^S)$. The DSD asymptotics are valid for NTWE which captures both MLWE and NTRU. This analysis suggests any such lattice with a sublattice of dimension n will lead to a fatigue point. The density of the sublattice, depending on the distribution and size of the secret key, determines the fatigue point. It just so happens for MLWE that the opportunity to ignore the last $n - 1$ columns (setting $j = 1$) makes the SKR attack significantly easier than DSD in the full lattice. Indeed, we can recover the asymptotics for NTRU, MLWE and NTWE from these most general expressions. Keeping in mind that $d = 0$ yields NTRU from NTWE, we see the following.

- Setting $d = 0$ and $J = 1$ collapses the expressions to

$$B_{\text{SKR}} = \frac{2Q}{(Q - S + 1)^2}, \quad B_{\text{DSD}} = \frac{8S}{Q^2 + 1},$$

  the expressions for NTRU (when $S < 1$).

- Setting $J = 0$ and keeping d yields

$$B_{\text{SKR}} = \frac{2Qd}{(Q - S + 1)^2}, \quad B_{\text{DSD}} = 2\frac{\sqrt{4S^2 + 4QSd - d^2} + dQ + 2S}{Q^2 + 1},$$

  the expressions for MLWE when analysed in $O(n)$.

- Setting $J = 1$ (i.e. $j = n$) and keeping d yields

$$B_{\text{SKR}} = \frac{2Q(d + 1)}{(Q - S + 1)^2}, \quad B_{\text{DSD}} = 2\frac{\sqrt{4S^2 + 4QSd - d^2} + dQ + 2S}{Q^2 + 1},$$

  the expressions for NTWE when analysed in $O(n)$.

7

Summary and Conclusion

This work concerns the impact of selecting a relatively large modulus q in the lattice-based primitives NTRU, module-LWE and NTWE. We consider NTRU with larger polynomial coefficients and propose a linear dependence between fatigue point and variance of the coefficient distribution. This is verified using an estimator and experiments for small n. The asymptotics show a non-linear behaviour for sufficiently large coefficients. Finding the point where the linear model stops being valid remains an open question. An asymptotically secure cryptosystem using overstretched parameters is proposed using the non-linear asymptotics.

The DSD model is applied to MLWE and NTWE. While certain MLWE instantiations are expected to result in DSD, we argue that the attack will be similar to SKR and lead to no significant advantage. NTWE is predicted to have a fatigue point which is verified by experiments. The DSD events seem to match the characteristics of DSD in NTRU.

We show that all theoretical results may be recovered from a general analysis of q-ary lattices. In this way, MLWE can be expected to have a very large asymptotic fatigue point for common error sizes.

The findings above show that the DSD model developed in [7] can be applied to other q-ary lattices than NTRU, demonstrating generality of the model. As such, we consider all research questions to be answered. Our study makes an initial contribution toward bridging this knowledge gap about overstretched instances of MLWE and NTWE while verifying the validity of the used heuristics.

Although the theory of overstretched parameters is successfully extended to two new primitives, the subject is not exhausted. We provide a list of open questions without any particular order.

- The precision of analytic estimates remains limited. Implementing an estimator in similarity to [7] would therefore allow for use of newer complexity estimates for SKR.
- An exhaustive inquiry into the behaviour of NTRU for large σ^2 is needed to characterise the accuracy of the asymptotics. Where are they valid and how well do they estimate complexity? Such a discussion could be preceded by fixing the $o(1)$ constant in the asymptotic estimates.

- Performing experiments on MLWE to verify the lack of overstretched instances.
- Performing additional experiments on NTWE with larger n to increase the quality of data and decrease variance within instantiations with the same parameters.
- Studying the validity of the extended ZGSA in comparison to experiments.

Bibliography

[1] P. Asplund, *Forskar om att framtidssäkra internet*, https://www.kth.se/om/nyheter/centrala-nyheter/forskar-om-att-framtidssakra-internet-1.1366813, 2024.

[2] J. Hoffstein, J. Pipher, and J. H. Silverman, "Ntru: A ring-based public key cryptosystem," in *Algorithmic Number Theory*, J. P. Buhler, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 267–288, ISBN: 978-3-540-69113-6.

[3] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, ser. STOC '05, Baltimore, MD, USA: Association for Computing Machinery, 2005, pp. 84–93, ISBN: 1581139608. DOI: 10.1145/1060590.1060603. [Online]. Available: https://doi.org/10.1145/1060590.1060603.

[4] G. Alagic, D. Apon, D. Cooper, *et al.*, *Status report on the third round of the NIST Post-Quantum Cryptography Standardization process*. Sep. 2022. DOI: 10.6028/nist.ir.8413-upd1. [Online]. Available: http://dx.doi.org/10.6028/NIST.IR.8413-upd1.

[5] M. Albrecht, S. Bai, and L. Ducas, "A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and graded encoding schemes," in *Annual International Cryptology Conference*, Springer, 2016, pp. 153–178.

[6] P. Kirchner and P.-A. Fouque, "Revisiting lattice attacks on overstretched ntru parameters," in *Advances in Cryptology – EUROCRYPT 2017*, J.-S. Coron and J. B. Nielsen, Eds., Cham: Springer International Publishing, 2017, pp. 3–26, ISBN: 978-3-319-56620-7.

[7] L. Ducas and W. van Woerden, "NTRU fatigue: how stretched is overstretched?" In *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV 27*, Springer, 2021, pp. 3–32.

[8] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Advances in Cryptology – EUROCRYPT 2010*, H. Gilbert, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–23, ISBN: 978-3-642-13190-5.

[9] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Designs, Codes and Cryptography*, vol. 75, no. 3, pp. 565–599,

Feb. 2014, ISSN: 1573-7586. DOI: 10.1007/s10623-014-9938-4. [Online]. Available: http://dx.doi.org/10.1007/s10623-014-9938-4.

[10] J. Gärtner, "NTWE: A Natural Combination of NTRU and LWE," in *Post-Quantum Cryptography*, T. Johansson and D. Smith-Tone, Eds., Cham: Springer Nature Switzerland, 2023, pp. 321–353, ISBN: 978-3-031-40003-2.

[11] S. D. Galbraith, *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012.

[12] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: a cryptographic perspective* (The Kluwer International Series in Engineering and Computer Science). Boston, Massachusetts: Kluwer Academic Publishers, Mar. 2002, vol. 671.

[13] R. Lindner and C. Peikert, "Better key sizes (and attacks) for lwe-based encryption," in *Topics in Cryptology – CT-RSA 2011*, A. Kiayias, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 319–339, ISBN: 978-3-642-19074-2.

[14] V. Lyubashevsky and D. Micciancio, "On bounded distance decoding, unique shortest vectors, and the minimum distance problem," in *Advances in Cryptology - CRYPTO 2009*, S. Halevi, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 577–594, ISBN: 978-3-642-03356-8.

[15] V. Lyubashevsky and G. Seiler, "Nttru: Truly fast ntru using ntt," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 3, pp. 180–201, May 2019. DOI: 10.13154/tches.v2019.i3.180-201. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/8293.

[16] M. R. Albrecht and L. Ducas, "Lattice attacks on ntru and lwe: A history of refinements," *IACR Cryptol. ePrint Arch.*, vol. 2021, p. 799, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235658005.

[17] C. P. Schnorr, "Lattice reduction by random sampling and birthday methods," in *STACS 2003*, H. Alt and M. Habib, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 145–156, ISBN: 978-3-540-36494-8.

[18] D. Micciancio and M. Walter, "Practical, predictable lattice basis reduction," in *Advances in Cryptology – EUROCRYPT 2016*, M. Fischlin and J.-S. Coron, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 820–849, ISBN: 978-3-662-49890-3.

[19] E. Alkim, L. Ducas, T. Pöppelmann, and P. 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. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim.

[20] C. Chen, O. Danba, J. Hoffstein, *et al.*, "Algorithm specifications and supporting documentation," *Brown University and Onboard security company, Wilmington USA*, 2019. [Online]. Available: https://ntru.org/f/ntru-20190330.pdf.

[21] M. R. Albrecht, B. R. Curtis, A. Deo, *et al.*, "Estimate all the {lwe, ntru} schemes!" In *Security and Cryptography for Networks: 11th International Con-*

ference, SCN 2018, Amalfi, Italy, September 5–7, 2018, Proceedings 11, Springer, 2018, pp. 351–367.

[22] R. Avanzi, J. Bos, L. Ducas, *et al.*, *Crystals-kyber algorithm specifications and supporting documentation*, https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf, 2020.

[23] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig, "Improved security for a ring-based fully homomorphic encryption scheme," in *Cryptography and Coding*, M. Stam, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 45–64, ISBN: 978-3-642-45239-0.

[24] J. R. Durbin, *Modern Algebra*, en, 6th ed. Chichester, England: John Wiley & Sons, Dec. 2008.

[25] Wikipedia contributors, *Hadamard's inequality — Wikipedia, the free encyclopedia*, [Online; accessed 5-May-2025], 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Hadamard%27s_inequality&oldid=1285635731.

[26] W. Banaszczyk, "New bounds in some transference theorems in the geometry of numbers," *Mathematische Annalen*, vol. 296, pp. 625–635, 1993.

