RADIX-3 NTT-BASED POLYNOMIAL MULTIPLICATION FOR LATTICE-BASED CRYPTOGRAPHY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CHENAR ABDULLA HASSAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

MAY 2022

Approval of the thesis:

RADIX-3 NTT-BASED POLYNOMIAL MULTIPLICATION FOR LATTICE-BASED CRYPTOGRAPHY

submitted by **CHENAR ABDULLA HASSAN** in partial fulfillment of the requirements for the degree of **Master of Science in Cryptography Department, Middle East Technical University** by,

Prof. Dr. A. Sevtap Kestel
Dean, Graduate School of **Applied Mathematics** __________

Prof. Dr. Ferruh Özbudak
Head of Department, **Cryptography** __________

Assoc. Prof. Dr. Oğuz Yayla
Supervisor, **Cryptography** __________

Examining Committee Members:

Assoc. Prof. Dr. Murat Cenk
Cryptography, IAM, METU __________

Assoc. Prof. Dr. Oğuz Yayla
Cryptography, IAM, METU __________

Assist. Prof. Dr. Erdem Alkım
Department of Computer Science, Dokuz Eylül University __________

Date: __________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: CHENAR ABDULLA HASSAN

Signature :

ABSTRACT

RADIX-3 NTT-BASED POLYNOMIAL MULTIPLICATION FOR LATTICE-BASED CRYPTOGRAPHY

Hassan, Chenar Abdulla
M.S., Department of Cryptography
Supervisor : Assoc. Prof. Dr. Oğuz Yayla

May 2022, 39 pages

The lattice-based cryptography is considered as a strong candidate amongst many other proposed quantum-safe schemes for the currently deployed asymmetric cryptosystems that do not seem to stay secure when quantum computers come into play. Lattice-based algorithms possesses a time consuming operation of polynomial multiplication. As it is relatively the highest time consuming operation in lattice-based cryptosystems, one can obtain fast polynomial multiplication by using number theoretic transform (NTT). In this thesis, we focus on and introduce a radix-3 butterfly operation to be used in NTT-based polynomial multiplication. In addition, utilizing the ring structure, we propose two parameter sets of CRYSTALS-KYBER, one of the four round three finalists in the NIST Post-Quantum Competition.

Keywords: Number Theoretic Transformation, Polynomial Multiplication, KYBER, Lattice-Based Cryptography.

ÖZ

KAFES-TABANLI KRİPTOGRAFİ İÇİN RADİX-3 NTT-TABANLI POLİNOM ÇARPMASI

Hassan, Chenar Abdulla
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Doç. Dr. Oğuz Yayla

Mayıs 2022, 39 sayfa

Kafes tabanlı kriptografi, kuantum bilgisayarlar devreye girdiği takdirde güvenli kalmayan şu anda kullanılmakta olan asimetrik şifreleme sistemleri yerine önerilen diğer kuantum-güvenli şemalar arasında güçlü bir aday olarak kabul edilmektedir. Kafes tabanlı algoritmaların en zaman harcayan işlemi polinom çarpmasıdır. Kafes tabanlı kriptosistemlerde görece olarak en fazla zaman alan işlem polinom çarpması olduğundan, sayı teorik dönüşüm (NTT) kullanılarak hızlı polinom çarpımı elde edilebilir. Bu tezde, NTT tabanlı polinom çarpmasında kullanılacak bir radix-3 kelebek işlemine odaklanıyor ve tanıtıyoruz. Ek olarak, NIST Post-Kuantum Yarışmasında üçüncü turunun dört finalistten biri olan CRYSTALS-KYBER için halka yapısını kullanarak iki parametre setini öneriyoruz.

Anahtar Kelimeler: Sayı Teorik Dönüşümü, Polinom Çarpımı, KYBER, Kafes-Tabanlı Kriptografi.

To my parents

ACKNOWLEDGMENTS

Certainly, my thesis supervisor Assoc. Prof. Dr. Oğuz Yayla has proved to be such an ideal mentor with his encouragement, advice, invaluable feedback, and confidence in me, sometimes even meeting late at night. I am very proud and fortunate to have had the opportunity of working under his excellent supervision. I would also like to thank the thesis jury members, Prof. Dr. Murat Cenk and Assoc. Prof. Dr. Erdem Alkım for their generosity and time invested in reading this thesis.

I am very fortunate to have met Ilir Sheraj who has had a great impact on me. A friend I could never forget. Thank you so much for your company. Countless fun times while sipping coffee and discussing mathematics despite being a biologist. Moreover, very special thanks goes to Esra Günsay and Hasan Bartu Yünüak for always being out there and whenever needed. Many more friends merit recognition for their support over the course of this master's study, I intend to leave them out and not name them purposefully!

Last, but definitely not least, my deepest gratitude and appreciation to my family for their continuous support and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

NIST	National Institute of Standards and Technology
PQC	Post Quantum Competition
KEM	Key Encapsulation Mechanism
PKE	Public Key Encryption
CPA	Chosen Plaintext Attack
CCA	Chosen Ciphertext Attack
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
NTT	Number Theoretic Transform
CRT	Chinese Remainder Theorem
LWE	Learning with Error
R-LWE	Ring-Learning with Error
M-LWE	Module-Learning with Error
pk	Public Key
ct	Ciphertext

CHAPTER 1

INTRODUCTION

A small portion of people notices the small padlock symbol that is visible on the left of the search bar in internet browsers which is significant while shopping, bank, email, and social media accounts are visited. It is a sign of whether the intended website uses encryption for communication on the internet. This symbol verifies that the data is guarded while traveling through the internet.

Symmetric and asymmetric schemes are the two cryptosystems that provide security for today's digital information. Symmetric Cryptography (also referred to as secret key cryptography) utilizes one previously agreed on and shared key between the sender and receiver to encrypt and decrypt messages among them. On the other hand, asymmetric cryptography, which is also known as public-key cryptography, makes use of two keys that are referred to as public and private keys. The public key (publicly known as the name suggests) is used by senders to encrypt and send over messages (known as plaintexts) and the receiver on the other end uses a unique and solely known private key to decrypt the message. These two oftentimes are used along with each other. For instance, internet browsers use public-key schemes for validation and obtainment of a shared key afterward symmetric key schemes for encrypting future communications.

The lack of security of any kind of credentials that goes through the internet, often referred to as an insecure channel, could lead to significant ramifications that are much more aggravating for government, intelligence communities, and business companies who handle quite sensitive information than it is for individuals. The reason for this is that the current computational power is not yet strong to compromise that security despite being still under development. Albeit, quantum computers can compromise that and it is anticipated in a decade or so they could become somewhat of a threat to the currently deployed cryptographic protocols. It is for this reason that government, intelligence agencies, bank companies, and researchers are all competing to create new methods and cryptographic protocols that would resist attacks that can utilize quantum computers.

It is well-known that for classical computers, it takes thousands of years to break the presently deployed cryptosystems for which we can conclude that they are practically secure and they cannot pose a great threat.

Quantum computers are machines that take advantage of quantum phenomena such as entanglement and superposition for solving difficult mathematical problems which are infeasible for classical computers. In fact, there has been a significant amount of research in the past few years. Moreover, having said that if a large enough quantum computer comes into play, that would in fact put in danger all the integrity and confidentially of all the digital information everywhere. Therefore, it is the primary goal of post-quantum cryptography (also known as quantum-safe cryptography) to advance a cryptographic scheme safe from attacks by classical as well as quantum computers.

As research institutes and companies are all racing for quantum supremacy, according to P. Shor's algorithm [28] that is published in 1999, the currently hard problems such as Integer Factorization and Discrete Log problems are all breakable in polynomial time by a quantum computer with high enough number of qubits. Although no efficient quantum computer is built yet, it is anticipated that it may happen in a decade or so. That's why it is thoroughly been worked on in industry and academia as well as by government intelligence agencies. That is why new crypto schemes safe toward quantum computer attacks are of great interest to the National Institute of Standards and Technology, NIST.

NIST furthermore demonstrates the significance of building post-quantum cryptography schemes with the statement "Historically, it has taken almost two decades to deploy our modern public key cryptography infrastructure. Therefore, regardless of whether we can estimate the exact time of the arrival of the quantum computing era, we must begin now to prepare our information security systems to be able to resist quantum computing [24]." Therefore, back in 2016 NIST started a quantum-safe standardization process and called for submission of cryptosystems that are quantum-safe. The scope of the competition included submissions of digital signatures and KEM/PKE schemes. For that reason, cryptosystems based on five main classes of quantum-resistant problems are proposed, namely, lattice-based, code-based, hash-based, multivariate-based, and supersingular elliptic curve isogeny-based cryptography. In the beginning, there were 86 submissions, and then 69 of the submissions were deemed qualified for further consideration in the first round that was held in 2016. The second round was held in 2019 and 26 of the submissions made it to the third round. Amongst the finalists of the second round were 12 lattice-based schemes, 3 digital signatures, and 9 KEM/PKEs. Performance and security were the two major criteria in the process. In 2020, the third round of the competition was carried out and there were 7 finalists along with 8 alternatives [25].

