On the Tightness Gap Analysis of Reductions of some Lattice problems to the Learning with Error problem

A thesis submitted to Indian Statistical Institute
in partial fulfillment of the thesis requirements for the degree of
Doctor of Philosophy in Computer Science

Author:
Subhadip Singha
subha_r@isical.ac.in

Supervisor:
Prof. Palash Sarkar
palash@isical.ac.in

Applied Statistics Unit
Indian Statistical Institute
203, B. T. Road, Kolkata,
West Bengal, India - 700 108.

Submitted on March, 2023

To Maaa, Baba and Trisha.

Acknowledgement

I would like to extend my sincere apologies to those individuals whose names may not be mentioned here, but who have provided valuable support and cooperation, contributing to the successful completion of my journey. Your contributions are deeply appreciated.

I am filled with joy as I express my heartfelt gratitude to Professor Palash Sarkar, my supervisor, whose unwavering support and encouragement throughout my Ph.D. program have been invaluable. I am truly grateful for the opportunity to work under his guidance, as it has been instrumental in achieving this significant milestone.

I would like to convey my deepest thanks to all the faculty members and non-teaching staff at ISI, Kolkata, for their exceptional cooperation throughout my journey. Their support has been invaluable. Special mention goes to the ISI Medical unit for their generous efforts in ensuring my physical and mental well-being.

My sincere gratitude goes to the Applied Statistics Unit at the Indian Statistical Institute, Kolkata, for providing me with the necessary research facilities.

I would like to express my heartfelt acknowledgment to our esteemed collaborator, Professor Neal Koblitz, whose exceptional work ethic and profound contributions to research have been an inspiration to me as a young researcher.

A special thanks to Dr. Subhabrata Samajdar (Butu-da) for his persistent support and assistance.

My beloved labmates, including Anirudhha, Koushik-da, Madhurima, Sebati-di, Amit-da, and Sreyosi, have been a constant source of support, and I am deeply grateful for their continuous encouragement.

I would like to thank Diptendu, Animesh, Susanta, Amit, Jyotinmoy, Arnab, Ayan, Laltu, Avishek, Samir, Mostafizar, Soumya, Chandranan, Rakesh, Anirban-da, and other juniors and friends who have made my long journey happy and fulfilling.

Dr. Sanjay Bhattacherjee (Sanjay-da) deserves my gratitude for his encouragement and continuous support.

Dr. Rishiraj Bhattacharyya (Rishi-da) has my deepest appreciation for his invaluable guidance during my master's thesis. Our insightful discussions provided me with the determination and motivation to embark on this research journey.

My family's presence and support have been integral to my journey, and I am immensely grateful. I extend my deepest appreciation to my parents, especially my mother, for their selflessness, unconditional love, and inexhaustible support. Their patience and guidance have been invaluable. I consider myself fortunate to have such supportive parents who have

instilled in me the values of being a good human being above all else. Additionally, I am thankful to my elder sisters, brothers-in-law, and nieces for their constant support.

A special thanks to my wife, Dr. Trisha Das, for her tireless support and boundless enthusiasm, which greatly contributed to making my journey smoother and more joyful. I am also immensely grateful to all my in-laws for their invaluable support throughout.

I would like to express my gratitude to my relatives and teachers whose blessings and good wishes have been instrumental in reaching this point.

Finally, I want to convey my deep affection for Foring and Bhutu, the beloved members of my extended cat family. These adorable feline companions have consistently brought joy and comfort to my life, particularly during challenging times. Their presence has been a constant source of positive energy.

Abstract

ON THE CONCRETE SECURITY OF LATTICE-BASED REDUCTIONS TO LWE

by **Subhadip Singha**

Lattice-based cryptography is a highly regarded contender for post-quantum standardization by NIST. NIST has already chosen "CRYSTALS-KYBER" a lattice-based public-key encryption and key-establishment algorithm and "CRYSTALS-DILITHIUM", a lattice-based digital signature algorithm. The current lattice-based schemes are based on Oded Regev's original construction, which sparked significant interest in the cryptographic community due to its post-quantum security and the equivalence between worst-case and average-case hardness.

Oded Regev's cryptographic scheme is built upon a problem called "Learning with Error" (LWE), which is a generalization of the "Learning Parity with Noise" (LPN) problem. This scheme is straightforward to implement and has gained attention for its simplicity. Previously, Mikolas Ajtai demonstrated the worst-case to average-case equivalence for a set of hard lattice problems. Regev's seminal paper demonstrated that the security of LWE-based cryptosystems could rely on the hardness of worst-case lattice problems. While this result is theoretically groundbreaking, the reduction from hard lattice problems to LWE is not tightly bound, which limits its practical applicability.

The tightness of a reduction is a critical factor often underestimated. The tightness gap of a reduction quantifies the concreteness of the reduction, and a tight reduction is valuable for translating theoretical hardness guarantees into practical scenarios. Cryptography, as a field, prioritizes practical applicability. Non-tight reductions lead to less efficient systems but they have practical applications. Regev's work has prompted numerous follow-up studies. One significant effort aimed to make the reduction classical, as the original reduction was quantum-based. Additionally, subsequent research has focused on enhancing the efficiency of LWE-based cryptosystems by utilizing various algebraic variants of lattices, such as ideal and module lattices.

We thoroughly investigate these major reductions to unveil their true significance in terms of reduction tightness. Furthermore, we conduct a concrete security analysis of these reductions and identify several areas for improvement.

List of Tables

List of Figures

Contents

Notation

$\mathbb{R}, \mathbb{Q}, \mathbb{Z}, \mathbb{C}, \mathbb{N}$	:	Reals, Rationals, Integers, Complex Numbers, Natural Numbers
H	:	Defined in 2.4.3 , inner product space isomorphic to $\mathbb{R}^n$
$\log x$	:	logarithm of x to the base 2
$\ln x$	:	natural logarithm of x
Λ, L	:	a lattice in H
$\lambda_1(\Lambda)$	:	minimum distance of the lattice Λ
$\lambda_n(\Lambda)$	:	the least real number such that Λ has n linearly independent vectors with the length of the longest being equal to this number
$\eta_\epsilon(\Lambda)$	:	smoothing parameter for a lattice Λ
K	:	underlying number field
M	:	Module over number field K
n	:	degree of the number field K, dimension of lattice Λ
$\mathcal{O}_K, R$	:	ring of integers of K
σ	:	canonical embedding of a number field into H
$\mathcal{I}^*$	:	the dual of the lattice $\sigma(\mathcal{I})$
$\mathcal{I}^\vee$	:	the conjugate dual of the lattice $\sigma(\mathcal{I})$
LWE	:	Learning with Error
SVP	:	Shortest Vector Problem
SIVP	:	Shortest Independent Vector Problem
GIVP	:	Generalized Independent Vector Problem
CVP	:	Closest Vector Problem
BDD	:	Bounded Distance Decoding Problem
GDP	:	Gaussian Decoding Problem
$\mathbf{a}, \mathbf{s}$	:	Vectors are written in bold letters

Chapter 1

Introduction

Effective and secure communication across different geographical locations has been a challenging endeavor throughout history. However, with the advent of the digital age, significant advancements have been made in this field. Achieving effective communication involves ensuring that messages reach their intended recipients accurately. On the other hand, the security aspect of communication focuses on guaranteeing that only the intended recipients can access and comprehend the messages.

In addition to effectiveness and security, the speed of communication has become a critical factor in the digital era. Faster communication is essential in various domains, such as streaming services, high-resolution video calling, online gaming, and more. Enormous amounts of data are transmitted over the internet in a matter of milliseconds.

Unfortunately, the importance of secure communication is often underestimated or overlooked by a large portion of users. However, cryptography plays a crucial role in addressing the challenges of digital communication, especially in ensuring confidentiality, integrity, and authenticity of data transmission.

In simple terms, cryptography is the science of ensuring secure communication. The word "cryptography" originates from two Greek words: "krypto's," meaning "hidden" or "secret," and "graphien," meaning "to write." Although the concept of cryptography has existed in society for a long time, we often do not explicitly acknowledge it.

Secure communication between two parties, such as a sender and a receiver, can be achieved through the use of a shared piece of information known only to them, known as the secret key. The sender encrypts a message or plaintext using an encryption function that takes two input parameters: the message and the secret key. The result of this process is known as a "ciphertext." The receiver then decrypts the ciphertext using a decryption function that takes the ciphertext and the shared secret key as inputs and produces the original plaintext.

To maintain the integrity of the communication, the decryption process must yield the same message as the original plaintext. It is assumed that the encryption and decryption functions are publicly available, while the secret key is known exclusively to the communi-

cating parties.

In this context, an adversary is someone who seeks to compromise the security of the communication by attempting to discover the secret key or intercept the message as it is transmitted through publicly accessible communication channels. The use of cryptography helps protect against such unauthorized access and ensures the confidentiality and authenticity of the communication between the parties involved.

Cryptography plays a vital role in ensuring various aspects of digital communication, including confidentiality, integrity, and authenticity.

Confidentiality in communication refers to the assurance that only authorized individuals can comprehend and access the secret message being communicated. It is essential to prevent adversaries from gaining any knowledge about the confidential message.

Data integrity ensures that the communicated message remains unchanged and unaltered during transmission. The recipient should be able to verify that the received message is identical to the one sent, without any unauthorized modifications.

Data authenticity is the assurance that the receiver of the message can verify the identity of the sender. This ensures that the message originates from the claimed source and has not been forged or tampered with by unauthorized parties.

In conventional cryptography, also known as symmetric key cryptography, the sender and receiver share a common secret, referred to as the secret key. This shared secret key is used for secure communication between them. Examples of symmetric key cryptography include widely used algorithms like AES [DR02], DES, RC4, Salsa20 [Ber08] etc.

By addressing these aspects of digital communication, cryptography provides a robust and secure framework for exchanging information while safeguarding the confidentiality, integrity, and authenticity of the transmitted data.

Asymmetric key cryptography, also known as public key cryptography, emerged after the development of symmetric key cryptography. In this cryptographic approach, the communicating parties do not share any secret data before communication begins. Each party possesses a pair of keys: a public key, which is publicly known, and a private key, which is kept confidential.

The generation of these key pairs relies on cryptographic algorithms that are based on mathematical problems referred to as “Trapdoor” functions. A Trapdoor function is a special type of “One-Way Function”, meaning it is easy to compute in one direction but challenging

to invert without specific information known as the "trapdoor".

To create a public key, the secret key is input into the trapdoor function. However, given only the public key and the trapdoor function, it is computationally infeasible to deduce the secret key. The sender encrypts the message using the recipient's public key, and the receiver decrypts the message using their private key.

The Diffie-Hellman protocol [DH76], introduced by Diffie and Hellman in 1976, is one of the oldest known cryptographic schemes based on public key cryptography. This protocol marked a significant breakthrough in the field of cryptography as it enabled secure communication between parties without the need for a shared secret key before initiating communication. During the key exchange process, two parties communicate over a public channel using their secret key and the other party's public key. Through this exchange, they establish a shared secret without having access to each other's secret keys.

The key exchange protocol in public key cryptography allows for secure communication between parties even when they have not previously shared any secret information. It has since become a fundamental component in modern cryptographic systems, providing a powerful and secure means of exchanging information over public channels.

A key notion that is closely connected to public key cryptography is "provable security". A cryptographic scheme is provably secure if the security of the scheme can be proved by a mathematical 'proof' or a 'reduction'. In a proof or a reduction, the adversarial capabilities of the attacker are defined by an adversarial model. The proof aims to show that the attacker must solve the underlying computational hard problem to breach the security of the cryptographic scheme. In the case of the Diffie-Hellman protocol, the hard problem is the computational Diffie-Hellman (CDH) problem. Another celebrated public key cryptosystem, "RSA" [RSA78], is based on the RSA assumption. These are number-theoretic problems that are assumed to be hard to solve for any adversary where the power of the adversary is computationally bounded. One distinct thing to notice is that the CDH problem and RSA assumption are hard to solve on classical computers. Implicitly, we assume that the adversary has access to only classical computers.

In 1996, Shor [Sho97] presented a polynomial-time algorithm for prime factorization and discrete logarithms on a quantum computer. These breakthroughs theoretically pose a significant threat to cryptographic systems like Diffie-Hellman and RSA. However, in practice, breaking these schemes with practical parameters requires a high-scale quantum computer, which has not been realized as of now. The development of a practical high-scale quantum computer is an ongoing challenge. The emergence of quantum computing has highlighted the

need to develop cryptographic algorithms that can withstand attacks from quantum computers. This field of study is known as post-quantum cryptography, where cryptographic protocols are designed to be quantum-safe.

Over the past two decades, cryptographic research has yielded promising results in the realm of post-quantum cryptography. Various candidates, such as code-based cryptography, lattice-based cryptography, and multivariate cryptography, have been proposed and are believed to be resistant to attacks from quantum computers.

In this thesis, the focus is on lattice-based cryptography, which is considered a strong candidate for post-quantum secure communication. By exploring lattice-based cryptographic schemes, this research aims to contribute to the development of quantum-safe cryptographic protocols to safeguard communications in the presence of powerful quantum computers.

In a seminal work, Regev [Reg09] introduced the Learning with Errors (LWE) problem and emphasized its significance in lattice-based cryptography. Informally, the LWE problem involves solving a system of linear equations with errors. Solving a system of linear equations is a relatively straightforward task, and the well-known Gaussian elimination method can achieve it in polynomial time relative to the dimension of the system. However, when errors are introduced in each equation, the problem becomes significantly harder.

A simplified version of the LWE problem is known as "Learning Parity with Noise" (LPN), which operates with a modulus of '2', unlike LWE, which can have any modulus. Naively solving the LPN problem using Gaussian elimination would require exponential time. However, Blum, Kalai, and Wasserman [BKW03] devised a more efficient algorithm that requires sub-exponential time relative to the dimension.

LWE-based cryptographic systems are readily implemented using the parameters of LWE, such as the dimension of LWE and the error parameter associated with it. The breaking of an LWE-based crypto-system translates into solving hard lattice problems through a series of reductions. To have high confidence in the security of an LWE-based cryptosystem, a tight reduction to a hard lattice problem is essential.

The "tightness" of a reduction refers to how closely the reduction preserves the security properties of the original cryptographic problem. In the context of cryptography, reductions are used to establish the hardness of a new problem by showing that breaking the new problem would allow an attacker to break an existing well-studied problem. If the reduction is "tight", it means that the hardness of the new problem is essentially equivalent to the hardness of the original problem, providing a strong security guarantee. On the other hand, if the

reduction is "loose", it means that the new problem's hardness is not as closely related to the original problem's hardness, leaving potential vulnerabilities and weaker security guarantees. A loose reduction may lead to overestimating the security level of the new problem or failing to fully understand its cryptographic implications. In summary, a tight reduction ensures that the security properties of a new cryptographic problem are well-founded and directly linked to the known hardness of an established problem, while a loose reduction may introduce uncertainties and weaken the overall security analysis. By exploring the LWE problem and its connections to lattice-based cryptography, researchers aim to develop robust and secure cryptographic protocols that can withstand various attacks and ensure the confidentiality and integrity of transmitted data.

Suppose we have a polynomial-time reduction from problem $\mathcal{P}$ to another problem $\mathcal{Q}$. Additionally, assume that we have an oracle $\mathcal{O}$ capable of solving problem $\mathcal{Q}$ in time T_1. The tightness gap of the reduction is denoted by G, and it signifies the relation between solving $\mathcal{P}$ and $\mathcal{Q}$ through the reduction, with the time taken estimated as $G{\cdot}T_1$. Now, let's consider the fastest known algorithm to solve problem $\mathcal{P}$, which requires time T_2 for the chosen parameters. The assumption is that the algorithm to solve $\mathcal{P}$ through the reduction will not surpass the efficiency of the known fastest algorithm for $\mathcal{P}$. Due to this assumption, we can derive the implicit relationship $T_1 \geq T_2/G$. By leveraging this relation, we can carefully select appropriate parameters such that $T_1 \geq 2^{128}$ is satisfied. Since problem $\mathcal{P}$ is a well-studied hard lattice problem, the time complexity T_2 for solving it with our chosen parameters is exponential. As a result, we can deduce that $T_2/G \geq 2^{128}$ when G is reasonably small or the reduction is tight. Consequently, this implies that $T_1 \geq 2^{128}$. This mathematical relationship places constraints on the lower bound of the algorithm's time complexity that attempts to solve $\mathcal{Q}$. It effectively translates the hardness of problem $\mathcal{P}$ to problem $\mathcal{Q}$ in a concrete manner.

In this thesis, we extensively discuss various lattice-based reductions, including those in [Reg09, BLP+13, LPR13, LS15, PRS17]. Despite their differences, all these reductions share a common structure characterized by a nested sequence of intermediate reductions. Throughout this chain of reductions, we observe that the tightness gaps multiply from one reduction to the next. When an algorithm A calls on algorithm B m times, and B in turn calls on algorithm C n times, there are mn calls on algorithm C. As a result, the cumulative tightness gaps in the reductions become significant, making the security guarantees practically meaningless for the chosen parameter values. This observation underscores the critical importance of carefully addressing and analyzing the tightness gaps at each step of the re-

duction chain. Without mitigating these issues, the overall security of the cryptographic scheme may be compromised, especially when considering practical parameter settings.

The importance of LWE in the context of lattice-based cryptography is underscored by the fact that several submissions made to the ongoing NIST process for selecting a new public key standard based their security on the LWE problem and several of its variants. Notable LWE-based proposals which are in various rounds of the NIST process are Frodo [ABD+19], Kyber [ABD+09], LAC [LLJ+19], NewHope [AAB+19], Round5 [BBF+19a] and Saber [DKRV19]. After the third round, NIST selected Kyber [ABD+09] as the finalist from the LWE-based cryptographic encryption category. Bernstein [Ber19] performed a comparative study of the provable security of these and other lattice-based proposals. For practical interest, we thoroughly investigate the aspect of concrete security for the LWE-based cryptosystems and related reductions in this thesis.

1.1 Overview of the Thesis

This thesis is focused on lattice-based cryptography, which is recognized as one of the primary candidates for post-quantum cryptography. The structure of the thesis is briefly outlined as follows. In Chapter 2, we set the notation and describe other prerequisite materials required for the rest of the thesis. Chapter 3 consists of a brief survey of the relevant works present in the literature.

Chapter 4 is the first contributory chapter. As mentioned earlier, Regev [Reg09] introduced the LWE problem and showed a reduction from a worst-case lattice problem to the breaking of a cryptosystem. This worst-case to average-case reduction has been later claimed to be a major theoretical advantage of cryptosystems based on lattices. The entire analysis in [Reg09] was done in an asymptotic setting where the lattice dimension n is allowed to go to infinity.

A later work [CKMS16] performed a concrete analysis of the reductions in [Reg09]. This exercise determined the tightness gap of the reduction in concrete terms as a function of n, the dimension of the lattice. It turned out that the tightness gap of reductions in [Reg09] can indeed be very large. For example, for $n = 1024$, it was argued that the tightness gap is about 2^{504} and so the worst-case to average-case reduction in [Reg09] cannot be used to argue about the security of cryptosystems with lattice dimension $n = 1024$.

The reduction in [Reg09] is a cascade of three smaller reductions. The first reduction is

from the Smallest Independent Vector Problem (SIVP) to the problem of Discrete Gaussian Sampling (DGS). The second reduction is from DGS to (search) LWE, while the third reduction is from search-LWE to average case decisional LWE (DLWE_{ac}). There is a further reduction from DLWE_{ac} to that of breaking the cryptosystem.

The second reduction, i.e., the one from DGS to LWE, is the main contribution of [Reg09]. A key step in this reduction consists of verifying solutions to LWE. This verification is done using a statistical test. It has been proved in [Reg09] that asymptotically the success probability of the statistical test is exponentially close to one. The statistical test is used many times in the entire reduction, and the success probability of the statistical test determines the overall success probability of the complete reduction. We take a close look at the success probability of the statistical test. Using the standard Hoeffding inequality, we determine an upper bound on the error of the statistical test. This in turn leads to a lower bound on the success probability of the test and then to a lower bound on the success probability of the entire reduction.

In Chapter 5, we focus our discussion on the module and ideal lattices and correspondingly on module-LWE and ring-LWE. Regev's discovery opened a new direction in the field of lattice-based cryptography. Many cryptographic applications were built based on this hardness assumption. These assumed to be quantum-safe cryptographic applications are inefficient to be used for practical purposes due to the size of the keys in these applications. This problem has been answered positively by Lyubashevsky et al. [LPR13] by using an algebraic variant of LWE and naming it ring-LWE. Many cryptosystems have based their security on the hardness of variants of the ring-LWE problem, namely module-LWE. Reductions in Lyubashevsky et al. [LPR13] are significant, concerning the practical implementation of lattice-based cryptosystems but the results presented in it are also asymptotic. As previously mentioned, asymptotic results may not always reflect practical scenarios when certain parameters are constrained. The concrete tightness gap analysis of the ring variant of LWE [LPR13] has been thoroughly investigated in [KSSS22].

Based on the results of [Reg09], [LPR13] and [BLP+13] Langlois and Stehlé proposed [LS15] which focuses on the worst-case to average-case reductions for module lattices. Langlois and Stehlé [LS15] aimed to achieve an optimal solution by combining the strengths of both general lattices, based on hardness assumption [Reg09], and ideal lattices, known for their implementation efficiency [LPR13]. Results presented in [LS15] is the stepping stone for the LWE-based crypto-systems, selected by NIST as the candidates for the post-quantum cryptosystem. That is why it is necessary to ask about the tightness of the reductions

in [LS15] to be used in practice.

Our purpose is to analyze this reduction in concrete terms. Previous analysis [CKMS16, SS21] shows that Regev's [Reg09] results were not tight enough for practical purposes. The reductions in [LS15] suffer from a significant tightness gap, making them unsuitable for practical purposes. This concern has been addressed and commented upon in [KSSS22]. We thoroughly analyze the issue related to the tightness gap of [LPR13] and [LS15] in this chapter. We also comment on the lower bound of values of the approximation factor γ of SIVP and also on the value of the modulus of the ring-LWE and module-LWE problems.

Chapter 6 completes our discussion on ring LWE and ideal lattices. In Chapter 5, we point out various concrete security issues in the reductions of [LPR13] other than the tightness gap. The main problem is the use of the cyclotomic number field. The motivation behind the choice of a cyclotomic number field is efficient computations. The cyclotomic number fields provide automorphisms between different embeddings of the number field, which is used for the security reductions in [LPR13]. On the other hand, the hardness of SVP_γ and SIVP_γ are equivalent in cyclotomic settings for ideal and module lattices. The hardness of SIVP_γ for general lattices is strictly greater than the hardness of SIVP_γ for ideal lattices. So, efficiency is given more importance over security. That may not be a popular school of thought in cryptography. Also, cyclotomic rings are a very narrow class of rings, and these rings are distributed very sparsely when we consider the set of all number fields. Cryptographic applications may need a general class of rings. Peikert, Regev, and Stephens-Davidowtz [PRS17] positively answer this problem by providing a reduction for any rings and modulus. As we work out the reductions to find out the tightness gap of the end-to-end reduction, we find that this reduction is too loose to be useful for practical purposes. The lower bound of approximation factor γ of ring-SIVP problem is same as in Chapter 5

Chapter 7 is dedicated to the concrete security analysis of results by Brakerski et al. [BLP+13] One major problem left open by Regev was whether there was a classical reduction from a worst-case lattice problem to LWE. The initial answer to this problem was provided by Peikert [Pei09]. While this represented progress, Peikert's reduction was not considered to be satisfactory since either an exponential size modulus is required or the lattice problem considered is not one of the standard problems. Later work by Brakerski et al. [BLP+13] built on Peikert's work to show a classical reduction from a standard lattice problem to LWE avoiding the exponential size modulus.

The works of Regev [Reg09], Peikert [Pei09] and Brakerski et al. [BLP+13] are all in the asymptotic setting hence the lattice dimension is allowed to go to infinity. However, practical

cryptosystems have a fixed value of the lattice dimension. So, it is of utmost interest to know what kind of security assurance one obtains from the results of [Reg09, Pei09, BLP+13] for practical cryptosystems. In Chapter 8, we follow up on [CKMS16, SS21] and perform a concrete security analysis of the tightness gap of the reduction in [BLP+13]. The reduction of Peikert [Pei09] is a step in the reduction performed by Brakerski et al. [BLP+13]. As a first step, we work out the tightness gap of Peikert's reduction. Then we follow the proof strategy in Brakerski et al. [BLP+13] and finally work out the end-to-end tightness gap of the classical reduction from the gap shortest vector problem to the LWE. There are two aspects to the concrete analysis. The first is a quadratic loss in the lattice dimension, and the second is a loss of tightness. The loss of tightness in this classical reduction is more than that of the original quantum reduction by Regev [Reg09]. The quadratic loss in the dimension was already pointed out in [BLP+13]. Due to this quadratic loss, Brakerski et al. put forward the open question of obtaining a reduction without such a loss mentioning that this would amount to a full de-quantization of Regev's reduction. However, the paper [BLP+13] does not consider the issue of the loss in tightness. Our analysis [SS20] shows that due to this loss of tightness, the reduction is not very meaningful in practice, especially for determining the sizes of the parameters of a cryptosystem that would purportedly enjoy the protection offered by the hardness of a well-studied worst-case lattice problem.

In Chapter 8, we extensively examine the tightness gap issues in all the lattice-based reductions presented in the previous chapters. We delve deeper into the significance of practice-oriented provable security and its necessity. We explore the potential consequences of loose reductions and highlight the importance of achieving tight security bounds. To provide a concrete analysis, we present tabulated results for various practical parameter ranges, offering a comprehensive understanding of the cryptographic schemes' security under real-world conditions. Moreover, we thoroughly discuss the drawbacks of employing structured lattices instead of general lattices in lattice-based cryptography, emphasizing the need to carefully consider the trade-offs between implementation efficiency and security guarantees.

Chapter 9 concludes the thesis and provides highlights of each chapter along with the importance of tightness gap analysis. Furthermore, our work includes providing valuable insights and directions for future research endeavors concerning concrete security aspects in the field of lattice-based cryptography. These directions aim to enhance the understanding of practical security guarantees and facilitate the development of more robust and efficient cryptographic protocols for real-world applications.

1.2 Publications

This thesis is based on the following works.

1. Palash Sarkar and Subhadip Singha. Verifying Solutions to LWE with Implications for Concrete Security. **Advances in Mathematics of Communications**, May 2021, 15(2): 257-266, https://www.aimsciences.org/article/doi/10.3934/amc.2020057.

2. Palash Sarkar and Subhadip Singha. Classical Reduction of GapSVP to LWE: A Concrete Security Analysis. **Advances in Mathematics of Communications**, April 2023, 17(2): 484-499, https://www.aimsciences.org/article/doi/10.3934/amc.2021004.

3. Neal Koblitz, Subhabrata Samajder, Palash Sarkar, Subhadip Singha. Concrete Analysis of Approximate Ideal-SIVP to Decision Ring-LWE Reduction. **Advances in Mathematics of Communications**, https://www.aimsciences.org/article/doi/10.3934/amc.2022082.

Chapter 2

Preliminaries

2.1 Introduction

This chapter focuses on the fundamental concepts that form the basis of this thesis. The central objects of interest are lattices and lattice-based cryptography. Lattices come in various variants, and we will explore their properties for all relevant types. Additionally, we delve into the preliminaries related to cryptography, with a specific focus on public key cryptography. Several lattice problems, known for their computational hardness and on which the hardness assumption of cryptographic constructions is based, will be discussed. To provide a comprehensive understanding of the ring and module version of the LWE problem, we explore the basics of algebraic number theory, which are essential for its analysis. Throughout this thesis, we rely on concepts from abstract algebra and linear algebra, and in this chapter, we present the fundamental ideas to ensure clarity and consistency in subsequent discussions.

2.2 Basic Definitions

We define a vector $\mathbf{v} = (v_1, \ldots, v_n)$ of dimension n in the column vector form. A row vector will be considered as the transpose of $\mathbf{v}$ or $\mathbf{v}^\mathsf{T}$. A $n \times n$ dimensional symmetric matrix M with real entries is positive-definite if the real number $\mathbf{v}^\mathsf{T} M \mathbf{v}$ is positive for every non-zero real row vector $\mathbf{v}$.

Definition 1 (Norm)**.** *Norms generalize the notion of length from Euclidean space. Norm on a vector space V over a subfield of $\mathbb{C}$ is a function from V to $\mathbb{R}$, denoted by $||v||$, satisfying the following properties.*

1. $||av|| = |a| \cdot ||v||$ *(positive homogeneity) where a is a scalar.*
2. $||u + v|| \leq ||u|| + ||v||$ *(triangle inequality).*
3. *If $||v|| = 0$, then $v = 0$ (separating point).*

Let n be a positive integer. For a vector $\mathbf{a} = (a_1, \ldots, a_n)$ in $\mathbb{R}^n$ or $\mathbb{C}^n$, the l_2-norm of $\mathbf{a}$ is defined to be $||\mathbf{a}||_2 = \left(|a_1|^2 + \cdots + |a_n|^2\right)^{1/2}$, the l_p-norm of $\mathbf{a}$ is defined to be $||\mathbf{a}||_p = \left(|a_1|^p + \cdots + |a_n|^p\right)^{1/p}$ and the l_∞-norm of $\mathbf{a}$ is defined to be $||\mathbf{a}||_\infty = \max_{i\in[n]}|a_i|$. We exclusively work with l_2 and l_∞ norm for most of this thesis.

We denote the open ball in $\mathbb{R}^n$ of unit radius by $\mathcal{B}_n$, i.e., $\mathcal{B}_n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < 1\}$.

Definition 2 (Distance or Metric)**.** *A distance function d is a non-negative function on $V \times V$, $d : V \times V \mapsto \mathbb{R}_{\geq 0}$ satisfying*

1. *$d(u, v) = d(v, u)$.*
2. *$d(u, v) \geq 0$ and $d(u, v) = 0$ if and only if $u = v$.*
3. *$d(u, v) \leq d(u, w) + d(w, v)$.*

, where V is a vector space.

Note 1. *In this case, V need not be over a subfield of the field of the complex numbers. Given a distance function, one can define the norm as a distance from 0.*

Definition 3 (Inner Product Space)**.** *Let V be a vector space over a field F (where F is either $\mathbb{R}$ or $\mathbb{C}$). An inner product is a function $\langle , \rangle : V \times V \mapsto F$ satisfying*

1. *$\langle u, v\rangle = \overline{\langle v, u\rangle}$, i.e., the complex conjugate.*
2. *$\langle u, v\rangle \geq 0$ and $\langle u, u\rangle = 0$ if and only if $u = 0$.*
3. *$\langle \alpha u + \beta v, w\rangle = \alpha\langle u, w\rangle + \beta\langle v, w\rangle$.*

Lemma 1. *If a, b, c are reals such that $a > 0$ and $a\lambda^2 + 2b\lambda + c \geq 0$ for all real λ, then $b^2 < ac$.*

Theorem 2 (Cauchy-Schwartz Inequality)**.** *If $u, v \in V$, where V is an inner product space, then*

$$|\langle u, v\rangle|^2 \leq ||u|| \times ||v||.$$

Definition 4 (Orthogonal Vectors)**.** *If $u, v \in V$, then u is orthogonal to v if $\langle u, v\rangle = 0$.*

Let, W be a subspace of V, s.t.,

$$W^\perp = \{x \in V : \langle x, w\rangle = 0 \text{ for all } w \in W\}.$$

Property: 1. *1. $W^\perp$ is a subspace of V.*

2. $W \cap W^\perp = \{0\}$, i.e., if $w \in W$ and $w \in W^\perp$, then $\langle w, w\rangle = 0 \Rightarrow w = 0$.

Definition 5 (Orthogonal Set)**.** *An orthogonal set $\{v_1, \ldots, v_n\} \subseteq V$ is a set such that $\langle v_i, v_j\rangle = 0$ for all $1 \leq i < j \leq n$.*

Definition 6 (Orthonormal Set)**.** *An orthonormal set is an orthogonal set such that each vector in the set has norm one.*

Property: 2. *An orthonormal set is linearly independent.*

Property: 3. *If $\{v_1, \ldots, v_n\}$ is orthonormal and $w \in V$, then*

$$u = w - \langle w, v_1\rangle v_1 - \cdots - \langle w, v_n\rangle v_n$$

is orthogonal to each $v_1, \ldots, v_n$.

Proposition 3. *If V is an inner product space, then $||v|| \stackrel{\Delta}{=} \sqrt{\langle v, v\rangle}$ defines a norm on V.*

2.2.1 Gram-Schmidt Orthogonalization

Gram-Schmidt Orthogonalization is the process to obtain an orthonormal basis for finite-dimensional inner product space. Orthogonalization has huge significance and usage in the field of lattices.

Theorem 4 (GSO)**.** *Any finite-dimensional inner product space has an orthonormal basis.*

Proof. Let $\{v_1, \ldots, v_n\}$ be any basis. Let $u_1 = v_1$ and

$$w_1 = \frac{v_1}{||v_1||}.$$

So, $||w_1|| = 1$. Let,

$$\begin{aligned}
u_2 &= v_2 - \langle v_2, w_1\rangle w_1, & w_2 &= \tfrac{u_2}{||u_2||} \\
u_3 &= v_3 - \langle v_3, w_1\rangle w_1 - \langle v_3, w_2\rangle w_2, & w_3 &= \tfrac{u_3}{||u_3||} \\
&\vdots & & \\
u_i &= v_i - \langle v_i, w_1\rangle w_1 - \cdots - \langle v_i, w_{i-1}\rangle w_{i-1}, & w_i &= \tfrac{u_i}{||u_i||}
\end{aligned}$$

The set of vectors $\{w_1, \ldots, w_n\}$ forms a orthonormal basis. □

Let $b_1, \ldots, b_n \overset{\text{GSO}}{\longmapsto} \widetilde{b}_1, \ldots, \widetilde{b}_n$. Since, $\widetilde{b}_1, \widetilde{b}_2, \ldots, \widetilde{b}_n$ are orthonormal, we have

$$\begin{aligned} \text{span}(\widetilde{b}_1)^{\perp} &= \text{span}(\widetilde{b}_2, \widetilde{b}_3, \ldots, \widetilde{b}_n) \\ \text{span}(\widetilde{b}_1, \ldots, \widetilde{b}_{i-1})^{\perp} &= \text{span}(\widetilde{b}_i, \ldots, \widetilde{b}_n) \end{aligned}$$

Notation: Given a basis $b_1, \ldots, b_n$, let π_i denote the projection on the space $\text{span}(\widetilde{b}_1, \ldots, \widetilde{b}_{i-1})^{\perp}$.

Theorem 5. *If W is a subspace of $\mathbb{R}^m$, then any $x \in \mathbb{R}^m$ can be written uniquely as $x = y+z$, where $y \in W$ and $z \in W^{\perp}$.*

In general, the GSO of $(b_1, \ldots, b_n)$ is $(\pi_1(b_1), \pi_2(b_2), \ldots, \pi_n(b_n)) = (\widetilde{b}_1, \widetilde{b}_2, \ldots, \widetilde{b}_n)$.

2.2.2 Algebraic Structures

Now we define a few algebraic structures like the ring, ideal, prime ideal, module, etc. Abstract Algebra by Dummit and Foote [DF04] provides these concepts in lucid ways.

Definition 7 (Ring)**.** *A ring is a non-empty set R together with two binary operators $+$ and $\cdot$ (commonly interpreted as addition and multiplication, respectively) satisfying the following conditions:*

1. *Additive associativity: For all $a, b, c \in R, (a + b) + c = a + (b + c)$,*
2. *Additive commutativity: For all $a, b \in R, a + b = b + a$,*
3. *Additive identity: There exists an element 0 in R such that for all $a \in S, 0 + a = a + 0 = a$,*
4. *Additive inverse: For every $a \in R$ there exists $-a \in R$ such that $a+(-a) = (-a)+a = 0$,*
5. *Left and right distributivity: For all $a, b, c \in S, a * (b + c) = (a * b) + (a * c)$ and $(b + c)\cdot a = (b * a) + (c\cdot a)$,*
6. *Multiplicative associativity: For all $a, b, c \in R, (a * b) * c = a * (b * c)$ (a ring satisfying this property is sometimes explicitly termed an associative ring).*

The set of integers $\mathbb{Z}$ forms a ring under usual addition and multiplication.

Definition 8 (Ideal)**.** *An ideal is a non-empty subset I of elements in a ring R that forms an additive group and has the property that, whenever $x \in R$ and $y \in I$, then $x \cdot y$ and $y \cdot x$ belong to I.*

For example, the set of even integers is an ideal in the ring of integers $\mathbb{Z}$.

Definition 9 (Prime Ideal)**.** *A prime ideal is a non-empty subset I of a ring R such that if $a \cdot b \in I$, then either $a \in I$ or $b \in I$.*

For example, in the integer ring $\mathbb{Z}$, the ideal $a = \langle p \rangle$ (i.e., the multiples of p) is prime ideal whenever p is a prime integer.

Definition 10 (Module)**.** *Let, R be a ring. A non-empty set M is said to be an R-module if M is an abelian group under an operation '+', s.t., for every $r \in R$, and $m \in M$, there is an element rm (defined as scalar multiplication) in M, s.t.,*

1. $r(a + b) = ra + rb$
2. $r(sa) = (rs)a$
3. $(r + s)a = ra + sa$

for all $a, b \in M$ and $r, s \in R$.

Modules are "vector spaces" defined over a ring.

2.3 Lattice

Lattice makes up most of this thesis. We not only discuss the basics of the lattices but also present standard results related to lattices. "An Introduction to the Geometry of Numbers" by Cassels [Cas59] provides thorough discussion on lattice. These standard results have been used rigorously in the later part of the thesis.

Definition 11 (Lattice)**.** *Given n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the* **lattice** *generated by them is*

$$L(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}.$$

We refer to $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ *as a basis of the lattice. Equivalently, if we define* $\mathcal{B}$ *as the* $m \times n$ *matrix whose columns are* $\mathbf{b}_1, \ldots, \mathbf{b}_n$*, then the lattice generated by* $\mathcal{B}$ *is* $L(\mathcal{B}) = L(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \{\mathcal{B}x | x \in \mathbb{Z}^n\}$*. We say that the rank of the lattice is* n *and its dimension is* m*. If* $n = m$*, the lattice is called a full-rank lattice.*

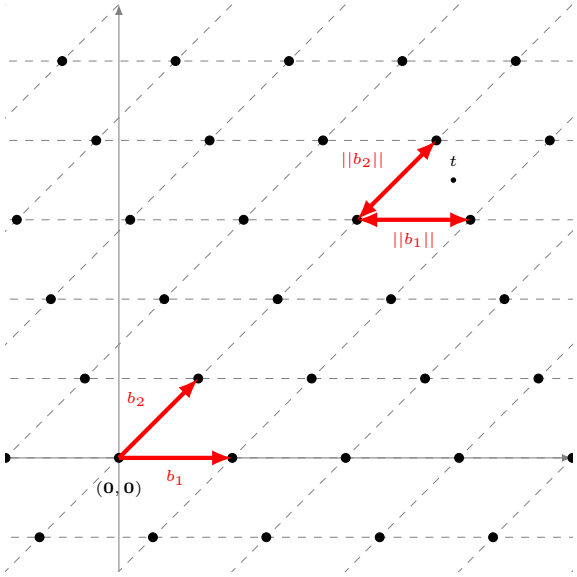


Figure 2-1: An example lattice.

A lattice is a $\mathbb{Z}$-module. A lattice can also be defined as a discrete additive subgroup of $\mathbb{R}^m$. We define the span of Lattice as follows

$$\begin{aligned} \text{Span of a lattice } L(\mathcal{B}) &= \mathsf{span}(L(\mathcal{B})) \\ &= \{\mathcal{B}y : y \in \mathbb{R}^n\}, \end{aligned}$$

In this thesis, we work with full-rank lattices only, i.e., $m = n$. Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ denote the columns of $\mathcal{B}$. The Gram-Schmidt orthogonalisation (GSO) of $\mathbf{b}_1, \ldots, \mathbf{b}_n$ will be denoted as $\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_n$. GSO of $\mathcal{B}$ will be needed in various places in this thesis.

Definition 12 (Fundamental Parallelepiped of a Lattice)**.**

$$\mathcal{P}(\mathcal{B}) = \{\mathcal{B}\mathbf{x} : \mathbf{x} \in \mathbb{R}^n, 0 \leq x_i < 1\};$$

where $\mathcal{B}$ *is a lattice basis.*

Property: 4 (Properties of Fundamental Parallelepiped)**.**

1. *The only lattice point in* $\mathcal{P}(\mathcal{B})$ *is the origin* $(0, \ldots, 0)^T$.

2. $\mathcal{P}(\mathcal{B})$ *induces a tiling of* $\mathsf{span}(L(\mathcal{B}))$, *i.e.,* $\mathsf{span}(L(\mathcal{B}))$ *can be written as*

$$\cup_{c \in L(\mathcal{B})} \{c + \mathcal{P}(\mathcal{B})\}.$$

Definition 13 (Determinant of a Lattice)**.** *Let* $\Lambda = L(\mathcal{B})$. *Then the determinant of* Λ *is the volume of* $\mathcal{P}(\mathcal{B})$, *i.e.,*

$$\det(\Lambda) = \sqrt{\det(\mathcal{B}^T\mathcal{B})}$$

or is equal to $|\det(\mathcal{B})|$ *if* $\mathcal{B}$ *is a full rank basis.*

Definition 14 (Successive Minima of a Lattice)**.** *Fix a lattice* Λ. *Define*

$$\lambda_1(\Lambda) = \min\{||x|| : x \in \Lambda, x \neq 0\}$$

So, $\lambda_1(\Lambda)$ *is the length of the shortest non-zero vector in* Λ. *Similarly define* $\lambda_2(\Lambda), \ldots, \lambda_n(\Lambda)$ *such that*

$$\lambda_i(\Lambda) = \inf\{r : \mathsf{dim}(\mathsf{span}(\Lambda \cap \bar{B}(0, r))) \geq i\};$$

where $\bar{B}(0, r)$ *denotes a closed ball of radius* r *centered at 0.*

Lattice problems like SVP, and SIVP deals with successive minima of a lattice. We state and prove theorems on successive minima in the following section.

2.3.1 Bounds on Successive Minima

Blichfeldt's Theorem

Theorem 6. *For any lattice* $\Lambda \subseteq \mathbb{R}^n$, *and a (measurable) set* $S \subseteq \mathbb{R}^n$ *such that* $\mathrm{vol}(S) > \det(\Lambda)$, *there exists points* $z_1, z_2 \in S$ *such that* $z_1 - z_2 \in \Lambda$.

Minkowski's Convex Body Theorem

We need the following definitions to prove the convex body theorem.

Definition 15. *(Convex Set) In a* convex set S, *if* $x, y \in S$, *then for* $t \in (0, 1)$, $tx + (1-t)y \in S$.

Definition 16. *(Centrally Symmetric Set) In a* centrally symmetric set S*, if* $x \in S$*, then* $-x \in S$*.*

Theorem 7. (**Minkowski's Convex Body Theorem**) *Let* Λ *be a full rank lattice of rank* n*. Then, for any centrally symmetric convex set* $S \subseteq \mathbb{R}^n$*, if* $\text{vol}(S) > 2^n \det(\Lambda)$*, then* S *will contain a point of* Λ*.*

Proof. Given the centrally symmetric convex set $S \subseteq \mathbb{R}^n$, let us define the set $\widehat{S}$ such that

$$\widehat{S} = \frac{1}{2}S = \{x : 2x \in S\}.$$

The volume of this set $\widehat{S}$ is

$$\text{vol}(\widehat{S}) = 2^{-n}\text{vol}(S) > \det(\Lambda).$$

By (Blichfeldt's) Theorem 6, since $\text{vol}(\widehat{S}) > \det(\Lambda)$, there are two points $y_1, y_2 \in \widehat{S}$ such that $y_1 - y_2 \in \Lambda$. Let $y_1 = z_1/2$ and $y_2 = z_2/2$ where $z_1, z_2 \in S$. Since S is a centrally symmetric set, $-z_2 \in S$. Also as S is a convex set, we have

$$\frac{z_1}{2} + \frac{(-z_2)}{2} = \frac{z_1}{2} - \frac{z_2}{2} \in S \Rightarrow y_1 - y_2 \in S.$$

□

2.3.2 Dual Lattice

Definition 17 (Dual Lattice)**.** *The dual of a lattice* L *is denoted as* L^* *and is defined to be the set of all vectors* $\mathbf{y} \in \mathbb{R}^n$*, such that* $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$ *for all* $\mathbf{x} \in L$*.*

If lattice $L = 2\mathbb{Z}^n = \{2y : y \in \mathbb{Z}^n\}$, then $(2\mathbb{Z}^n)^* = \frac{1}{2}\mathbb{Z}^n$. Also if lattice $L = \mathbb{Z}^n = \{y : y \in \mathbb{Z}^n\}$, then $(\mathbb{Z}^n)^* = \mathbb{Z}^n$

Definition 18 (Dual Basis)**.** *Let* $\mathcal{B}_{m\times n}$ *be a basis of a lattice* L*. The* **dual basis of** $\mathcal{B}$ *is defined to be the unique basis* $\mathcal{D}_{m\times n}$*, s.t.,*

1. $\mathsf{span}(\mathcal{B}) = \mathsf{span}(\mathcal{D})$.
2. $\langle b_i, d_j \rangle = \delta_{i,j} = \begin{cases} 1 & \text{if } i = j; \\ 0 & o.w.; \end{cases}$
   i.e., $\mathcal{B}^T\mathcal{D} = I_n$ *= identity matrix of rank* n*.*

Following are a few standard results related to a lattice and its dual.

Proposition 8. *If $\mathcal{D}$ is the dual basis of $\mathcal{B}$, then $(L(\mathcal{B}))^* = L(\mathcal{D})$.*

Proposition 9. *For any lattice Λ, $(\Lambda^*)^* = \Lambda$.*

Proposition 10. *For any lattice Λ, $\det(\Lambda^*) = \frac{1}{\det(\Lambda)}$.*

Proposition 11. *For any rank n lattice Λ, $\lambda_1(\Lambda) \cdot \lambda_1(\Lambda^*) \leq n$.*

Proposition 12. *For any rank n lattice Λ, $\lambda_1(\Lambda) \cdot \lambda_n(\Lambda^*) \geq 1$.*

The following theorem is very important regarding the successive minima of lattices and their dual and provides an upper bound on the value on $\lambda_1(\Lambda) \cdot \lambda_n(\Lambda^*)$.

Theorem 13 (Transference Theorem (Bauaszyczyk, 1993) [Ban93])**.** *For any rank n lattice Λ,*

$$1 \leq \lambda_1(\Lambda)\lambda_n(\Lambda^*) \leq n.$$

2.3.3 Smoothing Parameter

Let $\rho_s(x) = e^{-\pi||x/s||^2}$ be an n-dimensional Gaussian distribution (See Definition 23) over $\mathbb{R}^n$ and $\nu_s(x) = \frac{\rho_s(x)}{s^n}$ is the density function of a Gaussian distribution over $\mathbb{R}^n$. A vector chosen randomly following $\nu_1(x) = \rho_1(x) = \rho(x)$ has length at most $\sqrt{n}$ with probability $1 - 2^{-\Omega(n)}$. This result can be extended to show that a vector chosen randomly following $\nu_s(x)$ has length at most $s\sqrt{n}$ with probability at least $1 - 2^{-\Omega(n)}$. Suppose a lattice vector is chosen "uniformly" and some Gaussian noise is added to it. As the noise increases, the result begins to look like it is uniformly distributed.

Consider two distributions:

$\mathcal{D}_0$: A vector chosen from $\mathcal{P}(\mathcal{B})$ according to uniform density function on $\mathcal{P}(\mathcal{B})$.

$\mathcal{D}_1$: A vector is chosen from $\mathbb{R}^n$ following $\nu_s(x)$ and reduced modulo the lattice.

Lemma 14. *The statistical distance between $\mathcal{D}_0$ and $\mathcal{D}_1$ is at most $\frac{1}{2}\rho_{1/s}(\Lambda^* \backslash \{0\})$.*

The above discussion motivates the following definition.

Definition 19 (Smoothing Parameter [MR07])**.** *For any $\epsilon > 0$, the **smoothing parameter** of the lattice Λ is the minimum value of s, s.t.,*

$$\rho_{1/s}(\Lambda^* \backslash \{0\}) \leq \epsilon.$$

The smoothing parameter is denoted as $\eta_\epsilon(\Lambda)$.

To see why this is well-defined, notice that $\rho_{1/s}(\Lambda^* \setminus \{0\})$ is a continuous and strictly decreasing function of s with

$$\lim_{s \to 0} \rho_{1/s}(\Lambda^* \setminus \{0\}) = \infty \quad \text{and} \quad \lim_{s \to \infty} \rho_{1/s}(\Lambda^* \setminus \{0\}) = 0.$$

Using this definition, the above result can be restated as follows:

> *for any $s \geq \eta_\epsilon(\Lambda)$, the statistical distance between the uniform distribution on $\mathcal{P}(\mathcal{B})$ and the distribution obtained by sampling from ν_s and reducing the result modulo $\mathcal{P}(\mathcal{B})$ is at most $\frac{1}{2}\epsilon$.*

Informally speaking, the smallest amount of Gaussian noise that "smooths out" the discrete structure of a lattice Λ, is called the smoothing parameter of the lattice. The smoothing parameter is connected with various parameters of a lattice. The following results highlight a few of them.

Claim 15. *For any $\epsilon < \frac{1}{100}$, $\eta_\epsilon(\Lambda) \geq \frac{1}{\lambda_1(\Lambda^*)}$.*

Corollary 16. *For any $\epsilon < \frac{1}{100}$, $\eta_\epsilon(\Lambda) \geq \frac{\lambda_n(\Lambda)}{n}$.*

Claim 17 ([MR07, Lemma 3.2])**.** *For any $\epsilon \geq 2^{-n+1}$, $\eta_\epsilon(\Lambda) \leq \frac{\sqrt{n}}{\lambda_1(\Lambda^*)}$.*

Corollary 18. *For any $\epsilon \geq 2^{-n+1}$, $\eta_\epsilon(\Lambda) \leq \sqrt{n}\lambda_n(\Lambda)$.*

Lemma 19 ([Reg09, Claim 2.13])**.** *For any lattice L and any $\epsilon > 0$*

$$\eta_\epsilon(L) \geq \sqrt{\frac{\ln 1/\epsilon}{\pi}} \cdot \frac{1}{\lambda_1(L^*)} \geq \sqrt{\frac{\ln 1/\epsilon}{\pi}} \cdot \frac{\lambda_n(L)}{n}$$

2.4 Algebraic Number Theory

To work with ideal lattice, module lattice, ring LWE, and module LWE, we need some concepts of algebraic number theory.

2.4.1 Number Field

A number field is a finite extension of $\mathbb{Q}$, the field of rationals. Let K be a number field. Then $K = \mathbb{Q}(\zeta)$, where ζ is a complex root of a monic irreducible polynomial $f(x) \in \mathbb{Q}[x]$. K can be considered as a vector space over $\mathbb{Q}$. If $f(x)$ is a polynomial of degree n then K is an n-dimensional vector space with $\{1, \zeta^1, \zeta^2, \cdots, \zeta^{n-1}\}$ as a basis. This is also called the power basis of K over $\mathbb{Q}$. Here $f(x)$ is the minimal polynomial of ζ.

2.4.2 Ring of Algebraic Integers of a Number Field

An algebraic integer is a root of a monic polynomial with integer coefficients. The set of algebraic integers in a number field forms a ring under usual addition and multiplication operations. We call this ring, the ring of algebraic integers. Let R be the ring of integers of the number field K. R is a free $\mathbb{Z}$ module of rank n.

2.4.3 Space H

The space H is useful to work with ring LWE and module LWE.

Let s_1 and s_2 be non-negative integers such that $s_1 + 2s_2 = n$. The space $H \subset \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ is defined as

$$H = \{(x_1, \ldots, x_n) \in \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2} : x_{s_1+s_2+j} = \overline{x_{s_1+j}}, j = 1, \ldots, s_2\}. \tag{2.1}$$

Using the inner product on H induced on it by $\mathbb{C}^n$, it can be shown that H is isomorphic to $\mathbb{R}^n$ as an inner product space. For $j \in [n]$, let $e_j \in \mathbb{C}^n$ be the vector which has 1 in its j-th component and 0 elsewhere. An orthonormal basis for H is given by $\{h_i\}_{i\in[n]}$, where for $j \in [s_1], h_j = e_j$ and for $s_1 < j \leq s_1+s_2, h_j = (e_j+e_{j+s_2})/\sqrt{2}, h_{j+s_2} = \sqrt{-1}(e_j-e_{j+s_2})/\sqrt{2}$. When $\mathbf{x} \in H$ is written in terms of the orthonormal basis as $\mathbf{x} = \sum_{i=1}^{n} a_i \mathbf{h}_i$ with $(a_1, \ldots, a_n) \in \mathbb{R}^n$, the norm of $\mathbf{x}$ is simply $||(a_1, \ldots, a_n)||$.

All the previous definitions related to lattices can be defined on H in the same way these are defined on $\mathbb{R}^n$, because of the equivalence of H and $\mathbb{R}^n$ as an inner product space.

2.4.4 Canonical Embedding

Let $K = \mathbb{Q}(\zeta)$ be a number field and ζ is a complex root of a monic irreducible polynomial $f(x) \in \mathbb{Q}[x]$. $f(x)$ has n roots over $\mathbb{C}$. We know that complex roots come in pairs and let s_1 be the number of real roots and $2s_2$ be the number of complex roots so that $s_1 + 2s_2 = n$. Suppose that the roots are ordered as $\zeta_1, \ldots, \zeta_n$, where $\zeta_1, \ldots, \zeta_{s_1}$ are real and $\zeta_{s_1+j} = \overline{\zeta_{s_1+s_2+j}}$ for $j = 1, \ldots, s_2$. Let $\sigma_i : K \to \mathbb{C}$ be the embedding of K in $\mathbb{C}$ obtained by extending the map $\zeta \mapsto \zeta_i$. Thus, an n-dimensional number field K has exactly n injective ring homomorphism or embeddings $\sigma_i : K \to \mathbb{C}, i \in \{1, \ldots, n\}$ that fixes every element of $\mathbb{Q}$. The canonical embedding $\sigma : K \to \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ is defined as $\sigma(x) = (\sigma_1(x), , \cdots \sigma_n(x))$. Note that for any $x \in K$, and $i = 1, \ldots, s_2, \sigma_{s_1+i}(x) = \sigma_{s_1+s_2+i}(x)$, so that $\sigma(K) \subset H$.

2.4.5 Trace and Norm

The trace and norm of any $x \in K$ are respectively defined as $\mathsf{Tr}(x) = \sum_{i=1}^{n} \sigma_i(x)$ and $\mathsf{N}(x) = \Pi_{i=1}^{n} \sigma_i(x)$.

2.4.6 Ideals of a Number Field

Any ideal $\mathcal{I} \subseteq R$, of the ring of an algebraic integer of a number field K, is called an integral ideal. $\mathcal{I}$ is a non-empty additive subgroup of R that is closed under multiplication by elements of R. An integral ideal is also an n-dimensional free $\mathbb{Z}$ module which means that it can be generated by $\mathbb{Z}$ linear combinations of some basis $\{I_1, \ldots, I_n\} \subset R$.

A fractional ideal is a nonempty set $\mathcal{J} \subset K$, such that $d\mathcal{J} \subseteq R$ is an integral ideal for some $d \in R$. Like integral ideals, fractional ideals are also free $\mathbb{Z}$ modules.

A fractional ideal $\mathcal{J}$ also has $\mathbb{Z}$-basis $\{J_1, \ldots, J_n\} \subset K$. Fractional ideals form a multiplicative group in K with R as the identity element. The inverse of a fractional ideal $\mathcal{J}$ as an element of the multiplicative group is denoted as $\mathcal{J}^{-1} = \{x \in K : x\mathcal{J} \subseteq R\}$.