[27] D. Balbás, *The hardness of LWE and ring-LWE: A survey*, Cryptology ePrint Archive, Paper 2021/1358, 2021. [Online]. Available: https://eprint.iacr.org/2021/1358.

[28] K. Conrad, *The different ideal*. [Online]. Available: https://kconrad.math.uconn.edu/blurbs/gradnumthy/different.pdf.

[29] J. S. Milne, *Algebraic number theory (v3.08)*, Available at www.jmilne.org/math/, 2020.

[30] Wikipedia contributors, *Module (mathematics) — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Module_(mathematics)&oldid=1277500265, [Online; accessed 4-March-2025], 2025.

[31] Wikipedia contributors, *Cyclotomic polynomial — Wikipedia, the free encyclopedia*, [Online; accessed 1-May-2025], 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Cyclotomic_polynomial&oldid=1284543822.

[32] O. Regev, E. Kaplan, D. Sieradzki, V. Bronstein, and I. Haviv, https://cims.nyu.edu/~regev/teaching/lattices_fall_2009/index.html, Lecture notes in Lattices in Computer Science, 2009.

[33] Y. Chen, "Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe," Presented and defended publicly on November 13, 2013, Ph.D. dissertation, Université Paris Diderot, Nov. 2013. [Online]. Available: https://archive.org/details/PhDChen13.

[34] C. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Mathematical Programming*, vol. 66, no. 2, pp. 181–199, 1994. DOI: 10.1007/BF01581144.

[35] Y. Chen and P. Q. Nguyen, "Bkz 2.0: Better lattice security estimates," in *Advances in Cryptology – ASIACRYPT 2011*, D. H. Lee and X. Wang, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–20, ISBN: 978-3-642-25385-0.

[36] Y. Aono, Y. Wang, T. Hayashi, and T. Takagi, "Improved progressive bkz algorithms and their precise cost estimation by sharp simulator," in *Advances in Cryptology – EUROCRYPT 2016*, M. Fischlin and J.-S. Coron, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 789–819, ISBN: 978-3-662-49890-3.

[37] E. Mårtensson, "Solving ntru challenges using the new progressive bkz library," English, M.S. thesis, Department of Electrical and Information Technology, 2016.

[38] S. Bai and S. D. Galbraith, "Lattice decoding attacks on binary lwe," in *Information Security and Privacy*, W. Susilo and Y. Mu, Eds., Cham: Springer International Publishing, 2014, pp. 322–337, ISBN: 978-3-319-08344-5.

[39] V. Lyubashevsky, C. Peikert, and O. Regev, "A toolkit for ring-lwe cryptography," in *Advances in Cryptology – EUROCRYPT 2013*, T. Johansson and P. Q. Nguyen, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 35–54, ISBN: 978-3-642-38348-9.

[40] C. Peikert, O. Regev, and N. Stephens-Davidowitz, "Pseudorandomness of ring-lwe for any ring and modulus," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2017, Montreal, Canada: Association for Computing Machinery, 2017, pp. 461–473, ISBN: 9781450345286. DOI: 10.1145/3055399.3055489. [Online]. Available: https://doi.org/10.1145/3055399.3055489.

A

More Theory

This appendix collects theory that is background knowledge to the thesis while not being strictly related to the results. The goal is that the main text in conjunction with this appendix will provide a self-contained introduction to lattice-based cryptography for a reader inexperienced in abstract algebra. The chapter begins with an introduction to rings, followed by a discussion of polynomials in the context of abstract algebra. The next section collects extra lattice theory, with Appendix A.2.2 building on the algebra introduction. This is followed by a detailed exposition of reduction algorithms and a presentation of two additional LWE variants.

A.1 Algebra Primer

This section is intended as a gentle introduction to the algebra of lattice cryptography. It is written as a mathematical text concerning rings and polynomials. If these subjects feel familiar, the reader may proceed to the next section. Most of this content is based on [24].

A.1.1 Rings

In this subsection, rings and ideals are presented. A ring generalises a group. We note that for the integers, we have both addition and multiplication and that multiplication is defined as repeated addition. Clearly, there is extra structure here that can be formalised. Modelling from $\mathbb{Z}$, we define a ring as follows.

Definition A.1.1. *A ring is a set R together with two operations, addition $+$ and multiplication (denoted by juxtaposition), such that*

1. $a + (b + c) = (a + b) + c \quad \forall a, b, c \in R$,
2. $\exists\ 0 \in R$ *such that* $a + 0 = 0 + a = a \quad \forall a \in R$,
3. *Given* $a \in R$, $\exists\ -a \in R$ *such that* $a + (-a) = 0, \quad a \in R$,
4. $a + b = b + a \quad \forall a, b \in R$,
5. $a(bc) = (ab)c \quad \forall a, b, c \in R$, *and*
6. $a(b + c) = ab + ac$ *and* $(a + b)c = ac + bc \quad \forall a, b, c \in R$.

A ring is commutative if the multiplication is commutative, i.e. $ab = ba, \forall a, b \in R$.

Intuitively, this is an additive group with multiplication that is associative and distributive. However, note the definition does not include multiplicative inverses. A ring is therefore, in general, not a multiplicative group. As mentioned before $(\mathbb{Z}, +, \cdot)$ define a ring, but also $(2\mathbb{Z}, +, \cdot)$. This is also true for $(\mathbb{Q}, +, \cdot)$ and $(\mathbb{R}, +, \cdot)$.

In analogy with subgroups, a subset of a ring that is closed itself is a subring. We also define a special type of subrings that will become important later on.

Definition A.1.2. *A subset* $S \subset R$ *is a subring of* R *if* S *is a ring with the same operations as* R*. It is an ideal of* R *if*

$$sr \in S \text{ and } rs \in S \quad \forall r \in R, s \in S.$$

In other words, a subring that is closed under multiplication with ring elements is an ideal.

For example, $2\mathbb{Z}$ is a subring of $\mathbb{Z}$. Given an element $a \in R$, there are a special type of ideals that are the smallest possible ideal containing a.

Definition A.1.3. *Given a ring element* $a \in R$*, the set*

$$(a) = \{ar : r \in R\},$$

is the smallest ideal containing a*. It is called the principal ideal of* a*.*

Recall that the definition of a ring does not require that the elements form a group with respect to multiplication. With this restriction, we get a new type of structure.

Definition A.1.4. *A commutative ring* $(R, +, \cdot)$ *where* $(R, \cdot)$ *forms a group is called a field.*

Going back to our numbers, $(\mathbb{Z}, +, \cdot)$ is not a field, as $a \in \mathbb{Z}$ has inverse $1/a \notin \mathbb{Z}$. However, if we include the inverses, we get $(\mathbb{Q}, +, \cdot)$, which is a field. Furthermore, $(\mathbb{R}, +, \cdot)$ is as well. Combining facts about the multiplicative group and additive group over $\mathbb{Z}_q$, we can establish the following.

Lemma A.1.5. $\mathbb{Z}_q$ *is a field* $\iff$ q *is a prime.*

In analogy with group homomorphisms, we define ring homomorphisms as the following.

Theorem-Definition A.1.6. *Let* R *and* S *be two rings and* $\theta : R \to S$ *a mapping.* θ *is a ring homomorphism if*

$$\theta(a + b) = \theta(a) + \theta(b) \text{ and } \theta(ab) = \theta(a)\theta(b), \quad \forall a, b \in R.$$

A bijective ring homomorphism is called a ring isomorphism. Two rings are isomorphic if there exist ring isomorphisms between them. The image $\{\theta(r) : r \in R\} \subseteq S$ *is a subring of* S*.*

The kernel of θ*, denoted* $\ker(\theta)$ *is the set* $\{r \in R : \theta(r) = 0\}$*. The kernel is an ideal of* R*.*

Of course, we can write down the analogue of a quotient group for rings.

Theorem-Definition A.1.7. *Let R be a ring, and I an ideal of R. Further, let R/I denote the set of right cosets when considered as an additive group of R. Define the operations*

$$(I+a)+(I+b)=I(a+b) \text{ and } (I+a)(I+b)=I+(ab) \quad \forall I+a, I+b \in R/I.$$

The set R/I with addition and multiplication defined as above forms a ring called the quotient ring of R by I.

This concludes our exposition of rings, although we immediately continue by applying the theory.

A.1.2 Polynomials

This subsection discusses some concepts from Galois theory, starting with general polynomial arithmetic and ending with field extensions. Polynomials are an indispensable tool in mathematics. Abstract algebra provides a convenient framework for discussing properties of polynomials and exploring beyond the realm where coefficients are drawn from $\mathbb{R}$. As it turns out, polynomials are also the natural way to encode structure into lattice problems. First, we must define what they are.

Theorem-Definition A.1.8. *Let R be a commutative ring. A polynomial in indeterminate x over R is of the form*

$$a_0 + a_1 x + \ldots + a_n x^n,$$

where $a_i \in R$ for $i = 1, \ldots, n$ are called the coefficients of the polynomial. For $a_n \neq 0$, we define n as the degree of the polynomial and call a_n its leading coefficient. If $a_n = 1$, the polynomial is called monic. We denote the set of all polynomials over R by $R[x]$.