In contrast, lattice-based algorithms possess two very time-consuming operations; despite random number generation, polynomial multiplication is the other most time-consuming operation. Additionally, the RLWE and MLWE based schemes are built upon operations with polynomials from polynomial rings $\mathbb{Z}_p[x]/f(x)$ where p is a prime and f is an irreducible polynomial over $\mathbb{Z}_p$. Furthermore, as polynomial multiplication is relatively the highest time-consuming operation in lattice-based cryptosystems, by using Number Theoretic Transform (NTT) one can obtain fast polynomial multiplication if the rings and primes are selected in a specific way. Thus, it is of great interest in the research community to explore the utilization of NTT for this polynomial multiplication which reduces the time complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$ as well as to see how far can we optimize it in different settings.

The fast Fourier transform (FFT) [12] is perhaps one of the most elegant results of the twentieth century that computes the discrete Fourier transform (DFT) of a sequence. A lot of the currently used technologies rely on it such as GSM and WIFI communications and many more applications in science. In fact, any technologies using signal processing make use of the concept of DFT with different underlying rings. Moreover, if one restricts the underlying ring in the DFT algorithm to be a finite field, we then get a variant of DFT called number theoretic transform (NTT), which is primarily used for optimizing integer and polynomial multiplication.

In addition, despite its simplicity, the schoolbook polynomial multiplication has a quadratic time complexity and it takes a lot of processing cycles. To overcome that there are other polynomial multiplication approaches to be used such as Karatsuba [17], Toom-Cook [11], and more importantly the number-theoretic transform which is a very efficient way to compute polynomial multiplication for some lattice-based schemes.

Some of the proposed post quantum schemes which utilize NTT for polynomial multiplications are NewHope [2, 4], NewHope Compact [3], Kyber [5], NTTRU [21].

Our contribution in this thesis is that we define the radix-3 NTT, develop a Cooley-Tukey-like butterfly as well as utilizing it for polynomial multiplication along with a Python implementation. Moreover, we use these results to propose two parameter sets for CRYSTALS-KYBER, a post-quantum KEM.

This thesis is organized as follows. Followed by this introduction is Chapter 2 that gives the necessary mathematical background to this work. In Chapter 3, we present the radix-3 NTT, how it functions and arithmetic complexity. We propose two parameter sets for KYBER in Chapter 4. The benchmarking and comparison to the implemented radix-3 NTT is given in Chapter 5. And we conclude this work in Chapter 6.

CHAPTER 2

MATHEMATICAL BACKGROUND

In this chapter, we recall and give some of the basics and definitions to provide the reader with the necessary mathematical background to better understand the topics covered. Most of the concepts follow from [13].

2.1 Mathematical Definitions

Definition 1. *For a positive integer* n *and a finite field* $\mathbb{F}_p$*, the* n^{th} *root of unity is* $w \in \mathbb{F}$ *such that* $w^n = 1$*.*

Definition 2. *A primitive* n^{th} *root of unity for a positive integer* n *in the field* $\mathbb{F}$ *is a root of unity* $w \in \mathbb{F}$ *such that* $w^k \neq 1$ *for any* $k < n$*.*

Definition 3 (Irreducible Polynomial)**.** *A monic polynomial* $f(x)$ *in the field* $\mathbb{F}[x]$ *is called irreducible if it does not have nontrivial factors over* $\mathbb{F}[x]$*.*

Definition 4 (Cyclotomic Polynomial)**.** *For a positive integer* m*, the* m^{th} *cyclotomic polynomial is an irreducible polynomial defined over the field* $\mathbb{F}[x]$ *that divides* $x^m - 1$ *but not* $x^k - 1$ *for a positive integer* $k < m$*.*

Cyclotomic polynomials are monic as well as their roots are roots of unity.

It is well known that there exits a unique polynomial that interpolates $n+1$ distinct pairs $(a_i, b_i) \in \mathbb{F}$ for $i = 0, 1, \ldots, n$ of degree at most n [14].

Definition 5. *(Centered Binomial Distribution [6]) Let n be a positive integer, the centered binomial distribution is denoted by β_η and defined as follows.*

$$\textit{Sample } (a_1, \ldots, a_\eta, b_1, \ldots, b_\eta) \leftarrow \{0,1\}^{2\eta} \textit{ and output } \sum_{i=1}^{n} (a_i - b_i)$$

2.2 Lattice-Based Cryptography

The history of lattices goes a long time back to J. S. Lagrange and C. F. Gauss. However, it was in 1996 that Miklós Ajtai [1] introduced the use of lattices to build computationally hard problems that could be used for cryptographic purposes.

Assume $\mathcal{B}$ is a basis of the n dimensional Euclidean space, a lattice $\mathcal{L}$ is a linear combination of the basis elements in $\mathcal{B}$. To put it another way, for $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_2\}$ a basis of $\mathbb{R}^n$, the lattice $\mathcal{L}$ is defined as

$$\mathcal{L} = \left\{ \sum_{i=0}^{n} a_i \mathbf{b}_i \quad : \quad a_i \in \mathbb{Z} \right\}.$$

Shortest vector problem (SVP) and the closest vector problem (CVP) [22] are the two most studied mathematically hard problems based on lattices. The SVP is defined as finding the shortest vector in a lattice having been given a basis of the lattice whilst CVP is finding the closest vector to a vector outside the lattice.

2.2.1 Learning with Error (LWE) Problem

Introduced by Regev [27], Learning with Error problems are believed to be hard problems for quantum computers. Their security is proved to be as difficult as the worst-case lattice problems. The LWE problmes can be viewed as the problem of solving a linear system over $\mathbb{Z}_p$ except for which an error is added making it a hard problem to solve.

Let $A = [\mathbf{a}_1, \ldots, \mathbf{a}_m] \in \mathbb{Z}_p^{n \times m}$, and an error vector $\mathbf{e} \in \mathbb{Z}_p^m$ chosen according to ψ. The LWE problem is to find the secret vector $s \in \mathbb{Z}_p^n$ given $b = \mathbf{s}A + \mathbf{e}$.

2.2.2 Ring-Learning with Error (R-LWE) Problem

The R-LWE is very similar to the LWE problem in context. However, instead of now having vectors from $\mathbb{Z}_p^n$, we have polynomials of degree less than n chosen from a polynomial ring $\mathcal{R}_p$ [20]. The cryptographic algorithms based on R-LWE problem have a key size of order n unlike those based on LWE that has order n^2. In addition, computations can be accelerated using rings. Furthermore, it is proved that the R-LWE problem is as hard as the LWE on ideal lattices [20].

2.2.3 Module-Learning with Error (M-LWE) Problem

The M-LWE [18] is merely an extension of the R-LWE. However, this time a vector of k polynomials are selected rather than one polynomial as in R-LWE. This vector is called a module, and k is said to be the rank of this module.

2.3 Schoolbook Polynomial Multiplication

Schoolbook or naive polynomial multiplication is the conventional way of polynomial multiplication. Suppose that $f(x)$ and $g(x)$ be two n-term polynomials, that is of degree $n-1$. Then, their multiplication $h(x) = f(x)g(x)$ of $2n$-term is calculated as follows. $f(x)=\sum_{i=0}^{n-1} f_i x^i$ and $g(x)=\sum_{i=0}^{n-1} g_i x^i$, therefore,

$$h(x)=\sum_{i=0}^{2n-1} h_i x^i \text{ such that } h_i=\sum_{j=0}^{i} f_i g_{i-j}.$$

The multiplication $h(x)$ has a complexity of order n^2.

There are several approaches in the literature for polynomial multiplication such as the schoolbook method, Karatsuba method, Took-Cook method, NTT method, etc.

2.4 Radix-2 NTT

The NTT polynomial multiplication evaluates the two polynomials at roots of unity, multiply them coefficient-wise, and then uniquely interpolates the results to the resulting polynomial. Figure 2.1 below illustrates this process.