2.4.7 Norm of an Ideal

The norm of an integral ideal $\mathcal{I}$ is defined to be $\mathrm{N}(\mathcal{I}) = \#(R/I)$. As a fractional ideal $\mathcal{J} \in K$ is a set such that $d\mathcal{J}$ is an integral ideal of R for some $d \in R$, the norm of a fractional ideal $\mathcal{J}$ is defined to be $\mathrm{N}(\mathcal{J}) = \mathrm{N}(d\mathcal{J})/|\mathrm{N}(d)|$.

2.4.8 Dual of an Ideal

Dual of an ideal $\mathcal{J}$ in the number field K is defined as $\mathcal{J}^\vee = \{x \in K : \mathrm{Tr}(x\mathcal{J}) \subseteq \mathbb{Z}\}$. We have the relation $\mathcal{J}^\vee = \mathcal{J}^{-1} \cdot R^\vee$ between dual and inverse of an ideal in K. Every integral ideal of R including R is trivially a fractional ideal, so we can also define dual of R as $R^\vee$ using the same relation.

2.4.9 Ideal Lattice

If $\mathcal{J} \subset K$ is a fractional ideal, under canonical embedding $\sigma(\mathcal{J})$ forms a lattice in H, this we call an ideal lattice. Ideal lattice $\sigma(\mathcal{J})$ is of dimension n. The ideal lattice $\sigma(\mathcal{J})$ has basis $\{\sigma(J_1), \ldots, \sigma(J_n)\}$ where $\{J_1, \ldots, J_n\} \subset K$ is the $\mathbb{Z}$-basis of $\mathcal{J}$ in K. Note that the ideal lattice $\sigma(\mathcal{J}^\vee)$ corresponding to the ideal $\mathcal{J}$ is the same as the dual lattice up to complex conjugation, i.e. $\sigma(\mathcal{J}^\vee) = \overline{\sigma(\mathcal{J})^*}$. In practice, we often refer to $\mathcal{J}$ as an ideal lattice in place of $\sigma(\mathcal{J})$ as an abuse of notation.

2.4.10 Cyclotomic Polynomial

The m-th cyclotomic polynomial $\Phi_m(x)$ is defined as the product of the terms $x - \zeta$, where ζ ranges over all primitive m-th roots of unity in $\mathbb{C}$. Now an m-th root of unity is a primitive d-th root of unity for some divisor d of m, so $x^m - 1$ is the product of all cyclotomic polynomials $\Phi_d(x)$ with d a divisor of m. In particular, let $n = p^r$ be a prime power. Since a divisor of p^r is either p^r or a divisor of p^{r-1}, we have

$$\Phi_{p^r}(x) = \frac{x^{p^r} - 1}{x^{p^{r-1}} - 1} = \frac{t^p - 1}{t - 1} = 1 + t + \cdots + t^{p-1} \tag{2.2}$$

where $t = x^{p^{r-1}}$. If $x = 1$ then $t = 1$, and it follows that $\Phi_{p^r}(1) = p$.

2.4.11 Cyclotomic Number Field

Let $\Phi_m(x)$ be the m-th cyclotomic polynomial for $m \geq 1$. Let the degree of the m-th cyclotomic polynomial is $n = \varphi(m)$. The m-th cyclotomic number field K is $\mathbb{Q}(\zeta)$ where ζ is a root of $\Phi_m(x)$. Let, $\{1, \zeta, \zeta^2 \ldots, \zeta^{n-1}\}$ be the power basis of K over $\mathbb{Q}$. In case of cyclotomic number field, the power basis $\{1, \zeta, \zeta^2 \ldots, \zeta^{n-1}\}$ is also the $\mathbb{Z}$-basis of the ring of algebraic integer R of cyclotomic number field K, so $R = \mathbb{Z}[x]/\Phi_m(x)$.

Let q be a prime number such that $q = 1 \bmod m$ so that $q = km+1$ for some non-negative integer k. Noting that $\mathbb{Z}_q^* = \langle g \rangle$ for a generator g, it follows that the element $\omega = g^k$ has order m in $\mathbb{Z}_q^*$. The m-th cyclotomic polynomial factors over $\mathbb{Z}_q$ as $\Phi_m(x) = \prod_{i \in \mathbb{Z}_m^*}(x - \omega^i)$. Consequently, $\langle q \rangle = \prod_{i \in \mathbb{Z}_m^*} \mathfrak{q}_i$, where $\mathfrak{q}_i = \langle q, x - \omega^i \rangle$ is a prime ideal of R having norm q, so $\mathrm{N}(\mathfrak{q}_i) = q$. Note that the ideals $\mathfrak{q}_i$ have been indexed by elements of $\mathbb{Z}_m^*$ rather than by the integers $\{1, \ldots, n\}$.

The field K has n automorphisms $\tau_k(\zeta) = \zeta^k$, for $k \in \mathbb{Z}_m^*$. It follows that for $k \in \mathbb{Z}_m^*$, $\tau_k(\mathfrak{q}_i) = \mathfrak{q}_{ik^{-1} \bmod m}$ and $\tau_k^{-1} = \tau_{k^{-1} \bmod m}$. Also, for $k \in \mathbb{Z}_m^*$, R and $R^\vee$ are fixed by τ_k and so $\tau_k(R_q) = R_q$. Hence, if a is distributed uniformly in R_q, then $\tau_k(a)$ is also distributed uniformly in R_q. Here R_q represents the set of residue classes of R modulo qR.

For $i \in \mathbb{Z}_m^*$, it can be shown that the quotient group $R^\vee/(\mathfrak{q}_i R^\vee)$ has cardinality q and the representatives of the q distinct cosets can be taken to be the elements of the set $\{0, \ldots, q-1\}$. The cardinality of the set $R_q^\vee$ is q^n. Using the Chinese Remainder Theorem (CRT), it can be shown that there is an isomorphism $\mathfrak{I}$ from $R_q^\vee$ to $\oplus_{i \in \mathbb{Z}_m^*}(R^\vee/(\mathfrak{q}_i R^\vee))$. Further, $\mathfrak{I}$ can be efficiently computed in both the forward and the backward directions. For $i \in \mathbb{Z}_m^*$, let $w_i \in \{0, \ldots, q-1\}$ represent a coset of $R^\vee/(\mathfrak{q}_i R^\vee)$. Given $(w_i)_{i \in \mathbb{Z}_m^*}$, it is possible to efficiently construct $w \in R_q^\vee$ such that the i-th component of $\mathfrak{I}(w)$ is represented by w_i. For the sake of notational convenience, we let w denote $\mathfrak{I}^{-1}((w_i)_{i \in \mathbb{Z}_m^*})$.

2.4.12 Isomorphism in a Number Field

Using Chinese Remainder Theorem [Gau66], Lyubashevsky, Peikert and Regev [LPR13] have shown an isomorphism between R_q and $\mathcal{I}_q$, where $\mathcal{I}$ is any ideal of R, where q is an arbitrary positive integer. Similarly isomorphism can be shown between $R_q^\vee$ and $\mathcal{I}_q^\vee$.

2.5 Module

We have already defined the module in Section 2.2.2. Now, we define the module in the context of algebraic number theory. These definitions are equivalent but the following definition is needed for our analysis of the module and ideal lattice.

Definition 20. *Let K be a number field with R its ring of integers and $d, d' \in \mathbb{N}$. Let $M \subseteq K^d$. M is an R-module of rank $d' \leq d$ if it is closed under addition by elements of M and multiplication by elements of R.*

When $d' = d$, we refer to the module M as a full-rank R-module. In this thesis we work with full rank modules only.

It is a finitely generated module if there exists a finite family $\{b_k\}_k$ of vectors in K^d such that $M = \sum_k R{\cdot}b_k$. The dual of a module $M \subseteq K^d$ is denoted by $M^\vee$ and is defined by $M^\vee = \{x \in K^d : \mathrm{Tr}(\langle x, y\rangle) \subseteq \mathbb{Z}\ \forall y \in M\}$.

2.5.1 Module Lattice

We can define module lattices similarly to ideal lattices defined in Section 2.4.9 using canonical embedding. Using d canonical embeddings in Section 2.4.4, we define a map $\Sigma = (\sigma, \ldots, \sigma)$. Now the map Σ defines a canonical embedding from $K^d \to H^d$. The canonical embedding $\Sigma : K^d \to H^d$ is defined as $\Sigma(\mathbf{x}) = (\sigma(x_1), , \cdots \sigma(x_d))$ where $\mathbf{x} \in K^d$ and $\mathbf{x} = (x_1, \ldots, x_d)$. As H is isomorphic to $\mathbb{R}^n$, the canonical embedding Σ maps K^d into $\mathbb{R}^N$, where $N = n{\cdot}d$. Similar to ideal lattices, $\Sigma(M)$ is a module lattice. If the underlying number field K is of degree n and the rank of the module M is d, then the corresponding module lattice will have dimension $N = n{\cdot}d$. Similar to the dual ideal lattice, the module lattice $\Sigma(M^\vee)$ corresponding to the module M is the same as the dual lattice up to complex conjugation, i.e. $\Sigma(M^\vee) = \overline{\Sigma(M)^*}$ and similar to ideal lattice, we often refer M as a module lattice in place of $\Sigma(M)$ as an abuse of notation.

Norm of elements of a module: For any vector $\mathbf{x} \in K^d$ we define the Euclidean norm of $||\mathbf{x}||_2 = \left(\sum_{i\in[d]} \sum_j |\sigma_{j\in[n]}(x_i)|^2\right)^{1/2}$. We also define the infinity norm of $\mathbf{x} \in K^d$ in three different ways as follows $||\mathbf{x}||_\infty = \max_{i,j} |\sigma_j(x_i)|$, $||\mathbf{x}||_{\infty,2} = \max_i\left(\sum_j |\sigma_{j\in[n]}(x_i)|^2\right)^{1/2}$ and $||\mathbf{x}||_{2,\infty} = \max_j\left(\sum_{i\in[d]} |\sigma_j(x_i)|^2\right)^{1/2}$. Different definitions of the norm of an element of a module are consistent with the definition of norm defined in Definition 1. These different definitions are applicable in different scenarios of module lattice in the latter part of this thesis. Once the notion of module lattice is established, we can talk about all the lattice properties and results restricted to module lattices.

2.6 Statistical Results

Statistical results play a pivotal role in this thesis. Firstly, the error distribution for LWE is a Gaussian distribution. Gaussian distribution in various forms has been used, e.g, single

variable Gaussian distribution, multivariate Gaussian distribution, or discrete and continuous Gaussian distribution. In the case of multivariate gaussian distribution, the requirement varies from elliptical to spherical Gaussian distribution. Different forms of Chernoff-Hoeffding bound are required to estimate the probability of events in different parts of this thesis. Also, statistical hypothesis testing is used in various places. These statistical elements are defined in this section before using these in the contributory sections in the later part of this thesis.

Definition 21 (Normal Distribution)**.** *The normal distribution with mean μ and standard deviation σ is denoted as $\mathcal{N}(\mu, \sigma)$. The probability density function $f(x)$ of $\mathcal{N}(\mu, \sigma)$ is the follwing*

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2\}; \;\; -\infty \leq x \leq \infty.$$

For $\alpha \in (0, 1)$, Ψ_α is the probability distribution obtained by sampling from $\mathcal{N}(0, \alpha/\sqrt{2\pi})$ and reducing the result modulo 1.

Definition 22 (Multivariate Normal/Gaussian Distribution)**.** *The multivariate Gaussian distribution of k-dimensional random vector $\mathbf{X} = (X_1, \ldots, X_k)$ is denoted as $\mathcal{N}(\mu, \Sigma)$ where*

$$\mu = \mathrm{E}[\mathbf{X}] = (\mathrm{E}[X_1], \ldots, \mathrm{E}[X_k]).$$

and $k \times k$ co-variance matrix

$$\Sigma_{i,j} = \mathrm{E}[(X_i - \mathrm{E}[X_i])(X_j - \mathrm{E}[X_j])] = \mathrm{Cov}[X_i, X_j].$$

such that $1 \leq i \leq k$ and $1 \leq j \leq k$. The inverse of the covariance matrix is called the precision matrix and is denoted by $Q = \Sigma^{-1}$. The multivariate Gaussian distribution is said to be "non-degenerate" when the co-variance matrix Σ is positive definite. The probability density function $f(x_1, \ldots, x_k)$ is the following

$$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\mu)\right)}{\sqrt{(2\pi)^k|\Sigma|}}$$

where $\mathbf{x} = (x_1, \ldots, x_k)$ is a real k-dimensional column vector and $|\Sigma|$ is the determinant of Σ.

Definition 23 (Continuous Gaussian Distribution)**.** *For $s > 0$, the Gaussian function $\rho_s(\mathbf{x})$ is defined by*

$$\rho_s(\mathbf{x}) = \exp\left(-\pi||\mathbf{x}||^2/s^2\right), \;\; \mathbf{x} \in \mathbb{R}^n.$$

As H is isomorphic to $\mathbb{R}^n$ as an inner product space, continuous Gaussian distribution can be defined in the same way for $\mathbf{x} \in H$. The continuous Gaussian probability distribution over $\mathbb{R}^n$ or H is denoted by D_s and defined by

$$D_s(\mathbf{x}) = s^{-n}\rho_s(\mathbf{x}).$$

D_s is the n-dimensional normal distribution with mean vector $(0, \ldots, 0)$ and co-variance matrix $\text{diag}(\sigma^2, \ldots, \sigma^2)$ where $\sigma = s/\sqrt{2\pi}$. If X_1 and X_2 are two independent random variables following D_{s_1}, and D_{s_2} respectively, then $X_1 + X_2$ follows $D_{\sqrt{s_1^2+s_2^s}}$.

Definition 24 (Discrete Gaussian Distribution)**.** *For $\mathbf{x} \in \mathbb{R}^n$ and $s > 0$, define $\rho_s(\mathbf{x}) = \exp\left(-\pi||\mathbf{x}||^2/s^2\right)$. For a countable set A, define $\rho_s(A) = \sum_{\mathbf{x}\in A}\rho_s(\mathbf{x})$. The discrete Gaussian distribution $D_{A,s}$ on a countable set A, assigns to an element $\mathbf{v} \in A$ the probability as following*

$$D_{A,s}(\mathbf{v}) = \rho_s(\mathbf{v})/\rho_s(A)$$

.

The following two results of discrete Gaussian distribution over lattices are very useful for our analysis

Lemma 20 ([Ban93, Lemma 1.4(i)])**.** *For any lattice L and $a \geq 1$, $\rho_a(L) \leq a^n\rho(L)$.*

Lemma 21 ([Ban93, Lemma 1.5(i)])**.** *Let $\mathcal{B}_n$ denote the Euclidean unit ball. Then, for any lattice L and any $r > 0, \rho_r(L \setminus \sqrt{n}r\mathcal{B}_n) < 2^{-2n}\cdot\rho_r(L)$, where $L \setminus \sqrt{n}r\mathcal{B}_n$ is the set of lattice points of norm greater than $\sqrt{n}r$.*

While the first lemma is self-explanatory, the second one says that the probability of sampling lattice points outside the unit ball $\mathcal{B}_n$ in the Euclidean space using discrete Gaussian distribution is negligible. Hence, the output vectors will have a length less than $\sqrt{n}r$ with a probability exponentially close to 1.

Definition 25 (Statistical Distance)**.** *Let X be a random variable taking values in a set D and S be a subset of D. By $f_X(S)$ we denote the probability that X takes values in S. Given two random variables X and Y over D, the statistical distance between them is denoted as $\Delta(X, Y)$ and is defined to be $\Delta(X, Y) = \max_{S\subseteq D}|f_X(S) - f_Y(S)|$.*

2.6.1 Hoeffding Inequality

We describe Hoeffding's inequality [Hoe63] for the sum of independent random variables. This result presents the additive form of the Hoeffding bound.

Theorem 22 (Hoeffding Inequality)**.** *Let, $X_1, X_2, \ldots, X_\lambda$ be a finite sequence of independent random variables, such that for all $i = 1, \ldots, \lambda$, there exists real numbers $a_i, b_i \in \mathbb{R}$, with $a_i < b_i$ and $a_i \leq X_i \leq b_i$. Let $X = \sum_{i=1}^{\lambda} X_i$. Then for any positive $t > 0$,*

$$\Pr[X - E[X] \geq t] \leq \exp\left(-\frac{2t^2}{L_\lambda}\right) \quad \textit{and} \quad \Pr[X - E[X] \leq -t] \leq \exp\left(-\frac{2t^2}{L_\lambda}\right);$$

where $L_\lambda = \sum_{i=1}^{\lambda} (b_i - a_i)^2$.

2.6.2 Chebyshev's Inequality

We state Chebyshev's inequality [Als11] for a random variable taking real values.

Theorem 23 (Chebyshev's Inequality)**.** *Let X (integrable) be a random variable with finite expected value μ and finite non-zero variance σ^2. Then for any real number $k > 0$,*

$$\mathsf{Pr}(|X - \mu| \geq k) \leq \frac{\sigma^2}{k^2}.$$

2.7 Learning with Error (LWE)

Learning with Error (LWE) is one of the central objects of this thesis other than lattices. The LWE problem has been the basis of many lattice-based cryptographic schemes. We define LWE as related to different algebraic variants of lattices. We also discuss different search and decisional problems of LWE.

2.7.1 Average case Problem versus Worst case Problem

In cryptography, we often use these terms "**Average Case, Worst case**" with computationally hard problems. Public-key cryptography is based on one-way functions that are easy to compute but hard to reverse. For example, it's easy to multiply two large prime numbers

to get their product, but hard to factor that product back into the two numbers. The cryptographic constructions are based on factoring, the assumption is that it is hard to factor numbers chosen from a certain distribution. How do we choose this certain distribution? One obvious choice is not to choose from numbers with small factors or even factors. In this case, the problem of factoring can be solved easily. So, we need to fix the distribution for the factoring problem in the order for it to be hard to break. The distribution must cover a non-negligible fraction of the instances of the given problem. The "Average Case" simply refers to this situation. An average-case problem is "Hard" for instances of the problem chosen from a certain distribution. The "Worst Case" refers to the case where except with a non-negligible fraction of instances of a problem is hard to break.

Most conventional cryptosystems rely on average-case problems. Ajtai's [Ajt96] seminal work discovered that hard lattice problems can be made basis to construct the cryptographic secure system. The worst-case hardness guarantee of some hard lattice problems can be used for proving the security of cryptographic constructions. What it means is that if an adversary succeeds in breaking the cryptographic scheme, even with some small probability, then the adversary can also solve any instance of a certain lattice problem. Subsequently, Ajtai and Dwork [AD97] published cryptosystems whose security is based on the worst-case hardness of some lattice problems.

2.7.2 LWE Distributions

Regev introduced the LWE problem in his seminal paper [Reg09]. The LWE distribution is closely related to the LWE problem. In the LWE problem, a user is given polynomial samples as input from the LWE distribution. The requirement is to output the unknown of the LWE distribution. Here we define different LWE distributions.

Definition 26 (LWE Distribution)**.** *Let $p \geq 2$ be an integer. Let χ be a probability distribution on $\mathbb{Z}_p$. Let n be a positive integer and fix $\mathbf{s} \in \mathbb{Z}_p^n$. The distribution $A_{p,\mathbf{s},\chi}$ on $\mathbb{Z}_p^n \times \mathbb{Z}_p$ is defined as follows. Choose $\mathbf{a}$ uniformly at random from $\mathbb{Z}_p^n$; e from $\mathbb{Z}$ following χ and output $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$. Let ϕ be a probability density function on $\mathbb{T} = (0, 1)$. The distribution $A_{p,\mathbf{s},\phi}$ is defined as follows. Choose $\mathbf{a}$ uniformly at random from $\mathbb{Z}_p^n$; e from $\mathbb{T}$ following ϕ and output $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle / p + e)$, where the addition is performed modulo 1.*

We need this definition in Chapters 4 and 7 when dealing with Euclidean lattices. Equivalently we define LWE distributions related to a different algebraic variant of lattices as follows.

Definition 27 (Ring-LWE Distribution)**.** *Let K be a number field and R be the ring of an algebraic integer of K. For a fractional ideal $\mathcal{J}$ in K and an integer $q \geq 2$, let $\mathcal{J}_q$ denote the set of residue classes of $\mathcal{J}$ modulo $q\mathcal{J}$. Let $\mathbb{T} = H/\sigma(R^\vee)$ denote H modulo $\sigma(R^\vee)$. Suppose $s \in R_q^\vee$ and $a \in R_q$. There are elements $x \in R^\vee$ and $y \in R$ such that $s = x + qR^\vee$ and $a = y + qR$. Define the result of the operation $a{\cdot}s$ to be $xy + qR^\vee$ which is in $R_q^\vee$. One can show that the operation is well-defined. Similarly, the result of the operation $(a{\cdot}s)/q$ is defined to be $xy/q + R^\vee$ which is in $(1/q)R^\vee$ modulo $R^\vee$. By $\sigma((a{\cdot}s)/q)$ we will denote the element $\sigma(xy/q) + \sigma(R^\vee)$ of $\mathbb{T}$. For $s \in R_q^\vee$ and a positive real number r, a sample from the ring-LWE distribution $A_{s,r}^{(R)}$ over $R_q \times \mathbb{T}$ is $(a, \sigma((a{\cdot}s)/q) + \mathbf{e} \mod \sigma(R^\vee))$, where a is chosen uniformly at random from R_q and $\mathbf{e}$ is chosen from H following the distribution D_r .*

We use this definition in Chapter 5 and 6 when dealing with ideal lattices. Next, we define module-LWE distribution which we use in Chapter 5 when working with module lattices.

Definition 28 (Module-LWE Distribution)**.** *Let K be a number field, R be the ring of an algebraic integer of K, M be a module with rank d, and $M \subseteq K^d$. Let $\mathbb{T} = H/\sigma(R^\vee)$ denote H modulo $\sigma(R^\vee)$. For $\mathbf{s} \in (R_q^\vee)^d$ and a positive real number r, a sample from the* module-LWE *distribution $A_{\mathbf{s},r}^{(M)}$ over $(R_q)^d \times \mathbb{T}$ is $(\mathbf{a}, \sigma(\langle \mathbf{a}{\cdot}\mathbf{s} \rangle/q) + \mathbf{e} \mod \sigma(R^\vee))$, where $\mathbf{a}$ is chosen uniformly at random from $(R_q)^d$ and $\mathbf{e}$ is chosen from H following the distribution D_r .*

2.7.3 LWE Problems

Learning problem has been studied for ages from different research aspects. Regev [Reg09] made the "Learning with Error" (LWE) problem famous by showing cryptographic constructions based on the parameters of LWE and showing a reduction from hard lattice problems to it, thus projecting the security of the crypto-system. Here we define different worst-case and average-case LWE problems along with their search and decision versions.

Definition 29 (Search LWE)**.** *Fix a positive integer n and an integer $p \geq 2$. Let χ be a distribution on $\mathbb{Z}_p$. The learning with errors problem $\mathrm{LWE}_{p,\chi}$ is the following. For any $\mathbf{s} \in \mathbb{Z}_p^n$, given samples from $A_{p,\mathbf{s},\chi}$, it is required to output $\mathbf{s}$. Similarly, for a probability density function ϕ on $\mathbb{T}$, the $\mathrm{LWE}_{p,\phi}$ is the following. For any $\mathbf{s} \in \mathbb{Z}_p^n$, given samples from $A_{p,\mathbf{s},\phi}$, it is required to output $\mathbf{s}$.*

Let $\mathcal{A}$ be an algorithm which solves $\mathrm{LWE}_{p,\chi}$, has access to an oracle $W_{s,r}$ (say) where $r \leq \alpha$ is known. $W_{s,r}$ returns independent samples from $A_{s,\phi}$ when queried. $\mathcal{A}$ is allowed to to adaptively query $W_{s,r}$ a number of times and outputs $\mathbf{s}' \in \mathbb{Z}_p^n$. The success of the algorithm $\mathcal{A}$

is measured by the probability that $\mathbf{s}' = \mathbf{s}$. $\text{LWE}_{p,\chi}$ is solvable only if $\alpha < \eta_\epsilon(\mathbb{Z})$. Otherwise, LWE distribution essentially becomes a uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$. This argument is true for the following variants of LWE problems. Only the parameters change as per the problems.

If the number of samples is m, then the problem is denoted as $\text{LWE}_{n,m,p,\chi}$. Similarly, for a probability density function ϕ on $\mathbb{T}$, the $\text{LWE}_{n,m,p,\phi}$ problem is the following. For uniform random $\mathbf{s}$ in $\mathbb{Z}_p^n$, given samples from $A_{p,s,\phi}$, it is required to output $\mathbf{s}$. If the number of samples is m, then the problem is denoted as $\text{LWE}_{n,m,p,\phi}$. When $\phi = \Psi_\alpha$, the problem $\text{LWE}_{n,m,p,\phi}$ is more conveniently written as $\text{LWE}_{n,m,p,\alpha}$.

Next, we define different variants of the decision version of LWE. In the decision LWE problem, the job is to distinguish between two input distributions when samples from either LWE distribution or uniform distribution on the same domain are given as input.

Definition 30 (Decision LWE (Worst-Case))**.** *Let* $\mathbf{s}$ *be an arbitrary element of* $\mathbb{Z}_q^n$*. The worst-case decision version of the LWE problem is to distinguish the uniform distribution over* $\mathbb{Z}_q^n \times \mathbb{T}$ *from* $A_{q,\mathbf{s},\alpha}$ *with probability exponentially close to 1.*

Here in the worst-case problem, the success probability of an algorithm solving the problem has to be exponentially close to 1 and the algorithm has to solve the problem for any arbitrarily chosen value of unknown vector $\mathbf{s}$ from $\mathbb{Z}_q^n$.

Definition 31 (Decision LWE (Average-Case))**.** *The average-case version of the decision LWE problem,* $\text{decLWE}_{n,m,q,\alpha}$*, is to distinguish the uniform distribution* $\mathbb{Z}_q^n \times \mathbb{T}$ *from* $A_{q,\mathbf{s},\alpha}$ *for a non-negligible fraction of all possible* $\mathbf{s}$*, where a list of* m *independent samples of the relevant distribution is provided as input.*

Unlike the worst-case problem, the average-case problem is a bit relaxed. An algorithm solving
$\text{decLWE}_{n,m,q,\alpha}$, has to succeed for a non-negligible portion of all possible unknown vector $\mathbf{s}$ with a non-negligible probability. Regev [Reg09] showed a polynomial time reduction from the worst-case decision LWE problem to the average-case decision LWE problem.

Suppose $\mathbf{s}$ is chosen uniformly at random from $\{0,1\}^n$. The $\text{binLWE}_{n,m,q,\alpha}$ problem is to distinguish the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{T}$ from $A_{q,\mathbf{s},\alpha}$, where a list of m independent samples of the relevant distribution is provided as input. The difference between the decLWE and the binLWE problem lies in the method to select the secret $\mathbf{s}$. Given $n, q \geq 1$ and $\alpha \in (0,1)$, $\text{binLWE}_{n,m,q,\leq\alpha}$ is the problem which requires to solve $\text{binLWE}_{n,m,q,\beta}$ for any $\beta \leq \alpha$ [BLP+13].

2.7.4 Ring LWE Problems

Lyubashevsky, Peikert, and Regev [LPR10] first introduced the ring LWE problem. This is an algebraic variant of LWE problem [Reg09]. Ring LWE problem has been discussed in [LPR10, PRS17, LS15] extensively. Here in this section different variants of ring LWE problems are defined and discussed exclusively.

Definition 32 (Search Ring-LWE Problem)**.** *The search version of* ring-LWE *problem is denoted by* $\text{RLWE}_{q,\leq\alpha}$ *for a real number* $\alpha > 0$ *and* $q \geq 2$ *is the following. For any* $s \in R_q^\vee$*, and a positive real value* $r \leq \alpha$*, given access to arbitrarily many sample from* $A_{s,r}^{(R)}$*, find* s*.*

Definition 33 (Decision Ring-LWE Problem)**.** *Let* $r > 0$ *be a real number and* $q \geq 2$ *be an integer. The decision version of the ring-LWE problem, denoted ring-*$\text{DLWE}_{q,r}$*, is the following. Let* s *be chosen uniformly at random from* $R_q^\vee$*. The task is to distinguish with non-negligible advantage between arbitrarily many independent samples from* $A_{s,r}^{(R)}$ *and the same number of samples generated independently and uniformly from* $R_q \times \mathbb{T}$*, where* $\mathbb{T} = H/\sigma(R^\vee)$*.*

This definition of ring-$\text{DLWE}_{q,r}$ is basically for an average-case problem. In case of a worst-case ring-$\text{DLWE}_{q,r}$ problem, an algorithm solving the problem needs to solve the problem for any arbitrary value of s with probability exponentially close to 1.

2.7.5 Module LWE Problems

Module LWE problems are very similar to ring LWE problems. Inspired by ring LWE, Langlois and Stehlé came up with reductions for the module version of LWE [LS15]. We define different variants of module LWE problems in this section.

Definition 34 (search module-LWE Problem)**.** *The search version of* module-LWE *problem is denoted by* $\text{MLWE}_{q,\leq\alpha}$ *for a real number* $\alpha > 0$ *and* $q \geq 2$ *is the following. For any* $\mathbf{s} \in (R_q^\vee)^d$*, and a positive real value* $r \leq \alpha$*, given access to arbitrarily many sample from* $A_{s,r}^{(M)}$*, find* $\mathbf{s}$*.*

Definition 35 (decision module-LWE Problem)**.** *Let* $r > 0$ *be a real number and* $q \geq 2$ *be an integer. The decision version of the module-LWE problem, denoted module-*$\text{DLWE}_{q,r}$*, is the following. Let* $\mathbf{s}$ *be chosen uniformly at random from* $(R_q^\vee)^d$*. The task is to distinguish with non-negligible advantage between arbitrarily many independent samples from* $A_{s,r}^{(M)}$ *and the same number of samples generated independently and uniformly from* $(R_q)^d \times \mathbb{T}$*, where* $\mathbb{T} = H/\sigma(R^\vee)$*.*

Like ring-LWE problems, this definition of module-DLWE$_{q,r}$ is basically for an average-case problem. In case of a worst-case module-DLWE$_{q,r}$ problem, an algorithm solving the problem needs to solve the problem for any arbitrary value of $\mathbf{s}$ with probability exponentially close to 1.

2.8 Lattice Problems

Here in this section, we define some standard lattice problems which will be used extensively in the later part of the thesis. Standard lattice problems include Short Vector Problem (SVP), Short independent Vector Problem (SIVP), Generalized independent Vector Problem (GIVP), Discrete Gaussian Sampling (DGS), Closest Vector Problem (CVP) and Bounded Distance Decoding Problem (BDD) along with different variants of these problems.

Definition 36 (SVP)**.** *Given an n-dimensional lattice L, find a vector $\mathbf{x} \in L$ such that $||\mathbf{x}|| = \lambda_1(L)$.*

An algorithm solving the SVP problem has to output a non-zero shortest vector from the given lattice. SVP is considered to be a hard problem.

Definition 37 (GapSVP$_\gamma$)**.** *The problem GapSVP$_\gamma$ is the following: An instance is a pair $(\mathcal{B}, d)$, where $\mathcal{B}$ is a basis of an n-dimensional lattice $L = L(\mathcal{B})$ and $d > 0$ is a real number. The instance is a* **YES** *instance if $\lambda_1(L) \leq d$ and it is a* **NO** *instance if $\lambda_1(L) \geq \gamma(n) \cdot d$.*

GapSVP is a decision version of SVP, more precisely, a promise problem version of SVP. A promise problem is a variant of the decision problem. The lattice is presented by a basis $\mathcal{B}$. As per the definition, it is promised that $\lambda_1(L)$ is outside the range of $(d, \gamma(n) \cdot d)$. GapSVP is considered to be a standard lattice problem which is assumed to be hard for Euclidean lattices while this problem is solvable in polynomial time for ideal lattices.

Definition 38 (GapSVP$_{\zeta,\gamma}$)**.** *The problem ζ-to-γ-GapSVP (denoted as GapSVP$_{\zeta,\gamma}$) was introduced in [Pei09]. For functions $\zeta(n) \geq \gamma(n) \geq 1$, an instance of GapSVP$_{\zeta,\gamma}$ is a pair $(\mathcal{B}, d)$, where $\mathcal{B}$ is a basis of an n-dimensional lattice $L = L(\mathcal{B})$ for which $\lambda_1(L) \leq \zeta(n)$, $\min_i \|\widetilde{\mathbf{b}}_i\| \geq 1$, and $1 \leq d \leq \zeta(n)/\gamma(n)$. The instance is a* **YES** *instance if $\lambda_1(L) \leq d$ and it is a* **NO** *instance if $\lambda_1(L) > \gamma(n) \cdot d$.*

GapSVP$_{\zeta,\gamma}$ is considered to be a less standard lattice problem. Though, it has been used to show a classical reduction from a lattice problem to LWE by Peikert [Pei09]. It has been

shown in [Pei09] that for $\zeta(n) \geq 2^{n/2}$, the GapSVP$_{\zeta,\gamma}$ is equivalent to the standard GapSVP$_\gamma$. Showing a classical reduction from a standard lattice problem to the LWE has been one of the open problems till today. Peikert [Pei09] used GapSVP$_{\zeta,\gamma}$ to show the existence of a classical reduction but eventually falls short due to a restriction over the modulo parameter of LWE as the conditions $\zeta(n) \geq 2^{n/2}$, makes the modulo exponential.

Definition 39 (SIVP)**.** *Given an n-dimensional lattice L, the algorithm must output n linearly independent lattice vectors $(v_1, v_2, \ldots, v_n)$ so that $\max||v_i|| \leq \max_{\mathcal{B}}||b_i||$ where the right hand side considers all basis $\mathcal{B} = \{b_1, \ldots, b_n\}$ of the lattice L.*

An algorithm solving the SIVP has to output n shortest independent lattice vectors, given a lattice by a basis $\mathcal{B}$. Basically in SIVP, we need to output the set of lattice vectors corresponding to the set of successive minima $(\lambda_1, \ldots, \lambda_n)$.

From the descriptions of SVP and SIVP, it is evident that these two problems are closely related. If we have an algorithm for SIVP, the SVP problem becomes trivial but the other way is not straightforward as SIVP is strictly harder than SVP.

Definition 40 (SIVP$_\gamma$)**.** *Let $\gamma(n) \geq 1$ be a function from the naturals to the naturals. The problem SIVP$_\gamma$ is the following: An instance is a basis $\mathcal{B}$ of an n-dimensional lattice $L = L(\mathcal{B})$ and the task is to obtain a set of n linearly independent vectors from L whose lengths are at most $\gamma(n) \cdot \lambda_n(L)$.*

SIVP$_\gamma$ is another search variant of SIVP. SIVP$_1$ is SIVP. As $\gamma(n) \geq 1$, SIVP$_\gamma$ is a bit relaxed version of the absolute SIVP problem.

Definition 41 (GIVP$^\varphi_\gamma$)**.** *Given an n-dimensional lattice L, the algorithm must output n linearly independent lattice vectors $v_1, v_2, \ldots, v_n$ so that $\max||v_i|| \leq \gamma(n) \cdot \varphi(L)$ where $\gamma(n) \geq 1$ is the approximation factor and φ denotes an arbitrary real-valued function on the lattice.*

GIVP$^\varphi_\gamma$ is a generalization of SIVP$_\gamma$. GIVP$^\varphi_\gamma$ coincides with SIVP$_\gamma$ when the real-valued function $\varphi(L) = \lambda_n(L)$. The GIVP is considered to be a less standard lattice problem. In Chapter 4 we discuss the reduction from GIVP to LWE problem.

As discussed previously in Section 2.4.9, a fractional ideal $\mathcal{I} \in K$, under canonical embedding $\sigma(\mathcal{I})$ forms a lattice (ideal lattice) in H. In general, we use $\mathcal{I}$ to denote the ideal lattice generated by fractional ideal $\mathcal{I}$. All the lattice problems can be defined similarly on ideal lattices. We define and discuss the ideal lattice problems that are used in this thesis.

Definition 42 (Ideal SVP)**.** *An instance of the γ-approximate shortest vector problem for an n-dimensional number filed K, denoted by K-SVP_γ, is a fractional ideal $\mathcal{I}$ in K and it is required to find a nonzero $x \in \mathcal{I}$ such that $||x|| \leq \gamma \cdot \lambda_1(\mathcal{I})$.*

Here, we try to solve the appropriate SVP problem over the ideal lattice $\mathcal{I}$. Similarly, we can define the SIVP_γ like before but on ideal lattices as follows.

Definition 43 (Ideal SIVP)**.** *An instance of the γ-approximate shortest independent vector problem, denoted by K-SIVP_γ, requires finding n linearly independent elements in $\mathcal{I}$ all of whose norms are at most $\gamma \cdot \lambda_n(\mathcal{I})$.*

Similar to ideal lattices, we define module lattices in Section 2.5.1. Using d canonical embeddings, the map $\Sigma : K^d \to H^d$ is defined. A module $M \in K^d$ forms an $n \cdot d$-dimensional lattice under the map Σ, which we call a module lattice. Here K is an n-dimensional number field. In general, we use M to denote the module lattice generated by module M. We define two module lattice problems that are used in the later part of this thesis.

Definition 44 (module SVP)**.** *An instance of the γ-approximate shortest vector problem for a module M, denoted by M-SVP_γ, is a module lattice M in K^d and it is required to find a nonzero $x \in M$ such that $||x|| \leq \gamma \cdot \lambda_1(M)$.*

Module SVP is the approximate version of SVP on module lattice M. Similarly, we define SIVP on module lattices as follows.

Definition 45 (module SIVP)**.** *An instance of the γ-approximate shortest independent vector problem, denoted by M-SIVP_γ, requires finding n linearly independent elements in M all of whose norms are at most $\gamma \cdot \lambda_n(M)$.*

One of the key problems that are discussed in the lattice reductions is the "Discrete Gaussian Sampling Problem" or the DGS problem in short. In the DGS problem, a lattice L or a basis $\mathcal{B}$ of a lattice is given as input along with a real value r. A discrete Gaussian distribution is defined over the set of lattice points of L with width r. An algorithm solving DGS needs to output a sample from lattice L, according to the discrete Gaussian distribution. The formal definitions of different variants of DGS are as follows.

Definition 46 (DGS)**.** *Let φ be a real-valued function defined on euclidean lattices. The discrete Gaussian sampling problem is denoted by DGS_φ is the following. An instance is a pair $(\mathcal{B}, r)$, where $\mathcal{B}$ is a basis of an n-dimensional lattice $L = L(\mathcal{B})$ and $r > \varphi(L)$ is a real number. The task is to obtain a sample from $D_{L,r}$.*

Similarly, we define ring DGS and module DGS as follows. The ring DGS is the discrete Gaussian sampling over ideal lattices and module DGS is the same over module lattices.

Definition 47 (Ring DGS)**.** *Let Γ be a real-valued function defined on fractional ideals of K. The discrete Gaussian sampling problem in K is denoted by K-DGS_Γ is the following: Given a fractional ideal $\mathcal{I} \in K$ and $r \geq \Gamma(\mathcal{I})$, the task is to obtain a sample from $D_{\mathcal{I},r}$.*

Definition 48 (Module DGS)**.** *Let Γ be a real-valued function defined on module M of K^d. The discrete Gaussian sampling problem in M is denoted by M-DGS_Γ is the following: Given a module $M \in K^d$ and $r \geq \Gamma(M)$, the task is to obtain a sample from $D_{M,r}$.*

The DGS problem in general is considered to be a standard hard lattice problem. In the reductions from lattice problems to LWE of different algebraic variants, the lattice problem is chosen to be different variants of DGS, as defined above. Lattice problems like SIVP or SVP are polynomial time reducible to DGS [Reg09].

Definition 49 (CVP)**.** *Let $\mathcal{B}$ be a basis of an n-dimensional lattice $L = L(\mathcal{B})$ in $\mathbb{R}^n$. For any $\mathbf{t} \in \mathbb{R}^n$, define*

$$\mathsf{dist}(\mathbf{t}, L(\mathcal{B})) \triangleq \min\{||\mathbf{t} - \mathbf{y}|| : \mathbf{y} \in L(\mathcal{B}).$$

The "Closest vector Problem" or CVP is, given a basis $\mathcal{B}$ for a lattice L and $\mathbf{t} \in \mathbb{R}^n$, find $y \in L(\mathcal{B})$, s.t., $\mathsf{dist}(\mathbf{t}, \mathbf{y}) = \mathsf{dist}(\mathbf{t}, L(\mathcal{B}))$.

An algorithm for CVP tries to find out the nearest lattice vector to a given point in the span of the lattice. CVP also comes in different variations. This definition of CVP is the search version. Likewise, we can define decision, promise, etc. versions of CVP, although the search version is enough for our analysis. We need the following version of CVP in most of the lattice-based reductions, analyzed in this thesis.

Definition 50 (BDD)**.** *A variant of the CVP is considered in [Reg09]. An instance is a triplet $(\mathcal{B}, d, \mathbf{x})$, where $\mathcal{B}$ is the basis of an n-dimensional lattice $L = L(\mathcal{B})$, d is a positive real number with $d < \lambda_1(L)/2$, and $\mathbf{x} \in \mathbb{R}^n$ which is within distance d of L. The task is to find the closest lattice point to $\mathbf{x}$.*

This problem is also known as the bounded distance decoding problem [LLM06]. Since $d < \lambda_1(L)/2$, the output closest vector is unique.

Definition 51 (GDP)**.** *Let $L \subset H$ be a lattice and $g > 0$ be a Gaussian parameter, then given a coset $\mathbf{e} + L$ where $\mathbf{e} \leftarrow \mathrm{D}_g$, the Gaussian decoding problem $\mathrm{GDP}_{L,g}$ is to find $\mathbf{e}$.*

2.9 Lattice Results

This section refers to a few results related to lattices that are used in various places in the thesis for our analysis of lattice-based reductions. Here we state and briefly describe the results for readability of the later part of the thesis.

We start with the improved bootstrapping theorem. This theorem is used in all the lattice-based reductions, we talk about in this thesis. Bootstrapping theorem is used to generate lattice vectors from discrete Gaussian distribution with a reasonably large width. The basic goal is to sample lattice vectors from discrete Gaussian distributions with a small width. The first step to achieve this goal is to sample from DGS with large width using the Bootstrapping algorithm then iteratively narrow down the width to get the desired output. Bootstrapping algorithm ensures that sampling from discrete Gaussian with specified width is possible.

Theorem 24 ([GPV08]). *Let $\mathcal{B}$ be an $n \times n$ basis matrix of an n-dimensional lattice $L = L(\mathcal{B})$ and $r \geq \max_i \|\widetilde{b_i}\| \cdot \omega(\sqrt{\log n})$. There exists a sampling algorithm which on input $\mathcal{B}$ and r returns a sample that is within negligible statistical distance from $D_{L,r}$.*

Here $\widetilde{b_i}$'s are GSO of the basis of lattice L with basis $\mathcal{B}$.

The next theorem is a reduction from a search LWE problem to a worst-case decision LWE problem. The width of the Gaussian error distribution in the decision version of LWE is more than that of the search LWE problem. This theorem is used in the analysis of the classical lattice-based reduction in Chapter 7.

Theorem 25 ([MP11, Theorem 3.1]). *Let q have prime factorization $q = {p_1}^{e_1} \cdots {p_k}^{e_k}$ for pairwise distinct* poly(n)*-bounded primes p_i with each $e_i \geq 1$ and let $0 < \alpha < 1/\omega(\sqrt{\log n})$. Let l be the number of prime factors $p_i < \omega(\sqrt{\log n})/\alpha$. There is a probabilistic polynomial time reduction from solving search-$\mathrm{LWE}_{q,\alpha}$ in the worst case with overwhelming probability to solving decision-$\mathrm{LWE}_{q,\alpha'}$ on the average with non-negligible advantage for any $\alpha' \geq \alpha$ such that $\alpha' \geq \omega(\sqrt{\log n})/{p_i}^{e_i}$ for every i, and $(\alpha')^l \geq \alpha \cdot \omega(\sqrt{\log n})^{1+l}$.*

This theorem is also used in Chapter 7. This theorem is particularly a reduction from decision LWE problems to another decision LWE problem with the unknown from the set $(0,1)^n$. The former decision LWE problem has its unknown in $(0,1)^k$, where $k \in O(\sqrt{n})$. Both the LWE problems have the same modulo and an equal number of LWE samples provided to them. The width of the Gaussian error distribution increases in the case of the later LWE problem.

Theorem 26 ([BLP+13, Theorem 4.1]). *Let $k, q \geq 1$, and $m \geq n \geq 1$ be integers, and let $\epsilon \in (0, 1/2), \alpha, \delta > 0$, be such that $n \geq (k+1)\log_2 q + 2\log_2(1/\delta), \alpha \geq \sqrt{\ln(2n(1+1/\epsilon))/\pi}/q$. There exist three (transformation) reductions from $\mathrm{LWE}_{k,m,q,\alpha}$ to $\mathrm{binLWE}_{n,m,q,\leq\sqrt{10n}\alpha}$, such that for any algorithm for the latter problem with advantage ζ, at least one of the reductions produces an algorithm for the former problem with advantage at least*

$$(\zeta - \delta)/(3m) - 41\epsilon/2 - \sum_{p|q,\ p \text{ prime}} p^{-k-1}$$

The next theorem is also referred to in Chapter 7.

Theorem 27 ([BLP+13, Corollary 3.2]). *For any $m, n \geq 1, q \geq q' \geq 1, (B, \delta)$-bounded distribution D over $\mathbb{Z}^n, \alpha, \beta > 0$ and $\epsilon \in (0, 1/2)$ such that*

$$\beta^2 \geq \alpha^2 + (4/\pi)\ln(2n(1+1/\epsilon))\cdot(B/q')^2,$$

there is an efficient reduction from $\mathrm{LWE}_{n,m,q,\leq\alpha}$ to $\mathrm{LWE}_{n,m,q',\leq\beta}$ that reduces the advantage by at most $\delta + 14\epsilon m$

Here a distribution $\mathcal{D}$ over $\mathbb{Z}^n$ is (B, δ)-bounded, for $B, \delta \in \mathbb{R}$, if the probability that $\mathbf{x} \leftarrow \mathcal{D}$ has norm greater than B is at most δ. Corollary 3.2 of [BLP+13] is stated in terms of (B, δ) distribution $\mathcal{D}$. In this context, $\mathcal{D}$ is the uniform distribution over $\{0, 1\}$ which is $(\sqrt{n}, 0)$-bounded.

The next result states that if we sample around n^2 number of samples from a discrete Gaussian distribution over a lattice, the sample set of n^2 samples will contain n linearly independent lattice vectors with probability exponentially close to 1. To achieve this goal the width of the discrete Gaussian distribution has to be more than two times the smoothing parameter of the lattice under consideration. This is an intermediate result for the reduction from SIVP to DGS problem.

Corollary 28 ([Reg09, Corollary 3.16]). *Let L be an n-dimensional lattice and let r be such that $r \geq 2\eta_\epsilon(L)$ where $\epsilon \leq 1/10$. Then, the probability that a set of n^2 vectors chosen independently from $D_{L,r}$ contains no n linearly independent vectors is exponentially small.*

Lemma 29 ([GG00]). *For any constants $c, d > 0$ and any $\mathbf{z} \in \mathbb{R}^n$ with $||\mathbf{z}|| \leq d$ and $d' = d\cdot\sqrt{n/(c\log n)}$, we have $\Delta(U(d'\cdot\mathcal{B}_n), U(\mathbf{z} + d'\cdot\mathcal{B}_n)) \leq 1 - 1/\mathrm{poly}(n)$, where U denotes the uniform distribution over the domain given as input parameter.*

For a real number d and $\mathbf{z} \in \mathbb{R}^n$, the open ball in $\mathbb{R}^n$ centered at $\mathbf{z}$ and of radius d will be denoted as $\mathbf{z} + d \cdot \mathcal{B}_n$. The notation $\mathbf{w} \xleftarrow{\$} \mathbf{z} + d \cdot \mathcal{B}_n$ denotes the choice of a vector $\mathbf{w}$ drawn uniformly from $\mathbf{z} + d \cdot \mathcal{B}_n$. Here U denotes the uniform distribution over the ball of radius d'.

The next result is a polynomial time reduction from the worst-case decisional LWE problem to the average-case decisional LWE problem. Here, we have a distinguisher that distinguishes the LWE distribution from a uniform distribution for a non-negligible portion of all possible values of the LWE unknown with non-negligible probability and the task to form a distinguisher that distinguishes the same two distributions for all possible values of the LWE unknown with probability exponentially close to 1. The cryptographic constructions are based on the parameters of the average case decisional LWE problem. The reduction from worst-case decisional LWE problem to average case decisional LWE problem is the last step in the cascade of reductions showing search LWE problem reduces to average case decisional LWE problem.

Lemma 30 ([Reg09, Lemma 4.1])**.** *Let $n, p \geq 1$ be some integers and χ be some distribution on $\mathbb{Z}_p$. Assume that we have access to a distinguisher W that distinguishes $A_{p,\mathbf{s},\chi}$ from a uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$ for a non-negligible fraction of all possible $\mathbf{s}$. Then there exists an efficient algorithm W' that for all $\mathbf{s}$ accepts with probability exponentially close to 1 on inputs from $A_{p,\mathbf{s},\chi}$ and rejects with probability exponentially close to 1 on inputs from a uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$.*

The last result mentioned here is a reduction from GIVP or SIVP to DGS problem. The reduction is one of the critical reductions to achieve the end-to-end reduction from the standard lattice problem to the decisional LWE problem. This reduction has been analyzed in greater depth in Chapter 4.

Lemma 31 ([Reg09, Lemma 3.17])**.** *For any $\epsilon = \epsilon(n) \leq 1/10$ and any $\phi(L) \geq \sqrt{2}\eta_\epsilon(L)$, there is a polynomial time reduction from $\mathrm{GIVP}_{2\sqrt{n}\phi}$ to DGS_ϕ, where L is the lattice under consideration.*

Chapter 3

Brief Literature Survey

3.1 Introduction

In this chapter, we present a concise literature survey relevant to the focus of this thesis, which revolves around lattice-based cryptography and its related challenges. The primary focus of this thesis is on lattice-based cryptography and the issues it addresses. This chapter aims to provide the reader with ample references to initiate their study in the field of lattice-based cryptography. The survey begins by exploring the need for lattice-based cryptography as a viable post-quantum cryptographic solution. We look at the security concerns surrounding conventional number-theoretic cryptographic schemes and the potential impact of quantum computing on their security. As a result, lattice-based constructions emerge as promising candidates due to the absence of known efficient quantum algorithms for certain lattice-based computational problems

Conventional number-theoretic cryptography, such as the Diffie-Hellman protocol [DH76], ElGamal cryptosystem [Gam84], and RSA cryptosystem [RSA78], relies on the presumed hardness of problems like the computational Diffie-Hellman problem or integer factorization in certain groups, and the RSA assumption. However, Shor's [Sho97] polynomial-time quantum algorithm for prime factorization or discrete log problem poses a theoretical threat to these cryptosystems. This breakthrough result implies that in a future where large-scale quantum computers become a reality, conventional number-theoretic systems will be insecure. On the other hand, lattice-based constructions offer promising solutions for post-quantum cryptography as there are currently no known efficient quantum algorithms for some computational problems based on lattices. This makes cryptographic constructions relying on the hardness of these lattice problems viable candidates for achieving security in a post-quantum world. Lattice-based constructions are characterized by their algorithmic simplicity and high parallelizability, involving linear operations on vectors and matrices modulo relatively small integers. Notably, Ajtai's [Ajt96] seminal work demonstrated the asymptotic equivalence between worst-case and average-case lattice problems, while the work of Cai and Nerurkar[CN97] provided further confidence in this equivalence. Consequently, there exists the possibility of cryptographic constructions that achieve security based on the hardness of

worst-case lattice problems, at least in an **asymptotic** sense.

In this thesis, we closely observe the hardness of a few hard lattice problems, viz, SVP, CVP, SIVP, etc. In the following sections, we provide a few major algorithms for these problems.

3.2 Algorithms for hard lattice problems

The lattice problems under consideration are the search and the decision versions of SVP and CVP. Informally in SVP, the job is to find a lattice vector of the shortest length, given a lattice, and in CVP, we try to find out the lattice vector, nearest to a given point over the span of the lattice. These problems are studied extensively over the last few decades and no polynomial-time classical or quantum algorithm has been found yet. Hence, these problems appear to be intractable. Few approximation algorithms have been discovered but they are also intractable for constant and polynomial approximation factors.

3.2.1 Shortest Vector Problem (SVP)

It is known that SVP is NP-hard [Ajt98] under randomized reductions to solve exactly and also NP-hard to approximate under randomized reductions within at least constant factors [DKS98, HR07, Kho05, Kho10, Mic00]. We can split the solutions of SVP into two classes of algorithms requiring super-exponential time $2^{\omega(n)}$ and poly(n) memory, and algorithms requiring both exponential time and space $2^{\Theta(n)}$ in the lattice dimension.

The first class of algorithms mostly includes lattice enumeration by Kannan [Kan83], Phost [Poh81], Fincke and Pohst [FP85] and by Schnorr and Euchner [SE94]. The first class of algorithms also includes random sampling reduction methods like by Schnorr [Sch03] and by Aono and Nguyen [AN17].

The second class of algorithms includes lattice sieving like by Ajtai, Kumar, and Sivakumar [AKS01], by Micciancio and Voulgaris [MV10], by Becker et al. [BDGL15] etc. This class of algorithms also contains methods by computing the Voronoi cell of the lattice [AEVZ02], [MV10] and methods by discrete Gaussian sampling [ADRS15]. A hybrid SVP algorithm using sieving, and enumeration proposed by Doulgerakis, Laarhoven, and Weger [DLdW20].

To solve the γ -approximation version SVP$_\gamma$ for $\gamma > 1$, the best-known approaches are based on using lattice basis reduction. For large $\gamma = 2^{\Omega(n)}$, the Lenstra-Lenstra-Lovász

(LLL) [LLL82] algorithm can find a solution in time polynomial in the lattice dimension. Like LLL, the variants of LLL [Sch87] and [AKS01] obtain sub-exponential approximation factors.

Schnorr [Sch91] showed that deterministic polynomial time algorithms can solve the GapSVP$_\beta$ for $\beta = 2^{O(n(\log\log n)^2/\log n)}$, whereas Ajtai, Kumar, and Sivakumar [AKS01] showed that probabilistic algorithms can achieve a slightly better approximation factor of $\beta = 2^{O(n\log\log n/\log n)}$.

Few known algorithms obtain polynomial or better approximation factors such as [AKS01, Kan83, MV10, ADRS15, KS01] either require super-exponential time or exponential time and space.

We discuss two well-known algorithms below briefly. First is the LLL and second is the greedy sieving algorithm. More detailed analysis of these algorithms can be found in lecture notes by Regev [Reg04].

LLL Algorithm

Lenstra-Lenstra-Lovasz (LLL) [LLL82] algorithm is an approximation algorithm for the SVP. The LLL algorithm runs in polynomial time and outputs an approximation within an exponential factor of the correct answer. It is a practical method with enough accuracy in solving integer linear programming, factorizing polynomials over integers, and breaking cryptosystems. Let $\mathcal{B} = \{b_1, b_2, \ldots, b_n\}$ be an n dimensional basis for a lattice L. Let $\{\widetilde{b}_1, \widetilde{b}_2, \ldots, \widetilde{b}_n\}$ be set of vectors we get after performing Gram-Schmidt Orthogonalization (GSO)[See Section 2.2.1] over the vectors of $\mathcal{B}$. We can write the following.