Furthermore, consider two polynomials $a(x) = a_0 + a_1 x + \ldots + a_n x^n$ and $b(x) = b_0 + b_1 x + \ldots b_m x^m$. We define

$$a(x) + b(x) = \sum_{i=0}^{n} (a_i + b_i) x^i + \sum_{j=n+1}^{m} b_j x^j,$$

if $m > n$ and analogously for $m \leq n$. Further,

$$a(x)b(x) = \sum_{i=1}^{m+n} c_k x_k, \quad c_k = \sum_{i,j>0, i+j=k} a_i b_j.$$

With respect to these operations, $R[x]$ is a commutative ring. We call it the ring of polynomials in x over R.

An overarching goal of the study of polynomials is to find their roots, but they do not always exist in the setting polynomials are defined.

Definition A.1.9. *Two polynomials $a(x)$ and $b(x)$ over a field $\mathbb{K}$ are associates if $a(x) = cb(x)$ for some $c \in \mathbb{K} \setminus \{0\}$. Each non-zero polynomial has exactly one monic polynomial among its associates and two trivial sets of divisors; its associates and the polynomials of degree zero. A polynomial with only trivial divisors is called irreducible.*

By this definition, an irreducible polynomial is always of the form $a(x) = c(x)d(x)$ where one factor is an associate and the other of degree zero.

Example A.1. The polynomial $x^2 - 2 \in \mathbb{Q}[x]$ is irreducible, as its roots are $\pm\sqrt{2} \notin \mathbb{Q}$. Consequently, it is reducible over $\mathbb{R}$; its non-trivial divisors are $x - \sqrt{2}$ and $x + \sqrt{2}$. The polynomial $x^2 + 1$ is irreducible over $\mathbb{R}$ but not over $\mathbb{C}$. •

The following theorem clearly illustrates that irreducible polynomials behave as prime numbers. Both only have trivial divisors and can be used to factor any other element in their respective sets.

Theorem A.1.10. *Every polynomial of degree $n \geq 1$ over a field $\mathbb{K}$ can be written on the form*

$$a(x) = c \prod a_i(x),$$

where $c \in \mathbb{K}$ and $a_i(x)$ are monic irreducible polynomials over $\mathbb{K}$. Up to order, this factorisation is unique.

As we established that some polynomials do not have roots in certain fields, but do have in others, a natural question is how to extend fields to provide roots to all polynomials. In general, a field extension of a field $\mathbb{K}_1$ is an embedding into another field $\mathbb{K}_2$ such that $\mathbb{K}_1 \subseteq \mathbb{K}_2$ and $\mathbb{K}_1$ is a subfield of $\mathbb{K}_2$. The following theorem describes how embeddings can be constructed for any irreducible polynomial.

Theorem A.1.11. *Let $\mathbb{K}$ be a field and assume that $a(x) \in \mathbb{K}[x]$ is irreducible over $\mathbb{K}$ and of degree n. Then*

$$E = \frac{\mathbb{K}[x]}{(a(x))}$$

is a field extension of $\mathbb{K}$ and $a(x)$ has a root α in E.

Any element in E can be expressed uniquely on the form

$$b_0 + b_1\alpha + \ldots + b_{n-1}\alpha^{n-1}, \quad b_i \in \mathbb{K}, i = 1, \ldots, n-1.$$

Addition of polynomials is performed as usual, and multiplication is performed by multiplying $b(\alpha)$ and $c(\alpha)$ as usual, dividing by $a(\alpha)$ and keeping the remainder as the result as $b(\alpha)c(\alpha)$.

Example A.2. Recall that $a(x) = 1 + x^2$ is irreducible over $\mathbb{R}$. We extend $\mathbb{R}$ to the quotient ring

$$\frac{\mathbb{R}}{(x^2+1)}.$$

Denoting one root of $a(x)$ by i, the above theorem tells us that any element in $\mathbb{R}/(x^2+1)$ can be written as $b_1 + b_2 i$. In fact, one can show $\mathbb{R}/(x^2+1) \approx \mathbb{C}$, and this is exactly how it is constructed. •

A.2 Additional Topics in Lattice Theory

This section collects additional theory on lattices. Among these, the Hadamard bound on the determinant and the definition of a dual lattice are central. Furthermore, Section A.2.2 provides the connection between the abstract formulation of a primitive and the equivalent lattice problem.

A.2.1 More Theorems on Lattices

In Section 2.1, we provided a definition of lattices in $\mathbb{R}^v$. A general definition of a lattice can be given as follows.

Definition A.2.1. *A lattice in a number field $\mathbb{K}$ is the $\mathbb{Z}$-span of a $\mathbb{Q}$-basis of $\mathbb{K}$.*

The volume of a lattice can be computed in multiple ways, for example

$$\det(\mathcal{L}(\boldsymbol{B})) = \mathrm{vol}(\mathcal{L}(\boldsymbol{B})) = \sqrt{\det(\boldsymbol{B}^T\boldsymbol{B})} = \prod_{i=1}^{v} ||\boldsymbol{b}_i^*||,$$

where $\boldsymbol{b}_i^*$ denotes the Gram–Schmidt orthogonalisation of $\boldsymbol{b}_i$, see Definition 2.1.5. For a full rank lattice it can also be computed as $|\det(\boldsymbol{B})|$. A useful bound for such a basis is the following.

Lemma A.2.2 (Hadamard's inequality [25])**.** *For a matrix $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_v]$, we have*

$$|\det(\boldsymbol{B})| \leq \prod_{i=1}^{v} ||\boldsymbol{b}_i||. \tag{A.1}$$

We saw earlier that the lattice in Figure 2.1 had two possible bases. In fact, there exist infinitely many bases to a given lattice.

Theorem-Definition A.2.3. *If two bases $\boldsymbol{B}$ and $\boldsymbol{B}'$ generate the same lattice, we say they are equivalent. Equivalent bases are related by an unimodular transformation $\boldsymbol{B}' = \boldsymbol{B}\boldsymbol{U}$, where $\boldsymbol{U}$ is a unimodular matrix (a square matrix with integer entries such that $\det(\boldsymbol{U}) = \pm 1$).*

As there exist infinitely many unimodular matrices (in fact, they form a group), a lattice basis is not unique. This also shows that the volume of a lattice is independent of the choice of basis.

We introduced $\lambda_1(\mathcal{L})$ as the length of the shortest lattice vector. In fact, there exists a sequence of them, called the successive minima of the lattice.

Definition A.2.4. *Let $\mathcal{L}$ be a lattice of dimension v and $B_v(r)$ an open ball of radius r centred in 0. Then the ith minimum of $\mathcal{L}$ is*

$$\lambda_i(\mathcal{L}) = \inf\{r : \dim(\mathrm{span}(\mathcal{L} \cap B_v(r))) \geq i\}.$$

In words, this is the radius of the smallest ball containing i linearly independent lattice vectors. The distribution of these provides information about the orthogonality of the lattice basis. If all λ_i are the same, the lattice basis is orthogonal. A

wide distribution of minima, i.e. $\lambda_1 \ll \lambda_\nu$, suggests a highly non-orthogonal basis. It can be shown that the successive minima are always attained, so the infimum can in fact be replaced by a minimum.

We described an upper bound on the shortest vector in in Theorem 2.1.4. The shortest vector is also bounded from below.

Theorem A.2.5 (Theorem 1.1 in [12]). *Let* $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 & \dots & \boldsymbol{b}_\nu \end{bmatrix}$ *be a lattice basis. Then*

$$\lambda_1 \geq \min_i ||\boldsymbol{b}_i^*|| > 0,$$

where $\boldsymbol{b}_j^*$ *denotes the Gram–Schmidt orthogonalisation of* $\boldsymbol{b}_j$.

Theorem 2.1.4 has a generalisation involving all minima.

Theorem A.2.6 (Minkowski's second theorem, Theorem 1.5 in [12]). *Let* $\mathcal{L}$ *be a lattice of dimension* ν. *Then*

$$\left(\prod_{i=1}^{\nu} \lambda_i(\mathcal{L})\right)^{1/\nu} < \sqrt{\nu}\,\mathrm{vol}(\mathcal{L})^{1/\nu}.$$

Minkowski's theorems are frequently used in analysis of lattices and lattice algorithms.

Another important construction derived from a lattice is its dual, which can be likened to the orthogonal complement of a basis in linear algebra.

Definition A.2.7. *Let* $\mathcal{L}$ *be a lattice. The dual of a lattice is*

$$\mathcal{L}^\vee = \{\boldsymbol{a} \in \mathbb{R}^\nu : \langle \boldsymbol{a}, \boldsymbol{x} \rangle \in \mathbb{Z}, \forall \boldsymbol{x} \in \mathcal{L}\}.$$

In other words, the dual is the set of all vectors that have integer inner product with lattice vectors. If $\boldsymbol{B}$ is a basis for $\mathcal{L}$, $\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{B})^{-1}$ is a basis for $\mathcal{L}^\vee$. Consequently, $\mathrm{vol}(\mathcal{L}^\vee) = (\mathrm{vol}(\mathcal{L}))^{-1}$ and $(\mathcal{L}^\vee)^\vee = \mathcal{L}$. Successive minima in $\mathcal{L}$ and $\mathcal{L}^\vee$ are also related. We have that $1 \leq \lambda_i(\mathcal{L})\lambda_{\nu-i+1}(\mathcal{L}^\vee) \leq \nu$ due to [26] and $\lambda_1(\mathcal{L})\lambda_1(\mathcal{L}^\vee) \leq \nu$ due to Theorem 2.1.4.