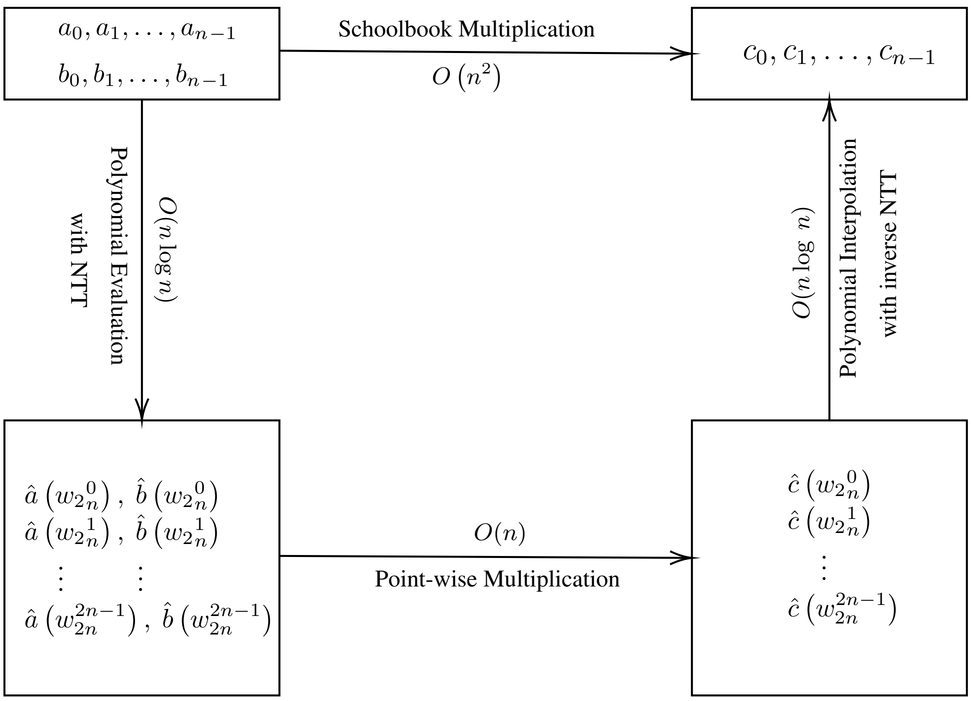

Figure 2.1: Discrete Fourier Transform Algorithm

In order to multiply two polynomials $g, h \in \mathcal{R}_p = \mathbb{Z}_p[x]/(x^n+1)$ for the parameters: an integer n power of 2 and a prime p such that $p \equiv 1 \mod 2n$ which provides the existence of w, a primitive $2n^{th}$ root of unity. Then one can efficiently compute the polynomial multiplication $f = gh \mod (x^n + 1)$ with NTT transformation. The multiplication is calculated by NTT as follows:

$$f = INTT(NTT(g) \circ NTT(h))$$

where NTT is forward NTT transformation, INTT is the inverse operation and $\circ$ is the componentwise multiplication of the vector elements. The formulas of NTT and INTT are given by

$$\hat{g} = NTT(a) = \sum_{i=0}^{n-1} \hat{g}_i x^i \text{ where } \hat{g}_i = \sum_{j=0}^{n-1} g_j w^{ij} \mod p,$$

$$g = INTT(\hat{g}) = \sum_{i=0}^{n-1} g_i x^i \text{ where } g_i = 1/n \sum_{j=0}^{n-1} \hat{g}_j w^{-ij} \mod p.$$

To carry out these transformations there are two separate methods. The first method is FFT (Fast Fourier Transform) method which uses the Cooley-Tukey butterfly algorithm [12] for calculating NTT and Gentleman-Sande butterfly algorithm [16] for INTT. The second method is called the Twisted FFT method [3] which uses the Gentleman-Sande butterfly algorithm for both NTT and INTT transformations.

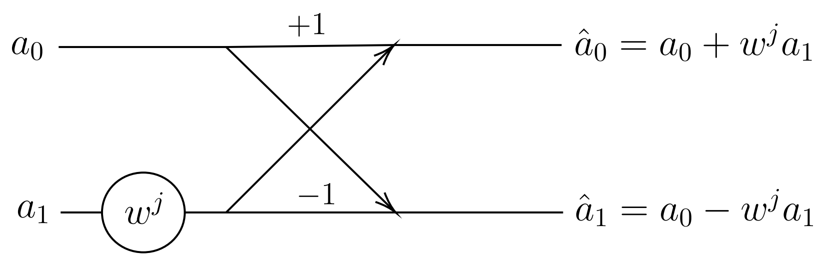

Figure 2.2: Cooley-Tukey Butterfly

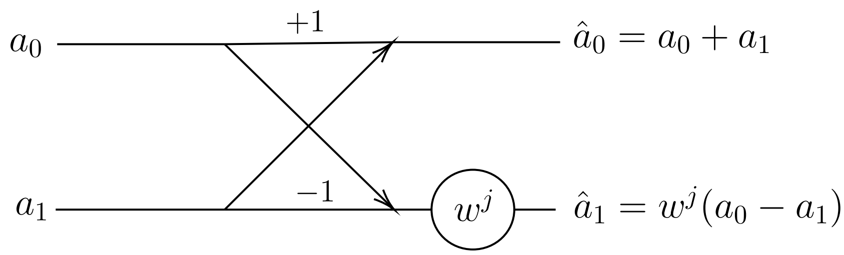

Figure 2.3: Gentleman-Sande Butterfly

CHAPTER 3

RADIX-3 NTT-BASED POLYNOMIAL MULTIPLICATION

3.1 Radix-3 NTT

The radix-3 NTT is defined on a modular polynomial ring $\mathcal{R}_p = \mathbb{Z}_p[x]/(f(x))$ where the modulus $f(x)$ is a cyclotomic polynomial allowing to evaluate a polynomial at its roots. The conventional definition is to evaluate a polynomial at the roots of unity w existing in the $\mathbb{Z}_p$, that is, for $f(x) = x^n - 1$ where $n = 3^\ell$, $\ell \in \mathbb{Z}$. However, computation over rings with such modulus $f(x)$ is not safe for cryptographic schemes based on the LWE problem as there are algebraic attacks against them [19]. Nevertheless, this could be utilized for schemes that do not require a cyclotomic modulus polynomial such as NTRU [9]. Here, our focus is on targeting cryptographic schemes based on LWE, and so we omit $f(x) = x^n - 1$. A cyclotomic polynomial that suits our goal here is $f(x)$ of the form $f(x) = x^{2n} + x^n + 1$ of degree $2 \cdot 3^\ell$. It is worth noting that, "Mixed-based FFT Multiplication Algorithms" [23] are introduced of length $n = k \cdot 2^\ell$ whereas considered multiples ℓ include 1 and the first three odd primes. Two recent works [3, 21] consider a ring of degree $3 \cdot 2^8$.

The ring $\mathcal{R}_p = \mathbb{Z}_p[x]/(x^{2n} + x^n + 1)$ allows efficient computation of full level NTT of polynomials at the roots of $x^{2n} + x^n + 1$ for $p \equiv 1 \mod 3n$. The ring possesses the following Chinese Remainder Theorem (CRT) isomorphism,

$$\mathcal{R}_p = \mathbb{Z}_p[x]/(x^{2n} + x^n + 1) \cong \mathbb{Z}_p[x]/(x^n - \alpha) \times \mathbb{Z}_p[x]/(x^n - \beta)$$

for $\alpha, \beta \in \mathbb{Z}_p$, $\alpha\beta = 1$ and $\alpha + \beta = -1$.

Therefore, if we have a polynomial in $\mathcal{R}_p$ and we would like to compute its NTT, we can instead map it to the frequency domain of the above CRT isomorphism, carry out its NTT over both rings, and finally use the inverse CRT map to get back the results in $\mathcal{R}_p$. Although this method would increase the arithmetic complexity, that is the best we could do.

The problem, now, has reduced to computing NTT in the rings of the form $\mathbb{Z}_p[x]/(x^n - \zeta)$. In the following section, we shall discuss this process.

3.2 NTT over $\mathbb{Z}_p[x]/(x^n - \gamma)$

Recall, NTT is just the evaluation of a polynomial at some special roots of the modulus $f(x)$ in the modular polynomial ring $\mathbb{Z}_p[x]/(x^n - \gamma)$. In this chapter, we develop what is called a radix-3 NTT over a finite field. Assume we have a polynomial $a(x)$ of length n that we would like to find its NTT where n is a power of 3. The algorithm evaluates the polynomial $a(x)$ at all the roots of $f(x)$. In order for the ring to allow computing NTT, it should contain a primitive n^{th} root of unity w and an n^{th} root of γ that we shall denote by ζ. Note that multiplying ζ by the powers of w yields all the roots of $x^n - \gamma$. As stated in the previous section, the modulus prime p is of the form $p \equiv 1 \mod 3n$.