- $\{b_1, b_2, \ldots, b_n\} \xrightarrow{\text{GSO}} \{\widetilde{b}_1, \widetilde{b}_2, \ldots, \widetilde{b}_n\}$
- $\mu_{i,j} = \frac{\langle b_i, \widetilde{b}_j\rangle}{\langle \widetilde{b}_i, \widetilde{b}_j\rangle}$
- $\widetilde{b}_i = b_i - \mu_{i,1}\widetilde{b}_1 - \cdots - \mu_{i,i-1}\widetilde{b}_{i-1}$

Let the basis $\mathcal{B}$ of the lattice L is said to be δ-LLL reduced if

1. $|\mu_{i,j}| \leq \frac{1}{2}$, for $1 \leq j < i \leq n$.
2. $\delta||\widetilde{b}_i||^2 \leq ||\mu_{i+1,i}\widetilde{b}_i + \widetilde{b}_{i+1}||^2 = \mu_{i+1,i}^2||\widetilde{b}_i||^2 + ||\widetilde{b}_{i+1}||^2$ (as $\widetilde{b}_i$ and $\widetilde{b}_{i+1}$ are orthogonal vectors), which implies that
$$||\widetilde{b}_{i+1}||^2 \geq (\delta - \mu_{i+1,i}^2)||\widetilde{b}_i||^2.$$

Thus, the goal of the LLL algorithm is the following. Given a basis for a lattice, transform the basis to one which is LLL-reduced.

Claim 32. *Let $b_1, \ldots, b_n$ be a δ-LLL reduced basis for a lattice L. Then $||b_1|| \leq (\frac{2}{\sqrt{4\delta-1}})^{n-1}\lambda_1(L)$.*

The algorithm works for $1 \leq \delta \leq \frac{1}{4}$. If we take $\delta = 3/4$, the above equation simplifies to $||b_1|| \leq 2^{\frac{n-1}{2}}\lambda_1(L)$. Thus b_1 is an approximation for the shortest vector within an exponential approximation factor. Now we present the LLL algorithm in the following manner.

Algorithm 1 LLL Algorithm

An integer basis $\mathcal{B} = \{b_1, \ldots, b_n\}$, i.e., $b_1, \ldots, b_n \in \mathbb{Z}^m$;
Start Compute $\widetilde{b}_1, \ldots, \widetilde{b}_n$ from $\mathcal{B}$ performing Gram-Schmidt Orthogonalization;
Reduction Step:
for $i \leftarrow 2$ down to n **do**
    for $j \leftarrow i-1$ down to 1 **do**
        if $|\mu_{i,j}| > \frac{1}{2}$ **then**;
            $q_{i,j} \leftarrow \left\lfloor \frac{\langle b_i, \widetilde{b}_j \rangle}{\langle \widetilde{b}_j, \widetilde{b}_j \rangle} \right\rceil$;
            $b_i \leftarrow b_i - q_{i,j} b_j$;
        end if
    end for
end for
Swap Step:
if $\exists$ a i, s.t. $\delta||b_i||^2 > ||\mu_{i+1,i}\widetilde{b}_i + \widetilde{b}_{i+1}||^2$ **then**;
    swap b_i and b_{i+1} **go to** the **Start** step;
end if

Note that when the algorithm terminates it returns a δ-LLL reduced basis. Next, we discuss the greedy sieving algorithm

Greedy Sieving Algorithm

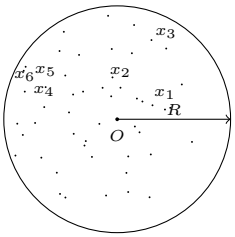
The greedy sieving Algorithm for finding the shortest vector of a lattice is an exact randomized algorithm for SVP in time $2^{\mathcal{O}(n)}$. This algorithm is due to Ajtai, Kumar, and Sivakumar [AKS01]. There is a deterministic $\mathcal{O}(2^{n \log n})$ algorithm by Kannan [Kan87] We start by noting that it is enough to solve SVP for instances where the length of the shortest vector is in some known range.

Lemma 33. *Given an algorithm $\mathcal{A}$ that finds a shortest nonzero vector in lattices for which*

$$2 \leq \lambda_1(L(\mathcal{B})) < 3,$$

we can find a shortest nonzero vector in any lattice in time that is greater by a factor of at most $\mathcal{O}(n)$.

Greedy Seiving Let $R > 0$. Suppose there are N points in $\mathcal{B}(0, R)$. The algorithm will mark some points as 'centers' such that, for any of the N points there will be a center within distance at most $R/2$.



Let the points be $x_1, \ldots, x_N$. Then the algorithm computes a function $\eta : \{1, \ldots, N\} \to \{1, \ldots, N\}$, s.t., $x_{\eta(i)}$ is the center corresponding to x_i. Following is the algorithm for "Greedy Sieving".

Algorithm 2 Greedy Seive

1: $x_1, \ldots, x_N$.
2: 'Grow" a set of centers $\mathcal{C}$.
3: $\mathcal{C} \leftarrow \{x_1\}$
4: **for** $x_i \in \{x_1, \ldots, x_N\}$ **do**
5: **if** $\exists\ j$, s.t., $x_j \in \mathcal{C}$ and $||x_i - x_j|| < R/2$ **then**
6: $\eta(i) = j$
7: **else** $\eta(i) = i$ and add x_i to $\mathcal{C}$
8: **end if**
9: **end for**

Lemma 34. *The number of centers returned by Algorithm 2 is* $\leq 5^n$.

We now describe the SVP algorithm.

3.2.2 Closest Vector Problem (CVP)

The closest vector problem (CVP) is at least as hard as SVP. Goldreich et al. [GMSS99] showed that hardness of approximate CVP implies hardness of approximate SVP. Ajtai, Kumar, and Sivakumar [AKS02] showed a Turing reduction from SVP to CVP. Though the

Algorithm 3 Greedy Seiving Algorithm for SVP

Input: $\mathcal{B}$, s.t., $\lambda_1(L(\mathcal{B})) \in [2, 3)$;
Output: v, s.t., $||v|| = \lambda_1(L(\mathcal{B}))$;
$R_0 \leftarrow n \times \max_i ||b_i||$;
Choose $N = 2^{8n} \log R_0$ points $x_1, \ldots, x_N$ independent and uniform at random from $\mathcal{B}(0, 2)$.;
Compute $y_i \equiv x_i \mod \mathcal{P}(\mathcal{B})$. Let $\mathcal{Z} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$;
$R \leftarrow R_0$;
while $(R > 6)$ **do**;
    Apply the sieving algorithm to the y vectors in $\mathcal{Z}$. The result is a set $\mathcal{C}$ of at most 5^n centers and a map $\eta : \{0, \ldots, N\} \mapsto \{0, \ldots, N\}$, s.t., $||y_i - y_{\eta(i)}|| \leq R/2$. Let $\mathcal{J} = \{(x_i, y_i) : i \in \mathcal{C}\}$. Then;
    $\mathcal{Z} \leftarrow \mathcal{Z} \backslash \mathcal{J}$;
    For (x_i, y_i) which remains in $\mathcal{Z}$, set (x_i, y_i) to $(x_i, y_i - (y_{\eta(i)} - x_{\eta(i)}))$. Note that $x_i - \{y_i - (y_{\eta(i)} - x_{\eta(i)})\} = \underbrace{(x_i - y_i)}_{L(\mathcal{B})} + \underbrace{(y_{\eta(i)} - x_{\eta(i)})}_{L(\mathcal{B})} \in L(\mathcal{B})$. $||y_i - (y_{\eta(i)} - x_{\eta(i)})|| = ||y_i - y_{\eta(i)} + x_{\eta(i)}|| \leq ||y_i - y_{\eta(i)}|| + ||x_{\eta(i)}|| \leq R/2 + 2$ [Since, $x_{\eta(i)} \in \mathcal{B}(0, 2)$];
    $R \leftarrow R/2 + 2$;
end while;
For all pairs $(x_i, y_i), (x_j, y_j) \in \mathcal{Z}$, consider the difference $(x_i - y_i) - (x_j - y_j)$ and output the shortest such vector.;

approximate version of CVP is also at least as hard as the approximate version of SVP but from the practical perspective, both are considered to be equally hard [Yas21] which is due to the embedding of Kanan [Kan87].

The hardness of the closest vector problem was thoroughly analyzed by Micciancio, which can be found in [Mic01a]. Babai's [Bab86] "Nearest Plane Algorithm" is one of the famous approximation solutions for CVP with approximation factor $2(2/\sqrt{3})^n$, where n is the lattice dimension. Aggarwal, Dadush, and Davidowitz's [ADS15] algorithm solves CVP in $2^{n+\mathcal{O}(n)}$ time and time. This is a randomized algorithm to solve exact CVP over Euclidean lattices. An approximate Voronoi cells-based solution for CVP is proposed by Doulgerakis, Laarhoven, and Weger [DLdW19]. Dadush, Regev, and Davidowitz's [DRS14] approximate algorithm for CVP with approximation factor $O(n/\sqrt{\log n})$ is better than the previously best-known algorithm for CVP with preprocessing by Lagarias, Lenstra Jr. and Schnorr [LJS90] which has approximation factor $O(n^{1.5})$.

Babai's Nearest Plane Algorithm

Let $\mathcal{B}^{m\times n}$ be the basis of a lattice $L \in \mathbb{R}^m$ and $\mathbf{t} \in \mathbb{R}^m$, find $\mathbf{x} \in L$, s.t.,

$$\mathsf{dist}(\mathbf{t}, \mathbf{x}) \leq 2^{n/2}\mathsf{dist}(\mathbf{t}, L(\mathcal{B})).$$

This is called the **Babai's nearest plane algorithm** [Bab86]. This is a polynomial time approximation algorithm for CVP with an exponential approximation factor. We present Babai's nearest plane algorithm in an algorithmic form in the following.

Algorithm 4 Babai's nearest plane algorithm.

1: **Input:** An integer basis $\mathcal{B} = \{b_1, \ldots, b_n\}$, i.e., $b_1, \ldots, b_n \in \mathbb{Z}^m$ and $\mathbf{t} \in \mathbb{Z}^m$;
2: **Output:** $\mathbf{x} \in L$, s.t., $\mathsf{dist}(\mathbf{t}, \mathbf{x}) \leq 2^{n/2}\mathsf{dist}(\mathbf{t}, L(\mathcal{B}))$.;
3: Run $\frac{3}{4}$-LLL on $\mathcal{B}$ and the output is over-written as $\mathcal{B} = \{b_1, \ldots, b_n\}$.
4: $\mathcal{B} = \{b_1, \ldots, b_n\} \xrightarrow{\text{GSO}} \{\tilde{b}_1, \ldots, \tilde{b}_n\}$
5: $\mathbf{b} \leftarrow \mathbf{t}$
6: **for** $j \leftarrow n$ down to 1 **do**
7: $\quad b \leftarrow b - cb_j$, where $c = \left\lfloor \frac{\langle b, \tilde{b}_j \rangle}{\langle \tilde{b}_j, \tilde{b}_j \rangle} \right\rceil$
8: **end for**
9: Return $(\mathbf{b} - \mathbf{t})$

3.3 Learning with Error Problem

The "Learning with error" problem or the "LWE" is a generalized version of the "Learning with parity with noise" problem or the "LPN" problem. The definition of LWE can be found in Section 2. It is the solving of a system of linear equations with errors. Generally, the Gaussian elimination method is used to solve the unknown for a system of linear equations when there is no error. In this case, the time required for solving the problem is polynomial in the problem parameter. The problem becomes significantly hard for non-zero error. We discuss extensively the reduction from hard lattice problems to the LWE problem. The LWE-based cryptographic schemes are the central object of this thesis. These problems have been studied extensively and are believed to be intractable.

A naive algorithm to solve LWE is to use the Gaussian elimination method to find a solution with low confidence and repeat the process exponentially many times to build confidence in the solution. This algorithm asymptotically requires $2^{O(n \log n)}$ samples and time.

The maximum likelihood algorithm [Muk20] performs better than the naive algorithm to solve LWE. It uses the idea that the approximate solution to the LWE using $O(n)$ equations is the correct one. That is why running time is exponential but only requires polynomial-many equations. This algorithm asymptotically requires $O(n)$ samples and $2^{O(n \log n)}$ time.

Blum, Kalai, and Wasserman [BKW03] (BKW) made a critical improvement to the LWE algorithms. Their algorithm requires $2^{O(n)}$ both time and samples asymptotically. In this algorithm, the n unknowns are partitioned into $(\log n)$ blocks of size $(n/\log n)$ each. Then it constructs the unknown vector recursively by finding a collision in $\log n$ blocks. Albrecht et al. [ACF+15] performed a concrete analysis of the asymptotic complexity of the BKW algorithm. Duc, Tramèr, and Vaudenay's [DTV15] algorithm for LWE is an improvement over Albrecht et al. [AFFP14]. Albrecht et al. [AFFP14] is a variant of the original BKW algorithm where the focus is on the secrets of small length in reduced time complexity.

Arora and Ge [AG11] showed a sub-exponential time algorithm for LWE and derived a relation between the error parameter and the time taken by the algorithm. The LWE problem is used to show the hardness of generalized learning problems [KS06] using the celebrated result of Regev [Reg05]. The cryptographic applications of LWE are [ACPS09, BFKL93, HB01, GRS08].

3.4 Cryptographic Schemes

In this section, we enlist a few of the renowned cryptographic schemes which are considered to be the pioneer and significant for the field of lattice-based cryptography. Though the content is not exhaustive, we provide enough schemes which are proposed over the last three decades.

The first lattice-based cryptographic scheme was proposed in 1996 by Ajtai [Ajt96]. Thereafter Ajtai and Dwork [AD97] gave the first lattice-based public key encryption scheme which has asymptotic worst-case security assurance which is considered the main flavor for lattice-based cryptosystems by many researchers. The collision-resistant hash function (CRHF) from lattices by Goldreich, Goldwasser, and Halevi's [GGH96] also came out subsequently. This security of the CRHF is based on the worst-case hardness of $O(n^3)$-approximate SIVP. Regev's [Reg03] early public key cryptosystem is also based on a hard lattice problem unique-SVP. The equivalence between unique-SVP and the standard lattice problem like GapSVP is proved by this results [LM09, Pei09]. Goldreich, Goldwasser, and Halevi

(GGH) [GGH97] proposed another public key encryption and a digital signature scheme whose security is based on the hardness of CVP. GGH has public key size and encryption time both in $O(n^2)$ where n is the security parameter. The work of Goldreich, Goldwasser, and Halevi was inspired by Ajtai [Ajt96] and McElieces code-based cryptosystem [McE78].

In 1998, Hoffstein, Pipher, and Silverman came up with NTRU [HPS98] which is a cryptographic scheme based on the lattice and uses polynomial rings. The famous NTRU is different than the other cryptographic scheme based on the lattice as NTRU has no security proof but it is efficient from a practical point of view. Recently a variant of NTRU is proven secure asymptotically by Stehlé and Steinfeld [SS11]. Based on the ideas of NTRU and GGH, two lattice-based digital signature schemes have been proposed [HPS01, HHP+03]. Nguyen performed the cryptanalysis of the GGH encryption scheme for practical parameters [Ngu99]. Later digital signature scheme of GGH and NTRU was broken by Nguyen and Regev [NR06, NR09]. Micciancio proposed the result [Mic01b] to improve lattice-based cryptosystems using the Hermite normal form.

Regev's [Reg05] work is considered to be groundbreaking in the field of lattice-based cryptography. He introduced the well-known "Learning with Errors" (LWE) problem and gave the first public key encryption scheme based on the hardness of LWE. Few other public key encryption schemes other than [Reg05] which are secure against chosen plaintext attacks (CPA) are [KTX07, PVW08]. LWE-based public key encryption schemes which are secure against chosen ciphertext attacks (CCA) are [PW08, Pei09]. Peikert, Vaikuntanathan, and Waters [PVW08] proposed an oblivious transfer protocol based on LWE. Few of the identity-based encryption schemes which are based on LWE are [CHKP10, ABB10, GPV08],

Regev showed that the average-case LWE problem is as hard as the worst-case lattice problems. This reduction from LWE to lattice problem is quantum in nature. Piekert [Pei09] provided a classical algorithm for lattice reduction but the reduction has serious limitations. The lattice problem it considers is not a standard lattice problem. If we make the lattice problem standard, the LWE modulo becomes exponential. Though it was a stepping stone to making the reduction fully classical. Brakerski, Langlois, Peikert, Regev, and Stehlé [BLP+13] proposed the improved classical reduction from LWE to lattice problem. Cryptographic constructions based on [Reg05] had two limitations, viz, speed and key size. This problem was answered by Lyubashevsky, Peikert, and Regev [LPR13] in a positive manner by using ideal lattice, an algebraic variant of lattices. Though cryptographic constructions based on [LPR13] are efficient and speedy, they are not as secure or versatile as the cryptographic constructions based on [Reg05]. Stehlé, Steinfeld, Tanaka, and Xa-

gawa's [SSTX09] is an example of public key encryption based on the hardness of ideal lattices. The gap between the two reductions [Reg05] and [LPR13] was eliminated by Langlois and Stehlé [LS15] by using module lattices.

Banerjee, Peikert, and Rosen [BPR12] came up with the concept of learning with rounding or LWR which is a deterministic error variant of LWE where the Gaussian error of LWE is replaced by rounding. LWR-based protocols do have the asymptotic worst to average case hardness guarantee [BPR12] and LWR is at least as hard as LWE asymptotically [AKPW13, BPR12, BGM+16, PRS17]. LWE-based proposals which are in the second round of the NIST process are Frodo [ABD+19, BCD+16], Kyber [ABD+09, BDK+18], LAC [LLJ+19, LLZ+18], NewHope [AAB+19, ADPS16], NTRU Prime [BCLvV17], Round5 [BBF+19a, BBF+19b, SBG+18], and Saber [DKRV19, DKRV18, KMRV18, BDK+21, KDB+22, GMK+22]. Kyber [ABD+09] is selected as the finalist in the fourth round and it is a module LWR-based public key encryption scheme where the security is based on the hardness of module lattices.

3.5 Concrete Analysis

Concrete security analysis of cryptographic schemes is often overlooked by a large portion of the cryptographic community. In this thesis, we focus on the tightness gap of reductions for lattice-based cryptographic constructions. This is one of the important aspects of concrete security analysis in cryptography. Bellare [Bel97, Bel98] was a pioneer in pointing out the need for concrete security as part of practice-oriented provable security for both symmetric and public key cryptographic constructions. Subsequently, Bellare, Desai, Jokipii, and Rogaway [BDJR97] studied symmetric key encryption schemes from a concrete security point of view and analyzed the concrete complexity of reductions among different notions of security. Bellare, Canetti, and Krawczyk [BCK96] analyzed the concrete security of pseudorandom functions, also [BR09] this work performs concrete security for Waters' IBE scheme. Koblitz and Menezes re-examined "Provable Security" from the view of concrete security in [KM07, KM06]. A similar kind of approach and analysis can be found in [KM08, Men12, CMS11, KM19]. Concrete security in lattice-based reduction was first talked about in [CKMS16] by Chatterjee, Koblitz, Menezes, and Sarkar. The tightness gap for lattice-based reductions is mainly focused in [CKMS16]. Walter [Wal17] studied extensively the concrete security of lattice-based cryptography.

Chapter 4

Quantum Reduction from GIVP to LWE

4.1 Introduction

In Regev's work [Reg09], he presented a polynomial-time reduction from the GIVP (a variant of the Shortest Independent Vectors Problem), a worst-case lattice problem to the average-case decisional LWE problem. In this chapter, we focus on evaluating the tightness of this reduction. This reduction is composed of multiple sub-reductions, each contributing to the overall tightness gap. We meticulously examine each sub-reduction and explore opportunities for optimization where feasible. By analyzing the tightness gap of each sub-reduction and combining them, we can derive the tightness gap as a function of lattice dimension, LWE error terms, and other relevant parameters. The security of LWE-based cryptosystems relies on the hardness of the average-case decisional LWE problem. As the parameters of the cryptosystem are directly linked to those of LWE, the tightness gap can be measured directly for a given cryptosystem. This measurement allows us to assess the practical feasibility of the cryptosystem. It is important to note that Regev's reduction involves a quantum step alongside classical ones. The quantum part significantly influences the overall tightness gap. In this chapter, we estimate the tightness gap of Regev's reduction, and in the concluding section, we discuss in detail the implications of a high tightness gap.

4.1.1 Outline of the Analysis

The reductions in [Reg09] are divided into three parts. The first part is a reduction from approximate GIVP to the DGS problem. The second part is the reduction between DGS to search LWE problem, while the third part describes the reduction from the search LWE problem to the decision LWE problem. The following sections describe the concrete analysis of each part. To obtain the end-to-end reduction from the approximate GIVP to decision LWE, we combine the three parts in the concluding section.

4.1.2 Parameters of the Reductions

We need to fix three parameters, a positive integer n that will denote the degree of the underlying lattice L; an integer $p \geq 2$ that we use to define the LWE problem; and a positive real number α such that $\alpha p \geq 2\sqrt{n}$. In the asymptotic setting, p and α are considered to be functions of n.

4.2 Reduction from GIVP to DGS

The first reduction (Lemma-3.7 of [Reg09]) is between two lattice problems, namely GIVP and DGS. We state the theorem here.

Theorem 35. *For any lattice L,$\epsilon = \epsilon(n) \leq \frac{1}{10}$ and any $\varphi(L) \geq \sqrt{2}\eta_\epsilon(L)$, there is a polynomial time reduction from* $\mathrm{GIVP}_{2\sqrt{n}}^{\varphi(L)}$ *to* $\mathrm{DGS}_{\varphi(L)}$.

The GIVP to DGS reduction follows from the following algorithms, $\mathcal{A}_0$ and $\mathcal{A}_1$ where $\mathcal{A}_0$ calls $\mathcal{A}_1$. We want to determine the number of times $\mathcal{A}_0$ calls $\mathcal{A}_1$. We briefly describe the algorithms below.

Algorithm $\mathcal{A}_0$: Solves $\mathrm{GIVP}_{2\sqrt{n}}^{\varphi(L)}$ for $\varphi(L) \geq \sqrt{2}\eta_\epsilon(L)$. The input to this algorithm is a lattice L. The output is a set of n linearly independent vectors of L and the longest of which is at most $2\sqrt{n}\varphi(L)$.

Algorithm $\mathcal{A}_1$: Solves $\mathrm{DGS}_{\varphi(L)}$ for $\varphi(L) \geq \sqrt{2}\eta_\epsilon(L)$. The input is a pair (L, r), where L is a lattice and $r \geq \varphi(L)$. The output is a sample from the distribution $D_{L,r}$.

We review the proof here. The objective is to output a set of n linearly independent vectors whose longest vector has length at most $2^n\lambda_n(L)$. The algorithm $\mathcal{A}_0$ uses LLL [LLL82] algorithm to obtain a set B_0 of n linearly independent vector such that the output vectors will have length at most $2\sqrt{n}\varphi(L)$. If d_0 is the length of the longest vector in B_0 then it will satisfy that $\lambda_n(L) \leq d_0 \leq 2^n\lambda_n(L)$ For $i = 0, \ldots, 2n$, let $u_i = d_0/2^i$. For each i in $\{0, \ldots, 2n\}$ algorithm $\mathcal{A}_0$ does the following. It invokes $\mathcal{A}_1$ a total number of n^2 times on the input (L, u_i) to obtain a set S_i of n^2 elements of L chosen independently from D_{L,u_i}. $\mathcal{A}_0$ looks for a set B_i of n linearly independent vectors in each of S_i and outputs the shortest set found.

We claim that with high probability $\mathcal{A}_0$ returns n linearly independent vectors whose longest vector length is at most $2\sqrt{n}\varphi(L)$. Firstly if $\varphi(L) \geq d_0$ then length of vectors in B_0 are at most $2\sqrt{n}\varphi(L)$ trivially. On the other hand, it can be shown that there exists an i in $\{0, \ldots, 2n\}$ such that $\varphi(L) < u_i \leq 2\varphi(L)$, using the lemma 19. The proof for the existence of such an i is as follows.

Proof. Here $\varphi(L) \geq \sqrt{2}\eta_\epsilon(L)$ and $\eta_\epsilon(L) \geq \sqrt{\frac{\ln 1/\epsilon}{\pi}}.\frac{\lambda_n(L)}{n}$ with $\epsilon \leq \frac{1}{10}$. Hence, we get $\varphi(L) \geq \sqrt{\frac{2\ln 10}{\pi}}.\frac{\lambda_n(L)}{n}$, which evaluates to $\varphi(L) \geq 1.21 \cdot \frac{\lambda_n(L)}{n}$.

Assume that $u_i > \varphi(L)$, which implies that $u_i = d_0 2^{-i} > \varphi(L)$

$$\begin{aligned} &\implies d_0 2^{-i} > \varphi(L) \\ &\implies \lambda_n(L) 2^{n-i} > \varphi(L) \\ &\implies \lambda_n(L) 2^{n-i} > 1.21 \cdot \frac{\lambda_n(L)}{n} \\ &\implies 2^{n-i} > \frac{1.21}{n} \end{aligned}$$

If $i = 0$, the relation $2^{n-i} > \frac{1.21}{n}$ holds for all $n > 0$ and if $i = 2n$, the relation $2^{n-i} > \frac{1.21}{n}$ does not hold for $n > 0$. Again $u_i = 2 \cdot u_{i+1}$, so the values of u_i are decreasing for $i \in [0, 2n]$. Hence we will get an i, for which $u_i > \varphi(L)$ and $u_{i+1} \leq \varphi(L)$, or equivalently $\varphi(L) < u_i \leq 2\varphi(L)$.

Let $\hat{i}$ be the value of i for which $\varphi(L) < u_{\hat{i}} \leq 2\varphi(L)$. Corollary 28 ensures that when we choose n^2 vectors independently from $D_{L,r}$, we get n linearly independent vectors with probability exponentially close to 1. The requirement of this result is to have $r \geq \sqrt{2}\eta_\epsilon$ and $\epsilon \leq 1/10$. In this particular case of ours, we choose n^2 vectors independently from $D_{L,u_{\hat{i}}}$. The two constraints $\epsilon \leq 1/10$ and $u_{\hat{i}} \geq \varphi(L) \geq \sqrt{2}\eta_\epsilon$ are also satisfied. So, we will get n linearly independent vectors from $D_{L,u_{\hat{i}}}$. Further, with high probability, the n linearly independent vectors are of length at most $u_{\hat{i}}\sqrt{n}$ due to Lemma 21, which states that discrete gaussian samples are of length at most $\sqrt{n}r$, where r is the width of the Gaussian distribution. Again, we have $u_{\hat{i}} \leq 2\varphi(L)$, which implies that output vectors for $\hat{i}$ is less than $2\sqrt{n}\varphi(L)$. □

Proposition 36. *We record the following results.*

1. $\mathcal{A}_0$ *invokes* $\mathcal{A}_1$ *a total of* n^3 *times. Hence the tightness gap of this reduction, namely*

GIVP to DGS is n^3.

2. *Based on the DGS parameter that is used in 37,(Section 4.3) $r \geq \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$, we may take the DGS parameter $\varphi(L) = \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$, as n is positive integer and $\alpha < 1$ and $\varphi(L)$ satisfies the condition that $\varphi(L) \geq \sqrt{2}\eta_\epsilon(L)$.*

4.3 From DGS to LWE

Reduction from DGS to LWE is the pivotal result of [Reg09]. We analyze the result in greater detail in this section from the perspective of tightness. The following is a restatement of Theorem 3.1 of [Reg09] which is the main result of [Reg09].

Theorem 37. *Let L be a lattice of dimension n, $p \geq 2$ be an integer and $\alpha \in (0, 1)$ be a real number. Given an oracle for $\mathsf{LWE}_{p,\Psi_\alpha}$, it is possible to sample from $D_{L,r}$ where $r \geq \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$, $\alpha p > 2\sqrt{n}$.*

For $r \geq \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$, define $r_i = r \cdot (\alpha p/\sqrt{n})^i$ for $i = 1, \ldots, 3n$. The proof of the theorem is provided in [Reg09] as a sequence of nested oracle calls. In the following, we rewrite the oracle calls and the other computations required for the proof in [Reg09] in an algorithmic form. The required subroutines and data structures are as follows. Let I be a polynomial in n, which is the dimension of the lattice L.

$\mathsf{solveLWE}_{p,\Psi_\alpha}(\mathcal{I})$**:** This is the oracle to solve $\mathsf{LWE}_{n,I,p,\Psi_\alpha}$. The list $\mathcal{I}$ consists of I samples from $A_{p,\mathbf{s},\Psi_\beta}$ for some $0 < \beta \leq \alpha$. Note that the oracle is guaranteed to work correctly if $\beta = \alpha$, otherwise it might return an incorrect result.

$\mathsf{verifyLWE}(\mathbf{s}', \mathcal{I})$**:** The input $\mathcal{I}$ contains I samples from $A_{\mathbf{s},\Psi_\beta}$. This algorithm returns true if $\mathbf{s} = \mathbf{s}'$, otherwise it returns false.

$\mathsf{solveCVP}^{(p)}(L^*, \mathcal{L}, \mathbf{z})$**:** Here L^* is the dual lattice of L; $\mathcal{L}$ contains I samples from D_{L,r_i} for some $i \in \{1, \ldots, 3n\}$; $\mathbf{z}$ is within distance $\lambda_1(L)/2$ of L^*. Returns the coefficient vector modulo p of the vector in L^* which is closest to $\mathbf{z}$.

$\mathsf{solveCVP}(L^*, \mathcal{L}, \mathbf{z})$**:** The inputs $L^*, \mathcal{L}$ and $\mathbf{z}$ are as in the case of $\mathsf{solveCVP}^{(p)}$. Returns a point of L^* which is closest to $\mathbf{z}$.

$\mathsf{quantumSample}()$**:** Uses $\mathsf{solveCVP}(L^*, \mathcal{L}, \cdot)$ as an oracle and some quantum computation to return a sample from $D_{L,r_{i-1}}$. The list $\mathcal{L}$ contains I samples from D_{L,r_i}.

$\mathsf{solveDGS}(p, \alpha, r)$**:** Uses the oracle $\mathsf{solveLWE}_{p,\Psi_\alpha}(\cdot)$ to return a sample from $D_{L,r}$ where $r \geq \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$. Note that the description of the algorithm $\mathsf{solveDGS}$ provides the proof of Theorem 37.

In the algorithm descriptions, we will make use of the following two subroutines mentioned below.

1. $\mathsf{bootstrap}(L, r)$**:** Here L is a lattice and $r > \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$. Returns a list $\mathcal{L}$ containing I independent samples from $D_{L,r_{3n}}$ where $r_{3n} = r \cdot ((\alpha p)/(\sqrt{n}))^{3n}$. $\mathsf{bootstrap}$ uses the LLL [LLL82] algorithm to find reduced basis for L. Let $\mathcal{B}$ be the new basis of L and $\mathcal{P}(\mathcal{B})$ be a fundamental parallelepiped of L. The sampling procedure samples a vector $\mathbf{x}$ from $\mathbb{R}^n$ following gaussian distribution of width r_{3n} then modulo $\mathbf{x}$ by $\mathcal{P}(\mathcal{B})$ and subtracts the resultant part from the original vector and outputs the result as the final output of $\mathsf{bootstrap}$. The output distribution of $\mathsf{bootstrap}$ subroutines is within a negligible statistical distance of $D_{L,r_{3n}}$.

2. $\mathsf{reconstruct}(x)$**:** This is used in $\mathsf{solveCVP}$ to reconstruct the closest vector by first applying a nearest neighbor algorithm and then retracing through the results returned by the repeated calls to $\mathsf{solveCVP}^{(p)}$. Following Algorithm 6, we use $\mathsf{solveCVP}^{(p)}$ on the input vector and find the nearest lattice vector modulo p, thereafter we subtract the modulo part from the original vector and divide the resultant vector by p, consequently, we get another vector which is closer to lattice than the initial vector by a factor of $1/p$. We repeat this procedure on the resultant vector and go on for a total of n times. The resultant vector at the end of n iterations becomes very close to a lattice point such that the Babai's [Bab86] algorithm can be employed to find the nearest lattice point of the resultant vector. Once the nearest lattice point is determined, we construct the closest lattice point of the original vector iteratively. At the first iteration, using the vector given by Babai's algorithm and the result by $\mathsf{solveCVP}^{(p)}$ at the n-th iteration, we construct the nearest lattice point for the $(n-1)$-th iteration. Using this vector and the result by $\mathsf{solveCVP}^{(p)}$ at the $n-1$-th iteration, we construct the nearest lattice point for the $(n-2)$-th iteration and so on. At the end of the n-th iteration, we get the closest lattice vector to the original vector.

4.3.1 Concrete Analysis

The number of times the oracle $\mathsf{solveLWE}$ is called is determined by the following factors.

Algorithm 5 Algorithm to solve DGS using an LWE oracle.

function solveDGS(p, α)
  $\mathcal{L} \leftarrow$ bootstrap(L, r);
  for $i \leftarrow 3n$ down to 1 **do**
    $\mathcal{L}' \leftarrow \{\}$;
    for $j \leftarrow 1$ to I **do**
      $\mathbf{y} \leftarrow$ quantumSample() (using solveCVP$(L^*, \mathcal{L}, \cdot)$ as an oracle);
      $\mathcal{L}' \leftarrow \mathcal{L}' \cup \{\mathbf{y}\}$;
    end for
    $\mathcal{L} \leftarrow \mathcal{L}'$; $r_{i-1} = r_i \cdot (\sqrt{n})/(\alpha p)$;
  end for
  return one element from $\mathcal{L}$.
end function.

Algorithm 6 Algorithm to solve CVP.

function solveCVP$(L^*, \mathcal{L}, z)$
  $\mathbf{z}_1 \leftarrow \mathbf{z}$;
  for $k \leftarrow 1$ to n **do**
    $\mathbf{a}_k \leftarrow$ solveCVP$^{(p)}(L^*, \mathcal{L}, \mathbf{z}_k)$;
    $\mathbf{z}_{k+1} \leftarrow (\mathbf{z}_k - L^*\mathbf{a}_k)/p$;
  end for
  $\mathbf{s} \leftarrow$ reconstruct$(\mathbf{z}_{n+1})$;
  return $\mathbf{s}$.
end function.

1. The loop in solveDGS has $3n$ iterations. In the i-th iteration I samples of D_{L,r_i} are used to generate I samples of $D_{L,r_{i-1}}$. Generating each sample of $D_{L,r_{i-1}}$ requires a call to quantumSample which in turn generates a call to solveCVP. So, the subroutines quantumSample and solveCVP are both called a total of $3n{\cdot}I$ times.

2. The loop in solveCVP has n iterations and in each iteration, a call to solveCVP$^{(p)}$ is made. So, each call to solveCVP generates n calls to solveCVP$^{(p)}$. One of the reviewers of this thesis has commented that **a single solveCVP$^{(p)}$ call is enough to execute solveCVP for a typical choice of p as we can use the samples from $D_{L,r}$**. While this is quite interesting to note, we don't have any proof for this claim.

3. In solveCVP$^{(p)}$, the set Z contains about I^2 values. So, the loop from Steps 9 to 21 makes about $n{\cdot}I^2$ calls to solveLWE and to verifyLWE. So, each call to solveCVP$^{(p)}$ generates $n{\cdot}I^2$ calls to solveLWE and to verifyLWE.

Algorithm 7 Algorithm to solve $\mathsf{CVP}^{(p)}$.

function $\mathsf{solveCVP}^{(p)}(L^*, \mathcal{L}, \mathbf{z})$
  $Z \leftarrow$ set of all integer multiples of $I^2\alpha^2$ in the range $(0, \alpha^2]$;
  $\mathcal{I} \leftarrow \{\}$;
  for $\mathbf{v}$ in $\mathcal{L}$ **do**
    $\mathbf{a} \leftarrow L^{-1}\mathbf{v} \bmod p$;
    $e \xleftarrow{\$} \mathcal{N}(0, \alpha/(2\sqrt{\pi}))$;
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{(\mathbf{a}, \langle \mathbf{z}, \mathbf{v} \rangle/p + e \bmod 1)\}$;
  end for
  for γ in Z **do**
    $\mathcal{I}' \leftarrow \{\}$;
    for $i \leftarrow 1$ to n **do**
      for $(\mathbf{a}, e) \in \mathcal{I}$ **do**
        $\varepsilon \xleftarrow{\$} \Psi_{\sqrt{\gamma}}$;
        $\mathcal{I}' \leftarrow \mathcal{I}' \cup \{(\mathbf{a}, e + \varepsilon)\}$;
      end for
      $\mathbf{s}' \leftarrow \mathsf{solveLWE}_{p,\Psi_\alpha}(\mathcal{I}')$;
      if $\mathsf{verifyLWE}(s', \mathcal{I}')$ returns true **then**
        return $\mathbf{s}'$;
      end if
    end for
  end for
end function.

4. Algorithm 7 ensures outputting correct $\mathbf{s}$ with probability at least $1/2$ and also ensures that when the input list $\mathcal{I}'$ is closest to $A_{\mathbf{s},\Psi_\alpha}$, the $\mathsf{solveLWE}_{p,\Psi_\alpha}$ outputs the correct $\mathbf{s}$ in one of n iterations with probability at-least $(1 - 2^n)$, thus making the the failure probability at most 2^{-n}. In the context of concrete security where n is around 1024, the failure probability is at most 2^{-1024} and that is way more than sufficient. In this case, a failure probability of 2^{-200} would be sufficient to consider. Hence the choice of the number of iterations in the inner loop is very pessimistic in favor of making the whole process more reliable. We can safely choose a large constant in place of n for the number of iterations in the inner loop, thus making the calls for $\mathsf{solveLWE}$ and $\mathsf{verifyLWE}$ limited to a constant multiple of I^2 instead of nI^2.

Proposition 38. *Algorithm* $\mathsf{solveDGS}$ *has the following properties.*

1. *The* $\mathsf{solveLWE}$ *oracle is called* $T = 3n^2I^3$ *times.*
2. *Algorithm* $\mathsf{verifyLWE}$ *is also called* T *times.*

Algorithm 8 Algorithm to verify an LWE solution.

function verifyLWE($\mathbf{s}', \mathcal{I}'$)
  Let $m \leq I$ be a positive integer;
  Choose m pairs $(\mathbf{a}_1, x_1), \ldots, (\mathbf{a}_m, x_m)$ from $\mathcal{I}'$;
  $w \leftarrow 0$;
  for $i \leftarrow 1$ to m **do**
    $y_i \leftarrow x_i - \langle \mathbf{a}_i, \mathbf{s}' \rangle / p$;
    $w \leftarrow w + \cos(2\pi y_i)$;
  end for
  $z \leftarrow w/m$;
  Let $t \in (0, 1)$;
  if $z > t$ **then**
    return true
  else
    return false
  end if
end function.

3. A total of $3n \cdot I$ quantum computations are required.

Remark: Each quantum computation in [Reg09] is on a state of $n \log R$ qubits where $R \geq 2^{3n} \lambda_n(L^*)$ is an integer. For example, if we take $I = n$, the number of quantum computations required is $3n^2$ where each computation is on at least $(3n^2 + n \log \lambda_n(L^*))$ bits. The cost of quantum computation increases quadratically with n. For $n = 1024$, about 3 million logical qubits will be required. In comparison, factoring a 2048-bit RSA modulus requires around 4000 to 5000 logical qubits.

The VerifyLWE algorithm is based on the proof of Lemma 3.6 of [Reg09]. We highlight two aspects of VerifyLWE that is not present in this proof.

1. The parameter m is not present in the proof. The proof starts by considering n samples. This is achieved by setting $m = n$ in VerifyLWE. Note that the set $\mathcal{I}'$ has cardinality I and so m can be at most I.

2. The parameter t is not present in the proof. The proof considers the rejection threshold to be 0.02. This is achieved by setting $t = 0.02$ in VerifyLWE.

The choices of $m = n$ and $t = 0.02$ are sufficient for asymptotic analysis. We show later that these choices are sub-optimal for concrete analysis.

Algorithm verifyLWE is essentially a test of the hypothesis. In verifyLWE, the pairs in $\mathcal{I}'$ are of the form $(a, \langle a, s\rangle/p + e)$ where e follows Ψ_β. The test statistic is the variable z. Let ξ_0 be the distribution of z when $s = s'$ and let μ_0 be the corresponding mean of z; let ξ_1 be the distribution of z when $s \neq s'$ and let μ_1 be the corresponding mean of z. The following have been proved by Regev [Reg09].

- $\xi_0 = \Psi_\alpha$ so that $\mu_0 = \exp(-\pi\alpha^2) \geq 0.04$ for $\alpha < 1$. Note that $\mu_0 > t = 0.02$.
- $\mu_1 = 0$.

The computation performed by verifyLWE is a test of hypothesis between $H_0 : s = s'$ versus $H_1 : s \neq s'$.

Two types of errors are to be considered.

$$e_0 = \Pr[\text{Type-1 error}] = \Pr[\text{reject } H_0 \text{ when it is true}] = \Pr_{z\sim\xi_0}[z \leq t]; \tag{4.1}$$

$$e_1 = \Pr[\text{Type-2 error}] = \Pr[\text{accept } H_0 \text{ when it is false}] = \Pr_{z\sim\xi_1}[z > t]. \tag{4.2}$$

A Type-1 error will result in the correct value of s' being rejected and so the entire reduction will not succeed. A Type-2 error will result in an incorrect value of s' being accepted. This incorrect value of s' will be passed on to verifyCVP$^{(p)}$ and then on to verifyCVP resulting in an incorrect solution to the CVP problem. So, again, the entire reduction will fail. So, it is required to ensure that both Type-1 and Type-2 errors are small.

For $i = 1, \ldots, m$, let $v_i = \cos(2\pi y_i)$. Then $v_1, \ldots, v_m$ take values in the interval $[-1, 1]$. Applying the Hoeffding inequality (Section 2.6.1 of Chapter 2) to $v_1, \ldots, v_m$ and $z = (v_1 + \cdots + v_m)/m$, provides the following upper bounds on e_0 and e_1.

$$e_0 = \Pr_{z\sim\xi_0}[z \leq t] = \Pr_{z\sim\xi_0}[z - \mu_0 \leq -(\mu_0 - t)] \leq \exp\left(-m(\mu_0 - t)^2/2\right); \tag{4.3}$$

$$e_1 = \Pr_{z\sim\xi_1}[z > t] = \Pr_{z\sim\xi_1}[z - \mu_1 > t - \mu_1] \leq \exp\left(-mt^2/2\right). \tag{4.4}$$

If $s' = s$, then the probability that verifyLWE makes an error is at most e_0; if $s' \neq s$, then the probability that verifyLWE makes an error is at most e_1. So, the probability that verifyLWE makes an error is at most $\max(e_0, e_1)$ and so the probability that verifyLWE is successful is at least

$$(1 - \max(e_0, e_1)) = (1 - \max(\exp\left(-m(\mu_0 - t)^2/2\right), \exp\left(-mt^2/2\right))).$$

Proposition 38 shows that verifyLWE is called a total of $3n^2I^3$ times by solveDGS. The probability that all of these calls are successful, is at least

$$P_S \;=\; \left(1-\max(\exp\left(-m(\mu_0-t)^2/2\right),\exp\left(-mt^2/2\right))\right)^{3n^2I^3}. \tag{4.5}$$

Again from Proposition 38, the number of calls to solveLWE made by solveDGS is $T=3n^2I^3$ and so the tightness gap of the reduction from DGS to LWE is at most

$$G \;=\; T/P_S = 3n^2I^3\cdot\left(1-\max(\exp\left(-m(\mu_0-t)^2/2\right),\exp\left(-mt^2/2\right))\right)^{-3n^2I^3}. \tag{4.6}$$

4.3.2 Numerical Results

To compute numerical values, we need to specify the values of the parameters m, I, and t. We consider two scenarios. The first scenario corresponds to the values used in the proof of Lemma 3.6 of [Reg09]. The second scenario corresponds to an alternative analysis where we change the value of the rejection threshold and consider values of α which occur in practice.

Concrete analysis of the proof Lemma 3.6 of [Reg09]. As mentioned earlier, the proof of Lemma 3.6 of [Reg09] corresponds to setting $m=n$ and $t=0.02$. Using $m=n$ and $t=0.02$, (4.6) simplifies to the following.

$$G \;=\; T/P_S = 3n^2I^3\cdot\left(1-\exp\left(-n/5000\right)\right)^{-3n^2I^3}. \tag{4.7}$$

Setting $I=n$ (which minimises the right hand side of (4.7)), we have evaluated T, P_S and G for various values of n. It turns out that for $n\leq 350000$, G is determined primarily by $1/P_S$ while for $n\geq 400000$, the value of $1/P_S$ becomes negligible. For $2\leq n\leq 350000$, the value of G remains very high, for example for $n=350000$, $\log_2(\log_2(G))\approx 11.68$. For $n=400000$, $\log_2(\log_2(G))\approx 6.83$. The parameter n is the dimension of the underlying lattice. So, if the lattice dimension is to be chosen based on the proof of Lemma 3.6 of [Reg09], then the value of n has to be at least 400000. The codes for the calculation can be found in section 4.7. We use SAGE [SJ05] and Magma [BCP97] software to perform the calculations.

Alternative concrete analysis. The statistical test performed by verifyLWE is essentially a test for the means μ_0 and $\mu_1=0$ of the distributions ξ_0 and ξ_1 respectively. A natural value of the rejection threshold t is the choice $\mu_0/2=\exp(-\pi\alpha^2)/2$. This makes G depend on the

value of α. A higher value of α makes the LWE problem more difficult but, also results in a worse tightness gap in the reduction from DGS to LWE. On the other hand, most practical cryptosystems consider α to be at most $1/\sqrt{n}$. To account for $\alpha \in (0, 1/\sqrt{n})$, one may set $t = \exp(-\pi/n)/2$. Using $t = \exp(-\pi/n)/2$ gives the following expression for G.

$$G \;=\; T/P_S = 3n^2I^3 \cdot (1 - \exp\left(-m\exp(-2\pi/n)/8\right))^{-3n^2I^3}. \tag{4.8}$$

As in the previous concrete analysis, setting $I = n$ and $m = n$, we have computed the values of T, P_S, and G for various values of n. In this case, we observe that the value of P_S becomes very close to 1 for values of n as small as 100. So, under this alternative concrete analysis, the value of G is determined entirely by T and is equal to $3n^6$ for most practical values of n.

To summarise, the value of the rejection threshold t plays an important role in the concrete analysis. If the value of t is set to be equal to 0.02 as in the proof of Lemma 3.6 of [Reg09], then extremely high lattice dimensions are required for an even somewhat meaningful tightness gap.

On the other hand, choosing the rejection threshold to be mid-way between the means of the two distributions and considering α to be at most $1/\sqrt{n}$, there is no noticeable effect of P_S on the tightness gap for reasonable values of n. In this case, the tightness gap is given entirely by the number of oracle calls T.

4.4 Search-LWE to Decision-LWE

This section is divided into three subsections. Each of the subsections contains a reduction for two intermediate problems. Combining these three reductions, we obtain a reduction from search LWE to decision LWE.

4.4.1 Search-LWE (Continuous) to Search-LWE (Discrete)

In the previous section, we dealt with the continuous version of the search LWE problem. The cryptographic applications are based on the discrete version of the decisional LWE problem. This section deals with the reduction from continuous search LWE to discrete search LWE problems. So, this is one step towards achieving the end goal. Regev's Lemma 4.3 of [Reg09] provides the procedure to do so. Let us have an oracle $\mathcal{W}_2$ to solve the discrete search version of the LWE problem. We need to find an algorithm $\mathcal{W}_1$ to solve the continuous search version

of the LWE problem. $\mathcal{W}_1$ is presented with $A_{\mathbf{s},\Psi_\alpha}$. Here, $A_{\mathbf{s},\Psi_\alpha} = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle / p + e)$, where the addition is performed modulo 1, $\mathbf{a}$ and $\mathbf{s}$ are from $\mathbb{Z}_p^n$, e is from $\mathbb{T}$ following Ψ_α. $\mathcal{W}_1$ performs discretization procedure on $A_{\mathbf{s},\Psi_\alpha}$ by multiplying p with the second element of $A_{\mathbf{s},\Psi_\alpha}$ and truncates it to nearest integer value. As, $b \in \mathbb{T}$, multiplying it with p and truncating it to its nearest integer produces elements in $\mathbb{Z}_p$. Thus the discretization procedure correctly transforms LWE samples from $A_{\mathbf{s},\Psi_\alpha}$ to $A_{\mathbf{s},\xi}$ where ξ follows Gaussian distribution over $\mathbb{Z}_p$. $\mathcal{W}_2$ takes the new samples and outputs correct $\mathbf{s}$.

Proposition 39. *We record the following observations.*

1. *Here $\mathcal{W}_1$ calls $\mathcal{W}_2$ only one time.*
2. *Number of samples needed for both $\mathcal{W}_1$ and $\mathcal{W}_2$ are the same.*

4.4.2 Search-LWE (Discrete) to Decision-LWE(Worst Case)

The following reduction is between search and decision variants of LWE problems. Both the problems under consideration are worst-case problems. Lemma 4.2 of [Reg09] describes an algorithm to reduce the discrete search version of LWE to the decisional LWE in the worst case. We describe this lemma to work out the tightness gap of this reduction. Let $\mathcal{W}_3$ be the distinguisher to solve the worst-case decision version of LWE (DLWE_{wc}). $\mathcal{W}_3$ takes a list as an input. The list contains samples from $A_{\mathbf{s},\xi}$ distribution or from the uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$. $\mathcal{W}_3$ accepts when the input list is from the distribution $A_{\mathbf{s},\xi}$, otherwise rejects. $\mathcal{W}_3$ works for an arbitrary value of $\mathbf{s} \in \mathbb{Z}_p^n$ with success probability exponentially close to 1. Here, ξ follows the gaussian distribution over $\mathbb{Z}_p$. We need to construct an algorithm for $\mathcal{W}_2$ to solve the search version of LWE on discrete gaussian error distribution ξ. $\mathcal{W}_2$ has a list of samples from $A_{\mathbf{s},\xi}$. The procedure works as follows. We find the elements of $\mathbf{s}$ one by one. First $\mathcal{W}_2$ tries to find s_1, or the 1-st element of $\mathbf{s}$. It takes a sample $(\mathbf{a}, b)$ from $A_{\mathbf{s},\xi}$ and transforms it to $(\mathbf{a}', b') := (\mathbf{a} + (l, 0, \ldots, 0), b + l \cdot k)$, where l is chosen uniformly at random from $\mathbb{Z}_p$ and $k \in \mathbb{Z}_p$. We see that $\mathbf{a}'$ is uniform over $\mathbb{Z}_p^n$ as $\mathbf{a}$ is uniform over $\mathbb{Z}_p^n$ and l is uniform over $\mathbb{Z}_p$. If b is uniform over $\mathbb{Z}_p$ this implies b' is also uniform over $\mathbb{Z}_p$. So, $(\mathbf{a}', b')$ is uniform over $\mathbb{Z}_p^n \times \mathbb{Z}_p$ if $(\mathbf{a}, b)$ is uniform over $\mathbb{Z}_p^n \times \mathbb{Z}_p$. On the other hand, if $(\mathbf{a}, b)$ is drawn from $A_{\mathbf{s},\xi}$ and $k = s_1$, $(\mathbf{a}', b')$ becomes a valid sample from $A_{\mathbf{s},\xi}$. The rest cases where $(\mathbf{a}, b)$ is drawn from $A_{\mathbf{s},\xi}$ and $k \neq s_1$, $(\mathbf{a}', b')$ maps to samples over uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$ as p is here is a prime integer. $\mathcal{W}_2$ provides $\mathcal{W}_3$ with transformed samples as input. It checks for which value of $k \in \mathbb{Z}_p$, $\mathcal{W}_3$ accepts. As $p = \mathsf{poly}(n)$, it is easy to repeat the procedure

for each value of k. Thus the first element of $\mathbf{s}$ can be found with probability exponentially close to 1 with the help of $\mathcal{W}_3$. The same procedure is repeated for each n element of $\mathbf{s}$ and $\mathbf{s}$ is fully recovered.

Proposition 40. *We record the following observations.*

1. *Here $\mathcal{W}_2$ calls $\mathcal{W}_3$ a total of $n{\cdot}p$ times.*

2. *Number of samples needed for both $\mathcal{W}_1$ and $\mathcal{W}_2$ are the same.*

4.4.3 Decision-LWE (Worst Case) to Decision-LWE (Average Case)

Here we describe the reduction from the worst case to the average case decision LWE problem. Lemma 4.1 of [Reg09] provides an algorithm for the same. Let $\mathcal{W}_4$ be a distinguisher for the average case decision-LWE problem(DLWE$_{ac}$). We need to construct a distinguisher $\mathcal{W}_3$ for the worst-case decision-LWE problem(DLWE$_{wc}$). For $\delta_1, \delta_2 \in (0, 1]$, we say that $\mathcal{W}_4$ is an (δ_1, δ_2)-distinguisher if $\mathcal{W}_4$ has advantage at least δ_2 for at least a proportion δ_1 of the set of possible $\mathbf{s} \in \mathbb{Z}_p^n$. The distinguisher $\mathcal{W}_4$ takes an input list of length l. Our goal is to construct a distinguisher $\mathcal{W}_3$ which will work for arbitrary $\mathbf{s} \in \mathbb{Z}_p^n$ with probability exponentially close to 1.

We describe the reduction in Algorithm 9 and elaborate on the same in detail. $\mathcal{W}_3$ has access to a list $\mathcal{L} = (\mathbf{a}, \mathbf{b})$ of polynomial many samples either from $A_{\mathbf{s},\xi}$ or from uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$, where ξ follows a Gaussian distribution over $\mathbb{Z}_p$. We define a transformation over the list $\mathcal{L}$ as follows and prepare a list $\mathcal{L}'$. The list $\mathcal{L}' = (\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle)$, where $\mathbf{t} \in \mathbb{Z}_p^n$. If $\mathcal{L}$ is from $A_{\mathbf{s},\xi}$, then $\mathcal{L}'$ is from $A_{\mathbf{s}+\mathbf{t},\xi}$. On the other hand if $\mathcal{L}$ is from a uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$, then $\mathcal{L}'$ is also from a uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$.

Algorithm 9 describes $\mathcal{W}_3$ which uses the distinguisher $\mathcal{W}_4$ as follows. It has two nested loops. The outer loop runs I_1 times and the inner loop runs I_2 times. In each iteration of the outer loop, $\mathcal{W}_3$ chooses a $\mathbf{t}$ uniformly at random from $\mathbb{Z}_p^n$. In each iteration the inner loop using samples from the list $\mathcal{L}$ a list $\mathcal{L}'$ is constructed as described above. Then another list $\mathcal{T}$ is created, which contains uniform samples from $\mathbb{Z}_p^n \times \mathbb{Z}_p$. In the last part of the inner loop, $\mathcal{W}_4$ is called with the inputs $\mathcal{T}$ and $\mathcal{L}'$ and captures the one-bit output in the variables cnt_0 and cnt_1 respectively. At the end of the inner loop, we capture the estimated probabilities in the variables $\hat{\mathfrak{p}_0}$ and $\hat{\mathfrak{p}_1}$ respectively. Here $\hat{\mathfrak{p}_0}$ and $\hat{\mathfrak{p}_1}$ are the estimated probabilities of $\mathfrak{p}_0$, $\mathfrak{p}_1$ that $\mathcal{W}_4$ accepts input from $\mathcal{T}$ and $\mathcal{L}'$ respectively. If any of the I_1 outer loop results in

$|\hat{\mathfrak{p}_0} - \hat{\mathfrak{p}_1}| \geq \delta_2/2$, the algorithm returns 1 and halts otherwise return 0 and halts when this condition is not satisfied by any of the I_1 iterations.