A.2.2 Modules, Algebraic Numbers and Ideal Lattices

The general definition of a lattice hints at a deeper connection between lattices and algebraic number theory. We explore this here. The main goal is to provide a link between the cryptosystems and their lattice formulations in Section 2.2. We begin by discussing modules which are a generalisation of vector spaces, move on to introducing number fields and tie it together by showing how number fields are vector spaces themselves. This subsection is mainly based on [27] and [28], but the interested reader may find [29] and [24] useful.

Definition A.2.8 (Modules, as presented in [30])**.** *Let R be a ring with multiplicative identity 1_R. A left R-module M is a commutative group $(M, +)$ together with an operation $\cdot : R \times M \to M$ such that*

1. $r \cdot (x + y) = r \cdot x + r \cdot y, \quad r \in R,\ x, y \in M,$
2. $(r + s) \cdot x = r \cdot x + s \cdot x, \quad r, s \in R,\ x \in M,$
3. $(rs) \cdot x = r \cdot (s \cdot x), \quad r, s \in R,\ x \in M,$
4. $1_R \cdot x = x.$

We call $\cdot$ scalar multiplication. A right R-module is defined mutatis mutandis with $\cdot$ from the right.

An R-module is finitely generated with range ν if $\exists x_1, \ldots, x_\nu \in M$ such that for any $v \in M$,

$$v = \sum_{i=1}^{\nu} r_i \cdot x_i, \quad r_i \in R.$$

An R-module is free if is isomorphic to a direct sum of copies of R. We denote this R^d for d copies.

Let N be a subgroup of M. Then N is a submodule if

$$r \cdot n \in N, \quad r \in R, n \in N.$$

In other words, N is a subgroup that is also closed under scalar multiplication.

We omit $\cdot$, even though this causes ambiguity between ring multiplication and scalar multiplication. If the base ring R is a field, the module is called an R-vector space. If I is a left ideal in R, then I is a left R-module. The same holds for right ideals, mutatis mutandis.

We now switch to discussing algebraic numbers, which may be recognised as solutions to polynomial equations. They are usually extensively studied in introductory analysis courses. Here we consider them in a more formal light.

Definition A.2.9. *A number $\alpha \in \mathbb{C}$ is an algebraic number if $\exists f(x) \in \mathbb{Q}[x]$ such that $f(\alpha) = 0$. Its minimal polynomial f_α is the monic irreducible polynomial of lowest degree with α as a root. The roots $\alpha_1, \ldots, \alpha_n$ of f_α are called the conjugates of α. The degree of α is the degree of f_α.*

Definition A.2.10. *If f_α has integer coefficients, we call α an algebraic integer.*

Definition A.2.11. *A number field $\mathbb{K}$ is a field extension constructed by adjoining $\mathbb{Q}$ with an algebraic number. The degree of $\mathbb{K}$ is the degree of α.*

It turns out that, as a consequence of Theorem A.1.11, any number field constructed as above with an algebraic number α is isomorphic to the field extension $\mathbb{Q}[x]/(f_\alpha(x))$. Therefore elements in number fields can be represented as polynomials.

We point out a type of polynomials common in cryptographic constructions, called cyclotomic polynomials. All three cryptosystems in this text use rings defined using (products of) these.

Definition A.2.12 (Cyclotomic polynomials, as given in [31])**.** *The nth cyclotomic polynomial is the minimal polynomial with integer coefficients of the field constructed by adjoining* $\mathbb{Q}$ *with any primitive nth root of unity. It is given as*

$$\Phi_n(x) = \prod_{\substack{1 \le l \le k \\ \gcd(l,n)=1}} x - e^{2\pi i l/n}.$$

Bringing it together, recall a module constructed from a field is a vector space. Thus a number field can be seen as a vector space of dimension n if we use $\{1, \alpha, \ldots, \alpha^{n-1}\}$ as a basis. In other words, given an element $g = g_0 + g_1\alpha + \ldots + g_{n-1}\alpha^{n-1}$ in a number field of degree n, one simply considers the coefficients $(g_0, g_1, \ldots, g_{n-1})$ as a vector. This basis is called the power basis and constructing a vector space this way is called coefficient embedding. Embeddings transform abstract vector spaces into geometric objects and allow us to define operators such as trace and norm. These are useful for geometric analysis of the structure of vector spaces.

Remark. *The choice of embedding may affect security in a cryptosystem and must be considered carefully. Traditionally, the naive coefficient embedding was standard, however the canonical embedding has proven useful in cryptographic proofs, mainly concerning ring-LWE. The naturalness of the canonical embedding is evident when working with cyclotomic polynomials. We present it here for context, and the interested reader is referred to [8].*

Definition A.2.13. *The canonical embedding is a map* $\sigma : \mathbb{K} \to \mathbb{R}^{\nu_1} \times \mathbb{C}^{2\nu_2}$ *defined as*

$$\sigma(x) = (\sigma_1(x), \ldots, \sigma_n(x)),$$

where

1. $\sigma_i(x) : \mathbb{K} \to \mathbb{C}$ *is a map such that* $\sigma_i(\alpha_1) = \alpha_i$*, i.e. it maps* α *to each of its conjugates,*
2. ν_1 *is the number of real conjugates such that* $\alpha_i \in \mathbb{R}$*,*
3. *and* $2\nu_2 = n - \nu_1$ *is the number of complex conjugates.*

We return to algebraic integers. They form a ring in $\mathbb{K}$, as the addition and multiplication of two integers results in an integer. This ring is called the ring of integers of $\mathbb{K}$ and denoted $\mathcal{O}_\mathbb{K}$. Notably, $\mathcal{O}_\mathbb{K}$ is a finitely generated $\mathbb{Z}$-module with some basis $\{o_1, \ldots, o_n\}$ that can also be used as a basis for $\mathbb{K}$ as a $\mathbb{Q}$-vector space.

We now arrive at the final piece in this exposition of algebraic number theory, which will allow us to make the connection between ideals and lattices. In this context, ideals (as defined in Appendix A.1) in $\mathcal{O}_\mathbb{K}$ are often called integral ideals to distinguish between them and the following definition. Recall an (integral)

ideal is a subring that is closed under multiplication, meaning if $a \in I$ and $b \in \mathcal{O}_{\mathbb{K}}$ then $ab \in I$.

Theorem-Definition A.2.14. *A fractional ideal is an $\mathcal{O}_{\mathbb{K}}$-module $\mathcal{I} \subset \mathbb{K}$ such that $d\mathcal{I}$ is an (integral) ideal for some $d \in \mathcal{O}_K$. In this context, this is the same as a finitely generated $\mathcal{O}_{\mathbb{K}}$-submodule of $\mathbb{K}$. The set of fractional ideals form a group under multiplication.*

We conclude by providing the connection to lattices. We stress that a fractional ideal $\mathcal{I}$ is a finitely generated $\mathcal{O}_{\mathbb{K}}$-submodule. Taking the basis $\{i_1, \ldots, i_\nu\}$, any element in $\mathcal{I}$ can be written

$$i = \sum_{j=1}^{\nu} r_j i_j, \quad r_j \in \mathcal{O}_{\mathbb{K}}.$$

Comparing to the definition of a lattice in $\mathbb{R}^\nu$, we see a striking resemblance. The ideal $\mathcal{I}$ is a lattice when viewed as the $\mathcal{O}_{\mathbb{K}}$-span of the basis $\{i_j\}_{j=1}^{\nu}$. This lattice is called an ideal lattice. When endowed with an embedding, be it coefficient or canonical, one can define trace and norm and discuss geometrical properties as above.

As $\mathcal{I}$ defines a lattice, $\mathcal{I}^\vee$ will define its dual, either defined geometrically (as in Definition A.2.7) or algebraically. In the latter, the dual ideal is given by $\mathcal{I}^\vee = \mathcal{I}^{-1}\mathcal{O}_{\mathbb{K}}^\vee$, where $\mathcal{I}^{-1}$ is the inverse of $\mathcal{I}$ in the group of fractional ideals, and $\mathcal{O}_{\mathbb{K}}^\vee$ is called the codifferent ideal which is not discussed here. More details about the algebraic structure of ideal lattices can be found in [28].

A.3 Lattice Reduction in More Detail

In this section, we discuss reduction algorithms at a deeper level. Building on each other, we begin with Gauss–Lagrange reduction, proceed with LLL and conclude with BKZ. As pointed out in the main text, the core of each algorithm is the definition of a reduced basis.

A.3.1 Gauss–Lagrange Reduction in Two Dimensions

For lattices in two dimensions, there exists an algorithm that solves SVP exactly in polynomial time. This is usually attributed to Gauss and Lagrange. We begin by defining what a "good" basis is in this context.

Definition A.3.1 (Reduced basis in $\mathbb{R}^2$)**.** *A lattice basis $\{\boldsymbol{b}_1, \boldsymbol{b}_2\}$ in $\mathbb{R}^2$ is reduced if*

1. $\mu_{21} \leq \frac{1}{2}$ *and*
2. $||\boldsymbol{b}_1|| \leq ||\boldsymbol{b}_2||$.