Since $w^n = 1 \mod p$, this implies that $w^n - 1 = 0 \mod p$ or $(w^{n/3} - 1)(w^{2n/3} + w^{n/3} + 1) = 0 \mod p$, and so only $(w^{2n/3} + w^{n/3} + 1) = 0 \mod p$ as $w^{n/3} \neq 1 \mod p$. For simplicity we use $\mu = w^{2n/3}$. Thus, it is not difficult to prove that $x^n - \zeta^n = (x^{n/3} - \zeta^{n/3})(x^{n/3} - \mu\zeta^{n/3})(x^{n/3} - \mu^2\zeta^{n/3})$. Therefore, according to CRT, the ring decomposes as follows:

$$\mathcal{R}_p = \mathbb{Z}_p[x]/(x^n - \zeta^n) \to \mathbb{Z}_p[x]/(x^{n/3} - \zeta^{n/3}) \\ \times \mathbb{Z}_p[x]/(x^{n/3} - \mu\zeta^{n/3}) \times \mathbb{Z}_p[x]/(x^{n/3} - \mu^2\zeta^{n/3}) \quad (3.1)$$

In order to find the NTT of a polynomial $a(x) \in \mathbb{Z}_p[x]/(x^n - \gamma)$, we need to reduce $a(x)$ iteratively according to this above CRT map, as shown in Example 1. However, we would rather prefer to follow the definition below to compute it. We will denote the NTT form of $a(x)$ by $\hat{a}(x)$.

Example 1. Suppose that $a(x)$ and $b(x)$ be two polynomials over $\mathbb{Z}_p[x]/(x^9 - 63)$ for $p = 109$, where $a(x) = x + 5x^2 + 2x^3 + 7x^4 + 100x^5 + 43x^6 + 105x^7 + 17x^7$ and $b(x) = 3 + 77x + 21x^2 + 99x^3 + 53x^4 + 29x^5 + x^6 + x^7 + 4x^9$. Note that $w = 16$ is 9^{th} root of unity and $\mu = 63, \mu^2 = 45$. Now, we would like to find $c(x) = a(x)b(x) \mod (x^9 - 63)$ using the NTT method. For that purpose, we first split the modulus as:

$$\begin{aligned} x^9 - 63 &= (x^3 - 27) \cdot (x^3 - 66) \cdot (x^3 - 16) \\ &= (x - 3)(x - 80)(x - 16) \\ &\quad \cdot (x - 81)(x - 89)(x - 48) \\ &\quad \cdot (x - 7)(x - 5)(x - 97) \end{aligned} \quad (3.2)$$

We divide the solution in a few steps, and in each step the polynomials are expressed in smaller rings.

Step 1: Write $a(x)$ and $b(x)$ in $\mathbb{Z}_p[x]/(x^3-27)$, $\mathbb{Z}_p[x]/(x^3-66)$ and $\mathbb{Z}_p[x]/(x^3-16)$:

$$a(x) \to [53 + 108x + 56x^2, 87 + 43x + 106x^2, 78 + 70x + 71x^2]$$

$$b(x) \to [14x^2 + 57x + 26, 66x^2 + 83x + 102, 92x^2 + 91x + 99].$$

Step 2: We now write each of these component in their rings, according to the order appears in Equation (3.2) :

$$NTT(a) = [9, 90, 60, 19, 98, 35, 14, 23, 88]$$

$$NTT(b) = [105, 10, 72, 36, 99, 62, 12, 20, 47].$$

In order to compute $NTT(c)$, we point-wise multiply $NTT(a)$ and $NTT(b)$ and obtain

$$NTT(c) = [73, 28, 69, 30, 1, 99, 59, 24, 103].$$

To compute $c(x)$, we now need to compute $INTT(NTT(c))$, which can be done in the reverse order and obtained as

$$c = [54, 25, 70, 6, 90, 29, 41, 37, 69].$$

Definition 6 ([26])**.** *The NTT form of a polynomial $a(x) \in \mathbb{Z}_p[x]/(x^n-\gamma)$ represented in coefficient form as $a = [a_0, a_1, a_2, \ldots, a_{n-1}]$ is defined by*

$$\hat{a}(x) = \sum_{i=0}^{n-1} \hat{a}[i]x^i \qquad \textit{where} \quad \hat{a}[k] = \sum_{j=0}^{n-1} a[j]\ \zeta^j\, w^{jk},$$

where w is a primitive n^{th}-root of unity and ζ is an n^{th} root of γ.

Therefore, from this definition,

$$\begin{aligned}
\hat{a}[k] &= \sum_{j=0}^{n-1} a[j]\,(\zeta w^k)^j \\
&= \sum_{j=0}^{n/3-1} a[3j]\,(\zeta w^k)^{3j} + \sum_{j=0}^{n/3-1} a[3j+1]\,(\zeta w^k)^{3j+1}
\end{aligned}$$

$$
\begin{aligned}
& + \sum_{j=0}^{n/3-1} a[3j+2]\,(\zeta w^k)^{3j+2} \\
&= \underbrace{\sum_{j=0}^{n/3-1} a[3j]\,(\zeta w^k)^{3j}}_{A[k]} + \zeta w^k \underbrace{\sum_{j=0}^{n/3-1} a[3j+1]\,(\zeta w^k)^{3j}}_{B[k]} \\
& + (\zeta w^k)^2 \underbrace{\sum_{j=0}^{n/3-1} a[3j+2]\,(\zeta w^k)^{3j}}_{C[k]} \\
&= A[k] + \zeta w^k\, B[k] + \zeta^2 w^{2k}\, C[k] \qquad (3.3)
\end{aligned}
$$

Remark 1. *From the definition of A, B and C we can derive equivalences for $A[k]$, $B[k]$ and $C[k]$:*

1. $A[k+n/3] = \sum_{j=0}^{n/3-1} a[3j](\zeta\, w^{k+n/3})^{3j} = \sum_{j=0}^{n/3-1} a[3j](\zeta\, w^k)^{3j} = A[k]$,
2. $A[k+2n/3] = \sum_{j=0}^{n/3-1} a[3j](\zeta\, w^{k+2n/3})^{3j} = \sum_{j=0}^{n/3-1} a[3j](\zeta\, w^k)^{3j} = A[k]$,

for $k = 1, 2, \ldots, n/3$. Note that $w^n = 1$. Similarly, the same equalities hold for $B[k]$ and $C[k]$.

From Remark 1, we obtain that,

$$
\begin{aligned}
\hat{a}[k+n/3] &= A[k] + \zeta w^{k+n/3} B[k] + \zeta^2 w^{2(k+n/3)} C[k] \\
&= A[k] + \mu\,\zeta\, w^k B[k] + \mu^2\,\zeta^2\, w^{2k}\, C[k] \\
\hat{a}[k+2n/3] &= A[k] + \mu^2\,\zeta\, w^k\, B[k] + \mu\,\zeta^2\, w^{2k}\, C[k].
\end{aligned}
$$

Therefore, for $0 \leq k < n/3$,

$$
\begin{aligned}
\hat{a}[k] &= A[k] + \zeta w^k\, B[k] + \zeta^2 w^{2k}\, C[k], \\
\hat{a}[k+n/3] &= A[k] + \mu\,\zeta\, w^k B[k] + \mu^2\,\zeta^2\, w^{2k}\, C[k], \\
\hat{a}[k+2n/3] &= A[k] + \mu^2\,\zeta\, w^k\, B[k] + \mu\,\zeta^2\, w^{2k}\, C[k]. \qquad (3.4)
\end{aligned}
$$

Moreover, we can trade two multiplications by μ^2 to four additions. Since we have that $\mu^2 + \mu + 1 = 0$ substituting this for μ^2 in (3.4), we get:

$$\begin{aligned} \hat{a}[k] &= A[k] + \zeta w^k B[k] + \zeta^2 w^{2k} C[k] \\ \hat{a}[k+n/3] &= A[k] - \zeta^2 w^{2k} C[k] + \mu (\zeta w^k B[k] - \zeta^2 w^{2k} C[k]) \\ \hat{a}[k+2n/3] &= A[k] - \zeta w^k B[k] - \mu(\zeta w^k B[k] - \zeta^2 w^{2k} C[k]) \end{aligned} \tag{3.5}$$

In Algorithm 1, we give a pseudo-code to computing the NTT according to (3.5).