Hypothesis testing can be used to analyze the errors that may occur in $\mathcal{W}_3$. Algorithm 9 may return an incorrect answer in two ways. First, when $\mathcal{L}$ is from uniform distribution and it returns 1 and secondly when $\mathcal{L}$ is from $A_{\mathbf{s},\xi}$ and it returns 0. The first is Type-1 failure and the second is Type-2 failure. When $\mathcal{L}$ is from uniform distribution both lists $\mathcal{T}$ and $\mathcal{L}'$ follow uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$. For each of the I_1 iterations of the outer loop we have

$$\Pr[\mathfrak{p}_0 - \delta_2/4 \leq \hat{\mathfrak{p}_0} \leq \mathfrak{p}_0 + \delta_2/4] \geq 1 - 2\exp(-I_2{\delta_2}^2/8)$$
$$\Pr[\mathfrak{p}_1 - \delta_2/4 \leq \hat{\mathfrak{p}_1} \leq \mathfrak{p}_1 + \delta_2/4] \geq 1 - 2\exp(-I_2{\delta_2}^2/8)$$

due to the additive form of the Chernoff-Hoeffding bound (Section 2.6.1 of Chapter 2). Both the lists $\mathcal{T}$ and $\mathcal{L}'$ follow uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$, so $\mathfrak{p}_0 = \mathfrak{p}_1$. It implies that $\Pr[|\hat{\mathfrak{p}_0} - \hat{\mathfrak{p}_1}| \leq \delta_2/2] \geq 1 - 4\exp(-I_2{\delta_2}^2/8)$. So, for all I_1 outer loop iterations the Type-1 failure probability becomes at most $4I_1\exp(-I_2{\delta_2}^2/8)$, which is exponentially close to 0 for I_2 to be a constant multiplier of ${\delta_2}^{-2}$. We can safely assume $I_2 = {\delta_2}^{-2}$ for the concrete analysis.

When $\mathcal{L}$ is from $A_{\mathbf{s},\xi}$, $\mathcal{L}'$ follows $A_{\mathbf{s}+\mathbf{t},\xi}$ and $\mathcal{T}$ follows uniform distribution irrespective of the list $\mathcal{L}$. Let $p_{\mathbf{s}+\mathbf{t},0}$ and $p_{\mathbf{s}+\mathbf{t},1}$ respectively denote the probabilities p_0 and p_1 corresponding to a particular value of $\mathbf{t}$. Similarly, let $\hat{p}_{\mathbf{s}+\mathbf{t},0}$ and $\hat{p}_{\mathbf{s}+\mathbf{t},1}$ respectively denote the estimates $\hat{p_0}$ and $\hat{p_1}$ corresponding to a particular value of $\mathbf{t}$. We say that a value $\mathbf{s}+\mathbf{t}$ is good if $|p_{\mathbf{s}+\mathbf{t},0} - p_{\mathbf{s}+\mathbf{t},1}| \geq \delta_2$. From the definition of an (δ_1, δ_2) distinguisher, the probability of a good $\mathbf{s}+\mathbf{t}$ is at least δ_1. If we use the additive form of the Chernoff-Hoeffding bound for good $\mathbf{s}+\mathbf{t}$, we get the following inequalities.

$$\Pr[p_{\mathbf{s}+\mathbf{t},0} - \delta_2/4 \leq \hat{p}_{\mathbf{s}+\mathbf{t},0} \leq p_{\mathbf{s}+\mathbf{t},0} + \delta_2/4] \geq 1 - 2\exp(-I_2{\delta_2}^2/8)$$
$$\Pr[p_{\mathbf{s}+\mathbf{t},1} - \delta_2/4 \leq \hat{p}_{\mathbf{s}+\mathbf{t},1} \leq p_{\mathbf{s}+\mathbf{t},1} + \delta_2/4] \geq 1 - 2\exp(-I_2{\delta_2}^2/8)$$

From above equations along with the condition that $|p_{\mathbf{s}+\mathbf{t},0} - p_{\mathbf{s}+\mathbf{t},1}| \geq \delta_2$ we get

$$\Pr[|\hat{p}_{\mathbf{s}+\mathbf{t},0} - \hat{p}_{\mathbf{s}+\mathbf{t},1}| \geq \delta_2/2] \geq 1 - 4\exp(-I_2{\delta_2}^2/8) \tag{4.9}$$

which is exponentially close to 1 for I_2 to be a constant multiplier of ${\delta_2}^{-2}$. As previously we take $I_2 = {\delta_2}^{-2}$.

Again from the definition of an (δ_1, δ_2) distinguisher, the probability of a good $\mathbf{s}$ is at least δ_1 for $|p_{\mathbf{s},0} - p_{\mathbf{s},1}| \geq \delta_2$. So, we have execute the outer loop ${\delta_1}^{-1}$ times to ensure a good s. Thus minimum value of I_1 becomes a constant multiplier of ${\delta_1}^{-1}$. We can take $I_1 = {\delta_1}^{-1}$ for our analysis.

Algorithm 9 Reducing DLWE_{wc} to DLWE_{ac}

function $\mathcal{W}_3(\mathcal{L})$
  for $k = 1$ to I_1 **do**;
    Choose a uniform $\mathbf{t} \in \mathbb{Z}_p^n$;
    $\text{cnt}_0 \leftarrow 0$, $\text{cnt}_1 \leftarrow 0$;
    for $l = 1$ to I_2 **do**;
      Obtain a list $\mathcal{T}$ of polynomial many samples from uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$;
      Compute $\mathcal{L}'$ from $\mathcal{L}$ and $\mathbf{t}$;
      $\text{cnt}_0 \leftarrow \text{cnt}_0 + \mathcal{W}_4(\mathcal{T})$, $\text{cnt}_1 \leftarrow \text{cnt}_1 + \mathcal{W}_4(\mathcal{L}')$;
    end for
    $\hat{\mathfrak{p}_0} \leftarrow \text{cnt}_0/I_2, \hat{\mathfrak{p}_1} \leftarrow \text{cnt}_1/I_2$;
    if then$|\hat{\mathfrak{p}_0} - \hat{\mathfrak{p}_1}| > \delta_2/2$;
      return 1;
    end if
  end for
  return 0;
end function.

Remark: In [Reg09], the values of I_1 and I_2 are taken to be ${\delta_1}^{-1}$ and ${\delta_2}^{-2}$ respectively to make the Type-1 and Type-2 probabilities of the above analysis asymptotically zero. This factor of an extra n is more than sufficient and this can be replaced by $\ln \ln n$ which is sufficient to make the Type-1 and Type-2 probabilities asymptotically zero and hence the success probability of this reduction is asymptotically close to 1. In concrete terms, we replace $\ln \ln n$ factor by a constant.

Proposition 41. *The following observations are related to the tightness of this reduction.*

1. $I_1 = {\delta_1}^{-1}$ *and* $I_2 = {\delta_2}^{-2}$.

2. *Here $\mathcal{W}_3$ calls $\mathcal{W}_4$ a total of $I_1 \cdot I_2$ times.*

3. *The number of LWE samples required by $\mathcal{W}_3$ is $I_1 I_2 l$.*

4.5 End to end Concrete Analysis

The complete reduction by Regev [Reg09] is from worst-case GIVP to average case decisional LWE problem (DLWE$_{ac}$). This reduction consists of three parts.

- GIVP to DGS with tightness gap $2n^3$.
- DGS to LWE with tightness gap G given by (4.6).
- From Proposition 39, 40, 41, the number of samples required by the search LWE problem is $I = I_1 I_2 l$ which comes to as $(\delta_1 {\delta_2}^2)^{-1} l$. So from search LWE to DLWE$_{ac}$ with tightness gap becomes $npI_1 I_2 = np(\delta_1 {\delta_2}^2)^{-1}$.

The main part of the entire reduction is the second reduction. For this case, we have incorporated into the tightness gap the success probability of the statistical test required by the algorithm for verifying LWE solutions. The above concrete analysis shows that by appropriately setting the value of the rejection threshold and using values of α that are used in practice, there is no noticeable effect of the success probability on the tightness gap.

So the tightness gap from GIVP to LWE or the search LWE is given by

$$G_1 = 6n^5(\delta_1 {\delta_2}^2)^{-3} l^3, \tag{4.10}$$

and the tightness gap from LWE to DLWE$_{ac}$ is the following

$$G_2 = np(\delta_1 {\delta_2}^2)^{-1}. \tag{4.11}$$

We summarise the tightness gap analysis in the context of the end-to-end reduction from GIVP to DLWE$_{ac}$ in the form of the following theorem.

Theorem 42. *Let L be a lattice of dimension n; p be a prime number greater than 2; ϵ be a positive real number and $\epsilon \leq \frac{1}{10}$. Suppose there is a (δ_1, δ_2) distinguisher $\mathcal{D}$ that solves* DLWE$_{ac}$ *with error drawn from a Gaussian distribution over $\mathbb{Z}_p$. Then there is a*

quantum algorithm $\mathcal{A}$ *requiring approximately* $3n^2$ *logical qubits to solve* $\mathrm{GIVP}^{\varphi(L)}_{2\sqrt{n}}$*, where* $\varphi(L) = \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$*. The number of times* $\mathcal{A}$ *calls* $\mathcal{D}$ *is about*

$$(2n^3)\cdot(3n^2I^3)\cdot(pnI_1I_2) \approx pn^6(\delta_1{\delta_2}^2)^{-4}l^3. \tag{4.12}$$

For practical cryptosystems p is taken to be a prime in the range between n^2 and $2n^2$ and l is taken to be $\tilde{\mathrm{O}}(n)$ as per the discussion by Regev in [Reg09]. Taking $p = n^2$ and $l = n$, the expression (4.12) reduces to

$$n^{11}\cdot(\delta_1{\delta_2}^2)^{-4}. \tag{4.13}$$

We have reevaluated the values of G_1 and G_2 using the expressions from (4.10) and (4.11). The results are as follows:

G_1 reduces to $n^8 \cdot (\delta_1{\delta_2}^2)^{-3}$, and G_2 reduces to $n^3 \cdot (\delta_1{\delta_2}^2)^{-1}$.

So, for this reduction the effect of G_1 is greater than that of G_2 in the context of the end-to-end reduction from GIVP to DLWE_{ac}. In [CKMS16], it was argued that for $n = 1024$, the tightness gap of the reduction from GIVP to DLWE_{ac} is about 2^{504}. Incorporating the revised number of oracle calls mentioned above, the tightness gap we get is the following. Taking $\delta_1 = 2^{-\beta_1}$ and $\delta_2 = 2^{-\beta_2}$, for $n = 1024$, the expression in (4.13) is $2^{110+4\beta_1+8\beta_2}$.

4.6 Conclusion

In this chapter, we have incorporated the success probability of the statistical test for verifying LWE solutions into an upper bound G on the tightness gap. If the rejection threshold is picked as in the proof of Lemma 3.6 of [Reg09], then the tightness gap G is very high for values of n as large as 350000, on the other hand, we show that by choosing a different value of the rejection threshold and considering α to be at most $1/\sqrt{n}$, there is no noticeable effect of the success probability on the value of G for reasonable values of n

Our analysis has highlighted the sensitivity of concrete security analysis to the values of underlying parameters in lattice-based cryptography. The choice of parameters plays a crucial role in determining the practical security of these schemes. We have also shown that the tightness gap of the end-to-end reduction is enormous for parameters within a practical range. This highlights the challenges and limitations of achieving practical security guarantees in lattice-based cryptosystems.

It is important to note that the tightness gaps we obtain from the reductions serve as upper bounds. In other words, they represent the worst-case scenario for the looseness of the reductions. However, there is a possibility to reduce the upper bound of the tightness gap through further analysis and optimizations. By carefully examining the reduction techniques, fine-tuning the parameter choices, and exploring alternative approaches, it may be possible to improve the tightness of the reductions. This involves delving deeper into the mathematical properties of the lattice problems and the underlying cryptographic constructions to identify potential areas for enhancement.

As we proceed with our investigation, we will determine the tightness gap of reductions for other algebraic variants of lattices and perform a comparative study with the results obtained in the following chapters. By analyzing a broader range of reductions and parameter settings, we aim to gain a comprehensive understanding of the practical feasibility and security implications of lattice-based cryptographic schemes.

4.7 Programs to Evaluate T, $1/P_S$ and G for $I = n$

4.7.1 SAGE

```
RR = Reals(1000)
n = 400000
calls = RR(3*n^6)
logcalls = RR(log(calls))/RR(log(2))
loglogcalls = RR(log(logcalls))/RR(log(2))
a = RR(-n/5000)
b = RR(1/(1-e^a))
logb = RR(log(b))/RR(log(2))
loglogb = RR(log(b))/RR(log(2))

logPSinv = RR(calls * logb)
loglogPSinv = RR( log(logPSinv) )/ RR(log(2))
logloggap = RR( log(logcalls + logPSinv) ) / RR(log(2))
print loglogcalls
print loglogPSinv
print logloggap
```

4.7.2 Magma

```
SetDefaultRealField(RealField(1000));

n:= 400000.0;
calls := 3*n^6;

a := (-n/5000.0);
b := 1/(1-Exp(a));

logPSinv := calls*Log(2,b);
loglogPSinv := Log(2,calls) + Log(2,Log(2,b));
loglogG := Log(2,Log(2,calls)+logPSinv);

print Log(2,Log(2,calls));
print loglogPSinv;
print loglogG;
```

Chapter 5

Reduction from module SIVP to module-LWE

5.1 Introduction

In the previous chapter, we delved into the concept of concrete security and its relation to the tightness gap in reductions for the approximate version of the GIVP to LWE problem. Unfortunately, our findings raised concerns as we discovered that the reduction is loose. This means that cryptographic applications relying on the hardness assumption of lattice problems, such as GIVP, exhibit inefficiency in practical usage due to the large size of keys required, and they inherently suffer from a loose tightness gap. Several works, including those by Chatterjee et al. [CKMS16, SS21, KSSS22], have addressed this issue. Also, a quadratic overhead is inevitable due to the use of LWE. However, an affirmative answer to this problem was provided by Lyubashevsky, Peikert, and Regev [LPR13], who proposed an alternative approach using an algebraic variant of LWE known as ring-LWE (Definition 32). The ring-LWE scheme offers more efficient cryptographic applications compared to conventional LWE-based schemes, making it a promising direction for overcoming the limitations imposed by the loose tightness gap.

The inspiration for the use of the ring variant, known as ring-LWE, comes from the NTRU cryptosystem [HPS98]. Ring-LWE has emerged as a significant advancement in the field of post-quantum cryptography because cryptographic applications built on this scheme are faster and require smaller keys compared to conventional LWE-based schemes. This motivated the transition from LWE to ring-LWE in the first place. The seminal work of Lyubashevsky, Peikert, and Regev [LPR13] provides a reduction of hard lattice problems for ideal lattices (Definition 2.4.9) to the DLWE problem over a ring of integers. However, the results presented in [LPR13] are asymptotic in nature, allowing the parameters, such as the lattice dimension, to go to infinity. In practical scenarios, real-world applications usually involve a specific set of parameters, for instance, a lattice with a dimension of 1024 would be of great interest.

While ring-LWE shows promise over LWE in terms of speed and key size, it still lacks a guarantee of tightness in the reduction. To address this concern, we conducted an investigation into the tightness gap of the reductions in Lyubashevsky et al. [LPR13], as documented in our paper [KSSS22]. Unfortunately, our findings did not provide much promise regarding the tightness gap of these reductions, which underscores the importance of further research in this area to establish concrete security guarantees for ring-LWE-based cryptographic schemes.

After three rounds of evaluation, NIST has selected "CRYSTALS-KYBER" as the finalist among the candidates for public-key encryption and key-establishment algorithm in their post-quantum cryptography standardization process. CRYSTALS-KYBER's security is based on the module-LWE hardness assumption, which is another algebraic variant of LWE. The security reductions for module-LWE follow a similar path as ring-LWE. Module-LWE's hardness assumption relies on standard hard problems like SIVP or GapSVP on module lattices. The hardness results of module lattices were presented in a seminal work by Langlois and Stehlé [LS15]. Their work focuses on worst-case to average-case reductions for module lattices, taking inspiration from previous works such as [Reg09, LPR13, BLP+13]. The selection of LWE-based cryptosystems by NIST for post-quantum cryptography candidates further underscores the significance of the work in [LS15].

However, it is essential to question the tightness of the reductions in [LS15] given that previous analyses [CKMS16, SS21] have shown that Regev's reductions were not tight enough for practical purposes. Similarly, the tightness gap of the reductions in [LS15] has also been found to be significant, as commented in [KSSS22]. These observations highlight the need for further research and analysis to ensure the practical feasibility and concrete security guarantees of module-LWE-based cryptographic schemes.

Here in this chapter, we elaborate on the analysis in further detail.

5.1.1 Outline of the Analysis

The reduction in [LS15] can be divided into two parts. The first part is a reduction from approximate module-SIVP to the search the module-LWE problem, while the second part is a reduction from the search module-LWE problem to the decision module-LWE problem. The concrete analysis of the first part is described in the first section and that for the second part is described in the next section. The two parts are combined and the end-to-end reduction from approximate module-SIVP to decision module-LWE is summarised in the concluding

section.

5.2 Reducing M-SIVP$_\gamma$ to search module-LWE$_{q,\leq\alpha}$

We need to fix four parameters, a positive integer n which denotes the degree of the underlying number field K; a positive integer d which denotes the rank of the module M; an integer $q \geq 2$ which is used to define the module-LWE problem; and a positive real number α such that $\alpha q \geq 2\sqrt{d}\cdot\omega(\sqrt{\log n})$ as mentioned in [LS15]. Once n and d are fixed, we can talk about the dimension of the module or the module lattice M as N, where $N = n\cdot d$. In the asymptotic setting, q and α are considered to be functions of n.

The M-SIVP$_\gamma$ to module-LWE$_{q,\leq\alpha}$ reduction is obtained from the following sequence of algorithms, in which $\mathcal{A}_i$ calls $\mathcal{A}_{i+1}$ for $0 \leq i \leq 4$. We briefly describe the algorithms below.

Algorithm $\mathcal{A}_0$: Solves M-SIVP$_\gamma$ where γ is the parameter of approximation factor. The input is a module lattice M and the output is a set of N linearly independent elements of M the longest of which is at most $\gamma\cdot\lambda_N(M)$.

Algorithm $\mathcal{A}_1$: Solves M-DGS$_\Gamma$, where Γ is the width of the Gaussian distribution. The input is a pair (M, r), where M is a module lattice of K and $r \geq \Gamma(M)$. The output is a sample from the distribution $D_{M,r}$.

Algorithm $\mathcal{A}_2$: This is a quantum algorithm which, given as input a module lattice M and a set of samples chosen independently from $D_{M,r}$, returns a sample from $D_{M,r'}$, where $r' \leq r/2$.

Algorithm $\mathcal{A}_3$: Solves M-BDD$_{M^\vee,\zeta}$. The input is a tuple $(M^\vee, \mathbf{y}, \mathfrak{I})$, where $M^\vee$ is module lattice in K^d, $\mathbf{y} \in K^d$, $\mathbf{y} = \mathbf{x} + \mathbf{e}_0$, $\mathbf{x} \in M^\vee$ and $\mathbf{e}_0 = \Sigma^{-1}(\mathbf{e})$ is a d dimensional error vector in K^d and $||\mathbf{e}||_{2,\infty} \leq \delta$ where $\delta < \lambda_1(M^\vee)/2$. Additionally, $\mathcal{A}_3$ has access to a set of samples $\mathfrak{I}$ chosen independently from $D_{M,r}$, here $D_{M,r}$ comes from the input of $\mathcal{A}_2$. The output is an $\mathbf{x}' \in M^\vee$ such that $\mathbf{x}' = \mathbf{x}$ except with negligible probability.

Algorithm $\mathcal{A}_4$: Solves q-BDD$_{M^\vee,\zeta}$. $\mathcal{A}_4$ has same input as $\mathcal{A}_3$. $\mathcal{A}_4$ outputs $\mathbf{x}' \mod q$ such that $\mathbf{x}' \equiv \mathbf{x} \mod q$ except with negligible probability.

Algorithm $\mathcal{A}_5$: $\mathcal{A}_5$ solves module-LWE$_{q,\leq\alpha}$. It has access to an oracle that generates samples from the module-LWE distribution $A_{s,r}^{(M)}$ (see Definition 28).

5.2.1 Reduction from M-SIVP$_\gamma$ to M-DGS$_\Gamma$

The first reduction (Lemma-16 of [LS15]) is between two lattice problems. This lemma is a module-restricted version of lemma 31. For this reduction, the relation between two parameters of the problems, i.e. γ and Γ are the following for module lattice M.

$$\Gamma(M) = \frac{\gamma \cdot \lambda_N(M)}{2\sqrt{N}} \tag{5.1}$$

Lemma 16 of [LS15] requires

$$\Gamma(M) = \sqrt{2d} \cdot \omega(\sqrt{\log n}) \cdot \eta_\epsilon(M)/\alpha \tag{5.2}$$

and

$$\gamma = \sqrt{8Nd} \cdot \omega(\sqrt{\log n}) \cdot \eta_\epsilon(M)/(\lambda_N(M) \cdot \alpha) \tag{5.3}$$

which is satisfied by (5.1). This reduction follows directly from lemma 31. The constraints of the lemma are $\epsilon \leq 1/10$ and $\Gamma(M) \geq \sqrt{2}\eta_\epsilon(M)$. The rank d of module M is at least 1, $\alpha < 1$ and $\omega(\sqrt{\log n}) > 1$, which implies that $\Gamma(M) \geq \sqrt{2}\eta_\epsilon(M)$. Lemma 16 of [LS15] follows all the conditions for the lemma 31. So, given an algorithm $\mathcal{A}_1$ to solve M-DGS$_\Gamma$, it is possible to construct an algorithm $\mathcal{A}_0$ to solve M-SIVP$_\gamma$.

We review the proof here. The objective is to output a set of N linearly independent vectors whose longest vector has length at most $2^N\lambda_N(M)$. Algorithm $\mathcal{A}_0$ uses the LLL algorithm to obtain a set B_0 of N linearly independent vector such that the output vectors will have length at most $2^N\lambda_N(M)$. If d_0 is the length of the longest vector in B_0, it will satisfy that $\lambda_N(M) \leq d_0 \leq 2^N\lambda_N(M)$ For $i = 0, \ldots, 2N$, let $u_i = d_0/2^i$. For each i in $\{0, \ldots, 2N\}$ algorithm $\mathcal{A}_0$ does the following. It invokes $\mathcal{A}_1$ a total number of N^2 times on the input (M, u_i) to obtain a set S_i of N^2 elements of M chosen independently from D_{M,u_i}. $\mathcal{A}_0$ looks for a set B_i of N linearly independent vectors in each of S_i and outputs the shortest set found.

The claim is that with high probability $\mathcal{A}_0$ returns N linearly independent vectors whose longest vector length is at most $2\sqrt{N}\Gamma(M)$. Firstly if $\Gamma(M) \geq d_0$ then length of vectors in B_0 are at most $2\sqrt{N}\Gamma(M)$ trivially. On the other hand, it can be shown that there exists an i

in $\{0,\ldots,2N\}$ such that $\Gamma(M) < u_i \leq 2\Gamma(M)$. Similar proof of this claim for general lattice can be found in section 4.2 using the lemma 19. Hence, we record the following proposition.

Proposition 43. *$\mathcal{A}_0$ invokes $\mathcal{A}_1$ a total of N^3 times.*

5.2.2 Reducing M-DGS$_\Gamma$ to module-LWE$_{q,\leq\alpha}$

The reduction from M-DGS$_\Gamma$ to module-LWE$_{q,\leq\alpha}$ is adapted from [LPR13] and following theorem is the restatement of the adapted module version, presented in lemma 17 of [LS15].

Theorem 44. *Let $M \subseteq K^d$ be a module of rank $d > 0$, over the ring of integers R of the number field K of n dimension. Let $\alpha = \alpha(N) \in (0,1)$, $q = q(N) \in \mathbb{N}$, such that $\alpha q \geq 2\sqrt{d} \cdot \omega(\sqrt{\log n})$. For some negligible $\epsilon = \epsilon(N) = N^{-\omega(1)}$, there is a probabilistic polynomial-time quantum reduction from M-DGS$_\Gamma$ to M-LWE$_{q,\leq\alpha}$.*

For $r \geq \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$, define $r_i = r \cdot (\alpha p/\sqrt{d \cdot \log n})^i$ for $i = 1,\ldots,3n$.

This theorem is analogous to the Theorem 3.1 of [Reg09], the main difference is that in [LS15] we work with more algebraic structures. We adopt the same methodology used in [SS21] to view the proof of the theorem in an algorithmic form. We need some subroutines and the data structures are as follows.

solveMLWE$_{q,\leq\alpha}(\mathcal{L})$: This is the oracle to solve MLWE$_{q,\leq\alpha}$. The list $\mathcal{L}$ consists of N^c samples from $A^{(M)}_{\mathbf{s},r}$, where $r < \alpha$.

solveqModBDD$(\mathcal{L}, M^\vee, \mathbf{z})$: Here $M^\vee$ is the dual lattice of M; $\mathcal{L}$ contains N^c samples from $D_{M^\vee,r_i}$ for some $i \in \{1,\ldots,3N\}$; $\mathbf{z}$ is within distance $\lambda_1(M^\vee)/2$ of $M^\vee$. Returns a lattice vector in $M^\vee$ modulo $qM^\vee$ which is closest to $\mathbf{z}$.

solveModBDD$(\mathcal{L}, M^\vee, \mathbf{z})$: The inputs $\mathcal{L}, M^\vee$ and $\mathbf{z}$ are as in the case of solveqModBDD. Returns a vector of $M^\vee$ which is closest to $\mathbf{z}$.

quantumSample(): Uses solveModBDD$(\mathcal{L}, M^\vee, \cdot)$ as an oracle and some quantum computation to return a sample from $D_{M^\vee,r_{i-1}}$. The list $\mathcal{L}$ contains N^c samples from $D_{M^\vee,r_i}$.

solveDGS(p,α,r): Uses the oracle solveMLWE$_{p,\leq\alpha}(\cdot)$ to return a sample from $D_{M,r}$ where $r \geq \sqrt{2d}\cdot\omega(\sqrt{\log(n)})\cdot\eta_\epsilon(M)/\alpha$. Note that the description of the algorithm solveDGS provides the proof of Theorem 44.

In the algorithm descriptions, we will make use of the following two subroutines mentioned below. The detailed descriptions of the following subroutines in the case of general lattices can be found in Section 4.3

1. $\mathsf{bootstrap}(M, r)$**:** Here M is a module lattice and $r \geq \sqrt{2d}\cdot\omega(\sqrt{\log(n)})\cdot\eta_\epsilon(M)/\alpha$. Returns a list $\mathcal{L}$ containing N^c independent samples from $D_{M,r_{3N}}$ where $r_{3N} = r \cdot ((\alpha p)/(\sqrt{d\log n}))^{3N}$.

2. $\mathsf{reconstruct}(x)$**:** This is used in solveModBDD to reconstruct the closest vector by first applying a nearest neighbor algorithm and then retracing through the results returned by the repeated calls to solveqModBDD.

The reduction from M-DGS$_\Gamma$ to MLWE$_{q,\leq\alpha}$ comprises different algorithms that we elaborate on now. Here DGS parameter Γ comes from (5.2). To solve the DGS problem with parameter Γ, the algorithm $\mathcal{A}_1$ needs to be called with input pair (M, r), where M is the module lattice and $r \geq \Gamma(M)$. Let us define $r_i = r\cdot(\alpha q/\omega(\sqrt{d\log n}))^i$ for $i = 0, \ldots, 3N$. Note that $r_i \geq 2^i r$ due to the assumption that $\alpha q > 2\omega(\sqrt{d\log n})$, hence $r_{3N} \geq 2^{3N} r$. Following the $\mathsf{bootstrap}(M, r)$ function, the algorithm $\mathcal{A}_1$ can sample from $D_{M,r_{3N}}$ in polynomial time without using any LWE oracle as $r_{3N} \geq 2^{2N}\lambda_N(M)$.

Algorithm 10 Algorithm to solve M-DGS$_\Gamma$ using an M-LWE$_{q,\leq\alpha}$

function solveDGS(q, α, y) // where $r \geq 2^{2N}\lambda_N(M)$
  $\mathcal{L} \leftarrow \{\mathsf{bootstrap}(M, r)\}$;
  for $i \leftarrow 3N$ down to 1 **do**
    $\mathcal{L}' \leftarrow \{\}$;
    for $j \leftarrow 1$ to N^c **do**
      $\mathbf{y} \leftarrow \mathsf{quantumSample}()$; // using solveModBDD$(\mathcal{L}, M^\vee, \mathbf{y})$ as an oracle
      $\mathcal{L}' \leftarrow \mathcal{L}' \cup \{\mathbf{y}\}$
    end for
    $\mathcal{L} \leftarrow \mathcal{L}'$;
    $r_{i-1} \leftarrow r_i \cdot (\sqrt{d\log n})/(\alpha q)$;
  end for
  return one element from $\mathcal{L}$;
end function.

Following the description of algorithm 10, $\mathcal{A}_1$ prepares a list of $N_1 = N^c$ samples S_{3N} from $D_{M,r_{3N}}$, referred to as initial list $\mathcal{L}$. $\mathcal{A}_1$ runs a loop starting $i = 3N$ to 1, where in each iteration $\mathcal{A}_2$ is called N_1 times with S_i and returns a sample from $D_{M,r_{i-1}}$, thus in total

creating N_1 samples from $D_{M,r_{i-1}}$, referred to as $\mathcal{L}'$. In the $3N^{th}$ iteration $\mathcal{A}_2$ produces N_1 samples from D_{M,r_0} or $D_{M,r}$ as desired output.

$\mathcal{A}_2$ is the only quantum part of algorithm 10. $\mathcal{A}_2$ works with the output of $\mathcal{A}_3$. $\mathcal{A}_3$ solves M-BDD problem using $\mathcal{A}_4$ which solves q-BDD problem. As already mentioned, the parameter set for both $\mathcal{A}_3$ and $\mathcal{A}_4$ is the same. At the i^{th} iteration of algorithm 10, $\mathcal{A}_3$ solves M-BDD$_{\delta_i}$. $\mathcal{A}_2$ uses the M-BDD$_{\delta_i}$ solver to produce samples from $D_{M,r_{i-1}}$. $\mathcal{A}_3$ solves M-BDD$_{\delta_i}$ using $\mathcal{A}_4$ solving q-BDD$_{\delta_i}$. The values of δ_i must be less than $\lambda_1(M^\vee)/2$.

We have $r \geq \Gamma(M) = \sqrt{2d}\cdot\omega(\sqrt{\log n})\cdot\eta_\epsilon(M)/\alpha$ and $\alpha q \geq 2\sqrt{d}\cdot\omega(\sqrt{\log n})$ from initial assumptions. Together they imply that $r > \sqrt{2}q\cdot\eta_\epsilon(M)$. Using $\delta_i = \frac{\alpha q\cdot\omega(\sqrt{\log n})}{\sqrt{2n}r_i}$ the lemma 19, it can be shown that $\delta_i < \lambda_1(M^\vee)/2$ for $i = 3N$ to 1.

We need to provide the algorithm $\mathcal{A}_3$ a vector $\mathbf{y}$ in $\mathbb{R}^N$ along with the lattice $M^\vee$ and find the closed lattice vector of the given vector $\mathbf{y}$. The offset in $\mathbf{y}$ is a vector $\mathbf{e}$, such that $||\mathbf{e}||_{2,\infty} \leq \delta_i$ in the i^{th} iteration of algorithm 10 through the call to $\mathcal{A}_2$. The same set of parameters is passed to $\mathcal{A}_4$ to solve M-BDD modulo q.

The description of $\mathcal{A}_4$ is shown in algorithm 11. $\mathcal{A}_4$ has three inputs, a module lattice $M^\vee$, an element $\mathbf{y} \in K^d$ such that $\mathbf{y} = \mathbf{x} + \mathbf{e}_0$, where $\mathbf{x} \in M^\vee$, $\mathbf{e}_0 = \Sigma^{-1}(\mathbf{e})$, here $\mathbf{e}_0 \in K^d$, $\mathbf{e} \in \Sigma(K^d)$ or $\mathbf{e} \in \mathbb{R}^{nd}$ and $||\mathbf{e}||_{2,\infty} \leq \delta_i$, the third input is a set of points from Gaussian distribution D_{M,r_i}. Here $\delta_i = \frac{\alpha q\cdot\omega(\sqrt{\log n})}{\sqrt{2n}r_i}$. $\mathcal{A}_4$ solves BDD problem using $\mathcal{A}_5$ which solves module-LWE$_{q,\leq\alpha}$.

Using D_{M,r_i} and $\mathbf{y}$, a list of LWE samples is prepared, where the unknown for the LWE problem is the lattice vector $\mathbf{x}$ which is closest to given vector $\mathbf{y}$. To make the samples compatible for the LWE solver $\mathcal{A}_5$, each LWE sample is added with an error vector $\mathbf{e}'$ chosen according to $D_{\alpha/\sqrt{2}}$.

The general task of $\mathcal{A}_4$ in solving an instance of q-BDD_{δ_i} is similar to that of $\mathcal{A}_5$ in solving module-$\mathsf{LWE}_{q,\leq\alpha}$. But there is a major difference. Algorithm $\mathcal{A}_4$ works with the lattice $M^\vee$ modulo q, whereas $\mathcal{A}_5$ works with the lattice $(R^\vee)^d$ modulo q. The reduction handles this by using an element $t \in M$ that gives an isomorphism from R mod q to M mod q and also in the other direction between the dual lattices $M^\vee$ mod q and $R^\vee$ mod q. Langlois and Stehlé [LS15] have extended the idea of isomorphism in number filed 2.4.12 and provided an isomorphism between R_q^d and M_q and similarly they have shown isomorphism between dual of these algebraic structures, i.e, between $(R_q^\vee)^d$ and $(M_q^\vee)^d$. Algorithm 12 describes how M-BDD solver can be formed using a solver of q-BDD.

Proposition 45. *We summarise the observations as follows.*

Algorithm 11 Algorithm to solve q-Mod-BDD$_{M^\vee,d}$ using oracles M-LWE$_{q,\leq\alpha}$ and D$_{M,r}$

function solveqModBDD($\mathcal{L}, M^\vee, \mathbf{z}$)
  $\mathfrak{I} \leftarrow \{\}$;
  for each $\mathbf{v} \in \mathcal{L}$ **do**
    $e' \xleftarrow{\$} D_{\alpha/\sqrt{2}}$;
    $\mathbf{a} \leftarrow \theta^{-1}(\mathbf{v} \mod qM)$; // where $\theta(\cdot)$ is an isomorphism from R_q^d to M_q
    $b \leftarrow \{\langle \mathbf{z}, \mathbf{v}\rangle/q + e'\} \mod R^\vee$;
    $\mathfrak{I} \leftarrow \mathfrak{I} \cup \{(\mathbf{a}, b)\}$;
  end for
  $\mathbf{s} \leftarrow$ solveMLWE$_{q,\leq\alpha}(\mathfrak{I})$;
  return $\theta^{-1}(\mathbf{s})$;
end function.

Algorithm 12 Algorithm to solve M-BDD$_{M^\vee,d}$ using a q-BDD$_{M^\vee,d}$ oracle

function solveModBDD($\mathcal{L}, M^\vee, \mathbf{z}$)
  $\mathbf{z}_1 \leftarrow \mathbf{z}$;
  for $i \leftarrow 1$ to N **do**
    $\mathbf{a}_i \leftarrow$ solveqModBDD($\mathcal{L}, M^\vee, \mathbf{z}_i$);
    $\mathbf{z}_{i+1} \leftarrow (\mathbf{z}_i - \theta(\mathbf{a}_i))/q$; // where $\theta(\cdot)$ is the same isomorphism that was used in solveqModBDD
  end for
  $\mathbf{s} \leftarrow$ reconstruct($\mathbf{z}_{N+1}$);
  return $\mathbf{s}$;
end function.

- $\mathcal{A}_1$ *invokes* $\mathcal{A}_2$ *a total of* $3N{\cdot}N_1$ *times, where* $N_1 = N^c$ *for positive integer* c. *We deduce the exact value of* N_1 *in the course of this analysis.*
- $\mathcal{A}_2$ *invokes the reverse circuit of* $\mathcal{A}_3$ *once.*
- $\mathcal{A}_3$ *invokes* $\mathcal{A}_4$ *a total of* N *times.*
- $\mathcal{A}_4$ *invokes* $\mathcal{A}_5$ *once.*

5.2.3 The tightness gap in M-SIVP$_\gamma$ to module-LWE$_{q,\leq\alpha}$

From propositions 43 and 45 we can infer that an algorithm $\mathcal{A}_0$ to solve M-SIVP$_{\sqrt{8Nd}\cdot\omega(\sqrt{\log n})\cdot\eta_\epsilon(M)/(\lambda_N(M)\cdot\alpha)}$ can be constructed using an algorithm $\mathcal{A}_5$ that solves M-LWE$_{q,\leq\alpha}$ and the number of times $\mathcal{A}_0$ calls $\mathcal{A}_5$ approximately $3N^5N_1$ times, where M

is a module over number field K of rank d, K is a number field of dimension n, $N = n{\cdot}d$, $\alpha \in (0,1)$ and $N_1 = N^c$ for positive integer c.

5.3 Reducing search module-LWE to module-DLWE

The main result that connects the search module-LWE problem to the decision module-LWE problem is the following.

Theorem 46. *[Modified Version of Theorem 10 of [LS15]] Let $\epsilon(N) = N^{-\omega(1)}$, $\alpha \in (0,1)$ and $q \geq 2$, a prime with $q \leq \text{poly}(N)$ and $q = 1 \mod m$, such that $\alpha q \geq 2\sqrt{d}{\cdot}\omega(\sqrt{\log n})$. There is a polynomial time classical reduction from solving M-LWE$_{q,\leq\alpha}$ to M-LWE$_{q,D_\xi}$, given only bounded samples l, $\xi = \alpha(NN_3l/\log(NN_3l))^{1/4}$, $N_3^{1/4} > \sqrt{N}$.*

Here, N is the dimension of the underlying module lattice $M \subseteq K^d$, and d denotes the rank of the module. Here $N = n{\cdot}d$, n is the degree of the number filed K. Hence, N is also the dimension of the unknown vector in module-LWE.

The factor M is not present in original Theorem 10 of [LS15]. This factor is the result of the concrete security analysis of this very theorem. We provide the full calculation in the later part of this chapter. Results for this section are applied to the Cyclotomic Number field. A few basics of the Cyclotomic number field have been covered in Section 2.4.11 of Chapter 2. More related results for the reductions in this section are discussed here.

Setup related to Cyclotomic Number Field

- $\zeta = \zeta_m$, the m^{th} primitive root of unity.
- $K = \mathbb{Q}(\zeta)$ of *even* degree $n = \varphi(m)$, cyclotomic number field.
- $R = \mathbb{O}_K = \mathbb{Z}[\zeta]$ is the ring of integers.
- $q \equiv 1 \mod m$ is a poly(n)-bounded prime.
- $\langle q \rangle = \prod_{i \in \mathbb{Z}_m^*} \mathfrak{q}_i$
- Field automorphism $\tau_k(\zeta) = \zeta^k$ for $k \in \mathbb{Z}_m^*$

Results needed for the Reduction: We describe the basic results related to the cyclotomic number field in section 2.4.11. Here we add some more specific results related to the following reductions. We know that cardinality of the set $R_q^\vee$ is q^n and cardinality of the set $R^\vee/(\mathfrak{q}_i R^\vee)$ is q for $i \in \mathbb{Z}_m^*$. The representatives of the q distinct cosets of $R^\vee/(\mathfrak{q}_i R^\vee)$ can be taken to be the elements of the set $\{0, \ldots, q-1\}$. Chinese Remainder Theorem (CRT) can be used to show that there is an isomorphism between $R_q^\vee$ and $\bigoplus_{i\in\mathbb{Z}_m^*}(R^\vee/(\mathfrak{q}_i R^\vee))$. Let $\mathfrak{I}$ be the isomorphism between $R_q^\vee$ and $\bigoplus_{i\in\mathbb{Z}_m^*}(R^\vee/(\mathfrak{q}_i R^\vee))$ and $\mathfrak{I}$. can be efficiently computed in both the forward and the backward directions. We write $\mathfrak{I}(w) = (w_i)_{i\in\mathbb{Z}_m^*}$, where $w \in R_q^\vee$ and $w_i \in (R^\vee/(\mathfrak{q}_i R^\vee))$ for $i \in \mathbb{Z}_m^*$. If we have w_i's for each $i \in \mathbb{Z}_m^*$ we can construct w such that $w = \mathfrak{I}^{-1}(w_i)_{i\in\mathbb{Z}_m^*}$. For $i \in \mathbb{Z}_m^*$, let $w_i \in \{0, \ldots, q-1\}$ represent a coset of $R^\vee/(\mathfrak{q}_i R^\vee)$. Given $(w_i)_{i\in\mathbb{Z}_m^*}$, it is possible to efficiently construct $w \in R_q^\vee$ such that the i-th component of $\mathfrak{I}(w)$ is represented by w_i. For the sake of notational convenience, we let w denote $\mathfrak{I}^{-1}((w_i)_{i\in\mathbb{Z}_m^*})$. Extending the same way we can define an isomorphism $\mathfrak{I}'$ between $(R_q^\vee)^d$ and $(\bigoplus_{i\in\mathbb{Z}_m^*}(R^\vee/(\mathfrak{q}_i R^\vee)))^d$. Here $\mathfrak{I}'$ is nothing but the simultaneous use of d $\mathfrak{I}$ isomorphisms. Extending the same logic we can say that $\mathfrak{I}$' can be efficiently computed in both the forward and the backward directions. Here the input to $\mathfrak{I}'$ is a vector $\mathbf{w} \in (R_q^\vee)^d$ of d elements. Let $\mathbf{w} = (\hat{w_1}, \hat{w_2}, \ldots, \hat{w_d})$. So the i-th $\mathfrak{I}$ of $\mathfrak{I}'$ will act on the element $\hat{w_i}$ the same way it is defined to work. Each $\mathfrak{I}$ will output n elements from $R^\vee/(\mathfrak{q}_i R^\vee)$ for $i \in \mathbb{Z}_m^*$. Hence $\mathfrak{I}'$ will concatenate the outputs of $\mathfrak{I}$ and provide $n{\cdot}d$ elements as output. Let $\mathbf{w} = (\hat{w_1}, \hat{w_2}, \ldots, \hat{w_d}) \in (R_q^\vee)^d$ as before, implies $\mathfrak{I}'(\mathbf{w}) = \mathfrak{I}(\hat{w_1}), \ldots, \mathfrak{I}(\hat{w_d})$. For an element $\hat{w_i} \in R_q^\vee$, we can have $\mathfrak{I}(\hat{w_i}) = (w_{ij})_{j\in\mathbb{Z}_m^*}$ where $w_{ij} \in (R^\vee/(\mathfrak{q}_j R^\vee))$ for $j \in \mathbb{Z}_m^*$. Again for the sake of notational convenience, we let $\mathbf{w}$ denote $(\mathfrak{I}^{-1}((\hat{w_{1i}})_{i\in\mathbb{Z}_m^*}), \ldots, \mathfrak{I}^{-1}((\hat{w_{di}})_{i\in\mathbb{Z}_m^*}))$.

Hybrid Distribution ($A^i_{\mathbf{s},r}$): Let the representatives of $\mathbb{Z}_m^*$ be chosen from the set $\{1, \ldots, m-1\}$ with the usual ordering. For $i \in \mathbb{Z}_m^*$, let $i-$ denote the largest element in $\mathbb{Z}_m^*$ which is less than i with the convention that $1-$ is taken to be 0. The distribution $A^i_{\mathbf{s},r} \in (R_q)^d \times \mathbb{T}$ for $i \in \mathbb{Z}_m^* \cup \{0\}$, an arbitrary $\mathbf{s} \in (R_q^\vee)^d$ and positive real number r is following. A sample from $A^{(M)}_{\mathbf{s},r}$ consists of a pair $(\mathbf{a}, \mathbf{b})$, where $\mathbf{a} \in (R_q)^d$ and $\mathbf{b} = \sigma(\langle \mathbf{a}{\cdot}\mathbf{s}\rangle/q) + \mathbf{e} \mod \sigma(R^\vee))$, where $\mathbf{e}$ is chosen from H following the distribution D_r. A sample from $A^i_{\mathbf{s},r}$ is a sample from $A^{(M)}_{\mathbf{s},r}$ whose k-th component for $k \leq i$ has been randomised by adding a uniform random $h_k \in \{0, 1, \ldots, q-1\}$ to the k-th component of $\langle \mathbf{a}{\cdot}\mathbf{s}\rangle$.

For $i \in \mathbb{Z}_m^*$, let $\chi(i)$ be the following distribution over $R_q^\vee$. For $k \in \mathbb{Z}_m^*$, choose $h_k \in \{0, \ldots, q-1\}$ as follows: $h_k = 0$ for $k > i$; and for $k \leq i$, the h_k's are chosen independently and uniformly; return $h = \mathfrak{I}^{-1}((h_k)_{k\in\mathbb{Z}_m^*})$.

A sample from $A^i_{\mathbf{s},r}$ is obtained as follows. For $i \in \mathbb{Z}^*_m$, let h be sampled from $\chi(i)$. Choose $(\mathbf{a}, \mathbf{b}) \leftarrow A_{s,r}$ and output $(\mathbf{a}, \mathbf{b} + \sigma(h)/q \bmod \sigma(R^\vee))$ as a sample from $A^i_{\mathbf{s},r}$; for $i = 0$, the distribution $A^0_{\mathbf{s},r}$ is defined to be $A^{(M)}_{\mathbf{s},r}$.

A sample from $A^i_{\mathbf{s},r}$ hides information about $\mathbf{s}$ with respect to the factors $\mathfrak{q}_k$ of $\langle q \rangle$ for $k \in \mathbb{Z}^*_m$ and $k \leq i$. For $i \in \mathbb{Z}^*_m$ or $i = 0$, as i increases from 0 to $m-1$, information about $\mathbf{s}$ is hidden in one more $\mathfrak{q}_i$-component than in the previous step. At the beginning, i.e. $i = 0$, all the components in the output of $\mathfrak{I}$ carry information about $\mathbf{s}$, while at the end, i.e., $i = m-1$, the element $\langle \mathbf{a} \cdot \mathbf{s} \rangle + h$ is a uniform random element of $R^\vee_q$ which is independent of both $\mathbf{s}$ and $\mathbf{a}$. So for a sample $(\mathbf{a}, \mathbf{b})$ drawn from $A^{m-1}_{s,r}$, $\mathbf{a}$ is uniform over R_q and $\mathbf{b}$ is independent of $\mathbf{a}$; further, $\mathbf{b}$ is the sum modulo $\sigma(R^\vee)$ of a uniform random element of $\sigma(R^\vee_q)/q$ and an element drawn from the distribution D_r. Consequently, a sample drawn from $A^{m-1}_{\mathbf{s},r}$ is almost uniform over $R_q \times \mathbb{T}$.

5.3.1 Some Intermediate Problems

We need a few intermediate problems to reduce the search module-LWE problem to the decision module-LWE problem.

$\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$: This is one of the variants of the search module-LWE problem. For $i \in \mathbb{Z}^*_m$ the $\mathfrak{q}_i$-MLWE$_{q,\alpha}$ problem is the following. Given access to $A^{(M)}_{s,r}$ for some arbitrary $\mathbf{s} \in (R^\vee_q)^d$ and and a positive real number $r \leq \alpha$ find $(\mathbf{s} \mod \mathfrak{q}_i R^\vee)$. Based on the result in the previous section, let $\mathbf{s} = (s_1, \ldots, s_d) \in (R^\vee_q)^d$. So, $\mathfrak{I}'(\mathbf{s}) = \mathfrak{I}(s_1), \ldots, \mathfrak{I}(s_d)$. We need to find i^{th} components of $\mathfrak{I}'(\mathbf{s})$ or the set $\{s'_1, \ldots, s'_d\}$, such that $s'_j = i^{th}$ component of $\mathfrak{I}(s_j)$ where $j \in [d]$ and $i \in \mathbb{Z}^*_m$.

Module DLWE (variable width) relative to $\mathfrak{q}_i$: For $i \in \mathbb{Z}^*_m$ and a positive real number α, the module-VDLWE$^i_{q,\leq\alpha}$ problem is the following. Given access to $A^i_{\mathbf{s},r}$ for $\mathbf{s} \in (R^\vee_q)^d$, positive real number $r \leq \alpha$ and $j \in \{i, i-\}$, the requirement is to find j. In other words, the solver must determine whether or not the i-th component of the distribution has been randomized.

Module DLWE (fixed width) relative to $\mathfrak{q}_i$ Let $i \in \mathbb{Z}^*_m$ and $r_0 > 0$ be a real number. The module-FDLWE$^i_{q,r_0}$ problem is the following. Choose $\mathbf{s}$ uniformly at random from $(R^\vee_q)^d$. The requirement is to distinguish between $A^i_{\mathbf{s},r}$ and $A^{i-}_{\mathbf{s},r}$.

Here we append F in front of DLWE to distinguish between the fixed-width error distribution and the variable-width error distribution. In case of later, we append V.

A distinguisher $\mathcal{D}_1$ is a probabilistic polynomial-time algorithm that solves module-FDLWE$^i_{q,r_0}$ problem. $\mathcal{D}_1$ takes a list $\mathcal{T}$ as an input. List $\mathcal{T}$ contains samples from $A^j_{\mathbf{s},r_0}$, where $j \in \{i, i-\}$. $\mathcal{D}_1$ tries to predict j.

For a fixed $\mathbf{s} \in (R_q^\vee)^d$, let $p_{s,0}$(resp. $p_{s,1}$) be the probability that $\mathcal{D}_1$ outputs 1 when $\mathcal{T}$ consists of samples from $A^{i-}_{\mathbf{s},r_0}$ (resp. $A^i_{\mathbf{s},r_0}$), where the probability is taken over all components of the input other than $\mathbf{s}$ as well as the internal coin tosses of $\mathcal{D}_1$. The advantage of the distinguisher is $|p_{s,0} - p_{s,1}|$. For $\epsilon_1, \epsilon_2 \in (0, 1]$, we say that $\mathcal{D}_1$ is an (ϵ_1, ϵ_2)-distinguisher if $\mathcal{D}_1$ has advantage at least ϵ_2 for at least a proportion ϵ_1 of the set of possible $\mathbf{s} \in (R_q^\vee)^d$.

Like the previous section, we define the algorithms that solve the intermediate problems.

- $\mathcal{A}_6$: an algorithm to solve $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$.
- $\mathcal{A}_7$: an algorithm to solve module-VDLWE$^i_{q,\leq\alpha}$.
- $\mathcal{D}_1$: a distinguisher to solve module-FDLWE$^i_{q,r_0}$.
- $\mathcal{D}_2$: a distinguisher to solve module-DLWE$_{q,r_0}$.

In the following sections, we describe each algorithm in greater detail to find the tightness gap of each reduction. We need some subroutines for algorithmic representation of the reductions, which are as follows.

solveQLWE$_{q,\leq\alpha}(\mathcal{I}, i)$: This is the oracle to solve $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$. The list $\mathcal{I}$ consists of N^c samples from $A^{(M)}_{\mathbf{s},r}$, where $r \leq \alpha$.

solveVDLWE$_{q,\leq\alpha}(\mathcal{I}, i)$: This is the oracle to solve module-VDLWE$^i_{q,\leq\alpha}$. The list $\mathcal{I}$ consists of N^c samples from $A^{(M)}_{\mathbf{s},r}$, where $r \leq \alpha$.

solveFDLWE$_{q,\leq\alpha}(\mathcal{I}, i)$: This is the oracle to solve module-FDLWE$^i_{q,r_0}$. The list $\mathcal{I}$ consists of N^c samples from $A^{(M)}_{\mathbf{s},D_{r_0}}$, where $r_0 > \alpha$.

5.3.2 Reducing module-LWE$_{q,\leq\alpha}$ to $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$

Algorithm 13 represents the reduction from module-LWE$_{q,\leq\alpha}$ to $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$. Here we provide a brief description of the algorithm. Algorithm $\mathcal{A}_6$ solves $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$. We want to

construct an algorithm $\mathcal{A}_5$ to solve module-LWE$_{q,\leq\alpha}$. $\mathcal{A}_5$ computes $\mathbf{s}$, whereas $\mathcal{A}_6$ computes i-th components of $\mathfrak{I}'(\mathbf{s})$. If all the component for $i \in \mathbb{Z}_m^*$ can be computed, $\mathbf{s}$ can be computed easily from the isomorphism $\mathfrak{I}'$. To compute all the components $\mathcal{A}_5$ uses automorphism. To get j-th components from i-th component, it does the following. For each $j \in \mathbb{Z}_m^*$, let $k = j \cdot i^{-1} \mod m$. $\mathcal{A}_5$ transforms the LWE samples by using automorphism on them, as $(\tau_k(\mathbf{a}), \tau_k(\mathbf{b}))$. $\mathcal{A}_5$ provides $(\tau_k(\mathbf{a}), \tau_k(\mathbf{b}))$ to $\mathcal{A}_6$ whenever $\mathcal{A}_6$ makes oracle query for LWE samples. Let $\mathcal{A}_6$ outputs $\mathbf{s}_j$ as the j-th components of $\mathbf{s}$. When all the components of $\mathfrak{I}'(\mathbf{s})$ are obtained, $\mathbf{s}$ is computed using $\mathfrak{I}'$ in the reverse direction.

Algorithm 13 Algorithm to solve M-LWE$_{q,\alpha}$ using an $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$ oracle.

function solveMLWE$(\mathcal{I}, i)$
  for $j \in \mathbb{Z}_m^*$ **do**
    $\mathcal{J} \leftarrow \{\}$;
    for each $(\mathbf{a} = (a_1, \ldots, a_d), \mathbf{b} = (b_1, \ldots, b_d)) \in \mathcal{I}$ **do**
      $\mathbf{a} \leftarrow (\tau_{j/i}(a_1), \tau_{j/i}(a_2), \ldots, \tau_{j/i}(a_d)) \in (R_q)^d$;
      $\mathbf{b} \leftarrow (\tau_{j/i}(b_1), \tau_{j/i}(b_2), \ldots, \tau_{j/i}(b_d)) \in \mathbb{T}$;
      $\mathcal{J} \leftarrow \mathcal{J} \cup \{(\mathbf{a}, \mathbf{b})\}$;
    end for
    $(t_{1j}, t_{2j}, \ldots, t_{dj}) \leftarrow$ solveQLWE$_{q,\leq\alpha}(\mathcal{J}, i)$;
    $t_{1j} \leftarrow \tau_{j/i}^{-1}(t_{1j}), \ldots, t_{dj} \leftarrow \tau_{j/i}^{-1}(t_{dj})$;
  end for
  for $k = 1$ to d **do**
    $s_k \leftarrow \mathfrak{I}^{-1}((t_{kj})_{j \in \mathbb{Z}_m^*})$;
  end for
  return $\mathbf{s} = (s_1, s_2, \ldots, s_d)$
end function.