Having in mind that subtracting integer multiples and permuting vectors preserves the lattice, this prescribes two operations we can perform to reduce any basis. The first condition suggests we subtract multiples of $\boldsymbol{b}_1$ from $\boldsymbol{b}_2$ until $\boldsymbol{b}_2$ is short enough. The second suggests swapping $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. Excluding edge cases,

Algorithm 1 : Gauss–Lagrange algorithm

Input: A lattice basis $\boldsymbol{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2\}$
Output: A reduced basis.
  while $\boldsymbol{B}$ is not reduced **do**
    Reduce: $\boldsymbol{b}_2 \leftarrow \boldsymbol{b}_2 - c\boldsymbol{b}_1$ where $c = \left\lceil \frac{\langle \boldsymbol{b}_2, \boldsymbol{b}_1 \rangle}{\langle \boldsymbol{b}_1, \boldsymbol{b}_1 \rangle} \right\rfloor$.
    Swap: If $||\boldsymbol{b}_1|| > ||\boldsymbol{b}_2||$ then $\boldsymbol{b}_1 \leftrightarrow \boldsymbol{b}_2$.
  end while

this is what the Gauss–Lagrange algorithm does, summarised in Algorithm 1. In more detail, we note that after a reduce step, $\mu_{21} \leq \frac{1}{2}$ by construction. As lengths have changed however, we might not have $||\boldsymbol{b}_1|| \leq ||\boldsymbol{b}_2||$. As $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are mere labels, they may be swapped to satisfy this condition, eventually voiding the first condition. When both conditions are satisfied, the algorithm exits as the basis is reduced. It can be shown that Algorithm 1 terminates in polynomial time [12, p. 31].

Example A.3. To provide intuition before introducing the more abstract LLL and BKZ algorithms, we illustrate above algorithm with an example. Let

$$\boldsymbol{b}_1 = \begin{bmatrix} 101 \\ 20 \end{bmatrix}, \boldsymbol{b}_2 = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

We calculate $c = \left\lceil \frac{5 \cdot 101 + 1 \cdot 20}{101^2 + 20^2} \right\rfloor = 0$, so there is nothing to be done. However, $||\boldsymbol{b}_1|| > ||\boldsymbol{b}_2||$, so we let

$$\boldsymbol{b}_1 = \begin{bmatrix} 5 \\ 1 \end{bmatrix}, \boldsymbol{b}_2 = \begin{bmatrix} 101 \\ 20 \end{bmatrix}.$$

In the second iteration we compute $c = \left\lceil \frac{5 \cdot 101 + 1 \cdot 20}{5^2 + 1^2} \right\rfloor = 20$, so we subtract $\boldsymbol{b}_2 \leftarrow \boldsymbol{b}_2 - 20\boldsymbol{b}_1$ to get $\boldsymbol{b}_2 = [1, 0]^T$. Again, $||\boldsymbol{b}_1|| > ||\boldsymbol{b}_2||$, so we swap to get

$$\boldsymbol{b}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{b}_2 = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

In the next iteration, $c = 5$ and we get $\boldsymbol{b}_2 \leftarrow \boldsymbol{b}_2 - 5\boldsymbol{b}_1 = [1, 0]^T$. Finally, both conditions are satisfied and the algorithm terminates with

$$\boldsymbol{b}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{b}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad \bullet$$

A.3.2 LLL Reduction

An exact solver for SVP that runs in polynomial time exists only for 2 dimensions. For higher dimensions, we resort to approximately solving SVP by "straightening out" a bad basis into a good basis. A first notion of "goodness" in higher dimensions is given by Lenstra–Lenstra–Lovasz (LLL) reduction, first described by Lenstra, Lenstra and Lovasz in 1982. This subsection is based on [32] and [12].

Algorithm 2 : The LLL algorithm, as presented in [32].

Input: A lattice basis $\boldsymbol{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$
Output: A δ-LLL reduced basis
  Start: Compute $\{\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_\nu^*\}$
  Reduce step:
  for $i = 2, \ldots, \nu$ **do**
    for $j = i-1, \ldots, 1$ **do**
      $\boldsymbol{b}_i \leftarrow \boldsymbol{b}_i - c_{ij}\boldsymbol{b}_j$ where $c_{ij} = \left\lceil \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^* \rangle}{\langle \boldsymbol{b}_j^*, \boldsymbol{b}_j^* \rangle} \right\rfloor$
    end for
  end for
  Swap step:
  if $\exists i$ such that $\delta||\boldsymbol{b}_i^*||^2 > ||\boldsymbol{b}_i^*||^2 > ||\mu_{i+1,i}\boldsymbol{b}_i^* + \boldsymbol{b}_{i+1}^*||^2$ **then**
    $\boldsymbol{b}_i \leftrightarrow \boldsymbol{b}_{i+1}$
    Go to Start
  end if
  Output $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}, \boldsymbol{U}$

Definition A.3.2 (LLL reduced, as presented in [32])**.** *A basis $\boldsymbol{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$ is said to be δ-LLL reduced if*

1. $|\mu_{ij}| \leq 1/2, \quad 1 < j \leq i \leq \nu$ *and*
2. $\delta||\boldsymbol{b}_i^*||^2 \leq ||\mu_{i+1,i}\boldsymbol{b}_i^* + \boldsymbol{b}_{i+1}^*||^2, \quad 1 \leq i \leq \nu.$

The definition is valid for any $1/4 < \delta < 1$. Comparing this to Algorithm 1, one can realise that, among other things, above conditions enforce the reduction condition (Definition A.3.1) on the local diagonal blocks

$$\begin{bmatrix} ||\boldsymbol{b}_i^*|| & \mu_{i+1,i}||\boldsymbol{b}_i^*|| \\ 0 & ||\boldsymbol{b}_{i+1}^*|| \end{bmatrix}.$$

The algorithm itself can be found in Algorithm 2. In a fashion similar to Gauss–Lagrange, the algorithm proceeds by first imposing condition 1 on all applicable vectors and then checks if condition 2 is violated anywhere. If it is at position i, $\boldsymbol{b}_i$ and $\boldsymbol{b}_{i+1}$ are swapped. Then the basis does not necessarily satisfy condition 1, which is enforced again. The conditions are alternately enforced until both are satisfied. Even though it is not obvious, LLL terminates in polynomial time and produces a δ-LLL reduced basis upon termination [12, p. 36].

As we know the lattice if δ-LLL reduced if LLL terminates, one can use Definition A.3.2 to provide guarantees for the reduced basis vectors. An important such guarantee is the length of the shortest vector in the following lemma. A proof is available in [32, Lec. 2].

Lemma A.3.3. *For a δ-LLL-reduced basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$ for a lattice $\mathcal{L}$, we have that*

$$||\boldsymbol{b}_1|| \leq \left(\delta - \frac{1}{4}\right)^{-(\nu-1)/2} \lambda_1(\mathcal{L}).$$

As a corollary, for $\delta = 3/4$, we get $||\boldsymbol{b}_1|| \leq 2^{(\nu-1)/2}\lambda_1(\mathcal{L})$, meaning that $||\boldsymbol{b}_1||$ is a solution to $2^{(\nu-1)/2}$-SVP in polynomial time.

A.3.3 BKZ Reduction

We discuss BKZ in more detail, presenting an algorithm description. This subsection is based on [33] and [16]. First, recall $\boldsymbol{B}_{[i:j]}$ denotes the local projected block $[\pi_i(\boldsymbol{b}_i), \ldots, \pi_i(\boldsymbol{b}_{j-1})]$.

The definition of a BKZ reduced basis, Definition 2.3.2, is a relaxation of the following intractably strong reduction condition.

Definition A.3.4 (HKZ reduced [33])**.** *A basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$ is Hermite Korkine-Zolotarev (HKZ) reduced if*

1. $|\mu_{ij}| \leq \frac{1}{2} \quad \forall 1 \leq j \leq i \leq \nu$ *and*
2. $||\boldsymbol{b}_i^*|| = \lambda_1(\mathcal{L}(\boldsymbol{B}_{[i:\nu]}) \quad \forall 1 \leq i \leq \nu.$

In other words, $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$ is HKZ reduced if it is size reduced and every GS vector is the shortest nonzero vector in the projected lattice $\mathcal{L}(\boldsymbol{B}_{[i:\nu]})$.

This is a strong condition, and an algorithm implementing it will make ν calls to an SVP algorithm with increasing size of the SVP problem. We may relax the second condition by instead considering locally HKZ reducing the basis, in blocks of size at most β, which leads to Definition 2.3.2. Note that for block sizes $\beta = 2$ in the BKZ conditions, we retrieve the definition of an LLL reduced basis. For $\beta = \nu$, these are the HKZ conditions. As such, BKZ can be viewed as an intermediary between LLL and HKZ.

The BKZ algorithm was, until BKZ 2.0 was introduced in [33], the best known algorithm for basis reduction. BKZ 2.0 will be discussed below, but we give the original BKZ algorithm as presented by Schnorr and Euchner in 1994 [34].

The main BKZ program can be found in Algorithm 3. It takes any basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$ and preprocesses the basis by running LLL on it. BKZ then runs rounds of BKZRound, found in Algorithm 4, until the basis is BKZ reduced. In BKZRound we find calls to another algorithm called ENUM, which performs a depth-first search to solve exact SVP. This brute-force search method is called enumeration, which is why BKZ and similar algorithms are called enumeration reduction algorithms. The flag `success` keeps track whether any successful enumeration is performed in a round. If no enumeration is successful, this means all possible shortest vectors have been found and the BKZ reduction conditions are satisfied. In that case, the program terminates.