Theorem 1. *The arithmetic complexity of calculating a radix-*3 NTT *of length* n *is given by the recurrence relation:*

$$T_{NTT}(n) = 3 \cdot T_{NTT}\left(\frac{n}{3}\right) + n \cdot M + \frac{7}{3} n \cdot A \tag{3.6}$$

where $T_{NTT}(1) = 0$*,* n *is a power of* 3*,* M *stands for multiplication and* A *stands for addition over* $\mathbb{Z}_p$*. Moreover,*

$$T_{NTT}(n) = \ell\, n \cdot M + \frac{7}{3}\, \ell\, n \cdot A, \tag{3.7}$$

for $n = 3^\ell$ *and* $\ell \in \mathbb{Z}^+$*.*

Proof. In order to calculate the complexity, we will follow Algorithm 1. The first line scrambles the polynomial according to a ternary reversal function. This helps in returning the resulting polynomial in normal ordering. In fact, the butterflies are in line 9, 10 and 11, the other lines can all be pre-computed. Assume $\delta_1 = w_j b, \delta_2 = w_j^2 c$ and $\delta_3 = \mu(\delta_1 - \delta_2)$. Thus, line 9 could be computed with 2 multiplications and 2 additions. Line 10, can be computed with 1 multiplication and 3 additions. Moreover, line 11 can be computed with just 2 additions. Note that some operations are carried out in the steps before. Therefore, it takes 3 multiplications and 7 additions, each

Algorithm 1 NTT algorithm based on a Cooley-Tukey-like butterfly

Input: A polynomial $a(x) \in \mathbb{Z}_p[x]/(x^n - \zeta^n)$ where $a[i], i < n$ is the i^{th} coefficient of $a(x)$. The list pre-computed Γ of $\zeta w^j, j \in [0, n/3)$. Also, $\mu = w^{n/3} \mod p$ where w is a primitive n^{th} root of unity.

Output: $A \leftarrow NTT(a) = [A_0, A_1 \ldots, A_{n-1}]$.

$A = scramble(a, n)$ ▷ To arrange a so that A will be in normal ordering.
$\ell \leftarrow \log_3 n$
for each $level \in [1, \ell + 1)$ **do**
  $m \leftarrow 3^{level}$
  for each $j \in [0, m/3)$ **do**
    $w_j \leftarrow \Gamma[j]^{n/m}$
    for each $k \in [0, n/m)$ **do**
      $A \leftarrow A[km + j], B \leftarrow A[km + j + \frac{m}{3}], C \leftarrow A[km + j + 2\frac{m}{3}]$
      $A[km + j] \leftarrow A + w_j B + w_j^2 C$
      $A[km + j + \frac{m}{3}] \leftarrow A - w_j^2\, C + \mu\, (w_j B - w_j^2\, C)$
      $A[km + j + 2\frac{m}{3}] \leftarrow A - w_j\, B - \mu(w_j\, B - w_j^2\, C)$
    end for
  end for
end for
return A

of length $n/3$. This completes the proof of (3.6). The result (3.7) can be directly obtained by solving (3.6) for $n = 3^\ell$.

□

3.3 Inverse NTT over $\mathbb{Z}_p[x]/(x^n - \gamma)$

In the previous section, we presented a Cooley-Tukey-like butterfly to compute the forward NTT. In the present section, we shall take this further and reverse it to give a similar butterfly to compute its inverse operation.

To find the inverse transformation, recall Equation (3.4):

$$\begin{aligned}
\hat{a}[k] &= A[k] + \zeta w^k\, B[k] + \zeta^2 w^{2k}\, C[k] \\
\hat{a}[k+n/3] &= A[k] + \mu\, \zeta\, w^k B[k] + \mu^2\, \zeta^2\, w^{2k}\, C[k] \\
\hat{a}[k+2n/3] &= A[k] + \mu^2\, \zeta\, w^k\, B[k] + \mu\, \zeta^2\, w^{2k}\, C[k]
\end{aligned}$$

In matrix-vector form, this can be written as

$$\begin{pmatrix} \hat{a}[k] \\ \hat{a}[k+n/3] \\ \hat{a}[k+2n/3] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \mu & \mu^2 \\ 1 & \mu^2 & \mu \end{pmatrix} \begin{pmatrix} A[k] \\ \zeta w^k B[k] \\ \zeta^2\, w^{2k}\, C[k] \end{pmatrix}$$

By inverting the intermediate matrix, we can obtain that

$$\begin{pmatrix} A[k] \\ \zeta w^k B[k] \\ \zeta^2\, w^{2k} C[k] \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \mu^2 & \mu \\ 1 & \mu & \mu^2 \end{pmatrix} \begin{pmatrix} \hat{a}[k] \\ \hat{a}[k+n/3] \\ \hat{a}[k+2n/3] \end{pmatrix}$$

equivalently

$$\begin{aligned}
A[k] &= \frac{1}{3}\left(\hat{a}[k] + \hat{a}[k+n/3] + \hat{a}[k+2n/3]\right) \\
B[k] &= \frac{1}{3}\zeta^{-1} w^{-k}\left(\hat{a}[k] + \mu^2 \hat{a}[k+n/3] + \mu \hat{a}[k+2n/3]\right) \\
C[k] &= \frac{1}{3}\zeta^{-2} w^{-2k}\left(\hat{a}[k] + \mu \hat{a}[k+n/3] + \mu^2 \hat{a}[k+2n/3]\right). \qquad (3.8)
\end{aligned}$$

for $0 \leq k < n/3$. Furthermore, we can again trade two multiplications by μ^2 to four additions. Since we have that $\mu^2 + \mu + 1 = 0$ substituting this for μ^2 in (3.8), we get:

$$A[k] = \frac{1}{3}\left(\hat{a}[k] + \hat{a}[k+n/3] + \hat{a}[k+2n/3]\right)$$

$$B[k] = \frac{1}{3}\zeta^{-1}w^{-k}\left(\hat{a}[k] - \hat{a}[k+n/3] - \mu(\hat{a}[k+n/3] - \hat{a}[k+2n/3])\right)$$
$$C[k] = \frac{1}{3}\zeta^{-2}w^{-2k}\left(\hat{a}[k] - \hat{a}[k+2n/3] + \mu(\hat{a}[k+n/3] - \hat{a}[k+2n/3])\right) \quad (3.9)$$

The above butterflies are inverse of the NTT transformation and it is called as Gentleman-Sande-like butterfly [16]. We prefer to follow the Gentleman-Sande-like butterflies to calculating the inverse of the NTT transformation. The definition of calculating the inverse NTT is provided in Definition 7.

Definition 7 ([26])**.** *The inverse NTT of a polynomial $\hat{a}(x) \in \mathbb{Z}_p[x]/(x^n - \gamma)$ represented in coefficient form as $\hat{a} = [\hat{a}_0, \hat{a}_1, \hat{a}_2, \ldots, \hat{a}_{n-1}]$ is defined by*

$$a(x) = \sum_{i=0}^{n-1} a[i]x^i \qquad \textit{where} \quad a[k] = \frac{1}{n}\sum_{j=0}^{n-1} \hat{a}[j]\ \zeta^{-j}\, w^{-jk},$$

for w is a primitive n^{th}-root of unity and ζ is an n^{th} root of γ.

In Algorithm 2, we propose a pseudo-code to compute the inverse NTT using the Gentleman-Sande-like butterflies according to Equation (3.9).

Theorem 2. *The arithmetic complexity of calculating a radix-3 inverse NTT of length n is given by the recurrence relation:*

$$T_{INTT}(n) = 3 \cdot T_{INTT}\left(\frac{n}{3}\right) + n \cdot M + \frac{7}{3}n \cdot A \quad (3.10)$$

where $T_{INTT}(1) = 0$, n is a power of 3, M stands for multiplication and A stands for addition in $\mathbb{Z}_p$. Moreover,

$$T_{INTT}(n) = \ell\, n \cdot M + \frac{7}{3}\,\ell\, n \cdot A, \quad (3.11)$$

for $n = 3^\ell$ and $\ell \in \mathbb{Z}^+$.

Proof. Line 8 can be computed with 2 additions. Line 9 can be computed with 2 multiplication and 3 additions. Moreover, Line 10 can be computed with 1 multiplication

Algorithm 2 Inverse NTT algorithm based on a Gentleman-Sande like butterfly

Input: A polynomial $A(x) \in \mathbb{Z}_p[x]/(x^n - \zeta^n)$ where $A[i], i < n$ is the i^{th} coefficient of $\hat{a}(x)$. The list pre-computed Γ^{-1} of $\zeta^{-1}w^{-j}, j \in [0, n/3)$. Also, $\mu^{-1} = w^{-n/3} \mod p$ where w is a primitive n^{th} root of unity.

Output: $a \leftarrow INTT(A) = [a_0, a_1, \ldots, a_{n-1}]$.

$\ell \leftarrow \log_3 n$
for each $level \in [\ell, -1)$ **do**
  $m \leftarrow 3^{level}$
  for each $j \in [0, m/3)$ **do**
    $w_j^{-1} \leftarrow \Gamma^{-1}[j]^{n/m}$
    for each $k \in [0, n/m)$ **do**
      $A \leftarrow A[km + j], B \leftarrow A[km + j + \frac{m}{3}], C \leftarrow A[km + j + 2\frac{m}{3}]$
      $A[km + j] \leftarrow \frac{1}{3}(A + B + C)$
      $A[km + j + \frac{m}{3}] \leftarrow \frac{1}{3}w_j^{-1}(A - B - \mu^{-1}(B - C))$
      $A[km + j + 2\frac{m}{3}] \leftarrow \frac{1}{3}w_j^{-2}(A - C + \mu^{-1}(B - C))$
    end for
  end for
end for
$a = scramble(A, n)$ ▷ To get a in normal ordering.
return a

and 2 additions. Therefore, it takes 3 multiplications and 7 additions, each of length $n/3$.

□

3.4 NTT Polynomial Multiplication over $\mathbb{Z}_p[x]/(x^n - \gamma)$ and Its Complexity

Let $a(x)$ and $b(x)$ be two polynomials in $\mathbb{Z}_p[x]/(x^n - \gamma)$, their multiplication $c(x) = a(x)b(x) \mod (x^n - \gamma)$ utilizing NTT can be carried out via

$$c = INTT(NTT(a) \circ NTT(b)), \tag{3.12}$$

where $\circ$ represents point-wise multiplication.

Therefore, in order to multiply two polynomials, it takes two NTT operations and one inverse NTT operation as well as n point-wise multiplication. From Equations (3.7) and (3.11), we can conclude that the cost of multiplying $a(x)$ and $b(x)$ is given by:

$$\begin{aligned} T(n) &= 2 \cdot T_{NTT}(n) + T_{INTT}(n) + n \cdot M \\ &= (3\ell + 1)n \cdot M + 7\,\ell\, n \cdot A. \end{aligned} \tag{3.13}$$

3.5 NTT Polynomial Multiplication over $\mathbb{Z}_p[x]/(x^{2n} + x^n + 1)$ and Its Complexity

Recall that

$$\mathcal{R}_p = \mathbb{Z}_p[x]/(x^{2n} + x^n + 1) \cong \mathbb{Z}_p[x]/(x^n - \alpha) \times \mathbb{Z}_p[x]/(x^n - \beta)$$

for $\alpha, \beta \in \mathbb{Z}_p$, $\alpha\beta = 1$ and $\alpha + \beta = -1$.

Hence, in order to multiply two polynomials in $\mathcal{R}_p$, we can map them to the frequency domain, and perform the multiplication there with the NTT multiplication developed in the last sections. And at the end, by inverse CRT map, we get the multiplication result in $\mathcal{R}_p$.

It is not so difficult to compute the aforementioned CRT map for $2n = 2 \cdot 3^\ell$. Let $a(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{2n-1} x^{2n-1} \in \mathcal{R}_p$, then we have

$$a(x) \mod (x^n - \alpha) = (a_0 + \alpha \cdot a_n) + (a_1 + \alpha \cdot a_{n+1})x + \ldots + (a_{n-1} + \alpha \cdot a_{2n-1})x^{n-1}$$

$$\begin{aligned} a(x) \mod (x^n + (\alpha + 1)) &= ((a_0 - a_n) - \alpha \cdot a_n) + ((a_1 - a_{n+1}) - \alpha \cdot a_{n+1})x + \ldots \\ &+ ((a_{n-1} - a_{2n-1}) - \alpha \cdot a_{2n-1})x^{n-1} \end{aligned} \tag{3.14}$$

with merely $n - 1$ multiplications, $n - 1$ additions and $2(n - 1)$ subtractions. Since addition and subtraction have the same cost, we can just say $n - 1$ multiplication and $3(n - 1)$ additions. Note that to reduce the number of multiplications in this CRT conversion we benefited from the fact that $\beta = -1 - \alpha$ and increased the number of additions in return. This is a good idea to perform as the cost of multiplication is much more than that of addition.

On the other hand, the inverse CRT map to $\mathcal{R}_p$ can be performed as follows. Let $c(x) \in \mathcal{R}_p$ and $c_\alpha(x) \equiv c(x) \mod (x^n - \alpha)$ and $c_\beta(x) \equiv c(x) \mod (x^n - \beta)$, $c(x)$ can be recovered with

$$c(x) = \frac{1}{\alpha - \beta}\, c_\alpha \cdot (x^n - \beta) + \frac{1}{\beta - \alpha}\, c_\beta \cdot (x^n - \alpha). \tag{3.15}$$

The arithmetic cost of calculating $c(x)$ can be done with only $2n$ multiplications and $2n$ additions, apart from the coefficients in front. The multiplication of c_α and c_β by x^n can be dealt with just a shift in coefficients of c_α and c_β, respectively.

The total complexity of multiplying two polynomials in $\mathcal{R}_p$ can now be computed as

$$\begin{aligned} PM(n) &= 2 \cdot CRT + 2 \cdot T(n) + ICRT \\ &= [6(\ell + 1)n - 2] \cdot M + [2(7\ell + 4)n - 6] \cdot A \end{aligned} \tag{3.16}$$

where $\ell = \log_3 n$.

CHAPTER 4

PARAMETER PROPOSALS FOR KYBER

Kyber [6, 8] is one of the round three finalists of the NIST post-quantum competition. It is a lattice based post-quantum key encapsulation mechanism that uses a modified Fujisaki-Okamoto Transformation [15]. KYBER depends on the Module-LWE [18] unlike the other cryptosystems such as Frodo [7] and NewHope [2, 4] that depend on LWE and RLWE problems, respectively, where operations are of the form $As + e$ for all the variables being polynomials from the underlying ring. In Kyber, those variables are no longer polynomials. Instead, A is a square matrix of polynomial components, and s and e are vectors of polynomials. The scheme utilizes NTT polynomial multiplication over an NTT friendly ring $\mathbb{Z}_p[x]/(x^n + 1)$ where n is a power of two and p is a prime such that $p - 1$ is divisible by $n.$ Moreover, the polynomial ring is of fixed length throughout all the parameter sets. This transition helps to scale the security level between different parameter sets mainly through the dimension of the matrix/vectors.

Furthermore, the NTT utilized in Kyber is radix-2 NTT. That is, the dimension of the modulus is a power of two. In this chpater, we attempt to change the polynomial ring to what we considered in the last chapter.

We start with providing the algorithmic specifications of Kyber [6], the key generation, encryption, and decryption algorithms are given in Algorithms 3, 4 and 5.

Algorithm 3 KYBER.CPAPKE.KeyGen(): key generation

1: $\rho, \sigma \leftarrow \{0,1\}^{256}$

2: $\mathbf{A} \sim R_p^{k\times k} := \mathsf{Sam}(\rho)$

3: $(\mathbf{s}, \mathbf{e}) \sim \beta_{\eta_1}^k \times \beta_{\eta_2}^k := \mathsf{Sam}(\sigma)$

4: $\mathbf{t} := \mathbf{As} + \mathbf{e}$

5: **return** $(pk := (\mathbf{t}, \rho), sk := \mathbf{s})$

Algorithm 4 KYBER.CPA.Enc($pk = (\mathbf{t}, \rho), m \in \mathcal{M}$): encryption

1: $r \leftarrow \{0,1\}^{256}$

2: $\mathbf{A} \sim R_p^{k\times k} := \mathsf{Sam}(\rho)$

3: $(\mathbf{r}, \mathbf{e}_1, e_2) \sim \beta_{\eta_1}^k \times \beta_{\eta_2}^k \times \beta_{\eta_2} := \mathsf{Sam}(r)$

4: $\mathbf{u} := \mathsf{Compress}_p(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1, d_u)$

5: $v := \mathsf{Compress}_p(\mathbf{t}^T\mathbf{r} + e_2 + \lceil \frac{q}{2} \rfloor \cdot m, d_v)$

6: **return** $c := (\mathbf{u}, v)$

Algorithm 5 KYBER.CPA.Dec($sk = \mathbf{s}, c := (\mathbf{u}, v)$): decryption

1: $\mathbf{u} := \mathsf{Decompress}_p(\mathbf{u}, d_u)$

2: $v := \mathsf{Decompress}_p(v, d_v)$

3: **return** $\mathsf{Compress}_p(v - \mathbf{s}^T\mathbf{u}, 1)$

The original parameter set of KYBER is given in Table 4.1. As discussed above, we are trying to change the modulus polynomial in the ring from $x^n + 1$ to $x^{2n} + x^n + 1$. This would mean having different parameters. For the security levels and communication cost, we refer the reader to the round three KYBER submission documentation [6].

Table 4.1: Original Parameter Sets for KYBER

	n	k	p	η_1	η_2	(d_p, d_u, d_v)	δ
KYBER256	256	2	3329	3	2	$(12, 10, 4)$	2^{-139}
KYBER512	256	3	3329	2	2	$(12, 10, 4)$	2^{-164}
KYBER1024	256	4	3329	2	2	$(12, 11, 5)$	2^{-174}

To examine the ring $\mathbb{Z}_p[x]/(x^{2n} + x^n + 1)$ for Kyber, we consider that n is a power of three. A suitable p that allows utilizing a full level NTT is of the form $p \equiv 1(\mod 3n)$. Throughout the rest of this work, we assume $n = 3^4$ and $n = 3^5$. Moreover, we consider $k = 2, 3, 5$ and 6 so that we obtain parameter sets as close as to what was proposed in Kyber submission. In addition, we denote the secret distribution by η_1 and the error distribution by η_2. Therefore, we consider the following two sets throughout the rest of this work.

Table 4.2: Proposed Parameter Set 1

	n	$2n$	k
KYBER486	81	162	3
KYBER810	81	162	5
KYBER972	81	162	6

Table 4.3: Proposed Parameter Set 2

	n	$2n$	k
KYBER972	243	486	2
KYBER1458	243	486	3

The natural question, now, is how do we compute the security level, failure probability and communication cost. In the sections ahead, we address these concerns.

4.1 Calculating Security Level

The security level does not depend on the ring structure and instead it depends mainly on the hardness of the module-LWE. The parameters that impact the security are

$2n \times k$ and the primes p as well as the secret distribution η_1.

In that respect, the Kyber scripts can still be used to calculate the security level. We investigate two primes $p = 1459$ and $p = 2917$. There are smaller primes that satisfy $p \equiv 1 \mod 3n$, but they would not give us a negligible probability of failure. We only consider the latter prime for the second parameter set. In Tables 4.4, 4.5 and 4.6 the security levels are shown for both sets.

Moreover, the message space is 162-bits for the first set, and 486-bits for the second set. In first set, $m \in \{0, 1\}^{162}$, if one decides to encode one key bit per polynomial coefficient, we get 162-bits of entropy. This is way smaller than acceptable for the securities of KYBER810 and KYBER972 as in this case the attacker does not need to try to solve LWE guessing the shared secret much more easier for higher security parameters, say, greater than 180-bits security. Nevertheless, the message space in the second set provides higher entropy and hence security levels in Table 4.6 are more realistic.

Table 4.4: Security estimates for $2n = 162$ the prime $p = 1459$

	BKZ block size β	Classical Hardness	Quantum Hardness
KYBER486			
Primal Attack	365	106	96
Dual Attack	361	105	95
KYBER810			
Primal Attack	678	198	179
Dual Attack	666	194	176
KYBER972			
Primal Attack	840	245	222
Dual Attack	823	240	218

Table 4.5: Security estimates for $2n = 162$ and the prime $p = 2917$

	BKZ block size β	Classical Hardness	Quantum Hardness
KYBER486			
Primal Attack	365	106	96
Dual Attack	363	106	96
KYBER810			
Primal Attack	620	181	164
Dual Attack	610	178	161
KYBER972			
Primal Attack	770	225	204
Dual Attack	757	221	200

Table 4.6: Security estimates for $2n = 486$ and the prime $p = 2917$

	BKZ block size β	Classical Hardness	Quantum Hardness
KYBER972			
Primal Attack	779	225	204
Dual Attack	757	221	200
KYBER1458			
Primal Attack	1237	361	328
Dual Attack	1208	353	320

4.2 Calculating Failure Probability

The decryption failure very much depends on the coefficients under multiplication in the decryption equation as well as the number of bits that are dropped in the Compress and Decompress functions of the ciphertexts **u** and v in the decryption equation. The nature of these functions is the same, except that the public key is not compressed anymore. However, the coefficients under multiplication will increase due to the extra middle term, that is, x^n. Therefore below we will analyze the effect of the coefficients.

The decryption equation receives the two ciphertexts **u** and v and calculates,

$$v - \mathbf{s}^T\mathbf{u} = \mathbf{e}^T\mathbf{r} + e_2 + \mathbf{c}_v + \mathbf{r} - \mathbf{s}^T\mathbf{e}_1 - \mathbf{s}^T\mathbf{c}_u \tag{4.1}$$

failure occurs if the coefficient of this equation, in absolute value, happens to be greater than $p/4$.

The NTTRU scheme [21] considers a similar modulus polynomial $x^d - x^{d/2} + 1$. Therefore to illustrate how these multiplications work, we follow the same strategy. Let $a(x)$ and $b(x)$ be two polynomials defined over $\mathbb{Z}_p[x]/(x^6 + x^3 + 1)$, and we would like to compute their multiplication $c(x) = a(x)b(x) \mod (x^6 + x^3 + 1)$. This modular polynomial multiplication can be seen as follows:

$$\begin{aligned} ab \mod (x^6 + x^3 + 1) &= a\left(\sum b_i x^i\right) \mod (x^6 + x^3 + 1) \\ &= \left(\sum a x^i \mod (x^6 + x^3 + 1)\right) b_i \end{aligned} \tag{4.2}$$

Writing this on matrix-vector form

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = \begin{pmatrix} a_0 & -a_5 & -a_4 & -a_3 & a_5 - a_2 & a_4 - a_1 \\ a_1 & a_0 & -a_5 & -a_4 & -a_3 & a_5 - a_2 \\ a_2 & a_1 & a_0 & -a_5 & -a_4 & -a_3 \\ a_3 & a_2 - a_5 & a_1 - a_4 & a_0 - a_3 & -a_2 & -a_1 \\ a_4 & a_3 & a_2 - a_5 & a_1 - a_4 & a_0 - a_3 & -a_2 \\ a_5 & a_4 & a_3 & a_2 - a_5 & a_1 - a_4 & a_0 - a_3 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix}$$

Observe that $c_5 = [a_5 b_0 + (a_2 - a_5) b_3] + [a_4 b_1 + (a_1 - a_4) b_4] + [a_3 v_2 + (a_0 - a_3) b_5]$. If you notice the last three rows, they are all sum of three independently randomly distributed variables of of the form

$$c = ab + (a' - a)b', \text{ where } a, a', b, b' \leftarrow \beta_\eta. \tag{4.3}$$

Moreover, the first three rows also follow a certain formula. However, the general distribution is established by the bottom three rows. In general, the multiplication of $a(x)$ and $b(x)$ in $\mathcal{R}_p = \mathbb{Z}_p[x]/(x^{2n} + x^n + 1)$ is the sum (or difference) of n randomly distributed variables as in (4.3), and so the tail bounds of this one is computed.

In order to calculate the decryption error over $\mathcal{R}_p$, we can still keep using the KYBER scripts having changed the coefficient distribution to (4.3) in the script. Tables 4.7, 4.8 and 4.9 present our proposed parameter set 1 and 2. The probability of failure is denoted by δ.

Table 4.7: Proposed Parameter Set 1, $p = 1459$

	n	$2n$	k	p	η_1	η_2	(d_p, d_u, d_v)	δ
KYBER486	81	162	3	1459	1	2	$(11, 11, 5)$	2^{-144}
KYBER810	81	162	5	1459	1	1	$(11, 11, 5)$	2^{-135}
KYBER972	81	162	6	1459	1	1	$(11, 11, 6)$	2^{-120}

Table 4.8: Proposed Parameter Set 1, $p = 2917$

	n	$2n$	k	p	η_1	η_2	(d_p, d_u, d_v)	δ
KYBER486	81	162	3	2917	2	2	$(10, 10, 5)$	2^{-130}
KYBER810	81	162	5	2917	1	1	$(10, 10, 3)$	2^{-152}
KYBER972	81	162	6	2917	1	1	$(10, 10, 4)$	2^{-171}

Table 4.9: Proposed Parameter Set 2, $p = 2917$

	n	$2n$	k	p	η_1	η_2	(d_p, d_u, d_v)	δ
KYBER972	243	486	2	2917	1	2	$(11, 10, 4)$	2^{-158}
KYBER1458	243	486	3	2917	1	1	$(11, 10, 4)$	2^{-135}

4.3 Calculating Communication Cost

For the communication cost, the public key pk size is composed of $(\rho, \mathbf{t})$, therefore it is calculated by $256 + k \cdot (2n) \cdot d_p$ where d_p is the number of bits in the prime p. Furthermore, the ciphertext consists of $\mathbf{u}$ and v, therefore the size of the ciphertext is the size of **u** and v together, that is, $k \cdot (2n) \cdot d_u + (2n) \cdot d_v$ where d_u is the number of bits in the **u** and d_v is the number of the bits in v. Tables 4.10 and 4.11 shows the communication cost for the two parameter sets.

Table 4.10: Communication Cost (byte) for Parameter Set 1

	Set 1, $p = 1459$		Set 1, $p = 2917$	
	pk	**ct**	**pk**	**ct**
Kyber486	700	749	639	708
Kyber810	1145	1215	1044	1073
Kyber972	1368	1458	1247	1296

Table 4.11: Communication Cost (byte) for Parameter Set 2

	Set 2, $p = 2917$	
	pk	**ct**
Kyber972	1368	1458
Kyber1458	2036	2065

CHAPTER 5

BENCHMARKS AND COMPARISON

In this section, we test our implementation of the radix-3 NTT polynomial multiplication developed in Section 3. In Table 5.2, we measure the runtime for our parameters and compare it to the runtime of the radix-2 NTT used for KYBER parameters. We perform the tests for both the parameter sets we proposed in Section 4.

Implementations and Details. The NTT implementations are in SageMath programming language and can be found in the GitHub repository `https://github.com/ChenarHassan/NTT`. The benchmarking was carried out on Windows operating system with an Intel Core i7-5600U processor running at a base speed of 2.6 GHz . Moreover, the mean, maximum and minimum runtime of 10000 executions are reported.

In order to calculate the security level, as pointed out before, one can still use the KYBER scripts. The KYBER scripts are available in `https://github.com/pq-crystals/security-estimates`. However, for the decryption failure, we modified the coefficient distribution functions in the file `proba_util.py` as well as a very minor change in the file `Kyber_failure.py`. We also put these files in the GitHub repository above.

Due to the coefficient increase under multiplication, we observe that the complexity of multiplication in $\mathbb{Z}_p[x]/(x^{2n} + x^n + 1)$ increases. Table 5.1 depicts the arithmetic complexity of radix-2 NTT where degree of the modulus polynomial is $n' = 2^k, k = 8$ and our radix-3 NTT where degree of the modulus polynomial is $d = 2n = 2\times 3^\ell, \ell = 4$ and $\ell = 5$

Table 5.1: Arithmetic complexity of radix-2 NTT and radix-3 NTT polynomial multiplication

	d	multiplications	additions
radix-2 NTT [3]	256	3328	6144
radix-3 NTT [this work]	162	2428	5178
radix-3 NTT [this work]	486	8746	18948

Table 5.2: The runtime of NTT multiplication for the proposed parameter sets and KYBER's parameter

		runtime (in seconds)		
d	prime p	minimum	average	maximum
162 [this work]	1459	0.021	0.024	0.113
162 [this work]	2917	0.021	0.024	0.274
256 [3]	3329	0.028	0.032	0.123
486 [this work]	2917	0.074	0.081	0.332

CHAPTER 6

CONCLUSION

Although it is not yet disclosed when the fourth round of the NIST competition will be, it is anticipated that the fourth round will be the last and that a scheme or two will be selected as standardized. So far, lattice-based cryptography is the most promising, and the finalists of round three of NIST's competition 3 out 4 Key Encapsulation Mechanism and 2 out of 3 Digital Signatures are based on lattices. The lattice-based cryptosystems thus far have proven to be the most promising candidate for a quantum-resistant scheme. The competitions for the post-quantum standardization project of NIST is still going on. There were 4 public key encryption finalists in the third round, namely, Classic McEliece [10], Kyber [6], NTRU [9], and Saber [29]. The last three of these are based on lattices in fact.

In this thesis, we investigated the polynomial multiplication using the number-theoretic transform which is a powerful method for polynomial multiplications for schemes based on RLWEs. Furthermore, improvements of the number-theoretic transform come in the form of more efficient modular reduction algorithms as well as different parameter sets of the underlying ring.

As discussed, the NTT is widely used for polynomial multiplication over cyclotomic rings of a certain structure. Most of the schemes, if not all, adopt what is called a radix-2 NTT variant. In this thesis, we considered the radix-3 NTT and introduced a Cooley-Tukey-like butterfly algorithm to compute it. Moreover, we defined the NTT in the ring $\mathbb{Z}_p[x]/(x^{2n}+x^n+1)$ which could be utilized for polynomial multiplication. Furthermore, we computed its computational complexity.

In Section 4, based on the radix-3 NTT, we proposed some parameters for Kyber and calculated their security level, probability of failure and communication cost. In addition to that, in the last section, we provided the required number of arithmetic operations as well as tested the runtime of the radix-3 NTT and compared it with the radix-2 NTT used in KYBER. We have observed that for the first parameter set, the radix-3 NTT is more efficient than the radix-2 NTT used in KYBER. However, this does not remain true for the second parameter set.

REFERENCES

[1] M. Ajtai, Generating hard instances of lattice problems, in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 99–108, 1996.

[2] E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. de la Piedra, T. Pöppelmann, P. Schwabe, and D. Stebila, NewHope: Algorithm specifications and supporting documentation.

[3] E. Alkım, Y. A. Bilgin, and M. Cenk, Compact and simple RLWE based key encapsulation mechanism, in *International Conference on Cryptology and Information Security in Latin America*, pp. 207–216, Springer, 2019.

[4] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, Post-quantum key exchange—a new hope, in *25Th {USENIX} security symposium ({USENIX} security 16)*, pp. 327–343, 2016.

[5] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, CRYSTALS-Kyber algorithm specifications and supporting documentation, NIST PQC Round, 2, p. 4, 2019.

[6] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, CRYSTALS-Kyber algorithm specifications and supporting documentation, NIST PQC Round, 2020.

[7] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila, Frodo: Take off the ring! practical, quantum-secure key exchange from LWE, in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 1006–1018, 2016.

[8] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM, in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 353–367, IEEE, 2018.

[9] C. Chen, O. Danba, J. Hoffstein, A. Hülsing, J. Rijneveld, J. M. Schanck, P. Schwabe, W. Whyte, and Z. Zhang, NTRU algorithm specifications and supporting documentation, in *Second PQC Standardization Conference*, 2019.

[10] T. Chou, C. Cid, S. UiB, J. Gilcher, T. Lange, V. Maram, R. Misoczki, R. Niederhagen, K. Paterson, and E. Persichetti, Classic McEliece: conservative code-based cryptography, 10 october 2020, 2020.

[11] S. A. Cook and S. O. Aanderaa, On the minimum computation time of functions, Transactions of the American Mathematical Society, 142, pp. 291–314, 1969.

[12] J. W. Cooley and J. W. Tukey, An algorithm for the machine calculation of complex fourier series, Mathematics of computation, 19(90), pp. 297–301, 1965.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, MIT press, 2022.

[14] P. J. Davis, *Interpolation and approximation*, Courier Corporation, 1975.

[15] E. Fujisaki and T. Okamoto, Secure integration of asymmetric and symmetric encryption schemes, in *Annual international cryptology conference*, pp. 537–554, Springer, 1999.

[16] W. M. Gentleman and G. Sande, Fast fourier transforms: for fun and profit, in *Proceedings of the November 7-10, 1966, fall joint computer conference*, pp. 563–578, 1966.

[17] A. A. Karatsuba and Y. P. Ofman, Multiplication of many-digital numbers by automatic computers, in *Doklady Akademii Nauk*, volume 145, pp. 293–294, Russian Academy of Sciences, 1962.

[18] A. Langlois and D. Stehlé, Worst-case to average-case reductions for module lattices, Designs, Codes and Cryptography, 75(3), pp. 565–599, 2015.

[19] V. Lyubashevsky, Basic lattice cryptography: Encryption and fiat-shamir signatures, `https://drive.google.com/file/d/1JTdW5ryznp-dUBBjN12QbvWz9R41NDGU/view`, 2020.

[20] V. Lyubashevsky, C. Peikert, and O. Regev, On ideal lattices and learning with errors over rings, in *Annual international conference on the theory and applications of cryptographic techniques*, pp. 1–23, Springer, 2010.

[21] V. Lyubashevsky and G. Seiler, NTTRU: truly fast NTRU using NTT, IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 180–201, 2019.

[22] D. Micciancio and S. Goldwasser, *Complexity of lattice problems: a cryptographic perspective*, volume 671, Springer Science & Business Media, 2002.

[23] R. T. Moenck, Practical fast polynomial multiplication, in *Proceedings of the third ACM symposium on Symbolic and algebraic computation*, pp. 136–148, 1976.

[24] NIST, Post-quantum cryptography PQC, `https://csrc.nist.gov/Projects/post-quantum-cryptography`, 2017.

[25] NIST, NIST PQC: Round 3 finalists, `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions`, 2020.

[26] L. R. Rabiner and B. Gold, Theory and application of digital signal processing, Englewood Cliffs: Prentice-Hall, 1975.

[27] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM (JACM), 56(6), pp. 1–40, 2009.

[28] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM review, 41(2), pp. 303–332, 1999.

[29] I. F. Vercauteren, SABER: Mod-LWR based KEM (round 2 submission).