Proposition 47. *$\mathcal{A}_5$ invokes $\mathcal{A}_6$ a total of n times. The numbers of LWE queries made by $\mathcal{A}_5$ and $\mathcal{A}_6$ are equal.*

5.3.3 Reducing $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$ to module-VDLWE$^i_{q,\leq\alpha}$

We have an oracle to solve module-VDLWE$^i_{q,\leq\alpha}$, i.e. algorithm $\mathcal{A}_7$. We need to find an algorithm $\mathcal{A}_6$ to solve $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$. Let $\mathbf{s} = (s_1, s_2, \ldots, s_d)$ be secret vector. The algorithmic representation of the procedure is described in algorithm 14, where we recover the i-th component of $\mathfrak{I}(s_1)$. The same procedure can be repeated d times along with the change in the vector $\mathbf{v}$ to find the i-th components of $\mathfrak{I}'(\mathbf{s})$, or a vector $(\mathfrak{I}(s_1), \ldots, \mathfrak{I}(s_d))$. Each component of $\mathfrak{I}(s_i)$ can be represented by an element from the set $\{0, 1, \ldots, q-1\}$. So

the actual output will be a vector of length d, where each element will be from the set $\{0, 1, \ldots, q-1\}$. Algorithm $\mathcal{A}_6$ does the following. For each $x \in \{0, 1, \ldots, q-1\}$, it computes an element $g \in R_q^\vee$ such that $\mathfrak{I}(g) = x$ in the i-th component and $\mathfrak{I}(g) = 0$ in all other components. $\mathcal{A}_6$ provides the LWE samples to $\mathcal{A}_7$ in following manner. Let $(\mathbf{a}, \mathbf{b}) \in A_{\mathbf{s},r}^{(M)}$ where $r \leq \alpha$. $\mathcal{A}_6$ computes a vector $\mathbf{v} \in (R_q)^d$ such that the i-th components of $\mathfrak{I}(y)$ are chosen uniformly at random from $\{0, 1, \ldots, q-1\}$ and all other components of $\mathfrak{I}(y)$ are equal to zero. $\mathcal{A}_6$ computes $\mathbf{a}' \leftarrow \mathbf{a} + \mathbf{v}$ and $\mathbf{b}' \leftarrow \mathbf{b} + \sigma((h + v{\cdot}g)/q)$, where $h \in R_q^\vee$ is used to randomises the first $i-$ components of the output sample. $\mathcal{A}_6$ provides $\mathcal{A}_7$ with the samples $(\mathbf{a}', \mathbf{b}')$. When the value of x is equal to the i-th component of $\mathfrak{I}(\mathbf{s_1})$, then the samples $(\mathbf{a}', \mathbf{b}')$, input to $\mathcal{A}_6$ are from the distribution $A_{\mathbf{s},r}^{i-}$, in all other cases the input samples are from $A_{\mathbf{s},r}^{i}$. Hence $\mathcal{A}_6$ returns the corresponding x as the desired output only when $\mathcal{A}_7$ returns $i-$. In order to find the i-th component of $\mathfrak{I}(s_j)$, we need to start the algorithm 14 with the initial vector $\mathbf{v} = (0, \ldots, y \ldots, 0) \in (R_q)^d$, i.e. y in the j-th component of $\mathbf{v}$ and all other $d-1$ components are set to 0.

Algorithm 14 Algorithm to solve $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$ using an module-VDLWE$^i_{q,\leq\alpha}$ oracle.

function solveQLWE$(\mathcal{I}, i)$
  Choose $y \in R_q$, s.t, y is uniformly random $\mod \mathfrak{q}_i$ and is equal to 0 $\mod \mathfrak{q}_j \; \forall \; j \neq i$, and $i, j \in \mathbb{Z}_m^*$;
  $\mathbf{v} = (y, 0, \ldots, 0) \in (R_q)^d$;
  Choose $h \in R_q^\vee$, s.t., h is uniformly random and independent $\mod \mathfrak{q}_j R^\vee \; \forall \; j < i$ and equal to 0 $\mod \mathfrak{q}_j R^\vee \; \forall \; j \geq i$ and $i, j \in \mathbb{Z}_m^*$;
  for each $g \in R_q^\vee$ **do**
    $\mathcal{J} \leftarrow \{\}$;
    for each $(\mathbf{a}, \mathbf{b}) \in \mathcal{I}$ **do**
      $\mathbf{a}' \leftarrow \mathbf{a} + \mathbf{v}$;
      $\mathbf{b}' \leftarrow \mathbf{b} + \sigma((h + v{\cdot}g)/q)$;
      $\mathcal{J} \leftarrow \mathcal{J} \cup \{(\mathbf{a}', \mathbf{b}')\}$;
    end for
    $j \leftarrow$ solveVDLWE$(\mathcal{J}, i)$;
    if $j == i$ **then**
      return g;
    end if
  end for
end function.

Proposition 48. *$\mathcal{A}_6$ invokes $\mathcal{A}_7$ at most $q{\cdot}d$ times. The numbers of LWE queries made by $\mathcal{A}_6$ and $\mathcal{A}_7$ are equal.*

5.3.4 Reducing module-$\text{VDLWE}^i_{q,\leq\alpha}$ to module-FDLWE^i_{q,r_0}

In this section we investigate a polynomial time reduction between module-$\text{VDLWE}^i_{q,\leq\alpha}$ problem to module-FDLWE^i_{q,r_0} problem. We have a distinguisher $\mathcal{D}_1$ to solve module-FDLWE^i_{q,r_0} problem. We need to device an algorithm $\mathcal{A}_7$ to solve module-$\text{VDLWE}^i_{q,\leq\alpha}$. The module-$\text{VDLWE}^i_{q,\leq\alpha}$ problem is said to be a worst-case problem as we need to solve this for an arbitrary value of $\mathbf{s} \in (R_q)^d$. On the other hand, module-FDLWE^i_{q,r_0} is said to be an average case problem as we need to solve this for the values of $\mathbf{s}$ chosen uniformly from $(R_q)^d$. Though these terms (worst-case, average-case) have not been used explicitly in [LS15], the main idea has been borrowed from [Reg09] and [LPR13]. We have distinguished these problems as variable-length error problems and fixed-length error problems respectively. The reduction between these two problems has been stated in [LS15] for elliptical Gaussian error distribution only. [LPR13] has presented this problem in both the settings, viz elliptical and spherical Gaussian distributions. For practical purposes, only spherical distribution is chosen. Hence we limit our focus only to spherical distributions (variable length or fixed length) for the whole analysis.

Here we restate the simplified version of the lemma analogous to lemma 5.16 of [LPR13]. Originally in this lemma, the module-VDLWE problem has been stated for the family of elliptical Gaussian distributions but here we consider the family of spherical Gaussian distributions.

Lemma 49. *For $\alpha > 0, l \geq 1$ and for all $i \in \mathbb{Z}^*_m$, there is a randomized polynomial time algorithm which reduces* module-$\text{VDLWE}^i_{q,\leq\alpha}$ *problem to* module-FDLWE^i_{q,r_0} *problem, where l is the number of LWE samples, ${N_3}^{1/4} > \sqrt{N}$ and*

$$r_0 = \alpha \cdot \left(\frac{N N_3 l}{\ln (N N_3 l)}\right)^{1/4}. \tag{5.4}$$

We describe the reduction in the algorithm 15 and elaborate on the same in detail here. $\mathcal{A}_7$ has access to a list $\mathcal{T} = ((\mathbf{a}_k, \mathbf{b}_k))_{1\leq k\leq l}$ of l samples from $A^j_{\mathbf{s},r}$, where $r \leq \alpha$ and $j \in \{i, i-\}$. We define another list $\mathcal{T}' = ((\mathbf{a}_k{}', \mathbf{b}_k{}'))_{1\leq k\leq l}$ where $\mathbf{a}_k{}' = \mathbf{a}_k$ and $\mathbf{b}_k{}' = \mathbf{b}_k + \sigma(\langle \mathbf{a}_k, \mathbf{t}\rangle) + \mathbf{f}_k (\mod \sigma(R^\vee))$. Here $\mathbf{t} \in (R^\vee_q)^d$ and $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_l$ are chosen independently from D_{r_0}. Error vector in $\mathbf{b}_k$ follows D_r and $\mathbf{f}_k$ follows D_{r_0}, hence error vector in $\mathbf{b}_k{}'$ follows $D_{r'}$ where $r' = \sqrt{r^2 + r_0^2}$. So, the samples from $\mathcal{T}'$ are from $A^j_{\mathbf{s}+\mathbf{t},r'}$. Now prepare another list $\mathcal{T}''$ from $\mathcal{T}'$. In $\mathcal{T}''$, $\mathbf{a}_k$ remains unchanged but $\mathbf{b}_k$ is modified as follows. $\mathbf{b}_k = \mathbf{b}_k + \sigma(h_k)/q (\mod (R^\vee))$, where h_k's are uniformly random and independent $\mod \mathfrak{q}_j R^\vee$ for $j \leq i$ and

h_k's are $0 \mod \mathfrak{q}_j R^\vee$ for $j > i$. So in $\mathcal{T}''$ we randomize first i components of $\langle \mathbf{a}_k, \mathbf{s}+\mathbf{t}\rangle$. So, it is clear that $\mathcal{T}''$ has samples from $A^i_{\mathbf{s}+\mathbf{t},r'}$.

We construct algorithm 15 to simulate $\mathcal{A}_7$ by using the distinguisher $\mathcal{D}_1$ as follows. It has two nested loops. The outer loop runs N_2 times and the inner loop runs N_3 times. In each iteration of the outer loop, $\mathcal{A}_7$ chooses $\mathbf{t}$ uniformly at random from $(R^\vee_q)^d$. In each iteration the inner loop using samples from $A^i_{\mathbf{s},r_0}$ a list $\mathcal{T}$ is constructed, then using $\mathcal{T}$ and $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_l$ another two lists ($\mathcal{T}'$ and $\mathcal{T}''$) are created. In the last part of the inner loop $\mathcal{D}_1$ is called with the inputs $\mathcal{T}'$ and $\mathcal{T}''$ and captures the one bit output in the variables cnt_0 and cnt_1 respectively. At the end of the inner loop, we capture the estimated probabilities in the variables $\hat{\mathfrak{p}_0}$ and $\hat{\mathfrak{p}_1}$ respectively. Here $\hat{\mathfrak{p}_0}$ and $\hat{\mathfrak{p}_1}$ is estimated probabilities of $\mathfrak{p}_0$, $\mathfrak{p}_1$ that $\mathcal{D}_1$ accepts input from $A^j_{\mathbf{s}+\mathbf{t},r'}$ and $A^i_{\mathbf{s}+\mathbf{t},r'}$ respectively. If any of the N_2 outer loop results in $|\hat{\mathfrak{p}_0} - \hat{\mathfrak{p}_1}| \geq \epsilon_2/4$, the algorithm returns $i-$ and halts else it returns i and halts when this condition is not satisfied by any of the N_2 iterations.

Here $\mathcal{D}_1$ is an (ϵ_1, ϵ_2) distinguisher which works for samples error drawn from $D^l_{r_0}$, where $\mathcal{T}'$ and $\mathcal{T}''$ contains error samples from $D^l_{r'}$. The following analysis shows how $\mathcal{D}_1$ works with the input samples drawn from $\mathcal{T}'$ and $\mathcal{T}''$. Methods of hypothesis testing can be adapted to analyze the errors that may occur in algorithm 15. Algorithm 15 may return an incorrect answer in two ways. First, when $j = i$ it returns $i-$, and second when $j = i-$ and it returns i. The first is Type-1 failure and the second is Type-2 failure. When $j = i$ both lists $\mathcal{T}'$ and $\mathcal{T}''$ follow $A^i_{\mathbf{s}+\mathbf{t},r'}$, so $\mathfrak{p}_0 = \mathfrak{p}_1$. For each of the N_2 iterations of the outer loop we have

$$\Pr[\mathfrak{p}_0 - \epsilon_2/8 \leq \hat{\mathfrak{p}_0} \leq \mathfrak{p}_0 + \epsilon_2/8] \geq 1 - 2\exp(-N_3\epsilon_2^2/32)$$
$$\Pr[\mathfrak{p}_1 - \epsilon_2/8 \leq \hat{\mathfrak{p}_1} \leq \mathfrak{p}_1 + \epsilon_2/8] \geq 1 - 2\exp(-N_3\epsilon_2^2/32)$$

due to the additive form of the Chernoff-Hoeffding bound. As $\mathfrak{p}_0 = \mathfrak{p}_1$, it implies that $\Pr[\hat{\mathfrak{p}_0} - \hat{\mathfrak{p}_1}] \geq 1 - 4\exp(-N_3\epsilon_2^2/32)$. So, for all N_2 outer loop iterations the Type-1 failure probability becomes at most $4N_2\exp(-N_3\epsilon_2^2/32)$. When $j = i-$, $\mathcal{T}'$ follows $A^{i-}_{\mathbf{s}+\mathbf{t},r'}$ and $\mathcal{T}''$ follows $A^i_{\mathbf{s}+\mathbf{t},r'}$. In any of the N_3 iterations of the inner loop let $\mathbf{z}_1, \ldots, \mathbf{z}_l$ be the errors in samples of $\mathcal{T}', \mathcal{T}''$. Let $\mathbf{z}$ be the concatenation of $\mathbf{z}_1, \ldots, \mathbf{z}_l$, so $\mathbf{z}$ follows $D^{N_3 l}_{r'}$. Suppose $\mathbf{z}$ follows $D^{N_3 l}_{r_0}$ instead of $D^{N_3 l}_{r'}$. Then we can use $\mathcal{D}_1$ appropriately. Here we change the Gaussian error width from r' to r_0, later we will have to compute the correction factor using Renyi Divergence. We denote the corresponding probabilities and their estimates by $p_0, p_1, \hat{p_0}$ and $\hat{p_1}$ respectively. Let $p_{\mathbf{s}+\mathbf{t},0}$ and $p_{\mathbf{s}+\mathbf{t},1}$ respectively denote the probabilities p_0

and p_1 corresponding to a particular value of $\mathbf{t}$. Similarly, let $\hat{p}_{\mathbf{s}+\mathbf{t},0}$ and $\hat{p}_{\mathbf{s}+\mathbf{t},1}$ respectively denote the estimates $\hat{p_0}$ and $\hat{p_1}$ corresponding to a particular value of $\mathbf{t}$. Lastly let $\hat{p}_{\mathbf{s}+\mathbf{t},\mathbf{z},0}$ and $\hat{p}_{\mathbf{s}+\mathbf{t},\mathbf{z},1}$ respectively denote the estimates $\hat{p_0}$ and $\hat{p_1}$ corresponding to a particular value of $\mathbf{t}$ and $\mathbf{z}$. We say that a value $\mathbf{s}+\mathbf{t}$ is good if $|p_{\mathbf{s}+\mathbf{t},0} - p_{\mathbf{s}+\mathbf{t},1}| \geq \epsilon_2$. From the definition of an (ϵ_1, ϵ_2) distinguisher, the probability of a good $\mathbf{s}+\mathbf{t}$ is at least ϵ_1. If we use the additive form of the Chernoff-Hoeffding for good $\mathbf{s}+\mathbf{t}$ it follows.

$$\Pr[p_{\mathbf{s}+\mathbf{t},0} - \epsilon_2/4 \leq \hat{p}_{\mathbf{s}+\mathbf{t},0} \leq p_{\mathbf{s}+\mathbf{t},0} + \epsilon_2/4] \geq 1 - 2\exp(-N_3\epsilon_2^2/8)$$
$$\Pr[p_{\mathbf{s}+\mathbf{t},1} - \epsilon_2/4 \leq \hat{p}_{\mathbf{s}+\mathbf{t},1} \leq p_{\mathbf{s}+\mathbf{t},1} + \epsilon_2/4] \geq 1 - 2\exp(-N_3\epsilon_2^2/8)$$

From above equations along with the condition that $|p_{\mathbf{s}+\mathbf{t},0} - p_{\mathbf{s}+\mathbf{t},1}| \geq \epsilon_2$ we get

$$\Pr[|\hat{p}_{\mathbf{s}+\mathbf{t},0} - \hat{p}_{\mathbf{s}+\mathbf{t},1}| \geq \epsilon_2/2] \geq 1 - 4\exp(-N_3\epsilon_2^2/8) \tag{5.5}$$

If we take N_3 a constant factor of ϵ_2^{-2} the difference $|\hat{p}_{\mathbf{s}+\mathbf{t},0} - \hat{p}_{\mathbf{s}+\mathbf{t},1}|$ will be at least $\epsilon_2/2$ with probability at most 1.

$$N_3 = O(\epsilon_2^{-2}) \tag{5.6}$$

Given a good $\mathbf{s}+\mathbf{t}$, we say that $\mathbf{z}$ is good if $|\hat{p}_{\mathbf{s}+\mathbf{t},\mathbf{z},0} - \hat{p}_{\mathbf{s}+\mathbf{t},\mathbf{z},1}| \geq \epsilon_2/4$ holds. The probability of a good $\mathbf{z}$ is at least $\epsilon_2/4$ (Refer Proposition 53). Now when we change the error distribution from $D_{r_0}^{N_3 l}$ to $D_{r'}^{N_3 l}$, the probability of a good $\mathbf{z}$ under $D_{r'}^{N_3 l}$ is at least $\epsilon_2^2/(256NN_3l)^{-1/2}$ (Refer Proposition 57). So the probability of a good pair $(\mathbf{s}+\mathbf{t}, \mathbf{z})$ where $\mathbf{z}$ follows $D_{r'}^{N_3 l}$ is at least $\epsilon_1\epsilon_2^2/(256NN_3l)^{-1/2}$. If N_2 is around $(256NN_3l)^{1/2}/\epsilon_1\epsilon_2^2$ then with probability exponentially close to 1 a good tuple will be encountered in one of the iterations of the outer loop. Type-2 failure can occur in two ways. The first way is that in none of the N_1 iterations, a good tuple is obtained. The second way is that for a good tuple, the condition $|\hat{\mathbf{p}}_{\mathbf{s}+\mathbf{t},\mathbf{z},0} - \hat{\mathbf{p}}_{\mathbf{s}+\mathbf{t},\mathbf{z},1}| \geq \epsilon_2/4$ does not hold. The above analysis shows that the probability of either of these errors is exponentially small.

Proposition 50. *$\mathcal{A}_7$ invokes $\mathcal{D}_1$ at most $N_2 \cdot N_3$ times, which is about $(256Nl)^{1/2}{N_3}^{3/2^{1/2}}/\epsilon_1\epsilon_2^2$ or $(\epsilon_1\epsilon_2^5)^{-1}(256Nl)^{1/2}$, where N_3 is about ϵ_2^{-2}. The number of times $\mathcal{A}_7$ calls its LWE oracle is about $(\epsilon_1\epsilon_2^5)^{-1}(256N)^{1/2}l^{3/2}$.*

Algorithm 15 Reducing module-VDLWE$^{i}_{q,\leq\alpha}$ to module-FDLWE$^{i}_{q,r_0}$

function solveFDLWE($\mathcal{L}, i$)
  for $k = 1$ to N_2 **do**;
    Choose a uniform $\mathbf{t} \in (R_q^\vee)^d$;
    $\text{cnt}_0 \leftarrow 0$, $\text{cnt}_1 \leftarrow 0$;
    for $l = 1$ to N_3 **do**;
      Obtain a list $\mathcal{T}$ of l samples from $A^i_{\mathbf{s},r_0}$;
      Choose $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_l$ independently from $D^l_{r_0}$;
      Compute $\mathcal{T}'$ and $\mathcal{T}''$ from $\mathcal{T}$ and $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_l$;
      $\text{cnt}_0 \leftarrow \text{cnt}_0 + \mathcal{D}_1(\mathcal{T}')$, $\text{cnt}_1 \leftarrow \text{cnt}_1 + \mathcal{D}_1(\mathcal{T}'')$;
    end for
    $\hat{\mathfrak{p}_0} \leftarrow \text{cnt}_0/N_3, \hat{\mathfrak{p}_1} \leftarrow \text{cnt}_1/N_3$;
    if then$|\hat{\mathfrak{p}_0} - \hat{\mathfrak{p}_1}| \geq \epsilon_2/4$;
      return $i-$;
    end if
  end for
  return i;
end function.

5.3.5 Reducing module-FDLWE$^{i}_{q,r_0}$ to module-DLWE$_{q,r_0}$

The last reduction is from module-FDLWE$^{i}_{q,r_0}$ problem to module-DLWE$_{q,r_0}$. We have already mentioned that distinguisher $\mathcal{D}_2$ is to solve module-DLWE$_{q,r_0}$. So we need to have distinguisher $\mathcal{D}_1$ from distinguisher $\mathcal{D}_2$ as an oracle. Let $\mathcal{D}_2$ be (δ_1, δ_2)-distinguisher. So, $\mathcal{D}_2$ has an advantage at least δ_2 on a δ_2 fraction of the set of possible values of $\mathbf{s}$. As per the lemma 5.14 of [LPR13], there exists an $i \in \mathbb{Z}^*_m$, we can construct a $(\delta_1/N, \delta_2/N)$-distinguisher $\mathcal{D}_1$, given $\mathcal{D}_2$ as an oracle. If $\mathcal{D}_1$ is a (ϵ_1, ϵ_2)-distinguisher, then we get $\epsilon_1 = \delta_1/N$ and $\epsilon_2 = \delta_2/N$. Lemma 5.14 of [LPR13] does not explicitly show how to construct $\mathcal{D}_1$ but rather proves its existence. In order to determine proper i, we need to perform the end-to-end (M-SIVP$_\gamma$ to module-DLWE$_{q,r_0}$) exercise for each possible i, i.e. N times, and select the set of independent vectors of the smallest maximum length. We have already shown that N_3 is around ϵ_2, so the value of N_3 comes around the following

$$N_3 = N^2 {\delta_2}^{-2} \tag{5.7}$$

5.3.6 The tightness gap in $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$ to module-DLWE$_{q,r_0}$

Here we need to find out the tightness gap of the end-to-end reduction from $\mathfrak{q}_i$-MLWE$_{q,\leq\alpha}$ to module-DLWE$_{q,r_0}$. From Proportion 47, we get that $\mathcal{A}_5$ invokes $\mathcal{A}_6$ a total of n times. From Proportion 48, we get that $\mathcal{A}_6$ invokes $\mathcal{A}_7$ at most $q{\cdot}d$ times. From Proportion 50, we get that $\mathcal{A}_7$ invokes $\mathcal{D}_1$ at most $N_2{\cdot}N_3$ times, which is about $(256Nl)^{1/2}{N_3}^{3/2^{1/2}}/\epsilon_1\epsilon_2^2$ or $(\epsilon_1\epsilon_2^5)^{-1}(256Nl)^{1/2}$, where N_3 is about ϵ_2^{-2}. $\mathcal{D}_1$ and $\mathcal{D}_2$ are basically identical with related parameters. So the number of times $\mathcal{A}_5$ calls $\mathcal{D}_2$ is about $(Nq){\cdot}(\epsilon_1\epsilon_2^5)^{-1}{\cdot}(256Nl)^{1/2}$. Putting $\epsilon_1 = \delta_1/N$ and $\epsilon_2 = \delta_2/N$ we get that $\mathcal{A}_5$ calls $\mathcal{D}_2$ is about $(N^{15/2}q){\cdot}(\delta_1\delta_2^5)^{-1}{\cdot}(256l)^{1/2} \approx (N^{15/2}q)(\delta_1\delta_2^5)^{-1}l^{1/2}$. Now, we need to find the value N_1, the number of queries made by $\mathcal{A}_5$, $\mathcal{A}_6$ and $\mathcal{A}_7$ are all equal as per Proposition 47, 48 and 50. So, the number of LWE queries made by $\mathcal{A}_5$ is about $(\epsilon_1\epsilon_2^5)^{-1}(256N)^{1/2}l^{3/2}$. Using the values of ϵ_1 and ϵ_2 the value of N_1 comes about $N^6(\delta_1\delta_2^5)^{-1}(256N)^{1/2}l^{3/2} \approx N^{13/2}(\delta_1\delta_2^5)^{-1}l^{3/2}$. Here we take $N_3 = (\delta_2/N)^{-2}$ from Equation 5.7.

5.3.7 The tightness gap in M-SIVP$_\gamma$ to module-DLWE$_{q,r_0}$

The tightness gap for the entire reduction is the number of times $\mathcal{A}_0$ calls $\mathcal{D}_2$. From Section 5.2.3, we get that $\mathcal{A}_0$ calls $\mathcal{A}_5$ approximately $3N^5N_1$. From Section 5.3.6, we get that $\mathcal{A}_5$ calls $\mathcal{D}_2$ approximately $(N^{15/2}q)(\delta_1\delta_2^5)^{-1}l^{1/2}$. Hence the total number of oracle calls is about $(N^{25/2}q)(\delta_1\delta_2^5)^{-1}{\cdot}l^{1/2}N_1$. Putting the estimated value of $N_1 = N^{13/2}(\delta_1\delta_2^5)^{-1}l^{3/2}$, we get the total tightness gap to be about $(N^{19}q)(\delta_1\delta_2^5)^{-2}l^2$. Additionally, we have a factor of N in the number of times $\mathcal{A}_0$ calls $\mathcal{D}_2$, that we get from 5.3.5. If we use $N = n{\cdot}d$ in the above figure we get the following. The number of times $\mathcal{A}_0$ calls $\mathcal{D}_2$ is around $(n^{20}qd^{20})(\delta_1\delta_2^5)^{-2}l^2$. The quantum part of the reduction requires about $3(nd)^2$ logical qubits.

Theorem 51. *Let K be the m-th cyclotomic number field having degree $n = \varphi(m)$ and $R = \mathcal{O}_K$ be its ring of integers. Let M be a d rank module and a subset of K^d. So, the dimension of the module M is $N = n{\cdot}d$. Let r_0 be a positive real number bounded from above by $O(\sqrt{\log N/N})$. Let $\delta_1, \delta_2 \in (0, 1]$. Let q be a prime greater than 2 such that $q \equiv 1 \bmod m$ and $q > (2\omega(\sqrt{\ln N})/r_0) \cdot (NN_3\ell/\ln(NN_3\ell))^{1/4}$, where N_3 is defined in the course of the proof and has magnitude $\tilde{O}(N^2/\delta_2^2)$, and ℓ is a positive integer. Suppose there is a (δ_1, δ_2)-distinguisher $\mathcal{D}_2$ which solves module-DLWE_{q,r_0} given ℓ samples. Then there is a quantum algorithm $\mathcal{A}_0$ requiring approximately $3N^2$ logical qubits to solve M-SIVP_γ, where $\gamma = \tilde{O}(N^{5/4}\ell^{1/4}/(r_0\delta_2^{1/2}))$. The number of times $\mathcal{A}_0$ calls $\mathcal{D}_2$ is about $(n^{20}qd^{20})(\delta_1\delta_2^5)^{-2}l^2$*

5.4 Reduction from ideal SIVP to ring-LWE

The reduction for SIVP for module lattice to module-LWE [LS15] is a generalization of SIVP for the ideal lattice to the ring-LWE by [LPR13]. In module lattice, the dimension of the lattice and degree of LWE is $N = n{\cdot}d$, where n is the degree of the number filed K and d is the rank of module M. If we restrict M for $d = 1$, the algebraic setting corresponds to ideal lattices of dimension $N = n$. The previous analysis is applied to the ring setting without any change. So, we can talk about the tightness result for ideal lattice and ring-LWE by the following theorem using Theorem 51. The analysis related to reduction from ideal SIVP to ring-LWE can be found here [KSSS22].

Theorem 52. *Let K be the m-th cyclotomic number field having degree $n = \varphi(m)$ and $R = \mathcal{O}_K$ be its ring of integers. Let r_0 be a positive real number bounded from above by $O(\sqrt{\log n/n})$. Let $\delta_1, \delta_2 \in (0, 1]$. Let q be a prime greater than 2 such that $q \equiv 1 \bmod m$ and $q > (2\omega(\sqrt{\ln n})/r_0) \cdot (nN_2\ell/\ln(nN_2\ell))^{1/4}$, where N_2 is defined in the course of the proof and has magnitude $\tilde{O}(n^2/\delta_2^2)$, and ℓ is a positive integer. Suppose there is a (δ_1, δ_2)-distinguisher $\mathcal{D}_2$ which solves ring-DLWE_{q,r_0} given ℓ samples. Then there is a quantum algorithm $\mathcal{A}_0$ requiring approximately $3n^2$ logical qubits to solve K-SIVP_γ, where $\gamma = \tilde{O}(n^{5/4}\ell^{1/4}/(r_0\delta_2^{1/2}))$. The number of times $\mathcal{A}_0$ calls $\mathcal{D}_2$ is about $qn^{20}\ell^2 \cdot (\delta_1\delta_2^5)^{-2}$.*

5.5 Details of the analysis in Section 5.3.4

We first show the lower bound on the probability of a good $\mathbf{z}$ for a good $\mathbf{s}+\mathbf{t}$.

Proposition 53. *For a good $\mathbf{s}+\mathbf{t}$, under the error distribution $D_{r_0}^{\ell N_3}$ the probability that $\mathbf{z}$ is good is at least $\epsilon_2/4$.*

Proof. Since $\mathbf{s}+\mathbf{t}$ is good, we have $|\hat{p}_{\mathbf{s}+\mathbf{t},0} - \hat{p}_{\mathbf{s}+\mathbf{t},1}| \geq \epsilon_2/2$. Without loss of generality, we assume $\hat{p}_{\mathbf{s}+\mathbf{t},0} \geq \hat{p}_{\mathbf{s}+\mathbf{t},1} + \epsilon_2/2$. Let Z denote the set of all $N_3\ell$-tuples $\mathbf{z}$, and let Y denote the subset of Z consisting of $\mathbf{z}$ such that $\hat{p}_{\mathbf{s}+\mathbf{t},\mathbf{z},0} \geq \hat{p}_{\mathbf{s}+\mathbf{t},\mathbf{z},1} + \epsilon_2/4$. We claim that Y has measure at least $\epsilon_2/4$. Assume the contrary. We then have

$$\begin{aligned}
\hat{p}_{\mathbf{s}+\mathbf{t},0} &= \int_Y \hat{p}_{\mathbf{s}+\mathbf{t},\mathbf{z},0} D_{r_0}^{\ell N_3}(\mathbf{z}) + \int_{Z\setminus Y} \hat{p}_{\mathbf{s}+\mathbf{t},\mathbf{z},0} D_{r_0}^{\ell N_3}(\mathbf{z}) \\
&< \int_Y 1 \cdot D_{r_0}^{\ell N_3}(\mathbf{z}) + \int_Z (\hat{p}_{\mathbf{s}+\mathbf{t},\mathbf{z},1} + \epsilon_2/4) D_{r_0}^{\ell N_3}(\mathbf{z}) \\
&\leq \epsilon_2/4 + \hat{p}_{\mathbf{s}+\mathbf{t},1} + \epsilon_2/4,
\end{aligned}$$

a contradiction. This shows that the probability of $\mathbf{z}$ being good is at least $\epsilon_2/4$. □

Next, we consider the effect of changing the error distribution from $D_{r_0}^{\ell N_3}$ to $D_{r'}^{\ell N_3}$. To do this, we introduce a quantity whose logarithm is the Rényi divergence of order 2. Let k be a positive integer. For two probability density functions[1] $P, Q : H^k \to \mathbb{R}_{\geq 0}$, let

$$\mathrm{R}(P||Q) = \int_{H^k} \frac{P(x)^2}{Q(x)} dx. \tag{5.8}$$

By an abuse of notation, we write $\mathrm{R}(D||D')$ to denote $\mathrm{R}(P||Q)$, where D and D' are the distributions corresponding to P and Q respectively. For a measurable subset B of H^k, we have,

$$\begin{aligned}
(\mathrm{Pr}_D[B])^2 &= \left(\int_B P(x)dx\right)^2 && (5.9)\\
&\leq \left(\int_B \frac{P(x)^2}{Q(x)}dx\right)\left(\int_B Q(x)dx\right) && (5.10)\\
&\leq \left(\int_{H^k} \frac{P(x)^2}{Q(x)}dx\right)\mathrm{Pr}_{D'}[B] \\
&= \mathrm{R}(D||D')\mathrm{Pr}_{D'}[B]. && (5.11)
\end{aligned}$$

The derivation of (5.10) from (5.9) is made using the Cauchy-Scharwz inequality[2].

Proposition 54. *The minimum value of c such that $x^2/\sqrt{2x^2-1}$ is less than $1+c(x-1)^2$ for $x > 1$ is $c = 2$.*

Proof. We first show that for $x > 1$, $x^2/\sqrt{2x^2-1} < 1 + 2(x-1)^2$. Let $p(x) = (2x^2-1)(1+2(x-1)^2)^2 - x^4$. Then $x^2/\sqrt{2x^2-1} < 1+2(x-1)^2$ if and only if $p(x) > 0$. The polynomial $p(x)$ factors as $p(x) = (8x^3 - 8x^2 + 3x + 9)(x-1)^3 = (8x^2(x-1) + 3x + 9)(x-1)^3$ which is a sum and product of positive numbers for $x > 1$. Hence, it follows that $p(x) > 0$ for $x > 1$.

We next show that if the 2 in $1 + 2(x-1)^2$ is replaced by $c < 2$, then the inequality $x^2/\sqrt{2x^2-1} < 1 + c(x-1)^2$ cannot hold when x is close to 1. We set $\epsilon = x - 1$ and

[1] In Claim 5.11 of [LPR13], the density functions are considered to be over $\mathbb{R}^n$. Here we consider density functions over H^k.

[2] In the Cauchy-Scharwz inequality of the form $\left(\int_B f(x)g(x)dx\right)^2 \leq \left(\int_B f(x)^2dx\right)\left(\int_B g(x)^2dx\right)$, take $f(x) = P(x)/\sqrt{Q(x)}$ and $g(x) = \sqrt{Q(x)}$.

$t = 4\epsilon + 2\epsilon^2$ and use the Taylor series $(1+t)^{-1/2} = 1 - t/2 + 3t^2/8 \pm O(t^3)$. We have

$$\begin{aligned} \frac{x^2}{\sqrt{2x^2-1}} &= (1+\epsilon)^2(1+4\epsilon+2\epsilon^2)^{-1/2} \\ &= (1+2\epsilon+\epsilon^2)(1-2\epsilon-\epsilon^2+3(4\epsilon+2\epsilon^2)^2/8 \pm O(\epsilon^3)) \\ &= 1+2\epsilon^2 \pm O(\epsilon^3). \end{aligned}$$

For $c < 2$ and small ϵ, this is greater than $1 + c\epsilon^2$. □

Proposition 55. *Let $k \geq 1$ be a positive integer and $r_1, r_2 \in \mathbb{R}^+$ be such that $1 < r_2/r_1 < 1 + \sqrt{\ln(nk)/(nk)}/2$. Let D_{r_1} and D_{r_2} be the continuous Gaussian distributions on H having widths r_1 and r_2 respectively. Then*

$$\mathrm{R}(D_{r_1}^k || D_{r_2}^k) \leq \left(1 + \frac{1}{2} \cdot \frac{\ln(nk)}{nk}\right)^{nk}. \tag{5.12}$$

Proof. Direct calculation from the definition of the continuous Gaussian distribution D_r on H shows that for $r > 0$ and $x > 1/\sqrt{2}$, $\mathrm{R}(D_r||D_{xr}) = (x^2/\sqrt{2x^2-1})^n$. For $x > 1$, from Proposition 54, we have $(x^2/\sqrt{2x^2-1})^n$ is smaller than $(1+2(x-1)^2)^n$. So $\mathrm{R}(D_{r_1}^k||D_{r_2}^k) = (\mathrm{R}(D_{r_1}||D_{r_2}))^k \leq (1+\ln(nk)/(2nk))^{nk}$. □

Setting $x = \alpha^2/r_0^2$ in the inequality $1 + x < (1 + x/2)^2$ for $x \neq 0$ and using (5.4), we have

$$1 \leq \frac{r'}{r_0} = \frac{\sqrt{r^2+r_0^2}}{r_0} \leq \sqrt{1+\frac{\alpha^2}{r_0^2}} < 1 + \frac{1}{2}\cdot\frac{\alpha^2}{r_0^2} = 1 + \frac{1}{2}\left(\frac{\ln(NN_3\ell)}{NN_3\ell}\right)^{1/2}. \tag{5.13}$$

Applying Proposition 55 with $k = N_3\ell$, $r_1 = r_0$ and $r_2 = r'$, we obtain

$$\mathrm{R}(D_{r_0}^{\ell N_3}||D_{r'}^{\ell N_3}) \leq (1+\ln(NN_3\ell)/(2NN_3\ell))^{NN_3\ell}. \tag{5.14}$$

Proposition 56. *The right hand side of (5.14) is about $(NN_3\ell)^{1/2}$.*

Proof. The approximation can be seen by setting $x = (2NN_3\ell)/(\ln(NN_3\ell))$ and $m = (\ln(NN_3\ell))/2$ in $(1+1/x)^{mx} \approx e^m$. □

Proposition 57. *For a good $\mathbf{s}+\mathbf{t}$, the measure of the set of good $\mathbf{z}$ under $D_{r'}^{\ell N_3}$ is at least about $\epsilon_2^2/(256NN_3\ell)^{1/2}$.*

Proof. From Section 5.3.4, we have that for a good $\mathbf{s}+\mathbf{t}$ the measure of the set of good $\mathbf{z}$ under $D_{r_0}^{\ell N_3}$ is at least $\epsilon_2/4$. In (5.11), considering B to be the set of good $\mathbf{z}$ and replacing k

by $N_3\ell$, we have

$$\Pr_{D_{r'}^{\ell N_3}}[B] \;\geq\; \frac{\left(\Pr_{D_{r_0}^{\ell N_3}}[B]\right)^2}{\mathrm{R}(D_{r_0}^{\ell N_3}||D_{r'}^{\ell N_3})} \geq \frac{\epsilon_2^2}{16\mathrm{R}(D_{r_0}^{\ell N_3}||D_{r'}^{\ell N_3})} \gtrapprox \frac{\epsilon_2^2}{(256NN_3\ell)^{1/2}}.$$

□

Remark: 1. *In [LPR13], the ratio r_0/α is defined to be $((N\ell)/\ln(N\ell))^{1/4}$. If we use this definition of r_0/α, and take $k = \ell$ in Proposition 55, then instead of (5.14) we would obtain*

$$\mathrm{R}(D_{r_0}^{\ell N_3}||D_{r'}^{\ell N_3}) \;\leq\; (\mathrm{R}(D_{r_0}^{\ell}||D_{r'}^{\ell}))^{N_3} \approx (N\ell)^{N_3/2}. \tag{5.15}$$

Since $N_3 > N^2$, this would lead to super-exponential running time $> N^{N^2}$.

5.6 Conclusion

Throughout this chapter, we have delved into the concept of the tightness gap of the reduction from the SIVP in a module lattice to the module-LWE problem. We have thoroughly explored the intricacies and implications of this reduction and examined various related issues. It is crucial to acknowledge that the reduction we discussed from the ideal SIVP to ring-LWE represents a specific and constrained version of the analysis. Nonetheless, the results and insights gained from this analysis can be directly adapted and extended to the more general case of module-LWE. Overall, this chapter serves as a valuable exploration of the tightness gap of the reduction from module SIVP to module-LWE, providing valuable insights that can be applied to other related reductions and cryptographic constructions.

Chapter 6

Ring LWE for any Ring and any modulus

6.1 Introduction

In the previous chapter, we focused on concrete security and discussed the tightness gap of reductions from SIVP on module lattices to the average case decision module LWE, as presented in the seminal paper by Langlois and Stehlé [LS15]. The main idea of this reduction is based on the work of Lyubashevsky, Peikert, and Regev [LPR13], where the main reduction is presented for ideal lattices and ring-LWE. The tightness gap of reductions in both cases, as discussed in [LPR13] and [LS15], is similar, and both reductions are constrained for cyclotomic number fields.

The choice of a cyclotomic number field for the reductions in [LPR13] and [LS15] is motivated by the desire for efficient computations. Cyclotomic number fields offer useful automorphisms between different embeddings of a number field, which are utilized in the security reductions. However, it should be noted that the hardness of certain lattice problems, such as SVPγ and SIVPγ, is equivalent in cyclotomic settings. While cyclotomic number fields provide certain advantages in terms of efficiency, they also come with limitations. For instance, in the case of Euclidean lattices, the hardness of SIVPγ is at least as hard as SVPγ. However, the hard lattice problems are not as hard as in general lattices, especially in the restricted settings of ideal lattices over a cyclotomic number field. Moreover, cyclotomic rings represent a narrow class of rings, and they are distributed sparsely when considering the set of all number fields. This limited generality poses challenges for ring LWE-based cryptographic applications that may require a more diverse set of rings. This was one of the major limitations of [LPR13]. Peikert, Regev, and Stephens-Davidowitz [PRS17] addressed this issue by providing a reduction for any ring and modulus, aiming to achieve greater generality. However, our investigation reveals that this reduction also lacks tightness, raising concerns about the practical feasibility of cryptographic constructions based on such reductions.

Both the results presented in [Reg09] and [LPR13] utilize similar reduction techniques

but in different ring settings. The main reduction in both the cases shows that the problem known as DGS reduces to the search version of the LWE problem. Furthermore, there is an additional reduction presented to further reduce the search version of LWE to the average-case decision version of LWE. These reductions play a crucial role in establishing the security of cryptographic constructions based on LWE and its ring variants. However, as mentioned earlier, concerns have been raised about the tightness of these reductions, which may impact the practical feasibility of the resulting cryptographic schemes.

In the work presented in [PRS17], the DGS problem is directly reduced to the decision version of the LWE problem. Unlike previous reductions, where DGS was reduced to the BDD problem as an intermediate step, this reduction takes a different approach. Instead of reducing DGS to BDD, it is reduced to a variant of BDD known as the "Gaussian Decoding Problem" (GDP). In the Gaussian Decoding Problem, the error vector $\mathbf{e} \in H$ is sampled from a spherical Gaussian distribution D_r, where r is a parameter (see Definition 51). This alternative reduction approach may offer advantages or insights into the hardness of certain problems in the context of lattice-based cryptography. However, as with any cryptographic reduction, the tightness of the reduction is a critical factor in assessing the practical security of the resulting cryptographic schemes.

To proceed further, we will need the following definitions and results.

Definition 52 (The Set G)**.** *Define the set* G *as*

$$\mathsf{G} \triangleq \{\mathbf{r} = (r_1, \ldots, r_n) \in (\mathbb{R}^+)^n : r_{s_1+s_2+i} = r_{s_1+i}, \forall i \in [1, s_2]\}.$$

where $n = s_1 + 2s_2$.

Here, G is a set of n-dimensional vectors, where each component of the vectors is a positive real number. For each vector, the last s_2 elements are the same as the s_2 elements preceding it.

Definition 53 ($\mathrm{W}_{r,\upsilon,T}$ [PRS17])**.** *For* $r > 0, \upsilon > 0$, *and* $T \geq 1$, *define* $\mathsf{W}_{r,\upsilon,T}$ *as the set of cardinality* $(s_1+s_2)\cdot(T+1)$ *containing for each* $i = 1, \ldots, s_1+s_2$ *and* $j = 0, \ldots, T$ *the vector* $\mathbf{r}_{i,j} \in \mathsf{G}$ *which is equal to* r *in all coordinates except in the* i^{th}, *and the* $(i+s_2)^{th}$ *if* $i > s_1$, *where it is equal to* $r \cdot (1+\upsilon)^j$.

Definition 54 ([PRS17])**.** *Fix an arbitrary* $f(n) = \omega(\log n)$. *For* $\alpha > 0$, *a distribution sampled from* Υ_α *is an elliptical Gaussian* $\mathsf{D}_{\mathbf{r}}$, *where* $\mathbf{r} \in \mathsf{G}$ *is sampled as follows:*

- *for* $i = 1, \ldots, s_1$, *sample* $x_i \leftarrow \mathsf{D}_1$ *and set*

$$r_i^2 \;=\; \alpha^2 \cdot \frac{x_i^2 + f^2(n)}{2}.$$

- *For* $i = s_{1+i}, \ldots, s_1 + s_2$, *sample* $x_i, y_i \leftarrow \mathsf{D}_{1/\sqrt{2}}$ *and set*

$$r_i^2 \;=\; r_{i+s_2}^2 \;=\; \alpha^2 \cdot \frac{x_i^2 + y_i^2 + f^2(n)}{2}.$$

These definitions are taken from [PRS17] as these are integral for the analysis presented in the following sections.

Random Self-Reduction of GDP The idea of random self-reduction is used to solve GDP. It says that if we can solve GDP with non-negligible probability then we can solve GDP with a probability exponentially close to 1 for a "smaller" error vector. The proposition below formalizes this result. Therefore the main reduction will focus on reducing DGS to GDP in the average case.

Proposition 58 ([PRS17])**.** *If we are given access to an oracle that solves* $\mathsf{GDP}_{L,r}$ *with some non-negligible probability over the choice of the coset, then we can devise an algorithm to efficiently solve* $\mathsf{GDP}_{L,\delta r}$ *with all but negligible probability, where* $\delta = \delta(n)$ *is any* $o(1)$ *function.*

Now, we define and discuss two problems that have a critical role in the reduction under consideration. The first one is the "Oracle Comparison Problem" or OCP and the second one is the "Oracle Hidden Center Problem" or the OHCP.

Oracle Comparison Problem (OCP): Let $\mathcal{O}$ be an oracle with domain S and range $\{0, 1\}$. So,

$$\mathcal{O} : S \to \{0, 1\}$$

and let

$$p(t) \;:=\; \Pr[\mathcal{O}(t) = 1]$$

for any $t \in S$. So, $p(t)$ measures the probability that the oracle $\mathcal{O}$ outputs 1 for a given input $t \in S$. If $S = \mathbb{R}$, then for any $s \in \mathbb{R}$, let $\mathcal{O}_s : \mathbb{R}^{\geq 0} \to \{0, 1\}$ denote the "suffix" oracle defined as

$$\mathcal{O}_s(t) \;:=\; \mathcal{O}(s + t).$$

The "suffix" oracle $\mathcal{O}_s$ works for input from non-negative real values. Here s is denoted a shift. The OCP is defined as follows. The (ϵ, r)-OCP is a decision problem and more specifically promise problem. The two input parameters of OCP are $\epsilon \geq 0$ which is considered an error parameter and the other one is $r > 0$. Here we work with two "suffix" oracles $\mathcal{O}_{s_1}$ and $\mathcal{O}_{s_2}$ where the shifts of the corresponding "suffix" oracles are $s_1, s_2 \in [-r, r]$. The underlying oracle is $\mathcal{O} : \mathbb{R} \to \{0, 1\}$. The (ϵ, r)-OCP outputs **YES** if $s_2 \leq s_1 - \epsilon$ and **NO** if $s_2 > s_1$. This definition is taken from [PRS17, Definition 4.1].

Oracle Hidden Center Problem (OHCP): Like OCP, OHCP deals with oracles that are randomized. Firstly $(\epsilon, \delta, \beta)$-OHCP is an approximate search problem that takes three parameters as input. The parameters ϵ and $\delta \in [0, 1)$, the third paramter $\beta \geq 1$. OHCP has access to the randomized oracles $\mathcal{O} : \mathbb{R}^k \times \mathbb{R}^{\geq 0} \to \{0, 1\}$. This randomized oracle outputs 1 with probability $p(t + \ln ||\mathbf{z} - \mathbf{z}^*||)$ when given $(\mathbf{z}, t)$ as input. Here $\mathbf{z}^* \in \mathbb{R}^k$ is the unknown hidden center with constraint over the norm of the vector as $\delta d \leq ||\mathbf{z}^*|| \leq d$, $||\mathbf{z} - \mathbf{z}^*|| \leq \beta d$, d is a scale parameter with values from positive real numbers, $d > 0$. The function p is also unknown here. The goal of the $(\epsilon, \delta, \beta)$-OHCP is to output some $\mathbf{z} \in \mathbb{R}^k$ such that $||\mathbf{z} - \mathbf{z}^*|| \leq \epsilon d$. This definition directly follows from [PRS17, Definition 4.3].

In the subsequent sections, we will elaborate on how these two problems are connected and how these are used to solve the GDP problem.

6.1.1 Outline of the Analysis

The analysis of the reductions in [PRS17] can be divided into two parts. The first part is a reduction from approximate ideal-SIVP to the DGS problem. The reduction from module-SIVP to the DGS problem over module lattices has been analyzed in Chapter 5. This reduction is a generalized version of the reduction from approximate ideal-SIVP to the DGS problem, as we have already mentioned that ideal lattices are module lattices with modules of rank 1. The second part is the reduction between DGS to average-case decision ring LWE problem. This is where the analysis differs from the analysis presented in Chapter 5 The concrete analysis of each part is described in the following sections. These two parts are combined and the end-to-end reduction from approximate SIVP to decision LWE is summarized in the concluding section of this chapter.

6.1.2 Reduction Overview

The main contribution of [PRS17] is a reduction from the DGS problem to average-case decision-RLWE for any ring and any modulus for RLWE. The reduction follows an iterative approach like [Reg09, LPR13]. The concrete security aspects of which are analyzed in Chapter 4 and Chapter 5 respectively. Here each iteration can be broadly divided into two steps. The first is a classical step and the second is a quantum step. The classical part uses a solver for GDP with the help of discrete Gaussian samples of larger width and an average-case decisional RLWE oracle. The GDP solver is then used in the quantum part to generate discrete Gaussian samples of a narrower width.

In concrete terms, the iterative step starts with a large value of $r > 2^{2n}\lambda_n(\mathcal{I})$ such that samples from $\mathrm{D}_{\mathcal{I},r}$ can be generated classically without the help of the average-case decision RLWE oracle. Here $\mathcal{I}$ is the ideal lattice generated by the fractional ideal $\mathcal{I}$ of the n-dimensional number field K by using canonical embeddings (Refer Section 2.4). Now, with help of the average-case decision RLWE oracle and polynomial many samples from $\mathrm{D}_{\mathcal{I},r}$, a solver for $\mathsf{GDP}_{\mathcal{I}^\vee,g}$ is devised. The next step of the iteration is quantum where $\mathsf{GDP}_{\mathcal{I}^\vee,g}$ is used to output polynomial many samples from $\mathrm{D}_{\mathcal{I},r'}$, where $r' \leq r/2$.

The iterations continue until r reaches its desired value and output discrete Gaussian samples of the desired width. Though this seems very similar to [Reg09, LPR13], the difference lies in the use of the "Oracle Hidden Center Problem" or the OHCP to solve GDP. The OHCP problem is solved using an oracle that solves the "Oracle Comparison Problem" (OCP), i.e. an OCP solver. The oracle for the OCP is simulated using the average-case decision RLWE oracle. To analyze the reduction and calculate the tightness gap, we need to look at how the above-mentioned problems are interlinked with each other to solve the underlying GDP problem.

6.2 Reducing K-SIVP$_\gamma$ to search ring-LWE$_{q,\leq\alpha}$

Fix three parameters, a positive integer n which denotes the degree of the underlying number field K; an integer $q \geq 2$ which is used to define the ring-LWE problem; and a positive real number α such that $\alpha q \geq 2\cdot\omega(1)$. In the asymptotic setting, q and α are considered to be functions of n.

The K-SIVP$_\gamma$ to ring-LWE$_{q,\leq\alpha}$ reduction is obtained from the following sequence of algorithms, in which $\mathcal{A}_i$ calls $\mathcal{A}_{i+1}$ for $0 \leq i \leq 3$. We briefly describe the algorithms.

Algorithm $\mathcal{A}_0$: Solves K-SIVP$_\gamma$(see Definition 43). The input is fractional ideal $\mathcal{I}$ and the output is a set of n linearly independent elements of $\mathcal{I}$ the longest of which is at most $\gamma \cdot \lambda_n(\mathcal{I})$.

Algorithm $\mathcal{A}_1$: Solves K-DGS$_\Gamma$(see Definition 47). The input is a pair $(\mathcal{I}, r)$, where $\mathcal{I}$ is a fractional ideal of K and $r \geq \Gamma(\mathcal{I})$. The output is a sample from the distribution $D_{\mathcal{I},r}$.

Algorithm $\mathcal{A}_2$: This is a quantum algorithm which, given as input a fractional ideal $\mathcal{I}$ and a set of samples chosen independently from $D_{\mathcal{I},r}$, returns a sample from $D_{\mathcal{I},r'}$, where $r' \leq r/2$.

Algorithm $\mathcal{A}_3$: Solves GDP$_{\mathcal{I}^\vee,\zeta}$. The input is a coset $(\mathcal{I}^\vee + e)$, where $\mathcal{I}$ is fractional ideal in K, $e \in K$ and $e = \sigma^{-1}(\mathbf{e})$ Each element of $\mathbf{e}$ is chosen according to the distribution D_ξ from H. Additionally, $\mathcal{A}_3$ has access to a set of samples chosen independently from $D_{\mathcal{I},r}$. The output is an $e' \in K$ such that $e' = e$ except with negligible probability.

Algorithm $\mathcal{A}_4$: $\mathcal{A}_4$ is a distinguisher ring-LWE$_{q,\leq\alpha}$. It can distinguish between independent samples over $R_q \times \mathbb{T}$, either from the ring-LWE distribution $A^{(R)}_{s,r}$ (Definition 28) or from uniform samples over $R_q \times \mathbb{T}$.

6.2.1 Reduction from K-SIVP$_\gamma$ to K-DGS$_\Gamma$

This analysis is similar to the analysis we have done in Section 5.2.1. We do not repeat it here, we only take the result of this part of the analysis in the following proposition.

Proposition 59. *$\mathcal{A}_0$ invokes $\mathcal{A}_1$ a total of n^3 times.*

6.3 Reducing K-DGS$_\Gamma$ to ring-LWE$_{q,\leq\alpha}$

In this section, we discuss two problems that will be integral to the reduction from DGS to the decision ring LWE problem. First, we start with the results related to OCP and OHCP and see how are these two problems connected.

6.3.1 The Oracle Comparison Problem(OCP)

We begin with a result that is related to the OCP. We have previously defined OCP. The following result states that we can have a polynomial time algorithm that solves OCP with certain conditions 60.

Lemma 60 ([PRS17, Lemma 4.2])**.** *There exsits a* $\mathsf{poly}(\tau)$*-time algorithm that takes as input the confidence parameter* $\tau \geq 200$ *and solves* $(1/\tau, \tau)$-OCP *except with probability at most* $\exp(-\tau)$*, provided that the oracle* $\mathcal{O}$ *and the two shifts* s_1, s_2 *corresponding to the instance of* OCP *satisfy the following conditions. There exists a* $p_\infty \in [0,1]$ *and* $t^* \geq s_1$ *such that*

1. $p(t^*) - p_\infty \geq 1/\tau$*;*
2. $|p(t) - p_\infty| \leq 2\exp(-t/\tau)$ *for all* t*; and*
3. $p(t)$ *is* τ*-Lipschitz, i.e.* $|p(t_1) - p(t_2)| \leq \tau|t_1 - t_2|$ *for all* $t_1, t_2 \in \mathbb{R}$*.*

This lemma ascertains the existence of an algorithm for $(1/\tau, \tau)$-OCP which runs in polynomial time in the parameter of τ under the stated conditions. The following algorithm solves the $(1/\tau, \tau)$-OCP using the oracles $\mathcal{O}_{s_1}(\cdot)$ and $\mathcal{O}_{s_2}(\cdot)$ where s_1 and s_2 are unknown, as per Lemma 60. The job is to determine the relation between s_1 and s_2.