One round of BKZ involves running reductions in a sliding window of size at most β, starting at j and ending at k. When j reaches the end of the basis, this marks the end of a round. For each block, we use enumeration (denoted ENUM) to find the shortest vector in the projected lattice $\mathcal{L}(\boldsymbol{B}_{[j:k]})$. If the search space is exhausted, ENUM will return $\boldsymbol{v} = (1, 0, \ldots, 0)$, indicating that it has failed. If

Algorithm 3 : The main BKZ algorithm, as presented in [33, p. 88]

Input: A basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$, block size β.
Output: A BKZ-β reduced basis.

$\boldsymbol{B}, \boldsymbol{U} \leftarrow \text{LLL}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu)$ ▷ Save work by running LLL first. $\boldsymbol{U}$ is GSO matrix.
success $\leftarrow$ True
while success **do**
  $\boldsymbol{B}, \boldsymbol{U}, \text{success} \leftarrow \text{BKZRound}(\beta, \boldsymbol{B}, \boldsymbol{U})$
end while
Output: $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$.

ENUM is successful, we insert the new shortest vector $\sum_{i=j}^{k} v_i \boldsymbol{b}_i$ as position j and run LLL, starting from stage j, to handle linear dependencies. This generates a new basis but keeps the shortest vector at position j. If enumeration fails, we just run LLL from stage $l-1$ to prepare for enumeration in the next block.

Algorithm 4 : BKZRound, one round of BKZ, as presented in [33, p. 89]

Input: A basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$, block size β, GSO matrix U.
Output: The basis reduced by one round of BKZ-β, `success` flag for enumeration.

$j \leftarrow 0$
`success` $\leftarrow$ False
for $j \in [0, \nu - 1]$ **do**
  $k \leftarrow \min(j + \beta - 1, \nu)$
  $l \leftarrow \min(k + 1, \nu)$ ▷ j, k and l define the local block.
  $\boldsymbol{v} \leftarrow \text{ENUM}(\boldsymbol{U}_{[j:k]}, \boldsymbol{b}_j, \ldots, \boldsymbol{b}_k, \boldsymbol{b}_j^*, \ldots, \boldsymbol{b}_k^*)$ ▷ Use enumeration to find shortest vector in $\mathcal{L}(\boldsymbol{B}_{[j:k]})$. The result is $\boldsymbol{v} = (v_j, \ldots, v_k)$.
  if $\boldsymbol{v} \neq (1, 0, \ldots, 0)$ **then**
    `success` $\leftarrow$ True
    $\text{LLL}(\boldsymbol{b}_1, \ldots, \sum_{i=j}^{k} v_i \boldsymbol{b}_i, \boldsymbol{b}_j, \ldots, \boldsymbol{b}_l, \boldsymbol{U})$ from stage j
  else
    $\text{LLL}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l)$ from stage $l-1$
  end if
end for
Output: $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$, `success`

BKZ also provides guarantees for the length of the shortest basis vector. Specifically, we have that

$$||\boldsymbol{b}_1|| \leq \gamma_\beta^{\frac{\nu-1}{2(\beta-1)}+\frac{1}{2}} \text{vol}(\mathcal{L}(\boldsymbol{B}))^{1/\nu},$$

in terms of the Hermite factor

$$\gamma_\beta = \sup\{\lambda_1^2(\mathcal{L}) | \mathcal{L} \subset \mathbb{R}^\beta \text{ s.t. } \text{vol}(\mathcal{L}) = 1\}.$$

This is defined as the square of the largest norm of any dimension β lattice of volume 1.

The runtime for BKZ is mainly dominated by calls to ENUM. As discussed before, LLL terminates in polynomial time, while enumeration has a super-exponential runtime in β, and so will BKZ.

A.3.4 BKZ 2.0 and Progressive BKZ

The original BKZ as proposed by Schnorr and Euchner [34] is not used in practice. The most recent enumeration reduction libraries use improvements described in [35], collectively referred to as BKZ 2.0, and [36], referred to as Progressive BKZ. We highlight the main improvements over original BKZ, as Progressive BKZ is implemented into `fplll`, the lattice reduction library used for experiments in this thesis. This part is based on [37].

For BKZ 2.0, the authors suggested the following improvements.

- Aborting BKZ after a fixed number of rounds. It was observed that most of the improvement to the basis occurs in the first few rounds. Aborting limits the number of possible enumeration calls, and thus provides an exponential speed-up to BKZ.
- The enumeration routine ENUM uses pruning to discard branches of the search tree before searching them if they are unlikely to contain a short vector. BKZ 2.0 optimises this pruning strategy. This version is called extreme pruning.
- In classic BKZ the basis is preprocessed using LLL. BKZ 2.0 preprocesses the basis by instead running BKZ with a smaller block size.

In progressive BKZ, further improvements were made.

- Picking optimal parameters for enumeration radius and success probability as functions of block size, current block, dimension and a simulated cost of enumeration. The enumeration radius is the target length of a short vector in the enumeration algorithm.
- Starting from a small block size, gradually increasing it each round until certain conditions are met. One may then consider what the optimal increase between each round should be.
- Applying extreme pruning to the entire basis and not just a local block.

Lattice reduction is an area of active research, and many more variants exist than the ones mentioned here. Similar in design to BKZ, the reduction algorithm with the best worst case guarantees is Slide reduction and is also implemented into `fplll`. There is another class of reduction algorithms based on removing long vectors instead of finding the shortest vector by enumeration. This method is called sieving, with the main library being `g6k`.

A.4 Additional LWE Variants

In this section, we present the two variants of LWE that were not discussed in the main text as they can be recovered as special cases of MLWE. This exposition is mainly for completeness and historical context.

A.4.1 Unstructured LWE

Unstructured LWE, commonly simply called LWE, was first presented in [3] as the first variant. Here, χ is a probability distribution over $\mathbb{Z}_q$ with variance σ^2. The most common choice is a discrete Gaussian.

Definition A.4.1 (LWE distribution)**.** *The discrete LWE distribution $A_{\boldsymbol{s},\chi}$, given a secret $\boldsymbol{s} \in \mathbb{Z}_q^d$, is the one obtained by sampling $\boldsymbol{a} \in \mathbb{Z}_q^d$ uniformly at random, $e \in \mathbb{Z}_q$ from χ and computing $(\boldsymbol{a}, b = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e \mod q) \in \mathbb{Z}_q^d \times \mathbb{Z}_q$. A pair $(\boldsymbol{a}, b)$ created this way is called and LWE sample.*

Remark. *There also exists a continuous LWE distribution which is often used due to its similarity to RLWE. Regev showed in [3] that the discrete and continuous versions are equivalent, so we only proceed with the discrete version to avoid confusion.*

From this, the computational problem itself can be introduced.

Definition A.4.2 (Search LWE)**.** *The search-LWE$_{d,q,\chi}$ problem is to recover $\boldsymbol{s} \in \mathbb{Z}_q^d$ given many independent samples on the form*

$$(\boldsymbol{a}_i, b_i = \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + e_i \mod q),$$

from the LWE distribution defined as above.

There is also a decision variant where the problem is to distinguish between LWE and uniform samples with a non-negligible advantage. There exist reductions between Search LWE and Decision LWE. For the purposes of this text, they can be viewed as equivalent problems. We will mainly be concerned with Search LWE.

Multiple LWE samples are usually collected in matrix form. If we consider m LWE samples, they can be collected as

$$\boldsymbol{A} = [\boldsymbol{a}_1 | \boldsymbol{a}_2 | \dots | \boldsymbol{a}_m]^T \in \mathbb{Z}_q^{m \times d}, \quad \boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} \in \mathbb{Z}_q^m.$$

Here, $\boldsymbol{A}$ is uniformly random and $\boldsymbol{e} \in \mathbb{Z}_q^m$ is drawn from χ^m. Solving $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$ can be viewed as finding a solution to a perturbed version of the equation $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{s}$. If the error $\boldsymbol{e}$ were $\boldsymbol{0}$, we would be able to find an exact solution (or the system would be overdetermined). If we define the lattice

$$\mathcal{L}(\boldsymbol{A}) = \{\boldsymbol{z} \in \mathbb{Z}^m : \exists \boldsymbol{s} \in \mathbb{Z}_q^d \text{ s.t. } \boldsymbol{z} = \boldsymbol{A}\boldsymbol{s} \mod q\},$$

the LWE problem can be viewed as being provided a lattice basis $\boldsymbol{A}$ and a perturbed vector $\boldsymbol{b}$ and searching for the closest vector $\boldsymbol{s}$, i.e. an instance of BDD.