The Algorithm 16 implements the $\mathsf{solveOCP}$ function which solves $(1/\tau, \tau)$-OCP. The $\mathsf{solveOCP}$ function takes two input parameters, $1/\tau$ and λ. Here τ is the OCP parameter and λ is used to determine the threshold value for the distinguisher that we are interested to make through this algorithm. The algorithm has two loops, one outer and one inner which is the nested loop. The outer loop runs for $1 + T_1$ times and the nested inner loop runs N times for each outer loop. The nested inner loop is used to count the number of '1' as output from $\mathcal{O}_{s_1}(\cdot)$ and $\mathcal{O}_{s_2}(\cdot)$ when given '$i\Delta$' as input at the i-th iteration of the outer loop. At the end of the inner loop the variable $\bar{p}_i^{(k)}$ captures the fraction of times $\mathcal{O}_{s_k}(i\Delta)$ outputs '1', where $k \in \{1, 2\}$ and $i \in \{0, 1, \ldots, T_1\}$. Here $\bar{p}_i^{(k)}$ is the estimated probability that the randomized oracle $\mathcal{O}_{s_k}(\cdot)$ outputs 1 for input $i\Delta$. At the end of the outer loop, the $\bar{p}_i^{(k)}$ variables are obtained for all the i's. Next $\bar{h}_k$'s are obtained as per the maximum value $(1+i\Delta)|\bar{p}_i^{(k)} - \bar{p}_{T_1}^{(k)}|$ for all possible i's. If the difference between $\bar{h}_1$ and $\bar{h}_2$ is bigger than some threshold value the function returns 1 else it returns 0. Here the function $\max_i(1 + i\Delta)|\bar{p}_i^{(1)} - \bar{p}_{T_1}^{(1)}|$ is a monotonically non-increasing function of i. From the definition of OCP 6.1, we can say that in case of **NO** instance where $s_2 > s_1$, we must have $\bar{h}_1 - \bar{h}_2 < 0$ due to the non-increasing

Algorithm 16 Algorithm to solve $(1/\tau, \tau)$-OCP

function solveOCP$(1/\tau, \lambda)$ // where $\tau \geq 200$
  for $i \leftarrow 0$ to $T_1 := \lfloor t_{\max}/\Delta \rfloor$ **do** // Take $T_1 = 1000\tau^{10}$ and $\Delta = \frac{1}{200\tau^8}$
    $\mathsf{cnt}_1 := 0$
    $\mathsf{cnt}_2 := 0$
    for $j \leftarrow 1$ to N **do** // Take $N = \frac{200 \ln T_1}{\Delta^2} = 8 \times 10^6 \tau^{16}(10 \ln \tau + 3 \ln 10)$
      $\mathsf{cnt}_1 := \mathsf{cnt}_1 + \mathcal{O}_{s_1}(i\Delta)$
      $\mathsf{cnt}_2 := \mathsf{cnt}_2 + \mathcal{O}_{s_2}(i\Delta)$
    end for
    $\bar{p}_i^{(1)} := \frac{\mathsf{cnt}_1}{N}$
    $\bar{p}_i^{(2)} := \frac{\mathsf{cnt}_2}{N}$
  end for
  $\bar{h}_1 := \max_i (1 + i\Delta)|\bar{p}_i^{(1)} - \bar{p}_{T_1}^{(1)}|$ // Requires another T_1 comparisons
  $\bar{h}_2 := \max_i (1 + i\Delta)|\bar{p}_i^{(2)} - \bar{p}_{T_1}^{(2)}|$ // Requires another T_1 comparisons
  if $(|\bar{h}_2 - \bar{h}_1| > \frac{1}{\tau^2} \cdot \frac{1}{1+100\tau \ln \tau} - \frac{2}{\lambda})$ **then**
    return 1;
  else
    return 0;
  end if
end function

nature of the function and for the **YES** instance $\bar{h}_1 - \bar{h}_2$ is non-negligible and more specifically $\bar{h}_1 - \bar{h}_2 > \frac{1}{\tau^2} \cdot \frac{1}{1+100\tau \ln \tau} - \frac{2}{\lambda}$. This threshold value is taken such that the distinguisher (solveOCP) solves the OCP problem with a probability exponentially close to 1. This has been captured in this algorithm. The conditions of the Lemma 60 justify this assertion.

The running time of this algorithm is one of the main concerns here. The solveOCP function will be used to solve the OHCP in the following section under certain conditions 61. Again the OHCP will be reduced to the decision ring LWE problem while solving the GDP problem. Through this chain of reductions, the oracles that are used in solveOCP functions are modeled by the oracles for the ring LWE problem. So it is necessary to count the time required to execute the solveOCP function for tightness gap analysis.

Running time: The running time of Algorithm 16 is proportional to

$$2 \cdot (T_1 + 1)N \ > \ 2 \cdot T_1 N$$

If we take the following parameters

$$T_1 \;=\; 1000\tau^{10}, \quad \Delta \;=\; \frac{1}{200\tau^8} \quad \text{and} \quad N = \frac{200\ln T_1}{\Delta^2} \;=\; 8\times 10^6\tau^{16}(10\ln\tau + 3\ln 10),$$

which were suggested in the STOC 2017 [PRS17] version of [LPR13], then the running time is greater than

$$8\times 10^9\tau^{26}(10\ln\tau + 3\ln 10) \;\geq\; 59.89\times 2^{29}\times 10^{61},$$

since $\tau \geq 200$, In general, the running time is proportional to

$$NT_1 \;\geq\; 500^2 t_{\max}^2\tau\lambda T_1\ln T_1.$$

Now,

$$T_1 \;=\; \frac{t_{\max}}{\Delta} \;=\; 100\tau\lambda t_{\max}^2 \;=\; 10^6\tau^3\lambda(\ln\tau\lambda)^2.$$

Therefore, we have

$$\begin{aligned} NT_1 \;&\geq\; 500^2(100\tau\ln\tau\lambda)^2\cdot\tau\lambda\cdot\left(10^6\tau^3\lambda(\ln\tau\lambda)^2\right)\ln(10^6\tau^3\lambda(\ln\tau\lambda)^2) \\ &=\; 25\times 10^{14}\cdot\tau^6\lambda^2\cdot(\ln\tau\lambda)^4\cdot(6\ln 10 + 3\tau\ln\lambda + 2\ln(\ln\tau\lambda))\,. \qquad (6.1) \\ &\approx\; 25\times 10^{14}\cdot\tau^7\lambda^2. \qquad (6.2) \end{aligned}$$

6.3.2 The Oracle Hidden Center Problem(OHCP)

We have already defined the OHCP in Section 6.1. Here we state an important result that indicates the existence of a polynomial time algorithm for the OHCP under certain conditions.

Proposition 61. *There exists is a* $\mathsf{poly}(\tau, k)$*-time algorithm that takes as input a confidence parameter* $\tau \geq 20\ln(k+1)$ *(and the scale parameter* $d > 0$*) and solves* $(\exp(-\tau), \exp(-\tau), 1+1/\tau)$-$\mathsf{OHCP}$ *in dimension* k *except with probability* $\exp(-\tau)$*, provided that the oracle* $\mathcal{O}$ *corresponding to the* OHCP *instance satisfies the following conditions. For some* $p_\infty \in [0,1]$ *and* $s^* \geq 0$*,*

1. $p(s^*) - p_\infty \geq 1/\tau$*;*
2. $|p(s) - p_\infty| \leq 2\exp(-s/\tau)$ *for any* s*; and*
3. $p(s)$ *is* τ*-Lipschitz in* s*,*

where $p(s)$ is the acceptance probability of $\mathcal{O}$ on input $(\mathbf{0}, s)$. Furthermore, each of the algorithm's oracle calls takes the form $\mathcal{O}(\cdot, i\Delta)$ for some $\Delta < 1$ that depends only on τ and k and $0 \leq i \leq \mathsf{poly}(\tau, k)$.

This result is taken from [PRS17, Proposition 4.4]. We describe the Algorithm 17 which solves the $(\exp(-\tau), \exp(-\tau), 1 + 1/\tau)$-OHCP problem.

Algorithm 17 Algorithm to solve $(\exp(-\tau), \exp(-\tau), 1+1/\tau)$-OHCP using an oracle $\mathcal{O}^*(\cdot)$ that solves the OHCP problem for inputs $(\mathbf{0}, s)$

function solveOHCP(τ, k) // where $\tau \geq 20\ln(k+1)$
  $\mathbf{z}_0 := \mathbf{0}$;
  Set $\lambda := 10\tau k^2 \ln T$;
  for $i \leftarrow 0$ to T **do** // where $T + 1 = 2000\tau^3 k^4$
    $j \xleftarrow{\$} \{1, 2, \ldots, k\}$
    $x \xleftarrow{\$} [0, 1]$
    $\sigma \xleftarrow{\$} \{-1, +1\}$
    Set $\mathbf{v}_i := \frac{\sigma \exp(-2\tau x)\mathbf{e}_j}{\sqrt{\tau^2 k}}$ // where $\mathbf{e}_j$ denotes the k-dimensional standard basis vector
    Set $\mathbf{x}_1 := \mathbf{z}_i$;
    Set $\mathbf{x}_2 := \mathbf{z}_i + \mathbf{v}_i$;
    if (solveOCP$(1/\lambda, \tau + \lambda) =$ YES) **then**
    else
      Set $\mathbf{z}_{i+1} := \mathbf{x}_1$;
    end if
  end for
  Return $\mathbf{z}_{T+1}$;
end function

The Algorithm 17 implements the solveOHCP function which solves the $(\exp(-\tau), \exp(-\tau), 1+1/\tau)$-OHCP using the solveOCP function. The solveOHCP function takes two input parameters, τ and k. Here τ is the OHCP parameter. This algorithm aims to find the hidden center $\mathbf{z}^*$ (OHCP Definition 6.1) of the oracle incrementally. The algorithm iterates $T + 1$ times. Each time, it tries to predict whether a vector $\mathbf{z}_i$ is nearer to the center of the hidden oracle $\mathbf{z}^*$ by using the solveOCP function. The solveOCP function is implemented as a decision function in Algorithm 16 and has given access to two oracles, having certain properties. Here in Algorithm 17, the solveOCP function is provided with two randomized oracles with appropriate parameters such that all the constraints of the Lemma 60 are satisfied. Hence the solveOCP function is used in finding the approximation of the hidden oracle center $\mathbf{z}^*$.

Next, we calculate the running time of this algorithm. The solveOHCP function will be used to solve the GDP in the following section.

Running time: The running time of Algorithm 17 is therefore proportional to $(T+1)\times$ time taken for one $(1/\tau, \tau+\lambda)$-OCP oracle call. Recall that

$$\begin{aligned} \lambda &= 10\tau k^2 \ln T \approx 10\tau k^2(3\ln 10 + \ln 2 + 3\ln\tau + 4\ln k) \\ \Rightarrow \quad \tau+\lambda &\approx \tau(1+10k^2(3\ln 10 + \ln 2 + 3\ln\tau + 4\ln k)) \\ &= \tau\vartheta \text{ (say)}. \end{aligned}$$

Then, by (6.1), the time taken for one $(1/\tau, \tau+\lambda)$-OCP oracle call is proportional to

$$\begin{aligned} & 25\times 10^{14}\cdot\tau^6(\tau+\lambda)^2\cdot(\ln\tau(\tau+\lambda))^4\cdot(6\ln 10 + 3\tau\ln(\tau+\lambda) + 2\ln(\ln\tau(\tau+\lambda))) \\ = \; & 25\times 10^{14}\cdot\tau^8\vartheta^2\cdot(\ln\tau^2\vartheta)^4\cdot\left(6\ln 10 + 3\tau(\ln\tau+\ln\vartheta) + 2\ln(\ln\tau^2\vartheta)\right). \end{aligned}$$

Then the running time of Algorithm 17 is proportional to

$$2\times 5^2\times 10^{17}\cdot k^4\cdot\tau^{11}\vartheta^2\cdot(\ln\tau^2\vartheta)^4\cdot\left(6\ln 10 + 3\tau(\ln\tau+\ln\vartheta) + 2\ln(\ln\tau^2\vartheta)\right).$$

Now we describe the main reductions of [PRS17] in further detail. As per our definition in Section 6.2, $\mathcal{A}_1$ is the algorithm to solve K-DGS, that prepares an initial list of N_0, DGS samples and goes through $3n$ iterations, where in each iteration $\mathcal{A}_1$ invokes a quantum circuit $\mathcal{A}_2$, N_0 times, and in each invocation, $\mathcal{A}_1$ provides $\mathcal{A}_2$ with a list of DGS samples and in return DGS samples with reduced by a factor of at least 2. Here, $\mathcal{A}_2$ is a quantum algorithm. Finally $\mathcal{A}_1$ returns a sample from the last list that it prepares. The number N_0 of DGS samples is equal to the number of RLWE samples required by the average-case decisional RLWE. $\mathcal{A}_1$ calls $\mathcal{A}_2$ a total of $3n\cdot N_0$ times. $\mathcal{A}_2$ applies the reverse of an algorithm $\mathcal{A}_3$ that solves the GDP problem. The construction of $\mathcal{A}_3$ is based on the distinguisher $\mathcal{A}_4$ which distinguishes ring-LWE distribution from the uniform distribution.

We now state the main theorem of [PRS17] and present an algorithmic representation of the reduction. This is a reduction from the DGS over ideal lattices to the decisional ring LWE for any ring and modulus.

Theorem 62 ([PRS17, Theorem 6.2])**.** *Let K be an arbitrary number field of degree n and R be the ring of an algebraic integer of K. Let $\alpha=\alpha(n)\in(0,1)$, and let $q=q(n)\geq 2$ be an integer such that $\alpha q\geq 2\cdot\omega(1)$. There is a polynomial-time quantum reduction from* K-DGS$_\gamma$

to average-case, decision $\mathsf{R\text{-}LWE}_{q,\Upsilon_\alpha}$ *for any*

$$\gamma \;=\; \max\left\{\eta(\mathcal{I})\cdot\frac{\sqrt{2}}{\alpha}\cdot\omega(1),\frac{\sqrt{2n}}{\lambda_1(\mathcal{I}^\vee)}\right\}. \tag{6.3}$$

As before, we assume that $\eta(\mathcal{I}) \geq \frac{\omega(\sqrt{\log n})}{\lambda_1(\mathcal{I}^\vee)}$ by Claim 19 and $\alpha < \sqrt{\log n/n}$ which is true for practical lattice-based crypto-systems, thus we get $\gamma \;=\; \eta(\mathcal{I})\cdot\frac{\sqrt{2}}{\alpha}\cdot\omega(1)$. This reduction is the main result of [PRS17] and is applicable for any number field and any modulus. Specifically, the reduction binds K-DGS to average-case decision RLWE, directly bypassing the requirement to first reduce to search RLWE and then to average-case decision RLWE like [Reg09, LPR13]. Algorithm 18 is the algorithmic representation of the reduction.

Algorithm 18 Algorithm to solve K-DGS_γ using an R-$\mathrm{LWE}_{q,\Upsilon_\alpha}$

function $\mathcal{A}_1(\mathcal{I}, r)$ // where $r \geq \lambda(\mathcal{I})$
  $\mathcal{S}_{3n} \leftarrow \{\mathsf{bootstrap}(\mathcal{I}, r_{3n})\}$;
  for $i \leftarrow 3n$ down to 1 **do**
    $\mathcal{S}_{i-1} \leftarrow \{\}$;
    for $j \leftarrow 1$ to N **do**
      $x \leftarrow \mathcal{A}_2(\mathcal{I}, \mathcal{S}_i, d_i')$; ($\mathcal{A}_2$ invokes $\mathcal{A}_3(\mathcal{I}^\vee, \mathcal{S}_i, y)$ for a suitable y); // $\mathcal{A}_2$ is the quantum function
      $\mathcal{S}_{i-1} \leftarrow \mathcal{S}_{i-1} \cup \{x\}$
    end for
    $r_{i-1} \leftarrow r_i \cdot (\sqrt{\log n})/(\alpha q)$;
  end for
  Return one element from $\mathcal{S}_0$;
end function

We now consider the reduction of $\mathsf{K\text{-}DGS}_\gamma$ to $\mathsf{R\text{-}LWE}_{q,\Upsilon_\alpha}$ in further details following the Algorithm 18. The input to an algorithm to solve $\mathsf{K\text{-}DGS}_\gamma$ is a pair $(\mathcal{I}, r)$, where $\mathcal{I}$ is an ideal lattice generated by the fraction ideal $\mathcal{I} \in K$ and $r \geq \gamma(\mathcal{I})$. The existence of an algorithm to solve $\mathsf{K\text{-}DGS}_\gamma$, is stated by Theorem 62. Algorithm 18 is an iterative algorithm. Each iteration of Algorithm 18, is represented by the quantum algorithm referred to by the Lemma 63. The algorithm referred to by the Lemma 63 is divided into two algorithms. One is referred to by the Lemma 64 and the other by the Lemma 65.

Now, we describe Algorithm 18 in detail. Let r be a real number satisfying $r \geq \gamma(\mathcal{I})$. For $i = 0, \ldots, 3n$, define

$$r_i \;=\; r\cdot(\alpha q/\omega(1))^i.$$

Theorem 62 follows from the iterative step, which is sketched below. Start with a very large value of $r \geq 2^{2n}\lambda_n(\mathcal{I})$, so that samples from $\mathsf{D}_{\mathcal{I},\mathbf{r}}$ for each $\mathbf{r} \in W_{r,\zeta,T}$ can be generated classically using the bootstrap function (see Section 4.3). This step is the same for all the lattice-based reductions which followed the path first indicated by Regev [Reg09]. This reduction from $\mathsf{K\text{-}DGS}_\gamma$ to $\mathsf{R\text{-}LWE}_{q,\Upsilon_\alpha}$ is iterative and the following lemma represents one iteration.

Lemma 63 ([PRS17, Lemma 6.5])**.** *Let α be a real number and q be a positive integer such that $\alpha \in (0,1)$ and $q \geq 2$. Suppose $\mathcal{I}$ be an ideal lattice generated by the fractional ideal $\mathcal{I}$ of a number filed K and let $r \geq 2q \cdot \eta(\mathcal{I})$ such that*

$$r' \;=\; \frac{r \cdot \omega(1)}{\alpha q} \;>\; \frac{2n}{\lambda_1(\mathcal{I}^\vee)}.$$

There exists a polynomial time quantum algorithm which takes polynomial many samples from discrete Gaussian distribution $\mathsf{D}_{\mathcal{I},\mathbf{r}}$ for each $\mathbf{r} \in \mathsf{W}_{r,\upsilon,T}$ as input and has access to an oracle that solves $\mathsf{R\text{-}LWE}_{q,\Upsilon_\alpha}$, outputs an independent sample from $\mathsf{D}_{\mathcal{I},\mathbf{r}'}$, where $\upsilon = 1/\mathsf{poly}(n)$, $T = \mathsf{poly}(n)$, and a vector $\mathbf{r}' \in G$ where $\mathbf{r}' \geq r'$.

The lemma says that if we are given polynomial many samples from discrete Gaussian distribution of certain width and an oracle that solves the RLWE problem in polynomial time, we can make a quantum circuit to produce polynomial many samples from discrete Gaussian distribution of width at most half of the width of the discrete Gaussian distribution of the given samples in polynomial time. As said before, one iteration of Algorithm 18 is represented by Lemma 63. This lemma is similar to the reductions we have already discussed in previous sections. This algorithm can be divided into two sub-algorithms, viz, one classical and another quantum. The quantum algorithm is dependent on the classical one. As mentioned earlier, the classical algorithm is represented by the Lemma 64, and the quantum algorithm is represented by Lemma 65. Hence Lemma 63 is the combination of Lemma 64 and 65. Now we state the two lemmas from [PRS17] and describe them in the following section. So, we can focus on the individual lemmas independently.

Lemma 64 ([PRS17, Lemma 6.6])**.** *There exists a probabilistic polynomial-time classical algorithm that given an oracle that solves average-case decision $\mathsf{R\text{-}LWE}_{q,\Upsilon_\alpha}$ and input a number $\alpha \in (0,1)$ and an integer $q \geq 2$ together with its factorization, a fractional ideal $\mathcal{I}$ in K, a parameter $r \geq \sqrt{2}q \cdot \eta_\epsilon(\mathcal{I})$, and polynomial many samples from the discrete Gaussian distribution $\mathsf{D}_{\mathcal{I},\mathbf{r}}$ for each $\mathbf{r} \in W_{r,\zeta,T}$ (for some $\zeta = 1/\mathsf{poly}(n)$ and $T = \mathsf{poly}(n)$), solves $\mathsf{GDP}_{\mathcal{I}^\vee,g}$ for any $g = o(1) \cdot \alpha q/(\sqrt{2}r)$.*

As said earlier, this result is a classical result. This is a polynomial time reduction from GDP to RLWE. This result is the main contribution of [PRS17] as this is where it differs from [LPR13, LS15]. According to this lemma, if we have an average case decision RLWE oracle and polynomial many samples from discrete Gaussian distribution, we can solve the GDP problem over the dual lattice of an ideal lattice $\mathcal{I} \in K$. The reduction uses discrete Gaussian samples over the dual ideal. Lemma 64 solves $\mathsf{GDP}_{\mathcal{I}^\vee,g}$ for $g = o(1) \cdot \alpha q/\sqrt{2}r$ by using an oracle that solves $\mathsf{GDP}_{\mathcal{I}^\vee,g'}$, using self reduction, where $g' = \alpha q/\sqrt{2}r$. In particular, the GDP is solved using an algorithm for OHCP. However, what we have with us is an RLWE oracle. So, we need to devise the algorithm for OHCP according to the Proposition 61 using the RLWE oracle. We discuss the proof of the Lemma 64 in detail to find out the tightness gap of this reduction. We have two reductions in hand. One is from GDP to OHCP and another is from OHCP to RLWE. We present these reductions in an algorithmic form in Algorithm 19 and Algorithm 20 respectively.

Lemma 65 ([PRS17, Lemma 6.7])**.** *There is an efficient quantum algorithm that, given any n-dimensional lattice L, a real $g < \lambda_1(L^*)/(2\sqrt{2n})$, a vector $\mathbf{r} \geq 1$, and an oracle that solves $\mathsf{GDP}_{L^*,g}$ (with all but negligible probability), outputs an independent sample from $\mathsf{D}_{L,\mathbf{r}/(2g)}$.*

This lemma is the only quantum part of the whole reduction. This result produces DGS samples of at most half of the given width, while it also takes a GDP solver over the dual lattice as input. This lemma ensures that at the end of every iteration of the main iterative procedure, we get DGS samples of a width of half of the width we start the iteration with. Each call to the quantum procedure produces one DGS sample, thus we need polynomial many calls to the quantum procedure to get polynomial many DGS samples.

6.3.3 GDP to OHCP Reduction

In this section, we describe the reduction from GDP to OHCP for a particular set of parameters of these problems. The OHCP oracle is used to approximate the error vector $\mathbf{e}$, where a coset $\mathbf{e} + \mathcal{I}^\vee \in H$ is given as input for GDP.

We are given a coset $\mathbf{e} + \mathcal{I}^\vee \in H$ as input for GDP. To solve it, we need to find the error vector $\mathbf{e} \in H$, where $\mathbf{e}$ is an n-dimensional vector. The OHCP oracle is used to approximate each coordinate individually. Algorithm 19 runs with the target vector e_i, the i^{th} coordinate of $\mathbf{e}$ and approximate each e_i in order to approximate $\mathbf{e}$. In other words, the approximation is done coordinate-wise.

Algorithm 19 Algorithm to solve $\mathrm{GDP}_{\mathcal{I}^\vee,g}$ using $(\exp(-\tau), \exp(-\tau), 1+1/\tau)$-OHCP oracle

function $\mathcal{A}_3(\mathcal{I}^\vee, \mathcal{S}, \mathbf{z})$ // where $\mathbf{z} = (z_1, \ldots, z_n)$
  $\mathbf{w} \leftarrow \mathbf{0}$;
  if $(\alpha > \exp(-n))$ **then**
    for $i = 1$ to $s_1 + s_2$ **do**
      if $(i \leq s_1)$ **then**
        $x_i \leftarrow \mathsf{solveOHCP}(\tau, 1)$; // solveOHCP is given access to oracle $\mathcal{O}(\mathcal{S}, i, z_i, t)$, where $\exp(t) = (1+\zeta)^j$
        $\mathbf{w} \leftarrow \mathbf{w} + (x_i \cdot \mathbf{e}_i)$;
      else
        $x_i \leftarrow \mathsf{solveOHCP}(\tau, 2)$; // solveOHCP is given access to oracle $\mathcal{O}(\mathcal{S}, i, z_i, t)$, where $\exp(t) = (1+\zeta)^j$
        $\mathbf{w} \leftarrow \mathbf{w} + (x_i \cdot \mathbf{e}_i + \bar{x_i} \cdot \mathbf{e}_{i+s_2})$;
      end if
    end for
  end if
  $\hat{\mathbf{e}} \leftarrow \mathsf{BabaiNearestPlane}(\mathcal{I}^\vee, \mathbf{z} - \mathbf{w})$;
  Return $\hat{\mathbf{e}} + \mathbf{w}$;
end function

Let $\mathcal{O}_i : \mathbb{R} \times \mathbb{R}^{\geq 0} \to \{0,1\}$ for $1 \leq i \leq s_1$ and $\mathcal{O}_i : \mathbb{C} \times \mathbb{R}^{\geq 0} \to \{0,1\}$ for $s_1 < i \leq s_1 + s_2$ be the oracles of the OHCP (Section 6.1) with hidden center e_i as the i-th coordinate of the vector $\mathbf{e}$. We know that the oracles $\mathcal{O}_i$ in OHCP are randomized. The probability that oracles outputs 1 or $\mathsf{Pr}[\mathcal{O}_i(z,t) = 1]$ depends only on $\exp(t)|z - e_i|$ (for $z \in \mathbb{C}$ with $|z - e_i|$ sufficiently small). For Oracles $\mathcal{O}_i$, $k = 1$ for $1 \leq i \leq s_1$ as the first s_1 coordinates of $\mathbf{e}$ are approximated using s_1, solveOHCP function calls and $k = 2$ for $s_1 < i \leq s_1 + s_2$ as the next $2s_2$ cordinates are approximated using with s_2, solveOHCP function calls . We implicitly identify $\mathbf{x}_i \in \mathbb{R}^2$ with $x_i \in \mathbb{C}$ in the natural way.

Let us define for $1 \leq i \leq s_1$, $k_i : \mathbb{R} \to H$ as $k_i(z) = z \cdot \mathbf{e}_i$, and for $s_1 < i \leq s_1 + s_2$ define $k_i : \mathbb{C} \to H$ as $k_i(z) = z \cdot \mathbf{e}_i + \bar{z} \cdot \mathbf{e}_{i+s_2}$, where $\mathbf{e}_i \in \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ has 1 in the i^{th} coordinate and 0 elsewhere.

Algorithm 19 starts with the assumption that $\alpha > \exp(-n)$. If $\alpha \leq \exp(-n)$, then with high probability $\mathbf{e}$ will satisfy that $||\mathbf{e}|| \leq 2^{-n}\lambda_1(\mathcal{I}^\vee)$. Hence, Babai's nearest plane algorithm 3.2.2 can be used to solve the problem efficiently.

In Algorithm 19, the $\mathbf{z} \in \mathbf{e} + \mathcal{I}^\vee$, i.e, $\mathbf{z}$ is an element from the coset. Algorithm 19 iterates over $i = 1$ to $s_1 + s_2$. For each i, $\mathsf{solveOHCP}(\tau, 1)$ represents a valid $(\exp(-\tau), \exp(-\tau), 1 + 1/\tau)$ OHCP instance with an oracle OHCP $\mathcal{O}_i$ with a confidence parameter τ, and a distance

bound $d' = d/(1+1/\tau)$ and hidden center as the i^{th} coordinate of vector $\mathbf{e}$, i.e. e_i. As x_i is the output by $\mathsf{solveOHCP}(\tau, 1)$ at the i^{th} iteration, $x_i \cdot \mathbf{e}_i$ represents n dimensional vector with i^{th} component being the approximation of the i^{th} coordinate of vector $\mathbf{e}$ for $i \leq s_1$. Similarly for $s_1 + s_2 \geq i > s_1$, $(x_i \cdot \mathbf{e}_i + \bar{x}_i \cdot \mathbf{e}_i)$ represents the approximation for i^{th} and $(i+s_2)^{th}$ coordinates. Hence $\mathbf{w}$ becomes the resulting approximation of the vector $\mathbf{z}$. We can then apply Babai's Nearest Plane algorithm as $||\mathbf{z} - \mathbf{w}|| \leq 2^{-n}\lambda_1(\mathcal{I}^\vee)$. Thus we recover $\mathbf{z}$ using $\mathsf{solveOHCP}$ function calls.

6.3.4 OHCP to RLWE Reduction

Now, we discuss the Algorithm 20 which shows how an OHCP oracle can be simulated using an average-case decision $\mathsf{R\text{-}LWE}$ oracle with error vectors drawn from an elliptical Gaussian distribution (See Definition 54).

Algorithm 20 Simulate $(\exp(-\tau), \exp(-\tau), 1+1/\tau)$-$\mathsf{OHCP}$ oracle using $\text{R-LWE}_{q,\Upsilon_\alpha}$

function $\mathcal{O}(\mathcal{S}, i, z, t, \mathbf{y})$
  $\mathcal{T} \leftarrow \{\}$;
  if $(i \leq s_1)$ **then**
    $\mathbf{c} \leftarrow (z \cdot \mathbf{e}_i)$;
  else if $(s_1 < i \leq s_1 + s_2)$ **then**
    $\mathbf{c} \leftarrow (z \cdot \mathbf{e}_i + \bar{z} \cdot \mathbf{e}_{i+s_2})$;
  end if
  Compute $t \in \mathcal{I}$, such that, $t \cdot \mathcal{I}^{-1}$ and $\langle q \rangle$ are co-prime;
  for each $\mathbf{v} \in \mathcal{S}$ **do**
    $\mathbf{err} \leftarrow \mathsf{D}_{\alpha f(n)/\sqrt{2}}$; // where $f(n) = \omega(\sqrt{\log n})$ with $f(n) \leq n$
    $a \leftarrow \theta_t^{-1}(v \mod q\mathcal{I})$; // where $\theta_t(\cdot)$ an isomorphism from R_q to $\mathcal{I}_q$ and $v \in K$ whose image in H is $\mathbf{v}$
    $\mathbf{b} \leftarrow (\mathbf{v} \cdot (\mathbf{y} - \mathbf{c})/q) + \mathbf{err} \mod \sigma(R^\vee)$;
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{(a, \mathbf{b})\}$;
  end for
  $b \leftarrow \mathsf{R\text{-}LWE}_{q,\Upsilon_\alpha}(\mathcal{T})$;
  if $(b = 1)$ **then**
    Return 1
  else
    Return 0
  end if
end function

Algorithm 20 employs samples from $D_{\mathcal{I},r_{i,j}}$, where $(1+\zeta)j = \exp(t)$, on input (z, t) to

OHCP oracles $\mathcal{O}_i$. On these samples, the coset $\mathbf{e} - k_i(z) + \mathcal{I}^\vee$, the parameter r, and the distance bound $d = f(n)\alpha q/(\sqrt{2}r)$ are then applied the transformation from Lemma 66 with the condition that $f(n) \leq n$. Let the generated samples be $A_{i,z,t}$. For $A_{i,z,t}$, $\mathcal{O}_i$ invokes the R-LWE oracle and outputs 1 if and only if it accepts. The following lemma is used by the Algorithm 20 to convert a GDP instance on an ideal lattice to Ring-LWE samples using discrete Gaussian samples over the dual lattice. This is how the solver for OHCP is simulated through the decisional RLWE oracle.

Lemma 66 ([PRS17, Lemma 6.8])**.** *There is an efficient algorithm that takes as input an integer $q \geq 2$ with known factorization, a fractional ideal $\mathcal{I}^\vee \subset K$, a coset $e + \mathcal{I}^\vee$ and bound $d \geq ||e||_\infty = \max_i |\sigma_i(e)|$, a parameter $r \geq 2q \cdot \eta_\epsilon(\mathcal{I})$, and samples from $\mathsf{D}_{\mathcal{I},\mathbf{r}}$ for some $\mathbf{r} \geq r$. It outputs samples that are within a negligible statistical distance of the Ring-LWE distribution $A_{s,\mathbf{r}'}$ for a uniformly random $s \in R_q^\vee$, where the coordinates of $\mathbf{r}'$ are given by*

$$(r_i')^2 \;=\; (r_i|\sigma_i(e)|/q)^2 + (rd/q)^2.$$

6.3.5 Number of Oracle Calls:

Our main goal is to find the tightness of the above reduction which will help in finding the tightness gap for end-to-end reduction. Let M_2 be the number of times $\mathcal{A}_3$ calls $\mathcal{A}_4$. Let each call to $\mathcal{A}_4$ require M_1 samples. So the total number of samples required in all M_2 calls is equal to M_1M_2. So, the N_0 number of DGS samples must be equal to M_1M_2, as this many samples are needed to provide to $\mathcal{A}_3$ so that it can generate the required number of LWE samples.

- Notice that Algorithm 19 calls solveOHCP calls s_1 times with parameters τ and $k = 1$. It is then followed by another s_2 many calls to solveOHCP but with parameters τ and $k = 2$.

- Algorithm 17 implements solveOHCP. Each solveOHCP function in turn calls solveOCP, $T+1 = 2000\tau^3k^4$ times. T is a function of (τ, k) and k varies for for different solveOHCP calls.

- Algorithm 16 implements solveOCP. Each solveOCP function calls the OCP oracle $2N \cdot T_1$ times. From the proof of Proposition 61, it is clear that the OCP oracles needed to solve $(\exp(-\tau), \exp(-\tau), 1 + 1/\tau)$-OHCP, using Lemma 60, are basically OHCP or-

acles. Hence these OCP oracle calls are implemented through R-LWE oracles that we have.

Hence, the number of times R-LWE is called is $(s_1{\cdot}(1+T){\cdot}2N{\cdot}T_1 + s_2{\cdot}(1+T){\cdot}2N{\cdot}T_1)$.

From Algorithm 17, if $k=1$, $1+T = 2000\tau^3$ and if $k=2$, $1+T = 32000\tau^3$.

From Algorithm 16, we get $\lambda = \mathsf{poly}(\tau)$, $N{\cdot}T_1 =$
$25 \times 10^{14} \cdot \tau^6\lambda^2 \cdot (\ln \tau\lambda)^4 \cdot (6\ln 10 + 3\tau \ln \lambda + 2\ln(\ln \tau\lambda))$

So, $(s_1{\cdot}(1+T){\cdot}2N{\cdot}T_1 + s_2{\cdot}(1+T){\cdot}2N{\cdot}T_1)$
$= 50 \times 10^{17} \cdot \tau^9\lambda^2 \cdot (s_1 + 16s_2) \cdot (\ln \tau\lambda)^4 \cdot (6\ln 10 + 3\tau \ln \lambda + 2\ln(\ln \tau\lambda))$.

So, M_2 is the number of oracle calls needed to solve $\mathsf{GDP}_{\mathcal{I}^\vee,g}$ from the $\mathsf{R\text{-}LWE}_{q,\Upsilon_\alpha}$ oracle. If we ignore the logarithmic terms and replace s_1+16s_2 with n, we get $M_2 = 50\times 10^{17}\cdot\tau^{10}\lambda^2\cdot n$ For simplicity of the calculation, we assume $\tau = \lambda$. And τ and λ are at-least $100n^2M_1$. Hence, the the value of M_2 becomes at-least $10^{43}{\cdot}n^{25} \cdot {M_1}^{12}$.

Proposition 67. *Here are the few observations we make*

1. *following Proposition 58, n^c $\mathsf{GDP}_{\mathcal{I}^\vee,g}$ oracle calls needed to solve $\mathsf{GDP}_{\mathcal{I}^\vee,g'}$, where $g = o(1)g'$ and c is positive integer.*
2. *M_2 becomes at-least $10^{41}{\cdot}n^{23} \cdot {M_1}^{11}$.*
3. *Algorithm 18 has $3n$ iterations. So the total number of* R LWE *oracle calls is $3n^{c+1}\cdot N_0$.*
4. *$N_0 = M_1{\cdot}M_2$*

6.4 Error from Spherical Gaussian Distribution

In this section, we will investigate a polynomial time reduction between ring-DLWE$_{q,\leq\alpha}$ problem to ring-DLWE$_{q,r_0}$ problem. Suppose, we have a distinguisher $\mathcal{D}_1$ to solve ring-DLWE$_{q,r_0}$ problem. We need to devise an algorithm $\mathcal{A}_4$ to solve ring-DLWE$_{q,\leq\alpha}$. The ring-DLWE$_{q,\leq\alpha}$ problem is a worst-case problem as one needs to solve it for any arbitrary value of $s \in R_q$. On the other hand, ring-DLWE$_{q,r_0}$ is an average-case problem as one needs to solve it for a non-negligible function of values of s when s is chosen uniformly at random from R_q. The main idea comes from [Reg09, LPR13]. We have distinguished worst-case, and average-case problems as variable length error problems (VDLWE) and fixed length error

problems (FDLWE) respectively. In [LPR13, PRS17] the problem has been presented in both the settings, viz elliptical and spherical gaussian distributions. We will limit our focus only to spherical distributions (variable length or fixed length) for the whole analysis.

Here we restate a simplified version of the lemma analogous to Lemma 7.2 of [LPR13]. Originally in the lemma, the ring-VDLWE problem has been stated as a family of elliptical Gaussian distributions but here we consider the family of spherical gaussian distributions.

Lemma 68. *For $\alpha > 0$ and $l \geq 1$, there is a randomized polynomial time algorithm which reduces ring-VDLWE$_{q,\leq\alpha}$ problem to ring-FDLWE$_{q,r_0}$ problem, where $r_0 = \alpha\cdot(\frac{nl}{\ln(nl)})^{1/4}$ and l is the number of* LWE *samples.*

We describe the reduction in Algorithm 21 and elaborate on the same in detail here. $\mathcal{A}_4$ has access to a list $\mathcal{T} = ((a_k, \mathbf{b}_k))_{1\leq k\leq l}$ of l samples over $R_q\times\mathbb{T}$. $\mathcal{A}_4$ wants to identify whether the samples are from $A_{s,r}$ or from uniform distribution over $R_q\times\mathbb{T}$, where $r \leq \alpha$. We define another list $\mathcal{T}' = ((a_k{}', \mathbf{b}_k{}'))_{1\leq k\leq l}$ where $a_k{}' = a_k$ and $\mathbf{b}_k{}' = \mathbf{b}_k + \mathbf{f}_k(\mod \sigma(R^\vee))$. Here $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_l$ are chosen independently from D_{r_0}. If $\mathcal{T}$ contains samples from $A_{s,r}$ then the error vector in $\mathbf{b}_k$ follows D_r,while $\mathbf{f}_k$ follows D_{r_0}, hence the error vector in $\mathbf{b}_k{}'$ follows $D_{r'}$ where $r' = \sqrt{r^2 + r_0^2}$. So, the samples from $\mathcal{T}'$ are from $A_{s,r'}$. If, on the other hand $\mathcal{T}$ contains samples from uniform distribution over $R_q\times\mathbb{T}$, the resultant distribution list $\mathcal{T}'$ contains samples from uniform distribution over $R_q\times\mathbb{T}$. Next, we prepare another list $\mathcal{T}''$. First we sample a list $\hat{\mathcal{T}}$ of l samples from uniform distribution over $R_q\times\mathbb{T}$ and add $\mathbf{f}_i$ to the second element of i-th sample of $\hat{\mathcal{T}}$. The resultant list is $\mathcal{T}''$.

In Algorithm 21, we incorporate a solver for $\mathcal{A}_4$ by using the distinguisher $\mathcal{D}_1$ as follows. It has two nested loops. The outer loop runs N_1 times and the inner loop runs N_2 times. In each iteration of the inner loop using samples from the list $\mathcal{T}$, $\hat{\mathcal{T}}$ and $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_l$ creates two lists $\mathcal{T}'$ and $\mathcal{T}''$. In the last part of the inner loop $\mathcal{D}_1$ is called with the inputs $\mathcal{T}'$ and $\mathcal{T}''$ and captures the one-bit output in the variables cnt_0 and cnt_1 respectively. At the end of the inner loop, we calculate the estimated probabilities $\hat{\mathfrak{p}_0}$ and $\hat{\mathfrak{p}_1}$ respectively. Here $\hat{\mathfrak{p}_0}$ and $\hat{\mathfrak{p}_1}$ are estimates of $\mathfrak{p}_0$ and $\mathfrak{p}_1$ with which $\mathcal{D}_1$ accepts the input $\mathcal{T}'$ and $\mathcal{T}''$ respectively. If any of the N_1 outer loop results in $|\hat{\mathfrak{p}_0} - \hat{\mathfrak{p}_1}| \geq \epsilon_2/4$, the algorithm returns 1 and halts otherwise it returns 0 and halts when this condition is not satisfied by any of the N_1 iterations.

Here $\mathcal{D}_1$ is a (ϵ_1, ϵ_2) distinguisher which works when error samples are drawn from $D_{r_0}^l$. But $\mathcal{T}'$ contains samples from $D_{r'}^l$. So, on input $\mathcal{T}'$, $\mathcal{D}_1$ can make both Type-I and Type-II errors. The following analyze how $\mathcal{D}_1$, a (ϵ_1, ϵ_2) distinguisher works for samples drawn from $A_{s,r'}$. Hypothesis testing methodology can be used to analyze the errors in Algorithm 21.

Algorithm 21 may return an incorrect answer in two ways. First, when $\mathcal{T}'$ follows uniform distribution and it returns 1 and second $\mathcal{T}'$ follows $A_{s,r'}$ and it returns 0. The first type of error is called the Type-I error and the second type of error is called is Type-II error. When $\mathcal{T}'$ follows uniform distribution, both the lists $\mathcal{T}'$ and $\mathcal{T}''$ follow uniform distributions over $R_q \times \mathbb{T}$, so $\mathfrak{p}_0 = \mathfrak{p}_1$. For each of the N_1 iterations of the outer loop we have

$$\Pr[\mathfrak{p}_0 - \epsilon_2/8 \leq \hat{\mathfrak{p}_0} \leq \mathfrak{p}_0 + \epsilon_2/8] \geq 1 - 2\exp(-N_2\epsilon_2^2/32)$$
$$\Pr[\mathfrak{p}_1 - \epsilon_2/8 \leq \hat{\mathfrak{p}_1} \leq \mathfrak{p}_1 + \epsilon_2/8] \geq 1 - 2\exp(-N_2\epsilon_2^2/32)$$

due to the additive form of the Chernoff-Hoeffding bound. As $\mathfrak{p}_0 = \mathfrak{p}_1$, it implies that $\Pr[\hat{\mathfrak{p}_0} - \hat{\mathfrak{p}_1}] \geq 1 - 4\exp(-N_2\epsilon_2^2/32)$. So, for all N_1 outer loop iterations the Type-1 failure probability becomes at most $4N_1\exp(-N_2\epsilon_2^2/32)$. When $\mathcal{T}$ follows $A_{s,r}$, $\mathcal{T}'$ follows $A_{s,r'}$. In any of the N_2 iterations of the inner loop let $\mathbf{z}_1, \ldots, \mathbf{z}_l$ be the errors in samples of $\mathcal{T}'$. Let $\mathbf{z}$ be the concatenation of $\mathbf{z}_1, \ldots, \mathbf{z}_l$, so $\mathbf{z}$ follows $D_{r'}^{N_2 l}$. Suppose $\mathbf{z}$ follows $D_{r_0}^{N_2 l}$ instead of $D_{r'}^{N_2 l}$. Then we can use $\mathcal{D}_1$ appropriately. Here we change the Gaussian error width from r' to r_0, later we will have to compute the correction factor using Renyi Divergence. We denote the corresponding probabilities and their estimates by $p_0, p_1, \hat{p_0}$ and $\hat{p_1}$. Let $p_{s,0}$ and $p_{s,1}$ respectively denote the probabilities p_0 and p_1 corresponding to a particular value of s. Similarly, let $\hat{p}_{s,0}$ and $\hat{p}_{s,1}$ respectively denote the estimates $\hat{p_0}$ and $\hat{p_1}$ corresponding to a particular value of s. Lastly let $\hat{p}_{s,\mathbf{z},0}$ and $\hat{p}_{s,\mathbf{z},1}$ respectively denote the estimates $\hat{p_0}$ and $\hat{p_1}$ corresponding to a particular value of s and $\mathbf{z}$. We say that a value s is good if $|p_{s,0} - p_{s,1}| \geq \epsilon_2$. From the definition of an (ϵ_1, ϵ_2) distinguisher, the probability of a good s is at least ϵ_1. If we use the additive form of the Chernoff-Hoeffding for good s it follows.

$$\Pr[p_{s,0} - \epsilon_2/4 \leq \hat{p}_{s,0} \leq p_{s,0} + \epsilon_2/4] \geq 1 - 2\exp(-N_2\epsilon_2^2/8)$$
$$\Pr[p_{s,1} - \epsilon_2/4 \leq \hat{p}_{s,1} \leq p_{s,1} + \epsilon_2/4] \geq 1 - 2\exp(-N_2\epsilon_2^2/8)$$

From above equations along with the condition that $|p_{s,0} - p_{s,1}| \geq \epsilon_2$ we get

$$\Pr[|\hat{p}_{s,0} - \hat{p}_{s,1}| \geq \epsilon_2/2] \geq 1 - 2\exp(-N_2\epsilon_2^2/8) \tag{6.4}$$

If we take N_2 a constant factor of ϵ_2^{-2} the difference $|\hat{p}_{s,0} - \hat{p}_{s,1}|$ will be at least $\epsilon_2/2$ with probability at most 1.

Given a good s, we say that $\mathbf{z}$ is good if $|\hat{p}_{s,\mathbf{z},0} - \hat{p}_{s,\mathbf{z},1}| \geq \epsilon_2/4$ holds. The probability of a good $\mathbf{z}$ is at least $\epsilon_2/4$ (Refer Proposition 53). Now when we change the error distribution

from $D_{r_0}^{N_2 l}$ to $D_{r'}^{N_2 l}$, the probability of a good $\mathbf{z}$ under $D_{r'}^{N_2 l}$ is at least $\epsilon_2^2/(256NN_2l)^{-1/2}$ (Refer Proposition 57). So the probability of a good pair $(s, \mathbf{z})$ where $\mathbf{z}$ follows $D_{r'}^{N_2 l}$ is at least $\epsilon_1\epsilon_2^2/(256nN_2l)^{-1/2}$. If N_1 is around $(256nN_2l)^{1/2}/\epsilon_1\epsilon_2^2$ then with probability exponentially close to 1 a good tuple will be encountered in one of the iterations of the outer loop. Type-2 failure can occur in two ways. The first way is that in none of the N_1 iterations, a good tuple is obtained. The second way is that for a good tuple, the condition $|\hat{\mathfrak{p}}_{s,\mathbf{z},0} - \hat{\mathfrak{p}}_{s,\mathbf{z},1}| \geq \epsilon_2/4$ does not hold. The above analysis shows that the probability of either of these errors is exponentially small.

Algorithm 21 Reducing ring-VDLWE$^i_{q,\leq\alpha}$ to ring-FDLWE$^i_{q,r_0}$

function solveFDLWE($\mathcal{L}$)
  for $k_1 = 1$ to N_1 **do**;
    $\text{cnt}_0 \leftarrow 0$, $\text{cnt}_1 \leftarrow 0$;
    for $k_2 = 1$ to N_2 **do**;
      Obtain a list $\mathcal{T}$ of l samples from $\mathcal{L}$;
      Choose $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_l$ independently from $D_{r_0}^l$;
      Compute $\mathcal{T}'$ and $\mathcal{T}''$ from $\mathcal{T}$, $\hat{\mathcal{T}}$ and $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_l$;
      $\text{cnt}_0 \leftarrow \text{cnt}_0 + \mathcal{D}_1(\mathcal{T}')$, $\text{cnt}_1 \leftarrow \text{cnt}_1 + \mathcal{D}_1(\mathcal{T}'')$;
    end for
    $\hat{\mathfrak{p}_0} \leftarrow \text{cnt}_0/N_2, \hat{\mathfrak{p}_1} \leftarrow \text{cnt}_1/N_2$;
    if then$|\hat{\mathfrak{p}_0} - \hat{\mathfrak{p}_1}| \geq \epsilon_2/4$;
      return 1;
    end if
  end for
  return 0;
end function.

Proposition 69. *We record the following observations.*

1. *$\mathcal{A}_4$ invokes $\mathcal{D}_1$ at most $N_1 \cdot N_2$ times, which is about $(256nl)^{1/2}{N_2}^{3/2^{1/2}}/\epsilon_1\epsilon_2^2$ or $(\epsilon_1\epsilon_2^5)^{-1}(256nl)^{1/2}$, where N_2 is about ϵ_2^{-2}.*
2. *The number of LWE samples needed by $\mathcal{A}_4$ is about $M_1 = (\epsilon_1\epsilon_2^5)^{-1}(256n)^{1/2}l^{3/2}$.*

Running time:

- Algorithm 21 makes $2N_1N_2$ calls to the R-LWE oracle in the worst case.
- Since, Algorithm 20 makes one call to Algorithm 21, therefore the number of R-LWE calls that Algorithm 20 makes is also $2N_1N_2$

6.5 End to end Concrete Analysis

The complete reduction by Regev [LPR13] is from worst-case ideal SIVP to average-case decisional RLWE problem with spherical gaussian error. This reduction consists of three parts.

- K-SIVP$_\gamma$ to K-DGS$_\Gamma$ is about n^3.
- Reducing K-DGS$_\Gamma$ to ring-LWE$_{q,\leq\alpha}$ is about $3n^{c+1} \cdot N_0$.
- Oracle calls for ring-LWE$_{q,\leq\alpha}$ elliptical to spherical reduction is about $(\epsilon_1\epsilon_2^5)^{-1}(256nl)^{1/2}$.
- $N_0 = M_1 \cdot M_2$ and $M_2 = 10^{43} \cdot n^{25} \cdot {M_1}^{12}$ from Proposition 67.
- $M_1 = (\epsilon_1\epsilon_2^5)^{-1}(256n)^{1/2} l^{3/2}$ from Proposition 69.

The overall tightness gap is given by the number of times $\mathcal{A}_0$ calls distinguisher $\mathcal{D}_1$. The number is about

$$\begin{aligned} n^3 \cdot 3n^{c+1} N_0 \cdot M_2 \cdot (\epsilon_1\epsilon_2^5)^{-1}(256nl)^{1/2} &\approx n^{c+4} \cdot M_1 {M_2}^2 \cdot (\epsilon_1\epsilon_2^5)^{-1}(256nl)^{1/2} \\ &\approx 10^{86} \cdot n^{c+54} \cdot {M_1}^{25} (\epsilon_1\epsilon_2^5)^{-1}(nl)^{1/2} \\ &\approx 10^{86} \cdot n^{c+54} \cdot ((\epsilon_1\epsilon_2^5)^{-1}(n^{1/2}l^{3/2}))^{25} \cdot (\epsilon_1\epsilon_2^5)^{-1}(nl)^{1/2} \\ &= 10^{86} \cdot n^{c+67} \cdot l^{38} \cdot (\epsilon_1\epsilon_2^5)^{-26} \end{aligned}$$

Theorem 70. *Let K be a number filed of dimension n and R be the ring of integer of K. Let r_0 be a positive real integer and $r_0 = \alpha \cdot (\frac{nl}{\ln(nl)})^{1/4}$, l is the number of* LWE *samples, q be a integer greater than 2 and c be any positive integer. Suppose there is a (ϵ_1, ϵ_2) distinguisher $\mathcal{D}$ that solves* DLWE$_{q,r_0}$*, then there is a quantum algorithm $\mathcal{A}$ requiring approximately $3n^2$ logical qubits to solve $K-$*SIVP$_{2\sqrt{2}\omega(\sqrt{n}/\alpha)\cdot\eta_\epsilon(\mathcal{I})/\lambda_n(\mathcal{I})}$*, The number of times $\mathcal{A}$ calls $\mathcal{D}$ is about*

$$10^{86} \cdot n^{c+67} \cdot l^{38} \cdot (\epsilon_1\epsilon_2^5)^{-26} \tag{6.5}$$

6.6 Conclusion

In our analysis, we carefully examined the tightness gap of the reduction for any ring and modulus, as presented in [PRS17]. The results of our analysis indicated that this reduction does not provide a meaningful guarantee of real-world security. To put it in perspective, we compared the tightness gap estimate of this reduction with those of other reductions for module-LWE, ring-LWE, or LWE. It became evident that this reduction is not tighter than the others by any means. A detailed discussion on the effect of tightness gap analysis is presented in the concluding chapter of our thesis. In that chapter, we extensively compare and contrast the tightness of various reductions and remark on their practical usability. By doing so, we aim to gain a comprehensive understanding of the practical security implications of the cryptographic constructions based on these reductions.

Chapter 7

Classical Reduction from SIVP to LWE

7.1 Introduction

In Chapter 4, we analyzed the tightness gap of the reductions in [Reg09]. These results are important from the theoretical aspects of the equivalence between the worst-case and average-case hard problems. From a practical point of view, these results/reductions lack a tightness guarantee. Cryptosystems based on the hardness of the LWE problem may be prone to security risk due to the large tightness gap of its reductions. Another distinct feature of the reductions obtained by Regev is quantum, i.e., the algorithm is required to make some quantum computations.

A problem left open by Regev [Reg09] was whether there is a classical reduction from a worst-case lattice problem to LWE. The initial answer to this problem was provided by Peikert [Pei09]. While this represented progress, Peikert's reduction was not considered to be satisfactory since either an exponential size modulus is required or, the lattice problem considered is not one of the standard problems. Later work by Brakerski et al. [BLP+13] built on Peikert's work to show a classical reduction from a standard lattice problem to LWE avoiding the exponential size modulus.

The works of Regev [Reg09], Peikert [Pei09] and Brakerski et al. [BLP+13] are all in the asymptotic setting where the lattice dimension is allowed to go to infinity. Practical cryptosystems, on the other hand, have a fixed value of the lattice dimension. So, it is interesting to know what kind of security assurance one obtains from the results of [Pei09, BLP+13] for practical cryptosystems.

In this Chapter, we perform a concrete security analysis of the tightness gap of the reduction in [BLP+13]. The reduction of Peikert [Pei09] is a step in the reduction performed by Brakerski et al. [BLP+13]. As a first step, we work out the tightness gap of Peikert's reduction. Then we follow the proof strategy in Brakerski et al. [BLP+13] and finally work out the end-to-end tightness gap of the classical reduction from the gap shortest vector problem to the LWE. There are two aspects to the concrete analysis. The first is a quadratic loss in the dimension of the lattice and the second is a loss of tightness. The loss of tightness in this

classical reduction is more than that of the original quantum reduction by Regev [Reg09]. The quadratic loss in the dimension was already pointed out in [BLP+13]. Due to this quadratic loss, Brakerski et al. put forward the open question of obtaining a reduction without such a loss mentioning that this would amount to a full de-quantization of Regev's reduction. The paper [BLP+13], however, does not consider the issue of the loss in tightness. Our analysis shows that due to this loss of tightness, the reduction is not very meaningful in practice, especially for determining the sizes of the parameters of a cryptosystem that would purportedly enjoy the protection offered by the hardness of a well-studied worst-case lattice problem.

7.1.1 Outline of the Analysis

The reductions in [BLP+13] can be divided into three parts. The first part is a brief recap of DGS to LWE reduction from Chapter 4. The second part is the reduction from the GapSVP to LWE problem [Pei09], while the third part is a reduction from the GapSVP to Decision LWE [BLP+13]. The concrete analysis of each part is described in the following sections.

7.2 Reducing DGS to LWE

Regev [Reg09] described a quantum algorithm that given access to an LWE oracle can solve the SIVP (or, the GapSVP). In the first step, the SIVP is reduced to the DGS problem using a classical algorithm. The main part of the proof is a quantum algorithm that reduces the DGS problem to the LWE problem. The proof given by Regev [Reg09] is in an asymptotic setting. A concrete analysis of the tightness gap in the reduction was carried out in [CKMS16] and in more detail in Chapter 4.

Let p be a positive integer and $\alpha \in (0,1)$. Assume that an oracle $\mathsf{solveLWE}_{n,I,p,\Psi_\alpha}(\mathcal{I})$ is available for some polynomial I of n. The input $\mathcal{I}$ to the oracle consists of I samples from $A_{p,\mathbf{s},\Psi_\beta}$ for some $0 < \beta \leq \alpha$. The oracle is guaranteed to work correctly if $\beta = \alpha$, otherwise it might return an incorrect result. Let $\mathbf{B}$ be an $n \times n$ basis matrix of an n-dimensional lattice $L = L(\mathbf{B})$ and r is a real number satisfying $r \geq \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$. The goal is to design an algorithm $\mathsf{solveDGS}(\mathbf{B}, r)$ which returns a sample from $D_{L,r}$ using the oracle $\mathsf{solveLWE}_{n,I,p,\Psi_\alpha}(\mathcal{I})$ where $\alpha p > 2\sqrt{n}$.

Let $r_i = r \cdot (\alpha p/\sqrt{n})^i$ for $i = 1, \ldots, 3n$. A list $\mathcal{L}$ containing samples from $D_{L,r_{3n}}$ can be created without using the LWE oracle. The algorithm $\mathsf{solveDGS}(\mathbf{B}, r)$ starts with such a list

and iterates a procedure over $3n$ steps with i going down from $3n$ to 1. The i-th step updates the list $\mathcal{L}$ consisting of I samples from D_{L,r_i} with I samples from $D_{L,r_{i-1}}$. At the end of the procedure, a sample from the final list $\mathcal{L}$ is returned. Each iteration updates the list $\mathcal{L}$ using a quantum sampling procedure I times. Each application of the quantum sampling procedure uses a classical algorithm $\mathsf{solveCVP}(L^*, \mathcal{L}, \mathbf{z})$, where L^* is the dual lattice of L, $\mathcal{L}$ contains n^c samples from D_{L,r_i} for some $i \in \{1, \ldots, 3n\}$, and $\mathbf{z}$ is within distance $\lambda_1(L^*)/2$ of L^*. The algorithm $\mathsf{solveCVP}$ solves the CVP problem for L^*. It is the algorithm $\mathsf{solveCVP}$ which invokes the oracle $\mathsf{solveLWE}_{n,I,p,\Psi_\alpha}(\mathcal{I})$. So, the main part of the DGS-to-LWE reduction is the design of the algorithm $\mathsf{solveCVP}$.

In Regev's reduction, $\mathsf{solveCVP}(L^*, \mathcal{L}, \mathbf{z})$ solves the unique closest vector problem on L^* using a list $\mathcal{L}$ of samples from $D_{L,\mathfrak{r}}$ with $\mathfrak{r} \geq \sqrt{2}p \cdot \eta_\epsilon(L)$, and $\mathbf{z}$ is within distance $\alpha q/(\sqrt{2}\mathfrak{r}) < \lambda_1(L^*)/2$ of L^*. As used in [Pei09], by interchanging the roles of L and L^*, it is possible to invoke $\mathsf{solveCVP}(L, \mathcal{L}, \mathbf{z})$ to solve the unique closest vector problem on L using a list $\mathcal{L}$ of samples from $D_{L^*,\mathfrak{r}}$ with $\mathfrak{r} \geq \sqrt{2}p \cdot \eta_\epsilon(L^*)$, and $\mathbf{z}$ is within distance $\alpha q/(\sqrt{2}\mathfrak{r}) < \lambda_1(L)/2$ of L. We record this as follows.

Proposition 71. *[Reg09, Pei09] Let $\mathbf{B}$ be an $n \times n$ basis matrix for an n-dimensional lattice $L = L(\mathbf{B})$, p be a positive integer, $\mathfrak{r}$ be a real number satisfying $\mathfrak{r} \geq \sqrt{2}p \cdot \eta_\epsilon(L^*)$ and $\alpha \in (0, 1)$ be such that $\alpha p > 2\sqrt{n}$. Let I be a polynomial in n. Given a list $\mathcal{L}$ consisting of I samples from $D_{L^*,\mathfrak{r}}$ and an oracle $\mathsf{solveLWE}_{n,I,p,\Psi_\alpha}(\mathcal{I})$, where $\mathcal{I}$ consists of I samples from $A_{p,\mathbf{s},\Psi_\beta}$ for some $0 < \beta \leq \alpha$, there is an algorithm $\mathsf{solveCVP}(L, \mathcal{L}, \mathbf{z})$, where $\mathbf{z}$ is within distance $\alpha q/(\sqrt{2}\mathfrak{r}) < \lambda_1(L)/2$ of L, which finds the unique vector in L which is closest to $\mathbf{z}$.*

From Chapter 4, we have the following facts.

1. Algorithm $\mathsf{solveCVP}$ calls the oracle $\mathsf{solveLWE}$ a total of nI^2 times.

2. The success probability of algorithm $\mathsf{solveCVP}$ is at least

$$\left(1 - \max(\exp\left(-m(\mu_0 - t)^2/2\right), \exp\left(-mt^2/2\right))\right)^{nI^2} \tag{7.1}$$

where m is a positive integer which is upper bounded by I, $\mu_0 = \exp(-\pi\alpha^2)$, and $t \in (0, \mu_0)$ are chosen so as to maximise (7.1). Setting $t = \mu_0/2$, the expression in (7.1) becomes

$$\left(1 - \exp\left(-m \exp\left(-2\pi\alpha^2\right)/8\right)\right)^{nI^2} \tag{7.2}$$

Using this lower bound for the success probability, it has been shown in [KSSS22] that an upper bound on the tightness gap of the DGS to LWE reduction is the following.

$$3n^2I^3 \cdot \left(1 - \exp\left(-m\exp(-2\pi\alpha^2)/8\right)\right)^{-3n^2I^3}. \tag{7.3}$$

For most practical cryptosystems, α is at most $1/\sqrt{n}$. Considering $\alpha = 1/\sqrt{n}$, the tightness gap given by (7.3) is essentially $3n^2I^3$. The tightness gap of the reduction from DGS to LWE has been extended to obtain the tightness gap of the reduction from SIVP to average-case decision LWE in Chapter 4 and is given by the following expression.

$$n^{11} \cdot (\delta_1 {\delta_2}^2)^{-4}. \tag{7.4}$$

Here δ_1 and δ_2 are non-negative integers such that average-case decision LWE can be solved for a fraction $n^{-\delta_1}$ of all the secrets with advantage at least $n^{-\delta_2}$ and subsequent calculation shows that $I \approx (\delta_1 {\delta_2}^2)^{-1} n$ (Refer section 4.5 of Chapter 4).

7.3 Reducing $\mathrm{GapSVP}_{\zeta,\gamma}$ to LWE

Peikert [Pei09] showed a classical reduction of $\mathrm{GapSVP}_{\zeta,\gamma}$ to $\mathrm{LWE}_{n,I,q,\Psi_\alpha}$, where $\gamma = \gamma(n) \geq n/(\alpha\sqrt{\log n})$, $q = q(n) \geq \zeta(n) \cdot \omega(\sqrt{\log n/n})$ and I is a polynomial in n. The reduction makes use of Proposition 71, i.e., it uses an LWE oracle to solve CVP.

Let $\mathbf{B}$ be an $n \times n$ basis matrix of an n-dimensional lattice $L = L(\mathbf{B})$ and $r \geq \max_i \|\widetilde{b_i}\| \cdot \omega(\sqrt{\log n})$. By $\mathsf{sample}(\mathbf{B}, r)$ we denote the sampling algorithm which on input $\mathbf{B}$ and r returns a sample which is within negligible statistical distance from $D_{L,r}$. (See Theorem 24).

The algorithm for reducing $\mathrm{GapSVP}_{\zeta,\gamma}$ to LWE given by Peikert [Pei09] is shown in Algorithm 22. The algorithm $\mathsf{solveCVP}$, in turn, calls the LWE oracle $\mathsf{solveLWE}$. So, overall $\mathsf{solveGapSVP}_{\zeta,\gamma}$ solves $\mathrm{GapSVP}_{\zeta,\gamma}$ by calling the LWE oracle $\mathsf{solveLWE}$. Algorithm $\mathsf{solveGapSVP}_{\zeta,\gamma}$ calls $\mathsf{solveCVP}$ a total of N times.

It has been noted in Section 7.2 that $\mathsf{solveCVP}$ calls $\mathsf{solveLWE}$ a total of nI^2 times. So, $\mathsf{solveGapSVP}_{\zeta,\gamma}$ calls $\mathsf{solveLWE}$ a total of $N \cdot nI^2$ times.

We now consider the success probability of $\mathsf{solveGapSVP}_{\zeta,\gamma}$. As in Section 7.2, assume that $\alpha = 1/\sqrt{n}$ and $t = \mu_0/2$. The probability that a single call to $\mathsf{solveCVP}$ is successful is at least ε, where using (7.2), $\varepsilon = \left(1 - \exp\left(-m\exp(-2\pi\alpha^2)/8\right)\right)^{nI^2}$. The N calls to $\mathsf{solveCVP}$ in Algorithm $\mathsf{solveGapSVP}_{\zeta,\gamma}$ are independent. Let E be the event that all these calls are

Algorithm 22 Reducing $\text{GapSVP}_{\zeta,\gamma}$ to $\text{LWE}_{q,\Psi_\alpha}$, where $\gamma = \gamma(n) \geq n/(\alpha\sqrt{\log n})$ and $q = q(n) \geq \zeta(n) \cdot \omega(\sqrt{\log n/n})$.

function $\mathsf{solveGapSVP}_{\zeta,\gamma}(\mathbf{B}, d)$
  Let $\mathbf{D}$ be the reverse dual basis of $\mathbf{B}$;
  $d' = d \cdot \sqrt{n/(4 \ln n)}$; $r = q\sqrt{2n}/(\gamma d)$;
  for $i \leftarrow 1$ to N **do**
    $\mathbf{w} \xleftarrow{\$} d' \cdot \mathcal{B}_n$; $\mathbf{x} = \mathbf{w} \bmod \mathbf{B}$;
    $\mathcal{L} \leftarrow \{\}$;
    for $j \leftarrow 1$ to n^c **do**
      $\mathcal{L} \leftarrow \mathcal{L} \cup \mathsf{sample}(D, r)$;
    end for
    $\mathbf{v} \leftarrow \mathsf{solveCVP}(\mathbf{B}, \mathcal{L}, \mathbf{x})$
    if $\mathbf{v} \neq \mathbf{x} - \mathbf{w}$ **then**
      return accept;
    end if
  end for
  return reject;
end function

successful and so $\Pr[E] \geq \varepsilon^N$.

For $i = 1, \ldots, N$, let S_i be the event that the event $\mathbf{v} \neq \mathbf{x} - \mathbf{w}$ holds in the i-th iteration. The events $S_1, \ldots, S_N$ are independent (even when conditioned on E).

First consider the instance $(\mathbf{B}, r)$ to be NO instance of $\text{GapSVP}_{\zeta,\gamma}$. Let succNO be the event that algorithm $\mathsf{solveGapSVP}_{\zeta,\gamma}$ is successful on a NO instance. Then $\Pr[\mathsf{succNO}] = \Pr[\overline{S}_1 \wedge \cdots \wedge \overline{S}_N] \geq \Pr[\overline{S}_1 \wedge \cdots \wedge \overline{S}_N | E] \Pr[E] = \Pr[E] \cdot \left(\prod_{i=1}^N \Pr[\overline{S}_i | E]\right)$
$\geq \varepsilon^N \cdot \left(\prod_{i=1}^N \Pr[\overline{S}_i | E]\right)$. It has been shown in [Pei09] that $\Pr[\overline{S}_i | E] \approx 1$, $i = 1, \ldots, N$, and so we may assume that $\Pr[\mathsf{succNO}]$ is lower bounded by ε^N.

Next consider the instance $(\mathbf{B}, r)$ to be a YES instance of $\text{GapSVP}_{\zeta,\gamma}$. Let $\mathsf{succYES}$ be the event that algorithm $\mathsf{solveGapSVP}_{\zeta,\gamma}$ is successful on a YES instance. So, $\mathsf{succYES}$ is the event $S_1 \vee (\overline{S}_1 \wedge S_2) \vee \cdots \vee (\overline{S}_1 \wedge \cdots \wedge \overline{S}_{N-1} \wedge S_N)$. For $i = 1, \ldots, N$, let δ be the common value of $\Pr[\overline{S}_i | E]$. It follows that

$$\Pr[\mathsf{succYES}] \;\geq\; \Pr[\mathsf{succYES} | E] \Pr[E] = (1 - \delta^N) \Pr[E] \geq (1 - \delta^N)\varepsilon^N.$$

It has been shown in [Pei09], that for a YES instance, $\delta = \Pr[\overline{S}_i | E] \leq 1 - 1/\mathsf{poly}(n)$. The $1 - 1/\mathsf{poly}(n)$ term arises from the asymptotic form of a result which states that for any

constants $c_1, d > 0$ and any $\mathbf{z} \in \mathbb{R}^n$ with $\|\mathbf{z}\| \leq d$ and $d' = d \cdot \sqrt{n/(c_1 \log n)}$ the statistical distance between the uniform distribution on $d' \cdot \mathcal{B}_n$ and the uniform distribution on $\mathbf{z} + d' \cdot \mathcal{B}_n$ is at most $1 - 1/\mathsf{poly}(n)$. This result is proved in Lemma 29 and the proof shows that the term $1 - 1/\mathsf{poly}(n)$ can be taken to be $1 - 3/n^2$. Using this we have $\delta \leq 1 - 3/n^2$. So, $\Pr[\mathsf{succYES}] \geq (1 - (1 - 3/n^2)^N)\varepsilon^N$.

Between the NO and YES instances, the lower bound on the success probability is less for YES instances. As a result, the upper bound on the tightness gap for YES instances is higher and this upper bound is taken to be the upper bound on the overall tightness gap of the reduction. So, an upper bound on the tightness gap of the $\mathrm{GapSVP}_{\zeta,\gamma}$ to LWE reduction is

$$\left(N \cdot nI^2\right) / \left((1 - (1 - 3/n^2)^N)\varepsilon^N\right). \tag{7.5}$$

Following Lemma 29, for $N = n^2$, $(1 - (1 - 3/n^2)^N) \approx 1$ and so the tightness gap in (7.5) becomes

$$N \cdot nI^2 \cdot \varepsilon^{-N} = n^3 I^2 \left(1 - \exp\left(-m \exp(-2\pi\alpha^2)/8\right)\right)^{-n^3 I^2}. \tag{7.6}$$

We note that the expression in (7.6) is almost the same as the expression in (7.3). It has been discussed in Chapter 4, that for $\alpha \leq 1/\sqrt{n}$, $\varepsilon \approx 1$ and so the tightness gap of $\mathrm{GapSVP}_{\zeta,\gamma}$ to $\mathrm{LWE}_{q,\Psi_\alpha}$ becomes

$$n^3 I^2. \tag{7.7}$$

The tightness gap of the reduction from $\mathrm{GapSVP}_{\zeta,\gamma}$ to LWE has been extended to obtain the tightness gap of the reduction from SIVP to average-case decision LWE in Chapter 4 and is given by the following expression.

$$n^8 \cdot (\delta_1 {\delta_2}^2)^{-3}. \tag{7.8}$$

Here δ_1 and δ_2 are non-negative integers such that average-case decision LWE can be solved for a fraction $n^{-\delta_1}$ of all the secrets with advantage at least $n^{-\delta_2}$ and elaborate calculations show that $I \approx (\delta_1 {\delta_2}^2)^{-1} n$. The tightness gap of search LWE to DLWE_{ac} with is $np(\delta_1 {\delta_2}^2)^{-1}$ where p is in the order of n^2 for practical cryptosystems. (Refer Section 4.5 of Chapter 4).

Remark: It is known [Pei09] that for $\zeta(n) \geq 2^{n/2}$, the problem GapSVP$_{\zeta,\gamma}$ is equivalent to the standard GapSVP$_\gamma$ problem. The reduction from GapSVP$_{\zeta,\gamma}$ to LWE$_{q,\Psi_\alpha}$ given in [Pei09] holds under the condition $q = q(n) \geq \zeta(n) \cdot \omega(\sqrt{\log n/n})$. So, for $q(n) \geq 2^{n/2} \cdot \omega(\sqrt{\log n/n})$, there is a classical reduction from GapSVP$_\gamma$ to LWE$_{q,\Psi_\alpha}$, where $\gamma = \gamma(n) \geq n/(\alpha\sqrt{\log n})$.

7.4 Reducing GapSVP$_\gamma$ to Decision LWE

In this section, we discuss the tightness of the reduction for the classical hardness of LWE. This reduction takes the result of [Pei09] as the starting point. The remark at the end of Section 7.3 shows that there is a classical reduction of GapSVP$_\gamma$ to LWE$_{q,\Psi_\alpha}$ for $q(n) \geq 2^{n/2} \cdot \omega(\sqrt{\log n/n})$. So, if the modulus of the LWE problem is exponential in the dimension of the lattice, then the result from [Pei09] provides a classical reduction of GapSVP$_\gamma$ to LWE. GapSVP$_\gamma$ problem is considered a standard hard lattice problem to have worst-case hardness. The work by Brakerski et al. [BLP+13] showed a reduction of GapSVP$_\gamma$ to a decision version of LWE with a polynomial-sized modulus. The reduction is quite intricate and is built by composing reductions between several pairs of problems. The goal of the present section is to perform a concrete security analysis of the reduction provided in [BLP+13].

The LWE problem considered in Section 7.3 is a search problem. For the classical reduction of GapSVP$_\gamma$ to LWE, the binLWE$_{n,m,q,\alpha}$ problem has been considered. We define the advantage of a distinguisher for binLWE$_{n,m,q,\alpha}$ in the following manner.

Let $\mathcal{D}_0$ be the distribution $A_{q,\mathbf{s},\alpha}$ and $\mathcal{D}_1$ be the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{T}$. For $i = 0, 1$, let $\mathcal{I} \xleftarrow{m} \mathcal{D}_i$ denote the selection of a list $\mathcal{I}$ of m independent samples from $\mathcal{D}_i$. Let $\mathcal{A}$ be a distinguisher for decLWE$_{n,m,q,\alpha}$. Let $\mathcal{A}(\mathcal{I}) \Rightarrow 1$ denote the event that $\mathcal{A}$ produces 1 as output. The advantage of $\mathcal{A}$ is the following.

$$\mathsf{Adv}(\mathcal{A}) \quad = \quad |\Pr[\mathcal{A}(\mathcal{I}) \Rightarrow 1 : I \xleftarrow{m} \mathcal{D}_0] - \Pr[\mathcal{A}(\mathcal{I}) \Rightarrow 1 : I \xleftarrow{m} \mathcal{D}_1]|. \tag{7.9}$$

The classical reduction in [BLP+13] reduces GapSVP to binLWE. This reduction is done in several steps. The first step is Peikert's reduction of GapSVP to LWE with exponential size modulus. The goal of the following steps is to reduce the LWE problem with exponential size modulus to binLWE problem with polynomial-size modulus. A trade-off is an increase in the dimension. The various steps of the overall reduction are as follows.

Reducing GapSVP$_\gamma$ to LWE$_{k,m_1,q_1,\alpha_1}$: This follows from section 7.3. Here $\alpha_1 \in (0,1)$, $q_1 \geq 2^{k/2} \cdot \omega(\sqrt{\log k/k})$ and $\gamma \geq k/(\alpha_1\sqrt{\log k})$. For simplicity, in the following, we will assume $q_1 = 2^{k/2}$.

Proposition 72. *Suppose W_0 is an algorithm to solve LWE$_{k,m_1,q_1,\alpha_1}$. Then following the analysis in Section 7.3, there is an algorithm W to solve GapSVP$_\gamma$ where the number of times W calls W_0 is $k^3{m_1}^2$ (which is obtained from (7.7) by replacing n with k and I with m_1).*

Reducing LWE$_{k,m_1,q_1,\alpha_1}$ to decLWE$_{k,m_1,q_1,\alpha_2}$: This follows as a special case of Theorem 25. Here $1/q_1 < \alpha_1 < 1/\omega(\sqrt{\log n})$ and $\alpha_2 = \alpha_1 \cdot \omega(\log k)$. To determine the tightness gap of the reduction, we follow the proof of Theorem 25 in the case where $q_1 = 2^{k/2}$. Let W_1 be an algorithm to solve decLWE$_{k,m_1,q_1,\alpha_2}$. The proof of Theorem 25 uses W_1 to first construct an algorithm W_1' following the construction used in Lemma 30 which has been discussed in section 4.4.3. Specifically, Lemma 30 shows how to boost the advantage of a distinguisher for the distributions $A_{q_1,\mathbf{s},\chi}$ and $U(\mathbb{Z}_{q_1}^n \times \mathbb{Z}_{q_1})$. The same method can be used to boost the advantage of a distinguisher for the distributions $A_{q_1,\mathbf{s},\alpha_2}$ and the uniform distribution on $\mathbb{Z}_{q_1}^n \times \mathbb{T}$. This is the situation considered in Theorem 25.

Let ζ_1 be the advantage of W_1 and c_1 and c_2 be such that W_1 is successful on a fraction k^{-c_1} of all possible secrets and

$$\zeta_1 \quad = \quad k^{-c_2}. \tag{7.10}$$

Following the method of Lemma 30, it is possible to construct W_1' which accepts with probability exponentially close to one on inputs from $A_{q_1,\mathbf{s},\alpha_2}$ and rejects with probability exponentially close to one on inputs from the uniform distribution over $\mathbb{Z}_{q_1}^n \times \mathbb{T}$. From the Section 4.4.3 we have that the algorithm W_1' calls the algorithm W_1 a total of $k^{c_1+2c_2+2}$ times.

The proof of Theorem 25 uses W_1' to construct an algorithm W_0 which solves LWE$_{k,m_1,q_1,\alpha_1}$. The secret $\mathbf{s} = (s_1, \ldots, s_k)$. The components $s_1, \ldots, s_k$ are determined one by one. Consider the determination of s_1. This is determined iteratively as $s_1 \bmod 2$, followed by $s_1 \bmod 2^2$, followed by $s_1 \bmod 2^3$, up to at most $s_1 \bmod 2^{k/2}$. Given the value of $s_1 \bmod 2^i$, there are only two possible values for $s_1 \bmod 2^{i+1}$. A single call to W_1' can be used to determine the correct value. So, to find s_1, at most $k/2$ calls to W_1' are required, and to find the entire vector $\mathbf{s}$, at most $k^2/2$ calls to W_1' are required. Each call to W_1' requires $k^{c_1+2c_2+2}$ calls to

W_1.

Proposition 73. *The number of times W_0 calls W_1 is*

$$k^{c_1+2c_2+4}. \tag{7.11}$$

Reducing decLWE$_{k,m_1,q_1,\alpha_2}$ to binLWE$_{n,m_1,q_1,\leq\sqrt{10n}\alpha_2}$: This reduction follows from Theorem 26. Here $n \geq (k+1)\log_2 q_1 + 2\log_2(1/\delta)$, $\alpha_2 \geq \sqrt{\ln(2n(1+1/\varepsilon_1))/\pi}/q_1$, where $\delta > 0$ and $\varepsilon_1 \in (0, 1/2)$. Suppose there is an algorithm W_2 for binLWE$_{n,m_1,q_1,\leq\sqrt{10n}\alpha_2}$ which has advantage ζ_2. Theorem 26 shows an algorithm W_1 for decLWE$_{k,m_1,q_1,\alpha_2}$ with advantage ζ_1 where

$$\zeta_1 \geq \frac{\zeta_2 - \delta}{3m_1} - \frac{41\varepsilon_1}{2} - 2^{-k-1}. \tag{7.12}$$

Proposition 74. *From the proof of Theorem 26, one obtains that W_1 calls W_2 once.*

Remark: We note a peculiarity in (7.12). The number of samples m_1 appears in the denominator of the right-hand side. If ζ_2 is fixed, then as m_1 increases, the right-hand side decreases. In other words, for a fixed value of ζ_2, as the number of samples increases, the lower bound on the advantage ζ_1 decreases. Intuitively, one may expect that as the number of samples increases, more information is obtained, so the advantage should be non-decreasing. This does not seem to hold for ζ_1. A possible explanation could be that m_1 and ζ_2 are positively correlated in which case, if m_1 increases, ζ_2 will also increase leaving the lower bound unchanged. Since the nature of the dependence of ζ_2 on m_1 is unknown, the issue cannot be definitively settled.

Reducing binLWE$_{n,m_1,q_1,\leq\sqrt{10n}\alpha_2}$ to binLWE$_{n,m_1,q_2,\leq\alpha_3}$: This reduction follows from the Theorem 27. Here $q_1 \geq q_2 \geq \sqrt{2\ln(2n(1+1/\varepsilon_2))} \cdot (\sqrt{n}/\alpha_2)$ and $\alpha_3^2 \geq 10n\alpha_2^2 + (4n/(\pi q_2^2))$ $\ln(2n(1+1/\varepsilon_2))$ where $\varepsilon_2 \in (0, 1/2)$.

Suppose there is an algorithm W_3 for binLWE$_{n,m_1,q_2,\leq\alpha_3}$ having advantage ζ_3. Theorem 27 shows an algorithm W_2 for binLWE$_{n,m_1,q_1,\leq\sqrt{10n}\alpha_2}$ with advantage ζ_2 where

$$\zeta_2 \geq \zeta_3 - 14\varepsilon_2 m_1. \tag{7.13}$$

Proposition 75. *W_2 calls W_3 once.*

Reducing binLWE$_{n,m_1,q_2,\leq\alpha_3}$ to binLWE$_{n,m_2,q_2,\alpha_3}$: This reduction follows from Lemma 77, the proof of which is worked out in section 7.5. Suppose there is an algorithm W_4 for binLWE$_{n,m_2,q_2,\alpha_3}$ having advantage ζ_4. Lemma 77 states that the algorithm W_3 for binLWE$_{n,m_1,q_2,\leq\alpha_3}$ has advantage ζ_3 where $\zeta_3 \geq 1/3$. Further, it is stated that both m_1 and the number of times W_3 calls W_4 are poly$(m_2, 1/\zeta_4, n, \log q_2)$. In Lemma 77, we show that $m_1 = \mathfrak{k} m_2$ and the number of times W_3 calls W_4 is $\mathfrak{k}(1 + 36m_2/\zeta_4)$ where $\mathfrak{k} \geq \max(32 \ln 12, 8 \ln(432 m_2/\zeta_4))/\zeta_4^2$. For simplicity, we take $\mathfrak{k} = 1/\zeta_4^2$. We assume that there are constants $d_1, d_2 > 0$, such that $m_2 = n^{d_1}$ and $\zeta_4 = n^{-d_2}$.

Putting together the various reductions, yields a reduction from GapSVP$_\gamma$ on a lattice of dimension k to binLWE$_{n,m_2,q_2,\alpha_3}$. The number of times C the algorithm W_4 (for solving binLWE$_{n,m_2,q_2,\alpha_3}$) is called by the algorithm W (for solving GapSVP$_\gamma$) is obtained from the above analysis (Proposition 72,73,74,75) to be the following.

$$\begin{aligned} C &= k^3 {m_1}^2 \cdot k^{c_1+2c_2+4} \cdot \frac{1}{\zeta_4^2}\left(1 + \frac{36m_2}{\zeta_4}\right) \\ &\approx k^3 {m_1}^2 \cdot k^{c_1+2c_2+4} \cdot \frac{m_2}{\zeta_4^3} \\ &= k^3 {m_1}^2 \cdot k^{c_1+2c_2+4} \cdot n^{d_1+3d_2}. \end{aligned} \tag{7.14}$$

Let the runtime of W_4 be T and the runtime of W be T'. Then $T'/T \approx C$. The advantage of W_4 is ζ_4 while the success probability of W is almost 1. The tightness gap of the reduction is $T'/(T/\zeta_4) = C\zeta_4$ which is equal to

$$G = k^3 {m_1}^2 \cdot k^{c_1+2c_2+4} \cdot n^{d_1+2d_2}. \tag{7.15}$$

The relations among the various parameters are as follows.

1. $\gamma \geq k/(\alpha_1\sqrt{\log k})$;
2. $q_1 = 2^{k/2}$;
3. $m_1 = k^c$ for some constant $c \geq 1$;
4. $1/q_1 < \alpha_1 < 1/\omega(\sqrt{\log n})$ and $\alpha_2 = \alpha_1 \cdot \omega(\log k)$;
5. The constants c_1 and c_2 are such that W_1 is successful on a fraction k^{-c_1} of all possible secrets and $\zeta_1 = k^{-c_2}$;
6. $n \geq (k+1)\log_2 q_1 + 2\log_2(1/\delta)$;

7. $\alpha_2 \geq \sqrt{\ln(2n(1+1/\varepsilon_1))/\pi}/q_1$, and $\zeta_1 \geq \frac{\zeta_2-\delta}{3m_1} - \frac{41\varepsilon_1}{2} - 2^{-k-1}$, where $\delta > 0$ and $\varepsilon_1 \in (0, 1/2)$;

8. $q_1 \geq q_2 \geq \sqrt{2\ln(2n(1+1/\varepsilon_2))} \cdot (\sqrt{n}/\alpha_2)$, $\alpha_3^2 \geq 10n\alpha_2^2 + (4n/(\pi q_2^2))\ln(2n(1+1/\varepsilon_2))$, and $\zeta_2 \geq \zeta_3 - 14\varepsilon_2 m_1$, where $\varepsilon_2 \in (0, 1/2)$;

9. $\zeta_3 \geq 1/3$;

10. $m_1 = m_2/\zeta_4^2$;

11. $m_2 = n^{d_1}$ and $\zeta_4 = n^{-d_2}$ for constants $d_1, d_2 > 0$.

Note that

$$\begin{aligned}\zeta_1 &\geq \frac{\zeta_2-\delta}{3m_1} - \frac{41\varepsilon_1}{2} - 2^{-k-1} \\ &\geq \frac{\zeta_3}{3m_1} - \frac{14\varepsilon_2}{3} - \frac{\delta}{3m_1} - \frac{41\varepsilon_1}{2} \\ &\geq \frac{1}{9m_1} - \frac{14\varepsilon_2}{3} - \frac{\delta}{3m_1} - \frac{41\varepsilon_1}{2},\end{aligned}$$

$$\alpha_3^2 \geq 10n\alpha_2^2 + \frac{4n}{\pi q_2^2}\ln(2n(1+1/\varepsilon_2)) \geq 10n\alpha_1^2\omega(\log^2 k) + \frac{4n}{\pi q_2^2}\ln(2n(1+1/\varepsilon_2)).$$

Performing a meaningful concrete security analysis with the exact form of the above relations is almost impossible. To simplify the analysis, we ignore logarithmic factors. Also, we will assume that the parameters ε_1, ε_2 and δ can be chosen in a manner (say, $1/\text{poly}(n)$) such that they do not have much effect on the concrete security analysis. Using these and other reasonable simplifications, we have the following relations.

$$\begin{aligned}&q_1 = 2^{k/2}; \quad n = k^2; \\ &\alpha_1 = \alpha_2 = \alpha_3/\sqrt{n} = \alpha_3/k; \\ &\gamma = k/\alpha_1 = k^2/\alpha_3; \\ &k^{-c_2} = \zeta_1 = 1/m_1 = k^{-c}, \\ &q_2 = \sqrt{n}/\alpha_2 = n/\alpha_3; \\ &k^c = m_1 = n^{d_1+2d_2}.\end{aligned} \tag{7.16}$$

From (7.16), we have $c_2 = c = 2d_1 + 4d_2$. As mentioned earlier, following Theorem 4.1 of [BLP+13], algorithm W_1 for decLWE$_{k,m_1,q_1,\alpha_2}$ is constructed from the algorithm W_2 for

$\text{binLWE}_{n,m_1,q_1,\leq\sqrt{10n}\alpha_2}$. The reduction shows that W_1 is successful for almost all secrets and so we take $c_1 = 0$. Using $c_2 = c = 2d_1 + 4d_2$ and $c_1 = 0$ in (7.15), the overall tightness gap is obtained to be

$$n^{3.5+5d_1+10d_2}. \tag{7.17}$$

The tightness gap given by (7.17) is to be compared to the tightness gap of Regev's reduction given by (7.4). While the numerical values of the tightness gaps for the two reductions can be compared, it should be kept in mind that the problems being connected by the two reductions are different. The following theorem encapsulates the analysis of the tightness gap of the end-to-end reduction.

Theorem 76. *If there is an algorithm which solves* $\text{binLWE}_{n,m_2,q_2,\alpha_3}$*, where* $q_2 = n/\alpha_3$*, for a fraction* n^{-d_1} *of the possible secrets and has advantage* n^{-d_2}*, then there is an algorithm to solve* $\text{GapSVP}_{k^2/\alpha_3}$ *on a lattice of dimension* $k = \sqrt{n}$ *with The tightness gap given by* $n^{3.5+5d_1+10d_2}$.

Regev [Reg09] had described a cryptosystem where the public key is a collection of $n^{1+\epsilon}$ LWE samples and the secret key is $\mathbf{s} \in \mathbb{Z}_q^n$. A successful adversary against the scheme can distinguish between encryptions of 0 and 1 with an advantage of at least n^{-d} for some $d > 0$. It was shown in [Reg09] that a successful adversary against the cryptosystem can be used to obtain an algorithm for the average case decision LWE problem such that the algorithm is successful for a fraction $1/(4n^d)$ of all secrets with an advantage at least $1/(8n^d)$.

The problem $\text{binLWE}_{n,m_2,q_2,\alpha_3}$ would be used as a basis for proving the security of cryptosystems. We consider $\alpha_3 = 1/\sqrt{n} = 1/k$. The security of any such cryptosystem would be given by a reduction of the type given by Regev for his cryptosystem. Suppose $\mathfrak{C}$ is such a cryptosystem and that an adversary is successful in breaking $\mathfrak{C}$ if it can distinguish between encryptions of 0 and 1 with an advantage at least $1/n^d$ for some $d > 0$. Following the reduction of Regev for his cryptosystem, we assume that a successful adversary for $\mathfrak{C}$ can be used to build algorithm W_4 for $\text{binLWE}_{n,m_2,q_2,\alpha_3}$ such that W_4 is successful on a fraction $\approx n^{-d}$ of the secrets with advantage at least n^{-d}. This suggests $d_2 \approx d$. (A similar approximation was made in [CKMS16].) We further assume that $d_1 \approx d$. As a numerical example, consider $n = 2^{10}$. Aiming at 128-bit security, ζ_4 would be 2^{-128} and so for $n = 2^{10}$, $d = 12.8$. In this case, the tightness gap in (7.17) is 2^{1960}. In other words, the quantitative effect of the reduction is the following. If T is the time required to solve $\text{binLWE}_{n,m_2,q_2,\alpha_3}$ on a lattice of dimension 2^{10}, then there is an algorithm to solve GapSVP_γ for a lattice of dimension

$k = \sqrt{n} = 2^5$ and $\gamma = k^3 = 2^{15}$ which takes time $2^{1910}T$. So, the tightness gap is 2^{1960}. In comparison, for $n = 2^{10}$ and 128-bit security, the tightness gap in [KSSS22] has been obtained to be 2^{1646}.

Note that the dimension of the lattice for which GapSVP is to be solved is $\sqrt{n}$ where n is the dimension of the lattice for which binLWE is to be solved. Brakerski et al. [BLP+13] mention this point. Due to the drawback of the quadratic loss in the dimension, they mention as an open problem the task of obtaining a reduction where such a quadratic loss does not occur. In their words, this would constitute a "full dequantization" of Regev's reduction.

The issue of the tightness gap has not been considered in [BLP+13]. For the GapSVP to binLWE reduction to be meaningfully used to derive parameters for practical cryptosystems, the tightness gap needs to be taken into consideration. So, for a full dequantization of Regev's reduction which can also be used in practice, one needs a *tight* reduction which does not suffer the quadratic loss in the dimension.

7.5 Reducing $\text{binLWE}_{n,m_1,q,\leq\alpha}$ to $\text{binLWE}_{n,m_2,q,\alpha}$

Suppose there is an algorithm $\mathcal{A}$ which has advantage θ in solving $\text{binLWE}_{n,m_2,q,\alpha}$. Lemma 2.15 of [BLP+13] states that using $\mathcal{A}$, it is possible to construct an algorithm $\mathcal{B}$ which solves $\text{binLWE}_{n,m_1,q,\leq\alpha}$ with advantage at least $1/3$ where both m_1 and the runtime of $\mathcal{B}$ are $\text{poly}(m_2, 1/\theta, n, \log q)$. In [BLP+13], it was mentioned that the proof is standard and is based on Lemma 3.7 of [Reg09]. The following brief idea of the proof was provided.

> "The idea is to use Chernoff bound to estimate $\mathcal{A}$'s success probability on the uniform distribution, and then add noise in small increments to our given distribution and estimate $\mathcal{A}$'s behavior on the resulting distributions. If there is a gap between any of these and the uniform behavior, the input distribution is deemed non-uniform."

Below we provide the details of the proof based on the above idea and also work out the dependence of m_1 on m_2 and θ.

Lemma 77. *Let $\mathcal{A}$ be an algorithm which has advantage at least θ in solving $\text{binLWE}_{n,m_2,q,\alpha}$. Using $\mathcal{A}$, it is possible to construct an algorithm $\mathcal{B}$ which has advantage 1/3 in solving $\text{binLWE}_{n,m_1,q,\leq\alpha}$, where $m_1 = \mathfrak{k}m_2$ with $\mathfrak{k}$ satisfying $\mathfrak{k} \geq \max(32\ln 12, 8\ln(432m_2/\theta))/\theta^2$. Further, $\mathcal{B}$ invokes $\mathcal{A}$ a total of $\mathfrak{k}(1 + 36m_2/\theta)$ times.*

Proof. An input to $\mathcal{A}$ is a collection of samples $\mathcal{I}$ of size m_2. By "$\mathcal{I}$ is real" we will mean that the samples are drawn independently from $A_{q,s,\alpha}$, while by "$\mathcal{I}$ is random" we will mean that the samples are drawn independently and uniformly from $\mathbb{Z}_q^n \times \mathbb{T}$. The output of $\mathcal{A}$ is a bit. The advantage of $\mathcal{A}$ is

$$\mathrm{Adv}_{\mathcal{A}} \quad = \quad |\Pr[\mathcal{A}(\mathcal{I}) \Rightarrow 1 : \mathcal{I} \text{ is real}] - \Pr[\mathcal{A}(\mathcal{I}) \Rightarrow 1 : \mathcal{I} \text{ is random}]|. \tag{7.18}$$

Let $p_\star = \Pr[\mathcal{A}(\mathcal{I}) = 1 : \mathcal{I} \text{ is real}]$ and $p_\$ = \Pr[\mathcal{A}(\mathcal{I}) = 1 : \mathcal{I} \text{ is random}]$. For the sake of convenience of the analysis, we will assume that $p_\star > p_\$$, the other case being similar. Since it is given that $\mathrm{Adv}_{\mathcal{A}}$ is at least θ, we have

$$\theta \quad \leq \quad p_\star - p_\$. \tag{7.19}$$

The construction of $\mathcal{B}$ using $\mathcal{A}$ is shown in Algorithm 23. The input to $\mathcal{B}$ is a collection of samples $\mathcal{J}$ of size m_1 where $m_1 = km_2$. By "$\mathcal{J}$ is real" we will mean that the samples are drawn independently from $A_{q,\mathbf{s},\beta}$ for some unknown $\beta \leq \alpha$, while by "$\mathcal{J}$ is random" we will mean that the samples are drawn independently and uniformly from $\mathbb{Z}_q^n \times \mathbb{T}$.

Steps 2 to 4 of Algorithm 23 compute an estimate $\hat{p}_\$$ of $p_\$$. From the additive form of the Chernoff-Hoeffding bound 2.6.1, we have

$$\Pr[p_\$ - \theta/4 \leq \hat{p}_\$ \leq p_\$ + \theta/4] \geq 1 - 2\exp(-2\mathfrak{k}(\theta/4)^2). \tag{7.20}$$

Consider the set Z defined in Step 6 and let $t = \#Z$. Note that $t = m_3^2$. The loop from Step 7 to 18 runs for t steps. For $i = 1, \ldots, t$, let p_i^{real} (resp. p_i^{rnd}) be the value of p computed at Step 14 in the i-th iteration of the loop when the input $\mathcal{J}$ is real (resp. random).

The loop in Steps 9 to 12 adds a certain amount of noise to the samples in $\mathcal{J}$ to obtain $\mathcal{J}'$. If $\mathcal{J}$ is random, then $\mathcal{J}'$ is also random and the inputs $\mathcal{J}_1, \ldots, \mathcal{J}_k$ on which $\mathcal{A}$ is invoked are also random. By the additive form of the Chernoff-Hoeffding bound, we have

$$\Pr[p_\$ - \theta/4 \leq p_i^{\mathrm{rnd}} \leq p_\$ + \theta/4] \geq 1 - 2\exp(-2\mathfrak{k}(\theta/4)^2). \tag{7.21}$$

For the case when $\mathcal{J}$ is real, we follow an argument from the proof of Lemma 3.7 of [Reg09]. In this case, the samples in $\mathcal{J}$ are from $A_{q,\mathbf{s},\beta}$, for some unknown $\beta \leq \alpha$. In other words, each element of $\mathcal{J}$ is a pair of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle / q + e)$, where e is drawn from Ψ_β. Step 11 converts such a pair to $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle / q + e + \varepsilon)$, where ε is drawn from Ψ_γ. This creates a pair

$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle/q + e')$, where $e' = e + \varepsilon$ and so, e' follows $\Psi_{\sqrt{\beta^2+\gamma}}$. Consider the smallest γ such that $\gamma \geq \alpha^2 - \beta^2$ and so $\gamma \leq \alpha^2 - \beta^2 + m_3^{-2}\alpha^2$. Suppose this γ is considered in the ℓ-th iteration of the loop in Steps 7 to 18. Let $\alpha' = \sqrt{\beta^2+\gamma}$ so that $\alpha \leq \alpha' \leq \sqrt{\alpha^2 + m_3^{-2}\alpha^2} \leq (1 + m_3^{-2})\alpha$. By Claim 2.2 of [Reg09], the statistical distance between Ψ_α and $\Psi_{\alpha'}$ is at most $9m_3^{-2}$. Consequently, the statistical distance between m_2 samples from Ψ_α and $\Psi_{\alpha'}$ is at most $9m_2m_3^{-2}$. So, in the ℓ-th iteration of the loop in Steps 7 to 18, for $j = 1, \ldots, \mathfrak{k}$, the statistical distance between $\mathcal{J}_j$ and m_2 samples from $A_{q,\mathbf{s},\alpha}$ is at most $9m_2m_3^{-2}$.

Let $\hat{p}_\star$ be the probability that $\mathcal{A}$ outputs 1 when the input consists of m_2 samples from a distribution whose statistical distance from $A_{q,\mathbf{s},\alpha}$ is at most $9m_2m_3^{-2}$. So, $|\hat{p}_\star - p_\star| \leq 9m_2m_3^{-2}/2$. In the ℓ-th iteration, for $j = 1, \ldots, \mathfrak{k}$, the probability that $\mathcal{A}$ outputs 1 on input $\mathcal{J}_j$ is $\hat{p}_\star$. Let $\epsilon_1 = \theta/4 - 9m_2m_3^{-2}/2$. By the additive form of the Chernoff-Hoeffding bound we have

$$\Pr[\hat{p}_\star - \epsilon_1 \leq p_\ell^{\mathrm{real}} \leq \hat{p}_\star + \epsilon_1] \geq 1 - 2\exp(-2\mathfrak{k}\epsilon_1^2). \tag{7.22}$$

Combining (7.22) with $|\hat{p}_\star - p_\star| \leq 9m_2m_3^{-2}/2$, we have

$$\Pr[p_\star - \epsilon_1 - 9m_2m_3^{-2}/2 \leq p_\ell^{\mathrm{real}} \leq p_\star + \epsilon_1 + 9m_2m_3^{-2}/2] \geq 1 - 2\exp(-2\mathfrak{k}\epsilon_1^2). \tag{7.23}$$

So,

$$\Pr[p_\star - \theta/4 \leq p_\ell^{\mathrm{real}} \leq p_\star + \theta/4] \geq 1 - 2\exp(-2\mathfrak{k}(\theta/4 - 9m_2m_3^{-2}/2)^2). \tag{7.24}$$

We define two sets of events. Suppose the input $\mathcal{J}$ to $\mathcal{B}$ is random. For $i = 1, \ldots, t$, let E_i be the event that the $|p_i^{\mathrm{rnd}} - \hat{p}_\$| > \theta/2$, i.e., the if-condition at Step 15 is satisfied in the i-th iteration on random input. Next, suppose that the input $\mathcal{J}$ to $\mathcal{B}$ is real. For $i = 1, \ldots, t$, let F_i be the event that the $|p_i^{\mathrm{real}} - \hat{p}_\$| > \theta/2$, i.e., the if-condition at Step 15 is satisfied in the i-th iteration on real input.

We consider the probability of $\overline{E}_i$. Let G_1 be the event $|\hat{p}_\$ - p_\$| \leq \theta/4$ and H_i be the event $|p_i^{\mathrm{rnd}} - p_\$| \leq \theta/4$. Note that G_1 and H_i are independent. Further, $G_1 \wedge H_i$ implies $\overline{E}_i$ and so using (7.20) and (7.21), we obtain

$$\begin{aligned} \Pr[\overline{E}_i] &\geq \Pr[G_1 \wedge H_i] \geq (1 - 2\exp(-2\mathfrak{k}(\theta/4)^2))^2 \geq 1 - 4\exp(-2\mathfrak{k}(\theta/4)^2) \\ &= 1 - \delta_1 \end{aligned} \tag{7.25}$$

where $\delta_1 = 4\exp(-2\mathfrak{k}(\theta/4)^2)$. Using $\mathfrak{k} \geq 8\ln(432m_2/\theta)/\theta^2$ and $m_3^2 = 36m_2/\theta$, we have

$$t\delta_1 \;=\; 4m_3^2\exp(-2\mathfrak{k}(\theta/4)^2) = \frac{144m_2}{\theta}\exp(-2\mathfrak{k}(\theta/4)^2) \leq 1/3. \tag{7.26}$$

Next, we consider the probability of F_ℓ. Let G_2 be the event $|p_\ell^{\text{real}} - p_\star| < \theta/4$. Note that G_1 and G_2 are independent events. We have G_2 to be the event $p_\star - \theta/4 \leq p_\ell^{\text{real}} \leq p_\star + \theta/4$; and G_1 to be the event $p_\$ - \theta/4 \leq \hat{p}_\$ \leq p_\$ + \theta/4$ which is equivalent to $-p_\$ + \theta/4 \geq -\hat{p}_\$ \geq -p_\$ - \theta/4$. So, if G_1 and G_2 both hold, we have $p_\star - p_\$ - \theta/2 \leq p_\ell^{\text{real}} - \hat{p}_\$$. Using $p_\star - p_\$ \geq \theta$, the last condition shows that $\theta/2 \leq p_\ell^{\text{real}} - \hat{p}_\$$ and so F_ℓ holds. This shows that $G_1 \wedge G_2$ implies F_ℓ and using (7.20) and (7.24), we obtain

$$\begin{aligned}
\Pr[F_\ell] \geq \Pr[G_1 \wedge G_2] \;&\geq\; (1 - 2\exp(-2\mathfrak{k}(\theta/4)^2)) \times \\
&\quad\; (1 - 2\exp(-2\mathfrak{k}(\theta/4 - 9m_2m_3^{-2}/2)^2)) \\
&\geq\; 1 - 2\exp(-2\mathfrak{k}(\theta/4)^2) - 2\exp(-2\mathfrak{k}(\theta/4 - 9m_2m_3^{-2}/2)^2) \\
&=\; 1 - \delta_2
\end{aligned} \tag{7.27}$$

where $\delta_2 = 2\exp(-2\mathfrak{k}(\theta/4)^2) + 2\exp(-2\mathfrak{k}(\theta/4 - 9m_2m_3^{-2}/2)^2)$. Using $m_3 = 6(m_2/\theta)^{1/2}$, we have $\theta/4 - 9m_2m_3^{-2}/2 = \theta/8$ so, $\delta_2 = 2\exp(-2\mathfrak{k}(\theta/4)^2)$ $+2\exp(-2\mathfrak{k}(\theta/8)^2) \leq 4\exp(-2\mathfrak{k}(\theta/8)^2)$. Using $\mathfrak{k} \geq 32\ln 12/\theta^2$, we have

$$\begin{aligned}
\delta_2 \;&=\; 2\exp(-2\mathfrak{k}(\theta/4)^2) + 2\exp(-2\mathfrak{k}(\theta/4 - 9m_2m_3^{-2}/2)^2) \leq 4\exp(-2\mathfrak{k}(\theta/8)^2) \\
&\leq\; 1/3.
\end{aligned} \tag{7.28}$$

We now compute the advantage of $\mathcal{B}$.

$$\begin{aligned}
\text{Adv}_{\mathcal{B}} \;&=\; |\Pr[\mathcal{B}(\mathcal{J}) \Rightarrow 1 : \mathcal{J} \text{ is real}] - \Pr[\mathcal{B}(\mathcal{J}) \Rightarrow 1 : \mathcal{J} \text{ is random}]| \\
&=\; |\Pr[F_1 \vee \cdots \vee F_t] - \Pr[E_1 \vee \cdots \vee E_t]| \\
&\geq\; |\Pr[F_\ell] - \Pr[E_1 \vee \cdots \vee E_t]| \\
&\geq\; |\Pr[F_\ell] - \sum_{i=1}^{t}\Pr[E_i]| \\
&\geq\; |1 - \delta_2 - t\delta_1| \quad \text{(from (7.25) and (7.27))} \\
&\geq\; \frac{1}{3} \quad \text{(from (7.26) and (7.28))}.
\end{aligned} \tag{7.29}$$

In Algorithm 23, $\mathcal{A}$ is called $\mathfrak{k}$ times in Step 4 and in each iteration of the loop in Steps 7

to 18, $\mathcal{A}$ is invoked $\mathfrak{k}$ times in Step 14. The loop in Steps 7 to 18 runs for $t = m_3^2$ iterations and so the total number of times $\mathcal{B}$ invokes $\mathcal{A}$ is $\mathfrak{k}(m_3^2 + 1) = \mathfrak{k}(1 + 36m_2/\theta)$. □

Algorithm 23 Construction of a distinguisher $\mathcal{B}$ for $\mathrm{binLWE}_{n,m_1,q,\leq\alpha}$ using a distinguisher $\mathcal{A}$ for $\mathrm{binLWE}_{n,m_2,q,\alpha}$. In the algorithm, θ is a known lower bound on the advantage of $\mathcal{A}$.

1: **function** $\mathcal{B}(\mathcal{J})$
2: let $\mathcal{S}$ be a collection of m_1 samples drawn independently and uniformly from $\mathbb{Z}_p^n \times \mathbb{T}$;
3: partition $\mathcal{S}$ as $\mathcal{S} = \cup_{i=1}^{\mathfrak{k}} \mathcal{S}_i$, such that $\#\mathcal{S}_i = m_2$, $i = 1, \ldots, \mathfrak{k}$;
4: let $\hat{p}_\$ = (\mathcal{A}(\mathcal{S}_1) + \cdots + \mathcal{A}(\mathcal{S}_\mathfrak{k}))/\mathfrak{k}$;
5: $m_3 \leftarrow 6(m_2/\theta)^{1/2}$;
6: let Z be the set of all integer multiples of $m_3^{-2}\alpha^2$ in the range $(0, \alpha^2]$;
7: **for** γ in Z **do**
8: $\mathcal{J}' \leftarrow \emptyset$;
9: **for** $(\mathbf{a}, e) \in \mathcal{J}$ **do**
10: sample ε from $\Psi_{\sqrt{\gamma}}$;
11: $\mathcal{J}' \leftarrow \mathcal{J}' \cup \{(\mathbf{a}, e + \varepsilon)\}$;
12: **end for**
13: partition $\mathcal{J}'$ as $\mathcal{J}' = \cup_{i=1}^{k} \mathcal{J}_i$, such that $\#\mathcal{J}_i = m_2$, $i = 1, \ldots, k$;
14: let $p = (\mathcal{A}(\mathcal{J}_1) + \cdots + \mathcal{A}(\mathcal{J}_\mathfrak{k}))/\mathfrak{k}$;
15: **if** $|p - \hat{p}_\$| > \theta/2$ **then**
16: return **1;**
17: **end if**
18: **end for**
19: return **0;**
20: **end function**.

7.6 Conclusion

In this chapter, we performed the concrete security analysis of the tightness gap in the classical reduction of the shortest vector problem to the LWE problem given by Brakerski et al. [BLP+13]. In the previous chapters, we already discussed the tightness gap of quantum reductions on general lattice, ring, or module lattices and pointed out that the tightness gap is huge in all cases. Our analysis shows that the tightness gap of the classical reduction by Brakerski et al. [BLP+13] is more than that of Regev's original quantum reduction. In the following chapter, we comment on the devastating effects of the reduction of the high tightness gap and related issues in more detail.

Chapter 8

Analysis of Concrete Security

8.1 Introduction

In the previous chapters, our main focus has been on conducting concrete security analyses for various reductions used in lattice-based cryptography. We have examined the tightness gap of these reductions, ranging from LWE to ring-LWE and module-LWE. The concrete analysis of these reductions is of paramount importance in the context of lattice-based cryptography. It provides a clear understanding of the practical security implications of cryptographic constructions that rely on these hardness assumptions. By assessing the tightness gap, we gain insights into how well the theoretical security reductions hold in real-world scenarios. The practical usability of lattice-based cryptographic schemes heavily relies on the tightness of these reductions. If the tightness gap is too large, it indicates that the theoretical security guarantees might not directly translate to practical security. In such cases, the cryptographic applications built upon these assumptions might have significant limitations in terms of efficiency and security. By performing concrete security analyses and examining the tightness gap, we can identify potential weaknesses and limitations in lattice-based cryptography. This information helps researchers and practitioners make informed decisions about the selection and design of cryptographic schemes, ensuring that they are robust and secure in practical settings. Moreover, understanding the tightness of reductions aids in the quest for more efficient and secure lattice-based cryptographic constructions, making progress toward achieving post-quantum security in the age of quantum computing.

The concept of tightness gap in concrete analysis of reductions allows us to quantitatively measure the looseness of a mathematical reduction. In the context of lattice-based cryptography, there are two main types of reductions: worst-case to average-case reductions for hard lattice problems and reductions from hard lattice problems to decisional-LWE.

The worst-case to average-case reduction is a self-reduction technique used to establish the hardness of solving random instances of a lattice problem. This reduction shows that if we can efficiently solve random instances of a hard lattice problem, then we can efficiently solve worst-case instances of the same problem.

On the other hand, the reduction from the worst-case lattice problem to decisional-LWE is a critical step in lattice-based cryptographic constructions. It establishes the security of LWE-based cryptosystems by showing that the hardness of the worst-case lattice problem implies the security of the LWE problem on which the cryptographic construction is based.

The breakthrough in lattice-based cryptography was first established by Regev in his seminal paper [Reg09], where he presented the result in an asymptotic setting. Subsequent works like [Pei09, BLP+13, LPR13, LS15, PRS17] and others followed the same approach, considering asymptotic treatments in their results.

However, real-world cryptographic scenarios operate with specific sets of parameters. Cryptosystems in practice work with practical values, such as a lattice with a dimension of 1024 or ensuring 128-bit security.

The concrete analysis of reductions, focusing on the tightness gap, becomes essential when evaluating lattice-based cryptographic constructions in practical terms. Understanding the practical implications of these reductions with real-world parameter choices helps in building efficient and secure post-quantum cryptographic schemes that are suitable for modern applications and technologies.

In the previous chapters, we have established that the tightness gap of each reduction is significantly large. Now, let us delve into the devastating effects of this high tightness gap, both in general and in specific cases. The concept of tightness of reduction is often overlooked and not given the importance it deserves, even though it is a crucial aspect of practical security.

The pioneers in the field of practice-based provable security, such as Bellare [Bel97] and Rogaway [BDJR97], have emphasized the need to analyze security from a practical viewpoint. While cryptographic protocols may offer theoretical assurances in the framework of asymptotic settings, it is not sufficient to solely rely on these theoretical guarantees. After all, cryptographic protocols are designed for real-world practical use, where specific parameter choices and efficiency play a crucial role.

Concrete security is a paradigm that addresses this concern, as it looks at security in concrete terms rather than asymptotic settings. In practical terms, it assesses the real-world security guarantees of cryptographic schemes, taking into account actual parameter choices and computational efficiency.

When the tightness gap of a reduction is high, it implies that the reduction does not provide meaningful security guarantees in practical scenarios. This has significant consequences

for the security of cryptographic constructions based on these reductions. It can lead to inefficiency in terms of key sizes, computational overhead, and other resources required for secure communications.

Therefore, understanding the tightness gap and its effects on the security of lattice-based cryptographic schemes is crucial for building practical, efficient, and secure post-quantum cryptographic systems that can withstand real-world threats. It highlights the importance of considering concrete security in the design and analysis of cryptographic protocols to ensure their effectiveness in real-world applications.

8.2 Practice Oriented Provable Security

Goldwasser and Micali [GM84] introduced the idea of provable security (Though, many researchers credit Shannon [Sha49] as he introduced the ideas of perfect secrecy and how to prove it much earlier). Though it was first meant for public key cryptography, later it is applied to different branches of cryptography. The provable security of a cryptographic scheme simply means that the security of that scheme can be proven. Here proof is a mathematical proof. A cryptographic system is provably secure if the security requirements can be formally stated in an adversarial model, where the adversary has a specific power of computation. This proof is generally called a "reduction". For example, the security of the well-known Diffie-Hellman [DH76] protocol for cryptographic key exchange is based on the hardness of computational Diffie-Hellman(CDH) assumption, and the security of RSA crypto-system [RSA78, BV98] is based on the RSA assumption. But it must be added that the reduction or the proof is a theoretical certificate of the security of the protocols. From the perspective of the adversary to break the hard problem is to break the cryptographic protocol. The hardness of the problem is guaranteed for a specific set of parameters of the problem. An instance of a cryptographic protocol is linked with an instance of the hard problem. An instance of the hard problem may not be hard at all. In this case, the cryptographic protocol can easily be broken by the adversary. For example, an RSA protocol with a small public exponent can be easily broken [FKM+06]. So it is of utmost importance to choose the parameters of the cryptographic protocol such that the adversary will have a hard instance of the underlying hard problem.

Another major part of practice-oriented provable security is the analysis of the tightness gap of reductions. The tightness gap analysis has been the main focus of this thesis. This notion of provable security tries to capture the quantitative nature of the security of any

protocol.

Let us recall the definition of the tightness gap of a reduction. A reduction is an algorithm $\mathcal{A}$ to reduce one problem $\mathcal{P}$ to another problem $\mathcal{Q}$ and there is an oracle to $\mathcal{O}$ to solve the problem $\mathcal{Q}$. Let $\mathcal{A}$ given access to the oracle $\mathcal{O}$ solve a problem $\mathcal{P}$ in time T with success probability P_S. Further, suppose $\mathcal{O}$ takes time T' and has success probability P'_S to solve the problem $\mathcal{Q}$. Thus the reduction algorithm $\mathcal{A}$ reduces problem $\mathcal{P}$ to problem $\mathcal{Q}$. Then the tightness gap of the reduction is $(T \cdot P'_S)/(T' \cdot P_S)$. The reduction is said to be tight if the tightness gap is 1 (or small) and is said to be loose if the tightness gap is 'large'. Implicitly, we need tight reductions.

A tight security reduction is evidence of confidence in the security of a protocol. If we eliminate the attacks outside the security model (such as side-channel attacks, duplicate-signature key selection attacks, etc.), it is guaranteed that the task of the adversary to break the protocol is at least as hard as solving a certain well-studied mathematical problem.

In this thesis, we rigorously analyzed the major reductions related to lattice-based cryptography. This analysis gave us different views to look at the reductions. Firstly, the tightness gap is huge in all the reductions whether the reduction is based on a general lattice or structured lattices like ring lattice or module lattice. Secondly, a thorough calculation shows that the concrete value of the approximation parameter (γ) of the hard lattice problem (SIVP) used in the reductions is unlikely to be NP hard [GG00]. It was simply hidden in the asymptotic descriptions. In the same way, it is found that the reduction puts a stricter bound on the value of the LWE modulus parameter q, which has not been captured by any previous result or the cryptographic schemes. We elaborate on these points in the subsequent sections. We also discuss the disadvantage of using structured lattices over general ones and also the effect of the quantum part of the reduction from the point of tightness gap.

8.2.1 Tightness Gap of different lattice-based reductions

The structure of the reductions that are discussed in this thesis is nested sequences of intermediate reductions. This gives rise to two difficulties from a practice-oriented perspective. In the first place, the tightness gaps multiply from one reduction to the next. If algorithm $\mathcal{A}$ calls on algorithm $\mathcal{B}$ m times, and $\mathcal{B}$ calls on $\mathcal{C}$ n times, then there are mn calls on $\mathcal{C}$ from $\mathcal{A}$.

Here we study the numerical values of tightness gaps for each of the theorems regarding tightness gaps in previous chapters and try to find the practical implications of it. In all cases,

the dimension of the lattices is set to be 1024 as this value is recommended in most of the lattice-based cryptosystems [DKRV19, ABD+09, ABD+19, LLJ+19, AAB+19, BBF+19a]. We tabulate the values of tightness gaps for the different parameters of decisional LWE on which the cryptosystem is based. Next, we try to figure out the meaning of having a large value of tightness gap concerning concrete security.

In Chapter 4, we evaluated tightness gap of the reduction from $\text{GIVP}_{\gamma}^{\phi(n)}$ to DLWE_{ac}. As per our previous notation, $\mathcal{P}$ is $\text{GIVP}_{\gamma}^{\phi(n)}$ and $\mathcal{Q}$ is DLWE_{ac}. Algorithm $\mathcal{A}$ is the reduction from $\mathcal{P}$ to $\mathcal{Q}$. The tightness gap of this reduction came out to be around $n^{11}\cdot(\delta_1{\delta_2}^2)^{-4}$ which we get from the equation (4.13). This simply means that if we have a (δ_1, δ_2) distinguisher that solves DLWE_{ac}, then we can devise a quantum algorithm to solve the approximate version of GIVP i.e., $\text{GIVP}_{\gamma}^{\phi(n)}$ where L is an n dimensional lattice over $\mathbb{R}^n$, $\alpha \in (0,1)$, $\epsilon < 1/10$, $\phi(n) = \sqrt{2n}\cdot\eta_{\epsilon}(L)/\alpha$ and $\gamma = 2\sqrt{n}$. As stated earlier, let us consider $n = 2^{10}$ and suppose $\delta_1 = 2^{-\beta_1}$, $\delta_2 = 2^{-\beta_2}$. Here β_1 and β_2 are the positive integers such that δ_1 and δ_2 are non-negligible functions of n. The tightness gap becomes approximately $n^{11}\cdot(\delta_1{\delta_2}^2)^{-4} = 2^{110+4\beta_1+8\beta_2}d$. We tabulate the values of the tightness gap for different practical values of β_1 and β_2 in Table 8.1

β_1	β_2	Tightness Gap
32	32	2^{494}
32	64	2^{750}
64	32	2^{622}
64	64	2^{878}
64	128	2^{1390}
128	64	2^{1134}
128	128	2^{1646}

Table 8.1: Tightness Gap for Theorem 42.

Similarly in Chapter 5, we evaluated the tightness gap of the reduction from $M\text{-SIVP}_{\gamma}$ to $\text{module-DLWE}_{q,r_0}$ and the value of the tightness gap is around $(n^{20}qd^{21})(\delta_1\delta_2^5)^{-2}l^2$. If we recall the Theorem 51, we get that if we have a (δ_1, δ_2) distinguisher that solves $\text{module-DLWE}_{q,r_0}$, we can solve $M\text{-SIVP}_{\gamma}$ through a quantum algorithm, where the lattice problem is considered over module lattices, and the decision LWE problem is considered for modules. Here the module under consideration is M that has rank d and dimension n, such that the module lattice will have dimension $n\cdot d$ and $n\cdot d$ is also the dimension for DLWE problem. Here q is the modulus for the DLWE problem and l is the number of samples that the

module-DLWE requires. Let us take $n{\cdot}d = 2^{10}$, $\delta_1 = 2^{-\beta_1}$, $\delta_2 = 2^{-\beta_2}$, $l = n{\cdot}d$ as l is generally of $O(n{\cdot}d)$ and $q = (n{\cdot}d)^2$ as q is a prime number which ranges from $(n{\cdot}d)^2$ to $2(n{\cdot}d)^2$. The tightness gap becomes approximately $(n^{24}d^{25})(2^{2\beta_1}2^{10\beta_2}) = 2^{240+2\beta_1+10\beta_2}d$. We tabulate the values of tightness-gap for different values of β_1, β_2 such that δ_1, δ_2 becomes non-negligible and different values of d in Table 8.2.

β_1	β_2	d	Tightness Gap
32	32	2	2^{625}
32	64	2	2^{945}
64	32	2	2^{689}
64	64	2	2^{1009}
64	128	2	2^{1649}
128	64	2	2^{1137}
128	128	2	2^{1777}

Table 8.2: Tightness Gap with rank 2 module for Theorem 51.

β_1	β_2	d	Tightness Gap
32	32	2	2^{626}
32	64	2	2^{946}
64	32	2	2^{690}
64	64	2	2^{1010}
64	128	2	2^{1650}
128	64	2	2^{1138}
128	128	2	2^{1778}

Table 8.3: Tightness Gap with rank 4 module for Theorem 51.

Next, in Chapter 6, we investigated the tightness gap of the reduction from K-SIVP$_\gamma$ to ring-DLWE$_{q,r_0}$. The evaluated tightness gap is approximately $10^{86}{\cdot}n^{c+67}{\cdot}l^{38}{\cdot}(\delta_1\delta_2^5)^{-26}$. According to Theorem 70, if we have a (δ_1, δ_2) distinguisher that solves ring-DLWE$_{q,r_0}$, then we can solve K-SIVP$_\gamma$ through a quantum algorithm, where the lattice problem is considered over ideal lattices, and the decision LWE problem is considered for rings. Here the ring under consideration is the ring of algebraic integer of number field K that has dimension n. The ideal lattice is also n dimensional. Here l is the number of samples that the ring-DLWE requires. Let us take $n = 2^{10}$, $\delta_1 = 2^{-\beta_1}$, $\delta_2 = 2^{-\beta_2}$, $l = n$ as l is of $O(n)$ and $c = 1$ as this is the minimum value for c. The tightness gap becomes approximately

$10^{86} \cdot 2^{1060} \cdot (2^{26\beta_1} 2^{130\beta_2}) \approx 2^{1345+26\beta_1+130\beta_2}$. We tabulate the values of the tightness gap in Table 8.4 for different values of β_1 and β_2 such that δ_1, δ_2 becomes non-negligible.

β_1	β_2	Tightness Gap
32	32	2^{6337}
32	64	2^{10497}
64	32	2^{7169}
64	64	2^{11329}
64	128	2^{19649}
128	64	2^{12993}
128	128	2^{21313}

Table 8.4: Tightness Gap for Theorem 70.

Next, in Chapter 7, tightness gap of the reduction from GapSVP_γ to DLWE_{ac} is evaluated. The evaluated tightness gap is seen to be $n^{3.5} \cdot ({\delta_1}^{-5} {\delta_2}^{-10})$. As per Theorem 76, we see that if we have a (δ_1, δ_2) distinguisher that solves DLWE_{ac}, then we can construct a quantum algorithm to solve an approximate version of GapSVP. Like before, we take $n = 2^{10}$, $\delta_1 = 2^{-\beta_1}$, $\delta_2 = 2^{-\beta_2}$. The tightness gap becomes approximately $2^{35} \cdot (2^{5\beta_1} 2^{10\beta_2}) = 2^{(35+5\beta_1+10\beta_2)}$. We tabulate the values of tightness-gap for different values of β_1 and β_2 in Table 8.5

β_1	β_2	Tightness Gap
32	32	2^{515}
32	64	2^{835}
64	32	2^{675}
64	64	2^{995}
64	128	2^{1635}
128	64	2^{1315}
128	128	2^{1955}

Table 8.5: Tightness Gap for Theorem 76.

From the tables 8.1,8.2,8.4,8.5, it is evident that the tightness gap is huge for all the reductions that we have discussed till now. Let G be the tightness gap of a reduction algorithm from problem $\mathcal{P}$ to problem $\mathcal{Q}$. Here, we analyze how the high tightness gap implies the security of a reduction. In our analysis, $\mathcal{P}$ is different versions of SIVP_γ problem according to the theorems and $\mathcal{Q}$ is different variations of DLWE that we have considered in the theorems. Again, $\mathcal{Q}$ is the problem that the cryptographic construction is based on.

Suppose, we have an oracle $\mathcal{O}$ to solve the problem $\mathcal{Q}$ in time T_1. If the tightness gap of the reduction is G, then the time taken to solve $\mathcal{P}$ through the reduction is $G{\cdot}T_1$. We assume that the fastest known algorithm ever devised to solve $\mathcal{P}$ for the parameters that we consider, is T_2. So, the relation $G{\cdot}T_1 \geq T_2$ or $T_1 \geq T_2/G$ is implicit if we assume that the algorithm to solve $\mathcal{P}$ through reduction will not out-perform the known fastest algorithm for $\mathcal{P}$. Exploiting the relation $T_1 \geq T_2/G$, we can choose parameters so that $T_1 \geq 2^{128}$, satisfies. Problem $\mathcal{P}$ is a well-studied hard lattice problem hence for our chosen parameters T_2 is exponential. So this makes $T_2/G \geq 2^{128}$ when G is reasonably small or the reduction is fairly tight. This implies that $T_1 \geq 2^{128}$. This mathematical relation puts constraints over the lower bound on the time of the algorithm that tries to solve $\mathcal{Q}$ or equivalently the problem on which the cryptographic construction is based. Thus it translates the hardness of the problem $\mathcal{P}$ to the problem $\mathcal{Q}$ in a concrete sense. Here, our implicit assumption is that G is reasonably small.

Now, we see what happens when we have reductions where the tightness gap or G is huge. We take $n = 2^{10}$ as before and calculate the lower bound of T_1 based on the known best algorithm for problem $\mathcal{P}$ or the SIVP_γ problem. Here T_2 is the time taken to solve SIVP_γ. We have discussed reductions on lattices where lattices are Euclidean lattices, ideal lattices, and module lattices. The hardness of SIVP_γ is different on different lattices. Rank 1 module lattice with n dimensional number field K is an n dimensional ideal lattice and rank n module lattice with 1-dimensional number field K is an n dimensional Euclidean lattice. So, module lattices carry both the characteristics of Euclidean and ideal lattices. In all the practical applications over module lattices rank of the module is taken as integers very close to 1. Hence these lattices are closer to ideal lattices than Euclidean lattices. Thus, for the analysis, we can focus on ideal lattices and Euclidean lattices only. So we omit module lattice for the time being as it is taken care of by ideal lattice approximately.

It is conjectured that SIVP_γ is at least as hard as SVP_γ for Euclidean lattices. But for ideal lattices, SIVP_γ and SVP_γ are equivalent with respect to hardness. SVP_γ on Euclidean lattice is presumably harder than SVP_γ on ideal lattices. If T_2 is the time to solve SIVP_γ on Euclidean lattice, T_2 is greater than the time to solve SVP_γ on Euclidean lattice. So, T_2 is greater than the time to solve SVP_γ on an ideal lattice. This implies that T_2 is greater than the time to solve SIVP_γ on an ideal lattice. So to make the analysis concrete we assume that the lower bound T_2 is the minimum time to solve SVP_γ or the best-known algorithm available for SVP_γ on Euclidean lattices.

The fastest classical algorithm for $\text{SVP}\gamma$, as shown by Becker et al. [BDGL16], has a

running time of approximately $2^{0.292n+o(n)}$, while the most efficient quantum SVPγ algorithm, developed by Laarhoven, Mosca, and Pol [LMvdP15], has a heuristic running time of around $2^{0.268n+o(n)}$. For the purpose of our analysis, we consider $o(n) = 50$, assuming it has a negligible impact on the exponent term. In the context of practical cryptosystems, where $n = 1024$, we can calculate the running times for both classical and quantum algorithms. For the classical algorithm, T_2 (time to solve SVP$_\gamma$) is greater than 2^{349}, and for the quantum algorithm, T_2 is greater than 2^{324}. Considering the smallest gap(G) value obtained, which is 2^{494}, we can determine the corresponding values of T_1 (time to break the DLWE problem). For the classical algorithm, T_1 is greater than 2^{-145}, and for the quantum algorithm, T_1 is greater than 2^{-170}. This implies that the lower bounds for breaking the DLWE problem using these algorithms are approximately 2^{-145} for the classical algorithm and 2^{-170} for the quantum algorithm.

It should be kept in mind that the above calculation is based on the minimum values obtained for G. The practical values are always greater than the minimum value by a big margin. In those cases, the lower bound on T_1 will have a much bigger negative exponent. This analysis reflects that the theoretical guarantee of the security of the cryptosystem based on the hardness of the hard lattice problem does not provide any meaningful assurance.

An alternative concrete security analysis by F. Gates [Gat18] tells us that a reasonable lower bound on T_1 can be achieved by increasing the value of n. F. Gates' work was performed on Euclidean lattices. If we try to incorporate this theory into practice, we would get a humongous value for G. An example of this claim is the following. If $n \approx 2^{17.5}$, then we have $q > 2^{85}, \gamma = 2^k > 2^{94}, n/k < 1970, G \approx 2^{1715}$, leading to a lower bound for T_1 of approximately 2^{255}.

However, there are many difficulties with choosing n so large. The practical needs are for restricted values of n. This huge value of n for a high lower bound of T_1 would make most applications useless. The main reason behind moving applications from Euclidean lattices to ideal lattices was efficiency. At this huge value of n, the question of efficiency using ideal lattices becomes irrelevant. One of the most interesting parts of the main reduction is that it is quantum. We already mentioned that the quantum part of the reduction requires at least $3n^2$ logical qubits. This is true for all the reductions we have discussed till now. Now, when we increase n to $2^{17.5}$, the required logical qubits become approximately 2^{36} which is roughly 20 million times as many as Shor's [Sho99] algorithm needs to factor a 2048-bit RSA modulus. So, the recommendations of F. Gates [Gat18] are fairly impractical from the point of view of this analysis.

8.2.2 The effect of γ

In all the reductions γ is the parameter for the approximate SIVP problem. The approximate SIVP is considered one of the hard lattice problems. The value of γ plays a major role in finding out the hardness of SIVP. According to Goldreich and Goldwasser [GG00], it is unlikely that the Shortest Independent Vectors Problem (SIVP_γ) is NP-hard when γ is greater than the lattice dimension n.

It is concluded by [GG00] that approximating the SVP within a factor of $\sqrt{n}$ is known to be in the complexity class NP $\cap$ coAM. However, it also implies that it is unlikely for these problems to be NP-hard when approximated to within a factor of $\sqrt{n}$. In the context of Euclidean lattices, it is known that the $\text{SIVP}_{\sqrt{n}\gamma}$ can be reduced in polynomial time to the SVP_γ [MG02]. However, for ideal lattices, the hardness of SIVP and SVP becomes equivalent. Therefore, in the reductions, it becomes crucial to concretely determine the value of γ in order to gain confidence in the hard lattice problems to be in the NP-hard class. The choice of γ directly influences the security and complexity guarantees of the cryptographic schemes, making it essential to carefully consider this parameter in practical applications.

Module Lattice: To have confidence in the security of a module-DLWE-based cryptosystem, we want to be sure that unless (δ_1, δ_2) is negligible, there is no efficient (δ_1, δ_2)-distinguisher for module-$\mathsf{DLWE}_{q,\mathfrak{r}}$, where q (the modulus) and $\mathfrak{r}$ (the distribution width) are parameters of our cryptosystem. We want the M-SIVP_γ problem that reduces to module-$\mathsf{DLWE}_{q,\mathfrak{r}}$ with this choice of δ_1, δ_2 to be hard. According to Theorem 51, the approximation factor γ in the SIVP is $\tilde{O}$ of the following expression:

$$\frac{\sqrt{N}}{\alpha} = \frac{\sqrt{N}}{\mathfrak{r}} \left(\frac{NN_3\ell}{\ln(NN_3\ell)} \right)^{1/4} \tag{8.1}$$

In the case of a module lattice with dimension N, $\mathfrak{r}$ is to be bounded above by $O(\sqrt{\log N/N})$, or else the distribution will be statistically indistinguishable from the uniform and no distinguisher will be possible.

Using (5.7) and the bound $O(\sqrt{\log N/N})$ on $\mathfrak{r}$ and ignoring log factors and constants, from (8.1) we have

$$\gamma > N(NN_3\ell)^{1/4} > N(N\ell N^2\delta_2^{-2})^{1/4} = N^{7/4}\ell^{1/4}\delta_2^{-1/2} > N^{7/4}\delta_2^{-1/2}. \tag{8.2}$$

Here we take $N_3 = (\delta_2/N)^{-2}$ from Equation 5.7.

Ideal Lattice: The same argument holds for reductions in Chapter 6 for ring LWE. Here the dimension of the lattice is n instead of N in theorem 70. The value of γ is greater than $n^{7/4}\delta_2^{-1/2}$ following (Equation 8.2).

Here the approximation factor of SIVP for module and ideal lattices is outside the range where the SIVP problem is considered to be hard. The hardness of the SIVP problem is supposed to provide evidence for the security of cryptographic systems but the SIVP problem that is chosen is not the desired one to be precise.

For example, choosing $n = 2^{10}$ and $\delta_2 = 2^{-\beta_2}$, we find that $\gamma > 2^{(35+\beta_2)/2}$. Now for $\gamma = 2^k$ the fastest classical algorithm known that solves SVP_γ (and hence also solves K-SVP_γ and its equivalent K-SIVP_γ) has running time $2^{\tilde{\theta}(n/k)}$ where $\tilde{\theta}$ suppresses a log factor [Pei16]. The requirement is that $2^{n/k}$ should be large. From $\gamma = 2^k > 2^{(35+\beta_2)/2}$, we have $k > (35+\beta_2)/2$. Suppose we are considering 128-bit security. If we are extra cautious, then we will choose $\beta_2 = 128$; if we are less cautious, then we may choose $\beta_2 = 50$; and if we are not particularly risk-averse we might choose $\beta_2 = 25$. The corresponding lower bounds on γ and upper bounds on $2^{n/k}$ are shown in Table 8.6. None of these values inspire confidence in the hardness of M-SIVP_γ and K-SIVP_γ. In particular, the approximation factors γ are very large, and the running times $2^{n/k}$ are too small.

8.2.3 The value of LWE modulus q

The value of LWE modulus q used in both [LPR13] and [LS15] has some strict lower bound. A condition for the reductions is that $q\alpha > 2\omega(\sqrt{\ln n})$ for ideal lattice and $q\alpha > 2\sqrt{d}\omega(\sqrt{\ln N})$ for module lattice. Using (5.4), for module lattice we obtain

$$q \;>\; \frac{2\sqrt{d}\omega(\sqrt{\ln N})}{\alpha} = \frac{2\sqrt{d}\omega(\sqrt{\ln n})}{\mathfrak{r}} \left(\frac{NN_3\ell}{\ln(NN_3\ell)}\right)^{1/4}. \tag{8.3}$$

Again ignoring constants and log-terms and using $\mathfrak{r} < O(\sqrt{\log N/N})$, we have

$$q \;>\; N^{5/4}\ell^{1/4}\delta_2^{-1/2} > N^{5/4}\delta_2^{-1/2}. \tag{8.4}$$

δ_2	2^{-128}	2^{-50}	2^{-25}
$\gamma = 2^k$	$2^{81.5}$	$2^{42.5}$	2^{30}
$2^{n/k}$	2^{12}	2^{24}	2^{34}
q	2^{76}	2^{37}	2^{34}

Table 8.6: For $n = 2^{10}$ the lower bounds on γ and upper bounds on $2^{n/k}$ along with lower bounds on q.

Similarly, we obtain the following for the ideal lattice ($d = 1$, rank of the module)

$$q \;>\; n^{5/4}\ell^{1/4}\delta_2^{-1/2} > n^{5/4}\delta_2^{-1/2}. \tag{8.5}$$

With our values $n = 2^{10}$, $\delta_2 = 2^{-\beta_2}$ we find that $q > 2^{(25+\beta_2)/2}$. The lower bounds for q corresponding to $\beta_2 = 128, 50$ and 25 are shown in Table 8.6. The cryptosystem would be quite inefficient with these values of the modulus.

The NIST-PQC proposals SABER and KYBER are based on module lattices. In Chapter 5, we considered the reduction for module lattices. The parameters of Kyber are chosen based on the parameters of the ideal lattices of the same dimension as that of module lattices. We make the same assumptions for SABER. There are several variants of SABER and Kyber, and the highest security variant for both sets the dimension $n = 1024$. Irrespective of the dimension, q for SABER and Kyber are 2^{13} and 3329 respectively. These values are much lower than the values of q in Table 8.6.

8.2.4 Problem with structured Lattices

Lattice-based cryptography got its importance due to the fact of its worst case to average case equivalence property [Ajt96] other than being quantum-safe. That means that lattice problems like SVP, SIVP or CVP, etc. on any randomly chosen lattice are at least as hard as on the worst-case lattice instances. This result of Ajtai and Dwork [AD97], confirms the above-stated fact with the first public key encryption system which enjoyed the hardness of the worst case of CVP. We already discussed the point from the tightness gap of concrete security perspective and found that the reductions lack desired tightness.

Now, we focus on the domain of lattices used to construct lattice-based applications. The main reason is that the practical applications chose a different algebraic variant of lattices rather than choosing Euclidean lattices. Here different algebraic variants are ideal lattice and

module lattice. Though ideal lattices are restricted versions of module lattices and module lattices are the generalized version of ideal lattices. The rationality behind this choice is efficiency. We get an advantage using ideal and module lattices in the size of the public key. The size of the public key is $O(n)$ times larger in the case of Euclidean lattices than that of ideal and module lattices. Again a single multiplication in ideal lattices is done through n multiplications in Euclidean lattices. But we need to remind that ideal lattices are a subclass of all lattices. In [LPR13] we work with ideal lattices over the cyclotomic number field. So, the argument for worst-to-average case equivalence comes under scrutiny when lattices are chosen from a special class, such as the class of ideal lattices or module lattices. We need to have the guarantee or the assurance of the worst to average case equivalence over the domain of lattices that are used in the applications in the name of achieving better efficiency. Thus the **"worst to average case equivalence"** is one the critical point in the evolution and endorsement of lattice-based cryptography and it should not be ignored against any other justification to work in a special subclass of lattices.

It is evident that ideal lattices and module lattices have much more structure than Euclidean lattices. These added features have been used extensively for the proof of security reductions for the ideal lattice as well as module lattices. We can have different embeddings of a number field but in the case of ideal lattices, canonical embedding is used. The fact that we can have isomorphisms between different embeddings of the number field is used as one of the key features in security reductions. In the cyclotomic number field and the case of non-cyclotomic Galois fields as well, these isomorphisms become automorphisms. The intricate security reductions in [LPR13] and [LS15] use these facts which are not present or needed in [Reg09]. The gain was efficiency.

Greater structure and symmetry can be a probable concern for weakness. The hardness of SVP is the basis of security for lattice-based cryptographic schemes. SVP is hard in Euclidean lattices but SVP is presumably not as hard as in Euclidean lattices as in the restricted settings in ideal lattices of a cyclotomic number field. In cyclotomic ideal lattices, SIVP_γ is trivially equivalent to SVP_γ as pointed out in [LPR13]. The hardness relation between SVP and SIVP is much weaker in Euclidean lattices. $\text{SIVP}_{\sqrt{n}\gamma}$ reduces in polynomial time to SVP_γ, where n is the dimension of the lattice [MG02]. This suggests that for Euclidean lattices SIVP and SIVP_γ are strictly harder than SVP and SVP_γ but they are equivalent when restricted to cyclotomic ideal lattices. To quote from [KSSS22].
"...even if SVP and approximate SVP for cyclotomic ideal lattices were to be as hard as for general lattices, SIVP and approximate SIVP for cyclotomic ideal lattices could be easier."

8.2.5 The Quantum Part

All the lattice based reductions starting from [Reg09] are quantum except [Pei09] and [BLP+13]. The need and the working of the quantum part are the same in all the reductions. The quantum part of the reduction in [LPR13] and [LS15] is largely taken from [Reg09]. The quantum part is needed only in one step in the proof of the main theorem.

In [Reg09]) the quantum algorithm has a state that is a linear combination of roughly 2^{3n^2} terms, each involving two entangled registers. The algorithm needs to "erase" the first of these entangled registers, which means "uncomputing" a closest vector in each summand. This is done by converting an algorithm for the closest vector problem (denoted CVP in [Reg09] and BDD in [LPR13]) into a quantum circuit and then reversing the circuit.

The quantum algorithm is based on Lemma 3.14 of [Reg09], which shows that $n \log R$ logical qubits are required for lattice L, where R is an integer which is at least $2^{3n}\lambda_n(L)$. Since $\lambda_n(L)$ is generally polynomial in n, it follows that the number of logical qubits required is about $3n^2$. For $n = 2^{10}$ about 3 million logical qubits will be required. In comparison, factoring a 2048-bit RSA modulus requires roughly 4000 to 5000 logical qubits [Sho99].

The reason to doubt the feasibility of the quantum part of the security reduction in [Reg09], [LPR13],[LS15] and [PRS17] is, even for $n = 1024$ the circuit size is many thousands of times the circuit size for Shor's algorithm to factor a 2048-bit RSA modulus. Since the number of qubits grows quadratically with n, the circuit size becomes much greater if one chooses n large enough to compensate for the tightness gap in the reduction. It questions the feasibility of the quantum part for its practical importance.

8.3 Conclusion

In this chapter, we have discussed various aspects of concrete security analysis, which are instrumental in our investigation. One of the key concepts we explored is the tightness gap, which serves to quantify the level of security provided by reductions in asymptotic settings. By measuring the tightness gap, we gain a more concrete understanding of the security guarantees offered by cryptographic constructions. Upon closer examination of the reductions, we observed that certain parameters, such as the LWE modulus and the approximation fac-

tor of SIVP, did not instill much confidence in their practicality. This further emphasizes the significance of concrete security analysis as a vital tool to bridge the gap between theoretical proofs and real-world practicality. By conducting concrete security analysis, we can assess cryptographic protocols based on specific parameter choices and efficiency considerations, ensuring that they provide meaningful security guarantees in practical scenarios. It allows us to validate the robustness of these protocols against potential real-world threats and resource constraints, such as key sizes and computational overhead. In conclusion, concrete security analysis is an essential aspect of cryptographic research, as it brings clarity and realism to the theoretical security guarantees provided by cryptographic constructions. It aids in designing efficient and secure post-quantum cryptographic systems that can be effectively deployed in real-world applications.

Chapter 9

Conclusion

Lattice-based cryptography has long been regarded as a promising contender for quantum-safe cryptography due to its strong theoretical security foundation, particularly the asymptotic evidence based on the worst-to-average case equivalence of hard lattice problems. This theoretical achievement cannot be overlooked, and it has driven significant progress in the field. However, the practical aspects of security and communication efficiency are equally crucial for real-world applications. Merely relying on asymptotic guarantees may not be sufficient to ensure the security and efficiency of cryptographic protocols when deployed in practical scenarios. Therefore, it becomes essential to delve beyond the theoretical perspective and thoroughly assess the practicality of these cryptographic systems before making them available for general use. This thesis follows the same line of thinking throughout its research journey. Its primary purpose is to re-evaluate the merits of lattice-based cryptography from a practical standpoint. By conducting concrete security analyses and investigating the tightness gap of reductions, the thesis aims to shed light on the real-world security and efficiency of lattice-based cryptographic constructions. The intention is to provide valuable insights into the practical feasibility of these cryptographic systems and offer guidelines for their effective and secure deployment in practical applications. In essence, the thesis seeks to strike a balance between the theoretical excellence of lattice-based cryptography and its practical applicability, ultimately contributing to the development of robust and efficient post-quantum cryptographic solutions.

Provable security is undeniably a crucial aspect that enhances the trustworthiness and reliability of cryptographic schemes. It offers a rigorous mathematical foundation, providing a certificate of hardness to cryptographic constructions, particularly in an asymptotic setting. This mathematical assurance is instrumental in establishing the security guarantees of cryptographic systems. Lattice-based cryptography, being one of the prominent candidates for post-quantum cryptography, is also built on provable security principles. Its security is grounded in the presumed hardness of certain lattice problems, which are believed to withstand attacks even from quantum adversaries. However, the analysis presented in this thesis reveals some limitations in the asymptotic hardness guarantee provided by the reductions in lattice-based cryptography. The tightness gap of these reductions suggests that the theo-

retical security assurances might not fully translate into practical guarantees. The practical feasibility and real-world security of lattice-based cryptographic constructions are affected by these gaps.

9.1 Summary

The thesis contributes to this understanding through five comprehensive chapters, each focusing on aspects of concrete security analysis and the tightness gap in different variants of lattice-based cryptography, which we summarize in the following.

- In Chapter 4, we discussed the quantum reductions for lattice-based cryptosystems. Lattice-based cryptographic schemes were shown as hard as worst-case hard lattice problems, namely GIVP. We analyzed each sub-reduction minutely and found the tightness gap of the end-to-end reduction. The tightness gap is shown as a function of the lattice dimension and the parameters of the average case LWE distinguisher. The tightness gap is not within practical range by any means.

- In Chapter 5, we focused on the quantum reduction for module and ideal lattice-based cryptographic schemes. Ideal lattices are a sub-class of lattices where cryptographic operations can be done more efficiently than a general or Euclidean lattice. Ideal lattices are chosen over Euclidean lattices for practical purposes for the computational efficiency aspect of it. Though it introduces some structural weakness to it which is not present in Euclidean lattices. Module lattices are an optimized version of the two types of lattices, viz, Euclidean and ideal lattices. Cryptographic designers tried to take the good out of both types of lattices and put them into one. Our analysis shows that the tightness gap is still an issue for module lattice-based cryptography. We calculated the tightness gap and tried to optimize the reductions whenever possible. But optimization could not change the overall analysis much.

- In Chapter 6, the point of discussion was ideal lattice-based reductions but here ideal lattices are more general than in the previous chapter. In the previous chapter module and ideal lattices are from the cyclotomic number field. These lattices are very skewed sub domain of lattices which restrict the scope for cryptographic designers. This reduction has more degrees of freedom as this reduction applies to any LWE modulus. But the asymptotic reduction still lacks the tightness.

- One major drawback of the lattice-based reductions is that they are all quantum. They all share one quantum step. The classical lattice reduction could solve the problem but that was not satisfactory from a practical perspective. We analyzed the classical reduction in Chapter 7. Our analysis gave new insights into this reduction in various ways. We could show that the tightness gap of the reduction is too big to use in practice.

- In Chapter 8, we discussed different aspects of concrete security analysis. We discussed what a high tightness gap implies in practice. We calculated the tightness gap for all the reductions that we discussed and explained the adverse effect of it. A concrete analysis shows that the approximation factor of SIVP in the case of ideal and module lattices lies beyond the permissible range for SIVP to be considered a hard problem. Also, the value of the modulus of the LWE problem is analyzed for the context of practical security. We also presented the problems of using structured lattices like ideal and module lattices in place of Euclidean lattices. The relationship between security and efficiency has been a challenging one. The optimization between both of them has been pointed out. The cost of the quantum step in the reduction is very high. The impact of the quantum step in the context of practical usability of the lattice-based reduction is a matter of concern, which has been described in the concluding chapter.

9.2 Future Directions

In the future, further research and work can be focused on improving the concrete security of lattice-based cryptography. Some potential areas for future work in this domain include:

- Tightness Gap Reductions: Investigate and develop new reductions that minimize the tightness gap in lattice-based cryptographic constructions. Finding tighter reductions would provide stronger guarantees of real-world security for these schemes.

- Parameter Selection: Analyze the impact of different parameter choices on the concrete security of lattice-based cryptosystems. Identifying optimal parameter sets that balance security and efficiency is crucial for practical implementations.

- New Mathematical Techniques: Explore novel mathematical techniques and tools that can enhance the concrete security analysis of lattice-based schemes. Developing innovative approaches to quantify the security guarantees could lead to more accurate

assessments.

- Post-Quantum Security: As quantum computers become more powerful, the security of lattice-based cryptography may be further challenged. Investigate the resilience of lattice-based schemes against quantum attacks and explore potential quantum-safe variants.

- Efficiency Improvements: Work on optimizing the efficiency of lattice-based cryptosystems without compromising security. Finding faster algorithms and reducing key sizes can make these schemes more practical for various applications.

- Cryptanalysis: Conduct cryptanalysis on lattice-based schemes to identify potential weaknesses and vulnerabilities. Understanding the security limitations of these schemes can guide the development of more robust cryptographic protocols.

By addressing these future research directions, the field of lattice-based concrete security can make significant progress in ensuring the practical viability and security of lattice-based cryptographic constructions in real-world applications. In conclusion, while lattice-based cryptography has a strong foundation in provable security, the concrete security analysis presented in this thesis raises important considerations for its practical implementation. By addressing these issues, the thesis strives to make lattice-based cryptography more reliable and suitable for real-world applications in the era of quantum computing.

Bibliography

[ADRS15] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2^n time using discrete gaussian sampling: Extended abstract. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 733–742. ACM, 2015.

[ADS15] Divesh Aggarwal, Daniel Dadush, and Noah Stephens-Davidowitz. Solving the closest vector problem in 2ˆn time - the discrete gaussian strikes again! In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 563–582. IEEE Computer Society, 2015.

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010.

[AEVZ02] Erik Agrell, Thomas Eriksson, Alexander Vardy, and Kenneth Zeger. Closest point search in lattices. *IEEE Trans. Information Theory*, 48(8):2201–2214, 2002.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108. ACM, 1996.

[Ajt98] Miklós Ajtai. The shortest vector problem in L_2 is *NP*-hard for randomized reductions (extended abstract). In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 10–19. ACM, 1998.

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In Frank Thomson Leighton and Peter W. Shor,

editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 284–293. ACM, 1997.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 601–610. ACM, 2001.

[AKS02] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity, Montréal, Québec, Canada, May 21-24, 2002*, pages 53–57. IEEE Computer Society, 2002.

[ACF+15] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. On the complexity of the BKW algorithm on LWE. *Des. Codes Cryptogr.*, 74(2):325–354, 2015.

[AFFP14] Martin R. Albrecht, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Lazy modulus switching for the BKW algorithm on LWE. In Hugo Krawczyk, editor, *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, volume 8383 of *Lecture Notes in Computer Science*, pages 429–445. Springer, 2014.

[AAB+19] Erdem Alkim, Roberto Avanzi, Joppe Bos, Leo Ducas, Antonio de la Piedra, Thomas Poppelmann, Peter Schwabe, Douglas Stebila, Martin R. Albrecht, Emmanuela Orsini, Valery Osheter, Kenneth G. Paterson, Guy Peer, and Nigel P. Smart. NewHope: algorithm specifications and supporting documentation. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions, 2019.

[ABD+19] Erdem Alkim, Joppe Bos, Leo Ducas, Patrick Longa, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. FrodoKEM: Learning With Errors key encapsulation. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions, 2019.

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*, pages 327–343. USENIX Association, 2016.

[Als11] Gerold Alsmeyer. *Chebyshev's Inequality*, pages 239–240. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[AKPW13] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 57–74. Springer, 2013.

[AN17] Yoshinori Aono and Phong Q. Nguyen. Random sampling revisited: Lattice enumeration with discrete pruning. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 65–102, 2017.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In Luca Aceto, Monika Henzinger, and Jirí Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.

[ABD+09] Roberto Avanzi, Joppe Bos, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien

Stehlé. CRYSTALS-Kyber: algorithm specifications and supporting documentation. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions, 2009.

[BBF+19a] Hayo Baan, Sauvik Bhattacharya, Scott Fluhrer, Oscar Garcia-Morchon, Thijs Laarhoven, Rachel Player, Ronald Rietman, Markku-Juhani O. Saarinen, Ludo Tolhuizen, Jose-Luis Torre-Arce, and Zhenfei Zhang. Round5: KEM and PKE based on (Ring) Learning With Rounding. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions, 2019.

[BBF+19b] Hayo Baan, Sauvik Bhattacharya, Scott R. Fluhrer, Óscar García-Morchón, Thijs Laarhoven, Ronald Rietman, Markku-Juhani O. Saarinen, Ludo Tolhuizen, and Zhenfei Zhang. Round5: Compact and fast post-quantum public-key encryption. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers*, volume 11505 of *Lecture Notes in Computer Science*, pages 83–102. Springer, 2019.

[Bab86] László Babai. On lovász' lattice reduction and the nearest lattice point problem. *Comb.*, 6(1):1–13, 1986.

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

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 719–737. Springer, 2012.

[BDGL15] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. *IACR Cryptology ePrint Archive*, 2015:1128, 2015.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert

Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 10–24. SIAM, 2016.

[BDK+21] Michiel Van Beirendonck, Jan-Pieter D'Anvers, Angshuman Karmakar, Josep Balasch, and Ingrid Verbauwhede. A side-channel-resistant implementation of SABER. *ACM J. Emerg. Technol. Comput. Syst.*, 17(2):10:1–10:26, 2021.

[Bel97] Mihir Bellare. Practice-oriented provable-security. In Eiji Okamoto, George I. Davida, and Masahiro Mambo, editors, *Information Security, First International Workshop, ISW '97, Tatsunokuchi, Japan, September 17-19, 1997, Proceedings*, volume 1396 of *Lecture Notes in Computer Science*, pages 221–231. Springer, 1997.

[Bel98] Mihir Bellare. Practice-oriented provable security. In Ivan Damgård, editor, *Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998*, volume 1561 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1998.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 514–523. IEEE Computer Society, 1996.

[BDJR97] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 394–403. IEEE Computer Society, 1997.

[BR09] Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for waters' IBE scheme. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 407–424. Springer, 2009.

[Ber08] Daniel J. Bernstein. The salsa20 family of stream ciphers. In Matthew J. B. Robshaw and Olivier Billet, editors, *New Stream Cipher Designs - The eS-*

TREAM Finalists, volume 4986 of *Lecture Notes in Computer Science*, pages 84–97. Springer, 2008.

[Ber19] Daniel J. Bernstein. Comparing proofs of security for lattice-based encryption. Cryptology ePrint Archive, Report 2019/691, 2019. https://eprint.iacr.org/2019/691.

[BCLvV17] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU prime: Reducing attack surface at low cost. In Carlisle Adams and Jan Camenisch, editors, *Selected Areas in Cryptography - SAC 2017 - 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers*, volume 10719 of *Lecture Notes in Computer Science*, pages 235–260. Springer, 2017.

[BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 278–291. Springer, 1993.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[BGM+16] Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 209–224. Springer, 2016.

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, volume 1403 of *Lecture Notes in Computer Science*, pages 59–71. Springer, 1998.

[BCD+16] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the

ring! practical, quantum-secure key exchange from LWE. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1006–1018. ACM, 2016.

[BDK+18] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 353–367. IEEE, 2018.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 575–584. ACM, 2013.

[CN97] Jin-yi Cai and Ajay Nerurkar. An improved worst-case to average-case connection for lattice problems. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 468–477. IEEE Computer Society, 1997.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, 2010.

[Cas59] John William Scott Cassels. *An Introduction to the Geometry of Numbers*. Springer, Berlin, Gttingen Heidelberg, 1959.

[CKMS16] Sanjit Chatterjee, Neal Koblitz, Alfred Menezes, and Palash Sarkar. Another look at tightness II: practical issues in cryptography. In Raphael C.-W. Phan

and Moti Yung, editors, *Paradigms in Cryptology - Mycrypt 2016. Malicious and Exploratory Cryptology - Second International Conference, Mycrypt 2016, Kuala Lumpur, Malaysia, December 1-2, 2016, Revised Selected Papers*, volume 10311 of *Lecture Notes in Computer Science*, pages 21–55. Springer, 2016.

[CMS11] Sanjit Chatterjee, Alfred Menezes, and Palash Sarkar. Another look at tightness. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 293–319. Springer, 2011.

[DRS14] Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. On the closest vector problem with a distance guarantee. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 98–109. IEEE Computer Society, 2014.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[DKRV18] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure KEM. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *Progress in Cryptology - AFRICACRYPT 2018 - 10th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 7-9, 2018, Proceedings*, volume 10831 of *Lecture Notes in Computer Science*, pages 282–305. Springer, 2018.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[DKS98] Irit Dinur, Guy Kindler, and Shmuel Safra. Approximating-cvp to within almost-polynomial factors is np-hard. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 99–111. IEEE Computer Society, 1998.

[DLdW19] Emmanouil Doulgerakis, Thijs Laarhoven, and Benne de Weger. Finding closest lattice vectors using approximate voronoi cells. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference,*

PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers, volume 11505 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2019.

[DLdW20] Emmanouil Doulgerakis, Thijs Laarhoven, and Benne de Weger. Sieve, enumerate, slice, and lift: - hybrid lattice algorithms for SVP via CVPP. In Abderrahmane Nitaj and Amr M. Youssef, editors, *Progress in Cryptology - AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20-22, 2020, Proceedings*, volume 12174 of *Lecture Notes in Computer Science*, pages 301–320. Springer, 2020.

[DTV15] Alexandre Duc, Florian Tramèr, and Serge Vaudenay. Better algorithms for LWE and LWR. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 173–202. Springer, 2015.

[DF04] David Steven Dummit and Richard M. Foote. *Abstract algebra*, volume 3rd edition. Hoboken NJ: Wiley., 2004.

[DKRV19] Jan-Pieter DAnvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. SABER: Mod-LWR based KEM (round 2 submission). https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions, 2019.

[FP85] U. Fincke and Michael Pohst. A new method of computing fundamental units in algebraic number fields. In B. F. Caviness, editor, *EUROCAL '85, European Conference on Computer Algebra, Linz, Austria, April 1-3, 1985, Proceedings Volume 2: Research Contributions*, volume 204 of *Lecture Notes in Computer Science*, pages 470–478. Springer, 1985.

[FKM+06] Pierre-Alain Fouque, Sébastien Kunz-Jacques, Gwenaëlle Martinet, Frédéric Muller, and Frédéric Valette. Power attack on small RSA public exponent. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, volume 4249 of *Lecture Notes in Computer Science*, pages 339–353. Springer, 2006.

[Gam84] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1984.

[Gat18] F. Gates. Reduction-respecting parameters for lattice-based cryptosystems. https://macsphere.mcmaster.ca/bitstream/11375/24466/2/gates_fletcher_m_finalsubmission2018october_msc.pdf, 2018.

[Gau66] Carl Friedrich Gauss. *Disquisitiones arithmeticae*, volume 157. Yale University Press, 1966.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206. ACM, 2008.

[GMK+22] Archisman Ghosh, Jose Maria Bermudo Mera, Angshuman Karmakar, Debayan Das, Santosh Ghosh, Ingrid Verbauwhede, and Shreyas Sen. A 334uw 0.158mm2 saber learning with rounding based post-quantum crypto accelerator. In *IEEE Custom Integrated Circuits Conference, CICC 2022, Newport Beach, CA, USA, April 24-27, 2022*, pages 1–2. IEEE, 2022.

[GRS08] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. How to encrypt with the LPN problem. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 679–690. Springer, 2008.

[GG00] Oded Goldreich and Shafi Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000.

[GGH96] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. *Electron. Colloquium Comput. Complex.*, TR96-042, 1996.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer, 1997.

[GMSS99] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inf. Process. Lett.*, 71(2):55–61, 1999.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[HR07] Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 469–477. ACM, 2007.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *Journal of the American Statistical Association*, volume 58, pages 13–30, 1963.

[HHP+03] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN: digital signatures using the NTRU lattice. In Marc Joye, editor, *Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings*, volume 2612 of *Lecture Notes in Computer Science*, pages 122–140. Springer, 2003.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.

[HPS01] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NSS: an NTRU lattice-based signature scheme. In Birgit Pfitzmann, editor, *Advances in Cryptology -*

EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding, volume 2045 of *Lecture Notes in Computer Science*, pages 211–228. Springer, 2001.

[HB01] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66. Springer, 2001.

[Kan83] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo, and Joel I. Seiferas, editors, *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 193–206. ACM, 1983.

[Kan87] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987.

[KMRV18] Angshuman Karmakar, Jose Maria Bermudo Mera, Sujoy Sinha Roy, and Ingrid Verbauwhede. Saber on ARM cca-secure module lattice-based key encapsulation on ARM. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):243–266, 2018.

[KTX07] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Multi-bit cryptosystems based on lattice problems. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography - PKC 2007, 10th International Conference on Practice and Theory in Public-Key Cryptography, Beijing, China, April 16-20, 2007, Proceedings*, volume 4450 of *Lecture Notes in Computer Science*, pages 315–329. Springer, 2007.

[Kho05] Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *J. ACM*, 52(5):789–808, 2005.

[Kho10] Subhash Khot. *Inapproximability Results for Computational Problems on Lattices*, pages 453–473. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[KS06] Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 553–562. IEEE Computer Society, 2006.

[KM06] Neal Koblitz and Alfred Menezes. Another look at "provable security". II. In Rana Barua and Tanja Lange, editors, *Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings*, volume 4329 of *Lecture Notes in Computer Science*, pages 148–175. Springer, 2006.

[KM07] Neal Koblitz and Alfred Menezes. Another look at "provable security". *J. Cryptol.*, 20(1):3–37, 2007.

[KM08] Neal Koblitz and Alfred Menezes. Another look at non-standard discrete log and diffie-hellman problems. *J. Math. Cryptol.*, 2(4):311–326, 2008.

[KM19] Neal Koblitz and Alfred Menezes. Critical perspectives on provable security: Fifteen years of "another look" papers. *Adv. Math. Commun.*, 13(4):517–558, 2019.

[KSSS22] Neal Koblitz, Subhabrata Samajder, Palash Sarkar, and Subhadip Singha. Concrete analysis of approximate ideal-sivp to decision ring-lwe reduction. *IACR Cryptol. ePrint Arch.*, page 275, 2022.

[KS01] Ravi Kumar and D. Sivakumar. On polynomial approximation to the shortest lattice vector length. In S. Rao Kosaraju, editor, *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA*, pages 126–127. ACM/SIAM, 2001.

[KDB+22] Suparna Kundu, Jan-Pieter D'Anvers, Michiel Van Beirendonck, Angshuman Karmakar, and Ingrid Verbauwhede. Higher-order masked saber. In Clemente Galdi and Stanislaw Jarecki, editors, *Security and Cryptography for Networks - 13th International Conference, SCN 2022, Amalfi, Italy, September 12-14, 2022, Proceedings*, volume 13409 of *Lecture Notes in Computer Science*, pages 93–116. Springer, 2022.

[LMvdP15] Thijs Laarhoven, Michele Mosca, and Joop van de Pol. Finding shortest lattice vectors faster using quantum search. *Des. Codes Cryptogr.*, 77(2-3):375–400, 2015.

[LJS90] J. C. Lagarias, Hendrik W. Lenstra Jr., and Claus-Peter Schnorr. Korkin-zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Comb.*, 10(4):333–348, 1990.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015.

[LLL82] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, Dec 1982.

[LLM06] Yi-Kai Liu, Vadim Lyubashevsky, and Daniele Micciancio. On bounded distance decoding for general lattices. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2006 and 10th International Workshop on Randomization and Computation, RANDOM 2006, Barcelona, Spain, August 28-30 2006, Proceedings*, volume 4110 of *Lecture Notes in Computer Science*, pages 450–461. Springer, 2006.

[LLJ+19] Xianhui Lu, Yamin Liu, Dingding Jia, Haiyang Xue, Jingnan He, Zhenfei Zhang, Zhe Liu, Hao Yang, Bao Li, and Kunpeng Wang. LAC: Lattice-based Cryptosystems. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions, 2019.

[LLZ+18] Xianhui Lu, Yamin Liu, Zhenfei Zhang, Dingding Jia, Haiyang Xue, Jingnan He, and Bao Li. LAC: practical ring-lwe based public-key encryption with byte-level modulus. *IACR Cryptol. ePrint Arch.*, page 1009, 2018.

[LM09] Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2009.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, 2010.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43:1–43:35, 2013.

[McE78] Robert J McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.

[Men12] Alfred Menezes. Another look at provable security. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, page 8. Springer, 2012.

[Mic00] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput.*, 30(6):2008–2035, 2000.

[Mic01a] Daniele Micciancio. The hardness of the closest vector problem with preprocessing. *IEEE Trans. Inf. Theory*, 47(3):1212–1215, 2001.

[Mic01b] Daniele Micciancio. Improving lattice based cryptosystems using the hermite normal form. In Joseph H. Silverman, editor, *Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001, Revised Papers*, volume 2146 of *Lecture Notes in Computer Science*, pages 126–145. Springer, 2001.

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems - a cryptograhic perspective*, volume 671 of *The Kluwer international series in engineering and computer science*. Springer, 2002.

[MP11] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. Cryptology ePrint Archive, Report 2011/501, https://eprint.iacr.org/2011/501, 2011. An abridged version of this paper appeared in the proceedings of Eurocrypt 2012.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

[MV10] Daniele Micciancio and Panagiotis Voulgaris. Faster exponential time algorithms for the shortest vector problem. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1468–1480. SIAM, 2010.

[Muk20] N. Mukhopadhyay. *Probability and Statistical Inference*. CRC Press, 2020.

[Ngu99] Phong Q. Nguyen. Cryptanalysis of the goldreich-goldwasser-halevi cryptosystem from crypto '97. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 288–304. Springer, 1999.

[NR06] Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 271–288. Springer, 2006.

[NR09] Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *J. Cryptol.*, 22(2):139–160, 2009.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342. ACM, 2009.

[Pei16] Chris Peikert. A decade of lattice cryptography. *Found. Trends Theor. Comput. Sci.*, 10(4):283–424, 2016.

[PRS17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 461–473. ACM, 2017.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David A. Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 187–196. ACM, 2008.

[Poh81] Michael Pohst. On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. *SIGSAM Bull.*, 15(1):37–44, 1981.

[Reg03] Oded Regev. New lattice based cryptographic constructions. In Lawrence L. Larmore and Michel X. Goemans, editors, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 407–416. ACM, 2003.

[Reg04] Oded Regev. Lattices in Computer Science. https://cims.nyu.edu/~regev/teaching/lattices_fall_2004/index.html, 2004.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93. ACM, 2005.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[SBG+18] Markku-Juhani O. Saarinen, Sauvik Bhattacharya, Óscar García-Morchón, Ronald Rietman, Ludo Tolhuizen, and Zhenfei Zhang. Shorter messages and faster post-quantum encryption with round5 on cortex M. In Begül Bilgin and Jean-Bernard Fischer, editors, *Smart Card Research and Advanced Applications,*

17th International Conference, CARDIS 2018, Montpellier, France, November 12-14, 2018, Revised Selected Papers, volume 11389 of *Lecture Notes in Computer Science*, pages 95–110. Springer, 2018.

[SS20] Palash Sarkar and Subhadip Singha. Classical reduction of gap SVP to LWE: A concrete security analysis. *IACR Cryptol. ePrint Arch.*, page 880, 2020.

[SS21] Palash Sarkar and Subhadip Singha. Verifying solutions to LWE with implications for concrete security. *Adv. Math. Commun.*, 15(2):257–266, 2021.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

[Sch91] Claus-Peter Schnorr. Factoring integers and computing discrete logarithms via diophantine approximations. In Donald W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 281–293. Springer, 1991.

[Sch03] Claus-Peter Schnorr. Lattice reduction by random sampling and birthday methods. In Helmut Alt and Michel Habib, editors, *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*, volume 2607 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2003.

[SE94] Claus-Peter Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66:181–199, 1994.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Syst. Tech. J.*, 28(4):656–715, 1949.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[SS11] Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *Advances in Cryptology*

- EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings, volume 6632 of *Lecture Notes in Computer Science*, pages 27–47. Springer, 2011.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635. Springer, 2009.

[SJ05] William Stein and David Joyner. SAGE: System for algebra and geometry experimentation. *ACM SIGSAM Bulletin*, 39(2):61–64, 2005.

[Wal17] Michael Walter. *On the Concrete Security of Lattice-Based Cryptography*. PhD thesis, University of California, San Diego, USA, 2017.

[Yas21] Masaya Yasuda. A survey of solving svp algorithms and recent strategies for solving the svp challenge. In Tsuyoshi Takagi, Masato Wakayama, Keisuke Tanaka, Noboru Kunihiro, Kazufumi Kimoto, and Yasuhiko Ikematsu, editors, *International Symposium on Mathematics, Quantum Theory, and Cryptography*, pages 189–207, Singapore, 2021. Springer Singapore.