Using Kannan embedding, this can be embedded into a lattice basis where the problem instead is to solve SVP. First, consider that the system $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} \mod q$ can equivalently be written

$$\begin{bmatrix} q\boldsymbol{I}_m & -\boldsymbol{A} \\ 0 & \boldsymbol{I}_d \end{bmatrix} \begin{bmatrix} \_ \\ \boldsymbol{s} \end{bmatrix} + \begin{bmatrix} \boldsymbol{b} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{s} \end{bmatrix},$$

where _ denotes an arbitrary vector which is reduced away modulo q. Rewriting this as a homogeneous system, we get

$$\underbrace{\begin{bmatrix} q\boldsymbol{I}_m & -\boldsymbol{A} & \boldsymbol{b} \\ 0 & \boldsymbol{I}_d & 0 \\ 0 & 0 & t \end{bmatrix}}_{\boldsymbol{C}} \begin{bmatrix} \_ \\ \boldsymbol{s} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{s} \\ t \end{bmatrix}. \tag{A.2}$$

Here, t is a free parameter, usually set to 1 [38]. Notably, this embedding means $(\boldsymbol{e}, \boldsymbol{s}, t)^T$ is a lattice vector. If σ^2 and t are selected small, this vector is short in the lattice spanned by $\boldsymbol{C}$. This is equivalent to discovering $\boldsymbol{s}$, meaning we solve LWE if we solve SVP.

A.4.2 Ring Learning With Errors

Ring Learning With Errors (abbreviated ring-LWE or RLWE) uses the same idea as unstructured LWE but lifts into a setting with algebraic structure. At the expense of the theory becoming more convoluted, this allows for smaller memory usage and more efficient computation due to the Number Theoretic Transform (NTT), a variant of the Fast Fourier Transform.

We parametrise RLWE by the ring of integers $R = \mathcal{O}_K$ of a number field K.The usual choice is the ring $R = \mathbb{Z}[X]/(p)$ with $p = x^n + 1$ with n a power of two. In this setting, NTT-based computation techniques provide significant speed-up. We instead use $p = x^n - 1$ where applicable.

As with LWE, there exists a continuous variant of RLWE. For the discrete Gaussian error distribution, it is shown in [39] that the discrete version presented below follows from the continuous version. We only present the discrete version for simplicity. Furthermore, general RLWE is defined in terms of the dual ideal $R^\vee$ and called dual RLWE. It is shown in [40] that primal RLWE as defined below is equivalent to dual RLWE up to a scaling factor if the error distribution is Gaussian.

Definition A.4.3 (RLWE distribution)**.** *The primal discrete RLWE distribution $A_{s,\chi}$ is, given a secret $s \in R_q$ and a distribution χ over R, the one given by sampling $a \in R_q$ uniformly at random, $e \in R$ from χ and computing $(a, b = a \cdot s + e \mod qR) \in R_q \times R_q$. The pair (a, b) is called an RLWE sample.*

Using this distribution, we can state the search problem for RLWE. The decision variant is similar to that of unstructured LWE, mutatis mutandis.

Definition A.4.4 (Search RLWE)**.** *The search-RLWE$_{q,\chi}$ problem is to recover $s \in R_q$ given many independent RLWE samples $(a_i, b_i = s \cdot a_i + e_i \mod qR)$.*

Search-RLWE, like search-LWE, can be interpreted as a BDD problem over the ideal lattice generated by R_q. Just as in unstructured LWE we can transform a BDD instance to an instance of SVP using Kannan's embedding. To embed m RLWE samples, recall that each polynomial a_i, b_i, e_i and s can be represented as an $n \times n$ matrix. As we have m polynomials, we stack these matrices on each other. Replacing all vectors in the LWE case with these matrices, we get the matrix equation

$$\underbrace{\begin{bmatrix} q\boldsymbol{I}_{mn} & -\boldsymbol{A} & \boldsymbol{B} \\ 0 & \boldsymbol{I}_n & 0 \\ 0 & 0 & t\boldsymbol{I}_n \end{bmatrix}}_{\boldsymbol{C}_R} \begin{bmatrix} \_ \\ \boldsymbol{S} \\ \boldsymbol{I}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} \\ \boldsymbol{S} \\ t\boldsymbol{I}_n \end{bmatrix}, \tag{A.3}$$

where $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{E}$ all are $mn \times n$ and $\boldsymbol{S}$ is $n \times n$. $\boldsymbol{C}_R$ will then be a square matrix of size $(mn + 2n) \times (mn + 2n)$. We see that instead of one short vector, this embedding admits a dense sublattice of rank n in the $mn + 2n$ dimensional lattice.

B

Additional Results for NTWE

This section contains results from the experiments described in Section 5.3. For each n, we present the following types of plots: First, the successful block sizes for each given q with the outcome of each experiment (grey plus signs and crosses), the average β (squares) and the ratio of DSD (colour of the squares) are found in Figures B.1–B.4. Note the horizontal scaling. Second, moving averages of the DSD ratio for each given q can be found in Figure B.5. In both instances, the computed fatigue point is shown as a green circle.

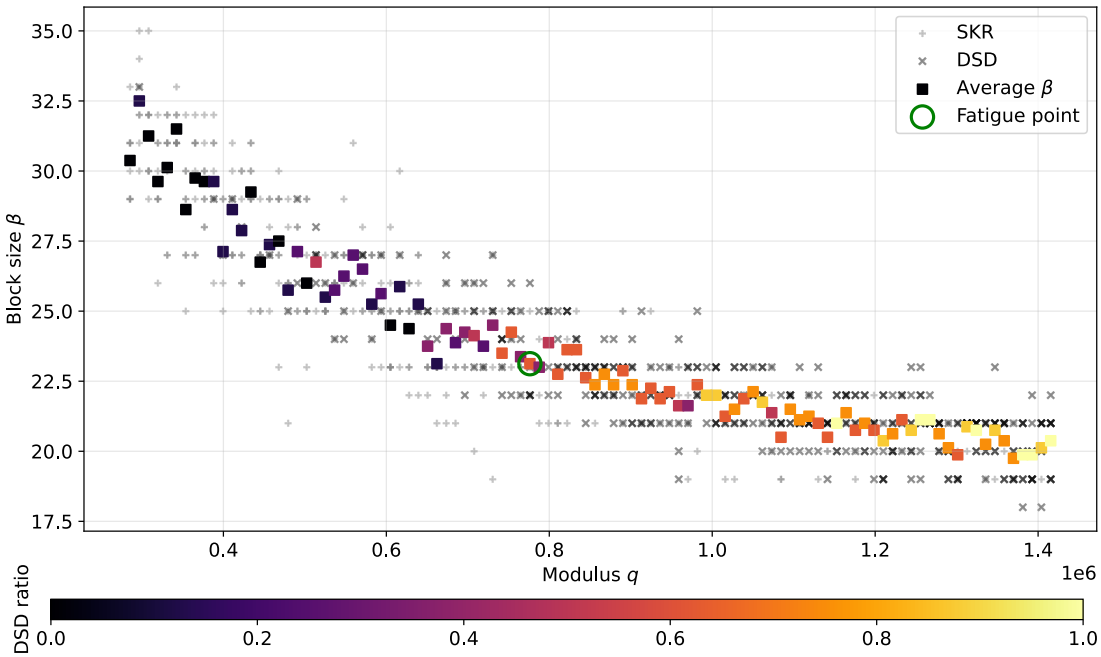


Figure B.1: Successful block sizes for $n = 97$.

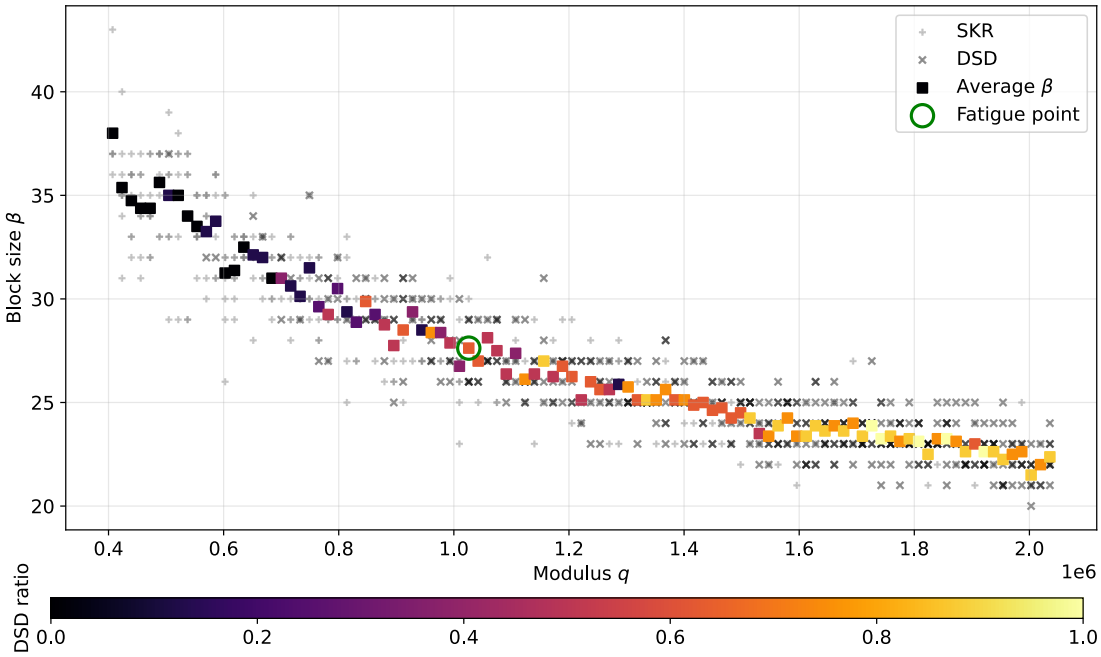


Figure B.2: Successful block sizes for $n = 103$.

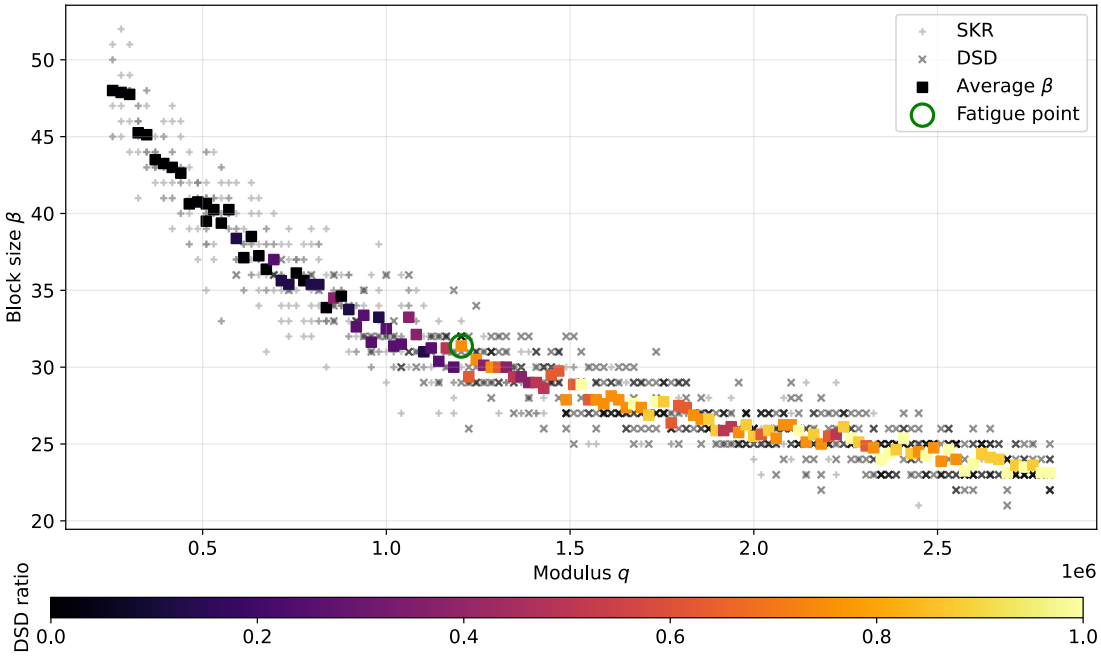


Figure B.3: Successful block sizes for $n = 107$.

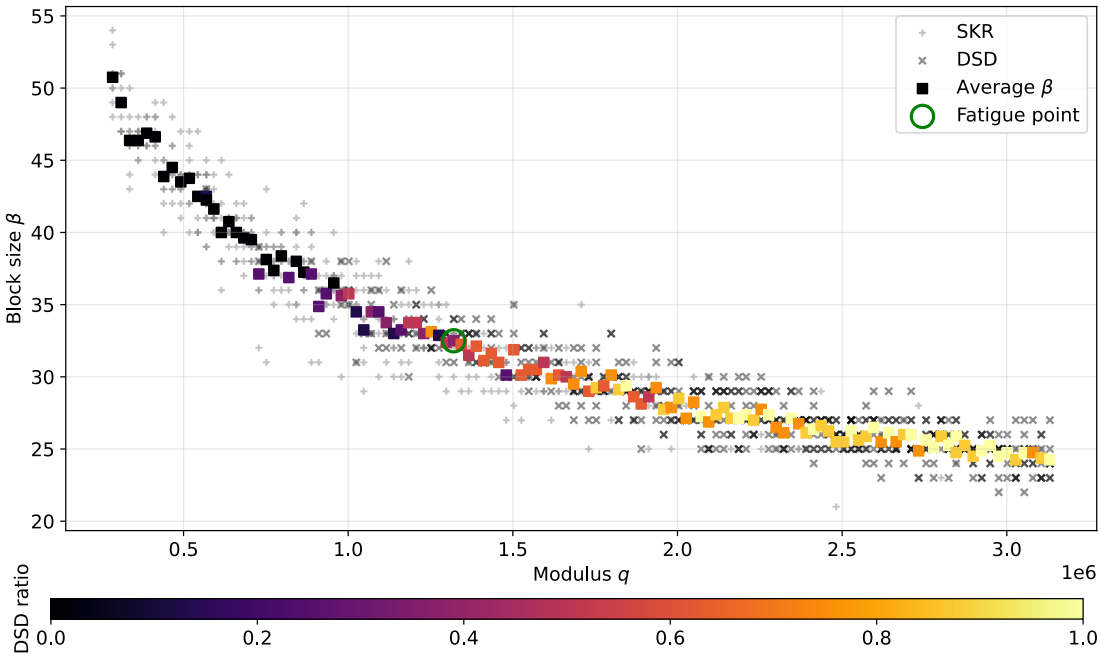
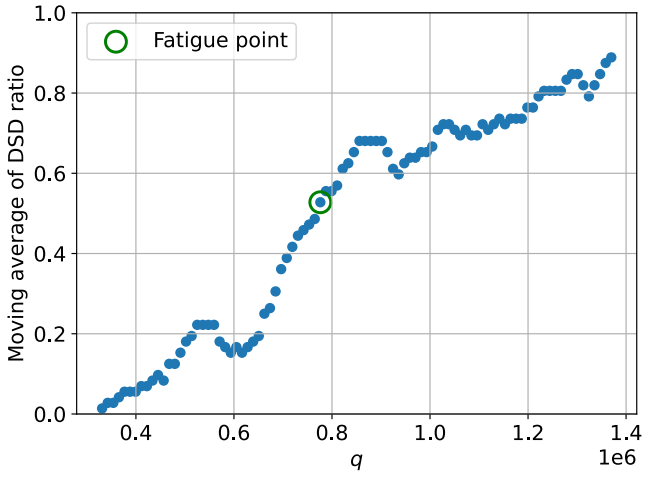
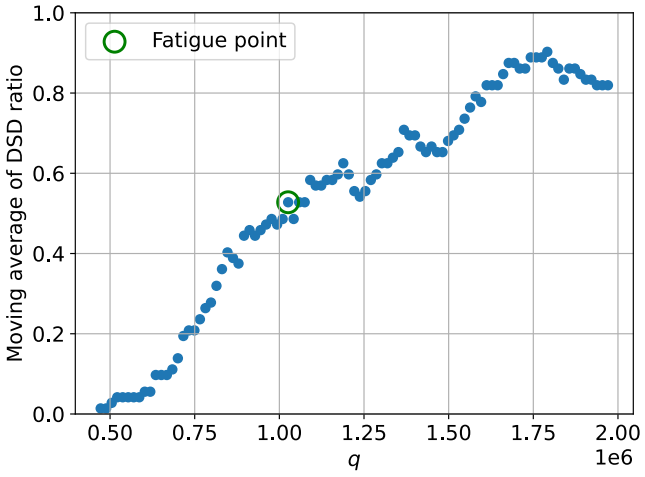


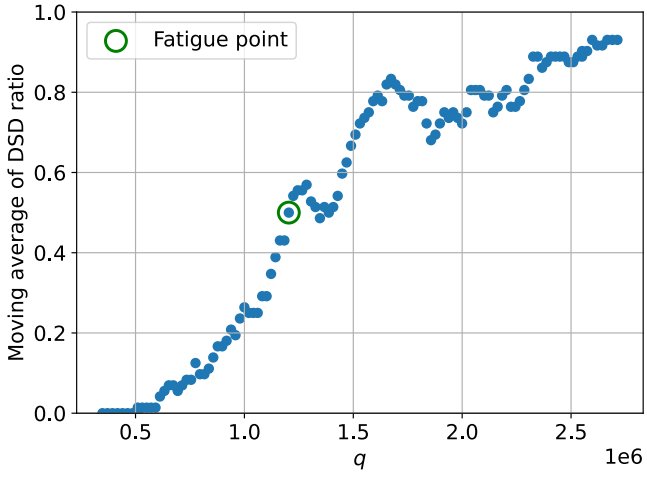
Figure B.4: Successful block sizes for $n = 109$.



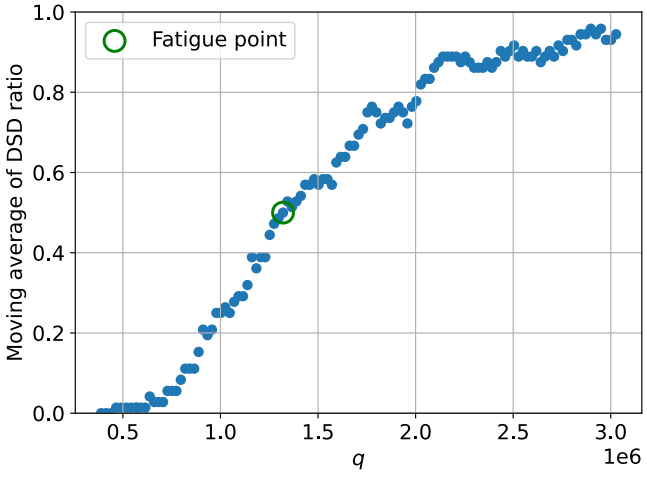
(a) $n = 97$.



(b) $n = 103$.



(c) $n = 107$.



(d) $n = 109$.

Figure B.5: Moving averages of the DSD ratio.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COMPUTING SCIENCE
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY