Lattice cryptography: A review of current literature

Alison Lin

July 28, 2021

1 Overview

The purpose of this report is to document the literature search findings, my reflection, and establish a landscape of the research trend on lattice cryptography in the past 10 years. Inspired by my preferred career path and job postings of particular interest, my research activities have primarily focused on 2 separate directions of lattice research: (a) implementation techniques of lattice-based crypto schemes, and (b) lattice-based zero knowledge proofs (ZKP). In the following sections, I outline and discuss my understanding and take-aways from the readings.

Lattice cryptography is one of the most popular areas of post-quantum cryptography. The security of lattice cryptosystems is based on the lattice problems, which are immune to any known attacks that utilize quantum computers. Thus, in recent years, there has been a strong interest in lattice cryptography.

The document is structured as follows: Section 2 will provide some background knowledge of lattice, LWE, and RLWE, which are relevant to the rest of this document. Section 3 and 4 document my findings for RLWE research trends related to implementation and ZKP respectively. Section 5 will outline some remaining questions about RLWE hardness reduction. Finally, Section 6 is my reflection about this literature research project.

This document will be uploaded onto my github link at [1].

2 Lattice, LWE, and RLWE

This section consists of the high level background and notation that will be used in this documents, and also the main efficiency barriers for RLWE - polynomial multiplication and Gaussian sampling. The details such as the exact formats of LWE and RLWE problems can be found in the articles listed in the reference section.

The security of lattice protocols are based on the hardness of these lattice problems - SVP(shortest vector problem), CVP (closest vector problem), and their variations such as SIVP (shortest independent vectors problem), SIS (short integer solution problem), sLWE (learning with error, search version), and dLWE (learning with error, decisional version). There has been a sequence of problem reduction studies to analyze the relation of their hardness, in the sense whether an algorithm that solves Problem A (if exists) can induce another algorithm to solve Problem B. For example, sLWE and dLWE are equally hard. And when using quantum algorithms, solving LWE is at least as hard as solving SIVP on any lattice (the worst case). This result is the so called "worst-case to average-case LWE reduction" [2].

Therefore, we know that LWE is hard enough as a foundation to generate crypto protocols. On the other hand, LWE is also quite versatile. Indeed, it has been used to generate a wide variety of crypto schemes, such as PKE (public key encrypion), IBE (identity based encryption), FHE (fully homomorphic encryption), ZKP, digital signature, KEX (key exchange), etc.

However, efficiency has been the problem with LWE schemes. In particular, its big key size and matrix-vector or matrix-matrix multiplication with $\mathcal{O}(n^2)$ complexity keeps LWE from being practical in reality. This is where RLWE (ring-LWE) came into play. Around year 2012, Vadim Lyubashevsky, Chris Peikert, and Oded Regev introduced RLWE, where the LWE concepts are converted from integers to polynomial rings [3]. The ring is usually chosen in the format of $\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1\rangle$ or $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$, where n is a power of 2, which implies

that $x^n + 1$ is a cyclotomic polynomial (explained in the below proposition). The ring (the q and n) is meant to be chosen in favor of NTT algorithm (Number Theoretic Transform) to reduce the polynomial multiplication complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$.

Proposition 2.1. $x^n + 1$ is irreducible in $\mathbb{Q}[x]$ if and only if n is a power of 2. In fact, for positive integer k, the cyclotomic polynomial $\Phi_{2^k}(x) = x^{2^{k-1}} + 1$.

Proof. If n is not power of 2, then n can be denoted as $n = km$ where m is odd. Thus $x^n + 1 = (x^k)^m + 1$ is reducible by the factor $x^k + 1$. Conversely, let $n = 2^k$ be a power of 2. We are going to show that $x^{2^{k-1}} + 1$ is in fact the 2^k-th cyclotomic polynomial $\Phi_{2^k}(x)$ and hence irreducible. Indeed, since $x^{2^k} - 1 = (x^{2^{k-1}} - 1)(x^{2^{k-1}} + 1)$ and the degree of $\Phi_{2^k}(x)$ is $\varphi(2^k) = 2^{k-1}$, $(x^{2^{k-1}} + 1)$ has to be $\Phi_{2^k}(x)$. □

Remark 2.2. For $\mathcal{R}_q$, since q is usually chosen to be prime, I'll just use the notation p instead of q, and write $\mathcal{R}_p$ instead of $\mathcal{R}_q$.

2.1 Polynomial Multiplication in RLWE and NTT

Remark 2.3. The naive polynomial multiplication scheme takes $\mathcal{O}(n^2)$ time. In particular, let $a = \sum_0^{n-1} a_k x^k, b = \sum_0^{n-1} b_k x^k \in \mathcal{R}_p = \mathbb{Z}_p[x]/\langle x^n + 1\rangle$, and $ab = c = \sum_0^{n-1} c_k x^k$. Then with basic calculation one can obtain that each $c_i = \sum_{j=0}^{i} a_j b_{i-j} - \sum_{j=i+1}^{n-1} a_j b_{n+i-j}$. It can also be written in the matrix format

$$(a_0 \cdots a_{n-1}) \begin{pmatrix} b_0 & b_1 & \cdots & b_{n-1} \\ -b_{-1} & b_0 & \cdots & b_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ -b_1 & -b_2 & \cdots & b_0 \end{pmatrix} = (c_0 \cdots c_{n-1}).$$

The idea of NTT is basically the CRT (Chinese Remainder Theory) over rings. In $\mathcal{R}_p = \mathbb{Z}_p[x]/\langle x^n + 1\rangle$ you first factorize $x^n + 1$ into irreducible polynomials over $\mathbb{Z}_p$. Thus by CRT, $\mathcal{R}_p$ can be decomposed into a cross product of fields (the NTT process). After performing the multiplication in each field component, you convert the result back to the original format $c = \sum_0^{n-1} c_k x^k$ (the NTT^{-1} process). The format of the formula

$$a \cdot b = NTT^{-1}(NTT(a) \odot NTT(b))$$

is self-explainable enough to reveal this CRT process.

Example 2.4. Taking $\mathcal{R}_{17} = \mathbb{Z}_{17}[x]/\langle x^4 + 1\rangle$ used in [4] as an example. Since $\Phi_8(x) = x^4 + 1 = (x + 2)(x - 2)(x + 8)(x - 8)$ in $\mathbb{Z}_{17}$, where $\{2, -2, 8, -8\}$ form the four 8th primitive roots of unity,

$$\mathcal{R}_{17} = \mathbb{Z}_{17}[x]/\langle x^4 + 1\rangle \cong \mathbb{Z}_{17}/\langle x + 2\rangle \times \mathbb{Z}_{17}/\langle x - 2\rangle \times \mathbb{Z}_{17}/\langle x + 8\rangle \times \mathbb{Z}_{17}/\langle x + 8\rangle \cong \mathbb{Z}_{17} \times \mathbb{Z}_{17} \times \mathbb{Z}_{17} \times \mathbb{Z}_{17}.$$

Hence, for $a = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ and $b = b_0 + b_1 x + b_2 x^2 + b_3 x^3 \in \mathcal{R}_{17}$, we can represent them as

$$NTT(a) = \begin{pmatrix} a(2) \\ a(-2) \\ a(8) \\ a(-8) \end{pmatrix} = \begin{pmatrix} 1 & 2 & 2^2 & 2^3 \\ 1 & -2 & (-2)^2 & (-2)^3 \\ 1 & 8 & 8^2 & 8^3 \\ 1 & -8 & (-8)^2 & (-8)^3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix},$$

$$NTT(b) = \begin{pmatrix} b(2) \\ b(-2) \\ b(8) \\ b(-8) \end{pmatrix} = \begin{pmatrix} 1 & 2 & 2^2 & 2^3 \\ 1 & -2 & (-2)^2 & (-2)^3 \\ 1 & 8 & 8^2 & 8^3 \\ 1 & -8 & (-8)^2 & (-8)^3 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}.$$

Thus we can calculate their product using pointwise multiplication in each of the $\mathbb{Z}_{17}$ field:

$$NTT(a) \odot NTT(b) = \begin{pmatrix} a(2) \cdot b(2) \\ a(-2) \cdot b(-2) \\ a(8) \cdot b(8) \\ a(-8) \cdot b(-8) \end{pmatrix}.$$

Remark 2.5. In the above example, the polynomial ring is in the format of $\mathcal{R}_{17} = \mathbb{Z}_p[x]/\langle x^4+1\rangle = \mathbb{Z}_{17}[x]/\langle \Phi_8(x)\rangle$. This is called the negacyclic convolution. If we define $\mathcal{R}_{17} = \mathbb{Z}_{17}[x]/\langle x^4-1\rangle$ then this is called the cyclic convolution and the NTT transformation matrix will look more straightforward. In particular, 4 is the 4th primitive root of unity and hence $\{1, 4, 4^2, 4^3\}$ form the roots of $x^4 - 1$. Therefore the NTT matrix for this cyclic convolution will be

$$\begin{pmatrix} 1 & 1 & 1^2 & 1^3 \\ 1 & 4 & 4^2 & 4^3 \\ 1 & 4^2 & (4^2)^2 & (4^2)^3 \\ 1 & 4^3 & (4^3)^2 & (4^3)^3 \end{pmatrix}.$$

Remark 2.6. The value of p can be chosen such that $\mathcal{R}_p = \mathbb{Z}_p[x]/\langle x^n+1\rangle$ or $\mathcal{R}_p = \mathbb{Z}_p[x]/\langle x^n-1\rangle$ are completely decomposed, i.e. such that $x^n + 1$ or $x^n - 1$ splits into linear factors in $\mathbb{Z}_p$ to enable NTT transformation. In particular, for $x^n - 1$, p can be chosen such that n divides $p-1$. Since $\mathbb{Z}_p^\times$ is a multiplicative cyclic group of order $p - 1$, the roots of $x^n - 1$ can be deduced from the generator of $\mathbb{Z}_p^\times$. Similarly, for $x^n + 1$, p can be chosen such that $2n$ divides $p - 1$. Thus the generator of $\mathbb{Z}_p^\times$ can induce the roots of $x^{2n} - 1$ and hence the roots of $x^n + 1$ because $x^{2n} - 1 = (x^n + 1)(x^n - 1)$.

In other words, by choosing p and n properly, we can ensure that $\mathcal{R}_p$ can be fully factorized into a number of $\mathbb{Z}_p$. This enables the polynomial multiplication to be performed pointwise in a number of fields, which is the essence of the NTT transformation.

The 2 most commonly used and efficient algorithms to compute NTT are the Cooley-Tukey [5], and Gentleman-Sande [6] algorithms. They are also called FFT (Fast Fourier Transform) sometimes. They achieve $\mathcal{O}(n \log n)$ time by employing the so called "butterfly operations" [7] where they carefully choose the pre-computed elements in the building blocks of NTT and NTT^{-1} processes, to minimize re-computation.

2.2 Guassian Sampling in RLWE

RLWE protocols often require sampling error polynomials from $\mathcal{R}_p$ in discrete Gaussian distribution. The most basic sampling algorithm is the rejection sampling whose idea is to sample elements from a distribution $g(x)$ and output/reject it with a pre-calculated probability, and then end up outputting samples in $f(x)$ distribution. The high rejection rate results in its inefficiency. Other than that, there are also the Ziggurat, CDT, Knuth-Yao, Bernoutlli, Binomial samplers. Each of them has different strengths in speed, size, flexibility, or suitable crypto schemes. Comparison between them can be found at some papers such as [8].

The quality of a sampler is given by the three tuple (σ, λ, τ) which respectively represents

- σ: standard deviation
- λ: precision parameter (statistical difference between a perfect and implemented sampler)
- τ: distribution tail-cut, i.e., random error e is sampled with the norm $|e| \in [0, \sigma \times \tau]$ instead of $[0, \infty)$.

In general, the smaller the σ, the less memory required to store pre-computed tables; the higher the λ, the more secure but the slower the sampler.

3 Implementation and Research Opportunities

There are lots of researches about implementing various RLWE protocols. In this section I'll list the types of implementations that I have ever seen in my literature search, an example for each type, and the research opportunities for the implementation work.

In the below categories the term "software implementation" indicates the programming level implementation using existing hardware (CPU/memory), where the programming can be the high level programming such as C/C++, or it can also be the low level programming such as assembly codes that optimize the instructions

operations on a selected processor. On the other hand, "hardware implementation" indicates dedicated hardware design for the algorithm, whose logics are performed directly with those designed gates (AND/OR/XOR, etc) or FPGA (field-programmable gate array), or hardware design for the space used to store the data in a certain way.

3.1 Recent Works

There are a number of studies for each type of recent implementation works. In this subsection I'll list one selected example for each of them. My findings could be biased because of my interest, but according to what I have seen so far,

(# of polynomial multiplication implementation) > (# of Gaussian sampling implementation)
(# of specific crypto scheme implementation) >> (# of generic operation implementation)

And in the specific crypto scheme implementation category,
(# of software implementation for specific scheme) >> (# of hardware implementation for specific scheme)

3.1.1 Implementation for Generic Operations

This kind of implementation proposes ideas to perform the generic processes in RLWE like polynomial multiplication and Gaussian sampling. Since they are not designed for any specific crypto scheme application, they can be used by most of the RLWE protocols.

- Polynomial multiplication

  It seems like that they are all based on FFT. At least I haven't seen anything using a fundamentally different algorithm on polynomial multiplication. From the implementation

  - Software implementation, for generic polynomial multiplication

    Paper: [9]

    In this study the authors introduced a new open source C++ library called NFLlib which contains optimized polynomial operations on $\mathcal{R}_p = \mathbb{Z}_p[x]/\langle x^n + 1\rangle$. The speed is achieved by

    * algorithm optimizations - fixed sized primes CRT, improved modular multiplication whose prime modulus are in a special range, Harvey's NTT algorithm [10]
    * programming level optimizations - take advantage of the parallel nature of polynomial operations by using SSE and AVX2 instruction sets.

    Then the authors use NFLlib for the RLWE encryption scheme and homomorphic encryption and compare their efficiency with other libraries such as NTL, FLINT, HElib.

  - Hardware implementation, for generic polynomial multiplication

    Paper: [11]

    In this study the authors proposed a pipelined architecture to obtain an efficient polynomial multiplier and experimented it on a Spartan-6 FPGA. The speed is achieved by

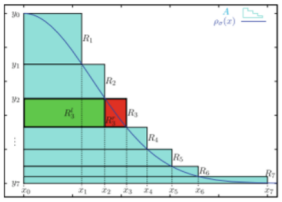
    * selecting the ring parameters (p and n) that meets security requirement, allows efficient modular reduction by p, and allows the existence of the components (ω, primitive n-th root of the unity in $\mathbb{Z}_p$ and ϕ with $\phi^2 \equiv \omega$) used in the negative wrapped convolution algorithm to calculate NTT.
    * taking advantage of the fact that NTT^{-1} is different from NTT in the multiplication by ω^{-i} instead of ω^i in the FFT algorithm, which allows a part of NTT processing unit to be reused to calculate NTT^{-1}.

- Gaussian sampling

- Software implementation, for generic Gaussian sampling

  Paper: [12]

  * Discrete Ziggurat sampler [12] implemented in C++ offers flexible tradeoff between memory, precision, and execution time. Below is the curve of a pdf (probability density function) function. The more refinement and more rectangles used, the more memory consumption, and the better performance/precision.



- Hardware implementation, for generic Gaussian sampling

  Paper: [13]

  There does exist some of them, such as this Roy's implementation [13] of Knuth-Yao sampler with small standard deviation (the σ) on a Xilinx Virtex5 FPGA. But I am not interested in hardware design for Gaussian sampling at all so didn't really spend time on searching this area.

3.1.2 Implementation for a Particular Crypto Scheme

This kind of implementation is more targeting on a specific RLWE crypto scheme instead of a general process, for instance, a specific public key encryption or a key exchange scheme such as NewHope. Therefore the enhancement could be more customized and specific to the needs based on the design of the scheme. Given that RLWE is versatile in constructing crypto schemes as mentioned earlier, there are overwhelmingly many implementation studies across a wide variety of RLWE-based crypto schemes in both software and hardware. In this section I'll list one software implementation study for public key encryption, and homomorphic encryption.

A lot of this kind of implementation apply a combination of both new and existing techniques that can enhance any component of the protocol. For example, [14] speeds up the PKE by using Knuth-Yao algorithm [15] for Gaussian sampling, and using [16] for the polynomial multiplication. Some of other implementation also use Barrett [17] or Montgomery [18] reduction for integer modulus reduction.

- Software implementation, for a PKE (public key encryption) scheme

  Paper: [14]

  In this study, the encryption scheme that it implements is the general one proposed along with the introduction RLWE paper mentioned earlier [3]. This is a software design implemented on the ARM Cortex-M4F, which is a popular embedded processor.

  - It speeds up the gaussian sampling using Knuth-Yao sampling algorithm [15]
  - It speeds up polynomial multiplication

    (1) by using negative-wrapped NTT along with computational optimizations from [16], which involves instruction-level parallelization.

    (2) by embedding multiple coefficients into one large word to allow load/store operations to be performed within one single instruction.

  Then it writes codes and compares its performance with other implementations of the same scheme.

  Remark 3.1. Based on the other similar studies that it compared to and listed, by the time this paper was written (2014), there had already existed a handful other similar researches. Considering that the RLWE and the PKE scheme was introduced only 2 year ahead of it in 2012, it implies that this area of study is moving very fast.

- Software implementation, for an HE (homomorphic encryption) scheme

  Paper: [19]

  This study introduced a C++ open-source library called HElib that implements a RLWE-based HE scheme named BGV, along with some optimizations to speed up the homomorphic operations on ciphertexts.

  In order to illustrate the 2 optimization approaches usedin HElib, let's take a brief look at the format of the secret key and ciphertext in BGV. The secret key $\mathtt{sk}$ is chosen from $\mathcal{R} = \mathbb{Z}[x]/\langle\Phi_N(x)\rangle$. BGV involves a selection of a sequence of decreasing moduli $q_L > q_{L-1} > \cdots > q_0$ and the ciphertext is an $L+1$ tuple with the format $v_i = (c_0, c_1) \in \mathcal{R}_{q_i}^2$, where $\mathcal{R}_{q_i} = \mathbb{Z}_{qi}[x]/\langle\Phi_N(x)\rangle$.

  Below are its 2 optimization approaches for ciphertext addition and multiplication.

  - Double-CRT

    The idea of "double" refers to decomposing both of the $\mathbb{Z}_{qi}$ and the $\Phi_N(x)$ part of $\mathcal{R}_{q_i} = \mathbb{Z}_{qi}[x]/\langle\Phi_N(x)\rangle$ using integer-CRT and ring-CRT respectively. In particular, pick small primes $p_0, ..., p_L$ such that $\Phi_N(x)$ splits (has $\phi(N)$ roots) over $\mathbb{Z}_{p_i}$ for all i. And define $q_\ell := \prod_{j=0}^{\ell} p_j$ for each $\ell = 0, ..., L+1$. This allows you to represent a polynomial in R_{q_ℓ} as an $(\ell+1) \times \phi(N)$ matrix, where each row is a ring-CRT representation, and each column is the integer-CRT representation. Thus the entries of the matrix are all coefficients, and the addition and multiplication in $\mathcal{R}_{q_\ell}$ are both entry-wise computation.

  - Key switching

    It provides a trick to convert a pair of ciphertext and its decryptable secret key $(c, \mathtt{sk})$ into another pair $(c', \mathtt{sk}')$ where c' can be decrypted by sk'. This conversion optimizes the process of getting a "canonical ciphertext" (defined below).

    Remark 3.2. A canonical ciphertext in BGV is a ciphertext $(c_0, c_1) \in \mathcal{R}_q^2$ (here q here means one of $q_0, ..., q_L$) such that $m := [c_0 + c_1\mathtt{sk}]_q$ in $\mathcal{R}_q$ is a polynomial with small norm (small coefficients), and the corresponding plaintext of this ciphertext is the binary polynomial $[m]_2 \in \mathcal{R}_2$.

3.2 Research Opportunities

There seems to be rich research opportunities in this area, not only because there keeps coming new RLWE-based crypto schemes and new enhancement techniques in these days. But also, a concept or technique introduced in one implementation may likely be useful in some other circumstances. For example, in 2016 the authors of [20] presented a new modular reduction and used it in their software implementations of NTT. Then in 2017 the new modular reduction then immediately replaced the Barrett reduction in a hardware implementation of NewHope in [21].

4 Zero Knowledge Proof and Research Opportunities

As mentioned earlier in this document, ever since RLWE was introduced [3], there has been an active track of RLWE-based protocols and their related researches, such as PKE, HE, digital signature, etc. ZKP drew my attention because of its high demands in the job market, and also its taking advantages of algebra structures, compared to other RLWE-based crypto schemes. Especially Vadim Lyubashevsky and his students/colleagues have been presenting a series of interesting results in ZKP in the past two years, which involves some interesting application of algebra and number theory.

In what follows I'll start by illustrating the approximate ZKP problem and why invertibility in $\mathcal{R}_p$ matters, since a few researches I am going to cover in this section have to do with these topics. Then I'll give some research samples to show how algebra and number theory were used to construct or enhance the efficiency of ZKP, based on my understanding of the papers. Then I'll briefly describe how they gave rise to some other useful ZKPs such as proving the knowledge of integer relation, followed by some research opportunities.

4.1 Approximate ZKP and invertibility

Let's say a prover wants to prove to a verifier that he knows a secret vector s over $\mathcal{R}_p = \mathbb{Z}_p[x]/\langle x^n + 1\rangle$ with short norm such that $As = t$ where A is a matrix over $\mathcal{R}_p$. The prover can first randomly pick another secret vector y and send $w := Ay$ to the verifier. Then the verifier picks a challenge value $c \in \mathcal{R}_p$ from a challenge set $\mathcal{C}$ and send back to the prover. Then the prover uses his secret s and y to calculate and send $z := sc + y$ back to the verifier (you can see that the purpose of y is to hide s). The verifier checks if z has small norm and $Az = ct + w$. The equation holds because $Az = A(sc + y) = cAs + Ay = ct + w$.

Assuming that a prover has shown that no matter the given challenge is c or c', he is always able to provide z or z' that passes the check $Az = ct + w$ or $Az' = c't + w$. However this doesn't necessarily show that the prover knew s and generated z using it. It at most shows that the prover knows some z and z' satisfying $Az = ct + w$ and $Az' = c't + w$, and hence $A(z - z') = (c - c')t$. In other words, the prover ends up proving the knowledge of a small norm vector $\bar{s}$ such that $A\bar{s} = \bar{c}t$, where $\bar{s} = z - z'$ and $\bar{c} = c - c'$.

Remark 4.1. However if this $\bar{c}$ is invertible, i.e., $\bar{c}$ has an inverse in $\mathcal{R}_p$, then this s must exist, because $A\frac{\bar{s}}{\bar{c}} = t = As$ which implies that $\frac{\bar{s}}{\bar{c}}$ has to equal s otherwise the difference of them is a solution to the ring-SIS problem. Therefore it is reasonable to conclude that an ideal challenge set $\mathcal{C}$ should at least satisfy 3 properties: the non-zero elements in the difference set $\mathcal{C} - \mathcal{C}$ are invertible, $|\mathcal{C}|$ is large enough, and the elements in $\mathcal{C}$ have small norms.

4.2 Interesting ZKP Researches

4.2.1 Invertibility on small norm elements in $\mathcal{R}_p$

In this subsection I'll illustrate how [22] views and addresses challenge set selection problems by their findings in the element invertiblility in $\mathcal{R}_p$.

Let $\mathcal{R}_p = \mathbb{Z}_p[x]/\langle x^{256} + 1\rangle$. Recall from the above remark that an ideal challenge set $\mathcal{C}$ has these properties: the non-zero elements in $\mathcal{C} - \mathcal{C}$ are invertible, $|\mathcal{C}|$ is large enough, and the elements in $\mathcal{C}$ have small norms.

Now here is the problem that this paper attempts to address.

The cost of collecting a challenge set $\mathcal{C}$ with at least 2^{256} invertible elements is that some $c \in C$ can have large norms. In particular, suppose $x^{256} + 1$ splits into k irreducible polynomials, then it can be proved that all of these irreducible polynomials have degree $\frac{256}{k}$, and hence $\mathcal{R}_p$'s "CRT component fields" all have degree $\frac{256}{k}$. A challenge set can be constructed by

$$\mathcal{C} := \{c \in \mathcal{R}_p \mid \deg(c) < \frac{256}{k},\ \|c\|_\infty \leq \gamma\}$$

where $\gamma \approx 2^{k-1}$, and $\|x\|_\infty = \|\sum_0^{n-1} a_i x_i\|_\infty = \max_0^{n-1} |a_i|$ represents the regular ℓ_∞ norm.

- Invertibility: by CRT these elements are non-zero in all of $\mathcal{R}_p$'s component fields and hence are invertible.
- Set size: with a little counting work one can tell that $|\mathcal{C}| = \gamma^{\lfloor \frac{256}{k} \rfloor - 1} \approx 2^{256}$.
- Norm: the problem is with the ℓ_2 norm $\|c\| = \sqrt{\gamma^2 + \cdots + \gamma^2} = \sqrt{\frac{256}{k}\gamma^2} = \sqrt{\frac{256}{k}}\gamma \approx \sqrt{\frac{256}{k}} 2^{k-1}$ which is big.

To summarize, the restriction on the degree and the set size leads to its big norm. It would be so ideal if one can simply just collect the small norm elements for $\mathcal{C}$ and they are magically invertible.

The paper addresses this problem by discovering some algebra properties in this kind of polynomial rings, and deducing that in some ways small norm itself guarantees invertibility! The below theorem and corollary are the main contribution in this paper, and the discovered algebra relation have also been used in other researches already [23].

Theorem 4.2. Let $m = \prod p_i^{e_i}$ for $e_i \geq 1$ and $z = \prod p_i^{f_i}$ for any $1 \leq f_i \leq e_i$. If p is a prime such that $p \equiv 1 \pmod{z}$ and $ord_m(p) = m/z$, then the polynomial $\Phi_m(x)$ factors as $\Phi_m(x) \equiv \prod_{j=1}^{\phi(z)} (x^{m/z} - r_j) \pmod{p}$ for distinct $r_j \in \mathbb{Z}_p^*$ where $x^{m/z} - r_j$ are irreducible in the ring $\mathbb{Z}_p[x]$. Furthermore, any y in $\mathbb{Z}_p[x]/\langle \Phi_m(x) \rangle$ that satisfies either $0 < \|y\|_\infty < \frac{1}{s_1(z)} \cdot p^{1/\phi(z)}$ or $0 < \|y\| < \frac{\sqrt{\phi(m)}}{s_1(m)} \cdot p^{1/\phi(z)}$ has an inverse in $\mathbb{Z}_p[x]/\langle \Phi_m(x) \rangle$.

Corollary 4.3. Let $n \geq k > 1$ be powers of 2 and $p = 2k + 1 \pmod{4k}$ be a prime. Then the polynomial $x^n + 1$ factors as $x^n + 1 \equiv \prod_{j=1}^{k} (x^{n/k} - r_j) \pmod{p}$ for distinct $r_j \in \mathbb{Z}_p^*$ where $x^{n/k} - r_j$ are irreducible in the ring $\mathbb{Z}_p[x]$. Furthermore, any y in $\mathbb{Z}_p[x]/\langle x^n + 1 \rangle$ that satisfies either $0 < \|y\|_\infty < \frac{1}{\sqrt{k}} \cdot p^{1/k}$ or $0 < \|y\| < p^{1/k}$ has an inverse in $\mathbb{Z}_p[x]/\langle x^n + 1 \rangle$.

The paper also analyzes the splitness of $x^n + 1$ over $\mathbb{Z}_p$. Suppose $x^n + 1$ can be factored into k irreducible polynomials. In general, the larger the k, the more the computation can be simplified using CRT (or FFT). However the larger the k, the smaller norm an ring element has to be to guarantee its invertibility based on the above Corollary, and hence the larger the prime p has to be. For example, the paper states that if it need to split into 16 or 32 factors, then p would need to be $p > 2^{48}$. The paper seems to believe the best option is $k = 8$ and p between 2^{20} to 2^{29}.

Remark 4.4. In this talk [24] Gregor Seiler gave an intuitive way to imagine how $x^n + 1$'s splitting too much could harm the invertibility. If it splits completely into linear factors, then each CRT component of $\mathcal{R}_p$ has only p elements. Hence if $|\mathcal{C}| > p$, there must exist $c, c' \in \mathcal{C}$ such that $c = c'$ in at least one of the CRT component ring. It implies that $\bar{c} = c - c'$ is 0 in that component and hence $\bar{c}$ cannot be invertible.

4.2.2 Practical ZKP for Multiplicative Relation

The papers [25] and [26] proposed a ZKP for the multiplicative relation $m_3 = m_1 m_2$, where m_1 and m_2 are "committed" elements in $\mathcal{R}_p = \mathbb{Z}_p[x]/\langle x^n + 1 \rangle = \mathbb{Z}_p[x]/\langle x^{128} + 1 \rangle$ (for the convenience of the explanation here, say $n = 128$).

Remark 4.5. "committed" is a term of commitment protocol that represents pre-selected un-changable values chosen by joint parties, which can be opened (or revealed) afterwards. Before the opening phase, the committed values remain unknown and can't be tampered. The application of a commit scheme can be something like coin flipping remotely. Some researches propose ZPK for various kinds of relation (e.g. linear, addition, product) for these committed values without revealing their contents. This paper is one of them.

Below I'll try to explain some of their contribution without giving the details of the ZKP protocol itself.

- Remove the restriction of invertibility and splitness

  The ZKP is constructed based on the same approximate ZKP and invertibility problems mentioned earlier in this section. Therefore recall from Remark 4.4 that in order to have large enough challenge set, $x^{128} + 1$ can not be factored into too many factors. Below I'll explain how the proposed ZKP allows $x^{128} + 1$ to split completely by making some non-invertible c's remain useful in the protocol.

  First of all, choose p such that $x^{128} + 1$ splits over $\mathbb{Z}_p$, i.e., $x^{128} + 1 = (x - r_1) \cdots (x - r_{128})$. Note that an element $c \in \mathcal{R}_p$ being invertible means that c is non-zero in every CRT component field of $\mathcal{R}_p$. The paper designs ZKP in a way such that it only requires the existence of c_j such that $c_j \mod x - r_j$ is non-zero in the component field $\mathbb{Z}_p[x]/\langle x - r_j \rangle$ for every j, and is still able to use the SIS problem (using module-SIS instead of ring-SIS) to ensure its security.

  The below explains how it collects such c using the algebra structure. The cyclotomic polynomial $\Phi_{256}(x) = x^{128} + 1$ can be factorized into

$$x^{128}+1=[x^4-\alpha_1][x^4-\alpha_2]\cdots[x^4-\alpha_{32}]=[(x-r_1)\cdots(x-r_4)]\cdots[(x-r_{125})\cdots(x-r_{128})]$$

over $\mathbb{Z}_p$ [refer to the website [7] for more concrete expression], and hence

$$\mathbb{Z}_p[x]/\langle x^{128}+1\rangle=\mathbb{Z}_p[x]/\langle x^4-\alpha_1\rangle\times\cdots\times\mathbb{Z}_p[x]/\langle x^4-\alpha_{32}\rangle=\mathbb{Z}_p[x]/\langle x-r_1\rangle\times\cdots\times\mathbb{Z}_p[x]/\langle x-r_{128}\rangle$$

Take $x^4-\alpha_1=(x-r_1)\cdots(x-r_4)$ as an example, let $\mathrm{Aut}(\mathcal{R}_p)$ the group of automorphisms on $\mathcal{R}_p$. The paper obtained a cyclic subgroup $\langle\sigma\rangle\subseteq\mathrm{Aut}(\mathcal{R}_p)$ that stabilizes every such ideal $\mathbb{Z}_p[x]/\langle x^4-\alpha_1\rangle$, and σ nicely rotates the value of an element modulo $\mathbb{Z}_p[x]/\langle x-r_1\rangle$, $\mathbb{Z}_p[x]/\langle x-r_2\rangle$, $\mathbb{Z}_p[x]/\langle x-r_3\rangle$, $\mathbb{Z}_p[x]/\langle x-r_4\rangle$. It's like the idea of Galois group but instead of on field extension this is on the ring $\mathcal{R}_p$ and it's CRT component rings. As long as $\bar{c}\neq 0 \mod x^4-\alpha_1$, there exists $j\in\{1,...,4\}$ such that $\bar{c}\neq 0 \mod \mathbb{Z}_p[x]/\langle x-r_j\rangle$, and because σ rotates the values modulo $\langle x-r_j\rangle$, for each j there exists $k\in\{1,...,4\}$ such that $\sigma^k(\bar{c})\neq 0 \mod \mathbb{Z}_p[x]/\langle x-r_j\rangle$.

Since the protocol is designed in a way that only requires $\sigma^k(\bar{c})$ to be non-zero in one of the CRT component field instead of all component fields like units, such non-units are still useful in the ZKP.

- Keep proof size small - have the verifier calculate a part of the proof materials

  Recall that the goal is to prove $m_1m_2=m_3$ without revealing any of them. It started with the idea to have the prover send $f_1:=y_1+cm_1, f_2:=y_2+cm_2$ to the verifier, where y_1 and y_2 are again used to hide m_1 and m_2, like all of the previous examples in this section. Thus $f_1f_2=(y_1y_2)+(y_1m_2+y_2m_1)c+(m_1m_2)c^2$. So if the prover also sends those "garbage terms" $g_0:=(y_1y_2)$, $g_1:=(y_1m_2+y_2m_1)$ and proves that $f_1f_2=g_0+g_1c+(m_3)c^2$, then he is able to prove that $m_1m_2=m_3$.

  However this approach requires many proof data to be sent, in particular the prover needs to provide those f_i's, y_i's and the garbage terms in order to hide m_i's, and also needs to prove that f_i has the format $f_i=y_i+cm_i$. The paper proposed a way to have the verifier calculate the f_i's using these ingredients from the prover: $z:=y+cr$, $t_i:=\langle b_i,r\rangle+m_i$ for $i=1,2,3$, and $t_4:=\langle b_4,r\rangle+\langle b_3,y\rangle-m_1\langle b_2,y\rangle-m_2\langle b_1,y\rangle$ for canceling the garbage terms [note: b_i is public, r is privately chosen by the prover, $\langle\cdot,\cdot\rangle$ is the inner product]. The verifier caculates $f_i:=\langle b_i,z\rangle-ct_i$. With some basic linear calculation, one can see that this f_i has the format $f_i=\langle b_i,y\rangle-cm_i$ which is automatically in the format of cm_i being hidden by some "y" where y is unknown to the verifier.

- Keep proof size small - reduce soundness error rate without repeating the proof

  Remark 4.6. In general, among the 3 properties (completeness, soundness, zero-knowledge) of a ZKP proof, soundness represents the mistakenly acceptance rate, i.e., the possibility of the verifier convinced with a false statement mistakenly. Therefore the lower the soundness rate the more robust the ZKP proof is. In general in order to decrease the soundness error rate, one can repeat the scheme multiple times, say, 10 times, but that way the proof size will go up 10 times.

  This improvement is an extension to the above contribution where it has the verifier to calculate $f_i:=\langle b_i,z\rangle-ct_i$ to reduce the proof communication size. Instead of repeating the above ZKP process and generating multiple sets of f_i's, it has the verifier calculate $f_i^{(j)}:=\langle b_i,z_j\rangle-\sigma^j(c)t_i$, where $z_j:=y_j+\sigma^j(c)r$, $t_i=\langle b_i,r\rangle+m_i$ for $i=1,2,3$, $j=0,1,2,3$, and $\sigma\in\mathrm{Aut}(\mathcal{R}_p)$ is the automorphism mentioned earlier in the subsection that nicely rotates the values modulo $\langle x-r_1\rangle$, $\langle x-r_2\rangle$, $\langle x-r_3\rangle$, $\langle x-r_4\rangle$. Similarly to the simplified version of the protocol mentioned above, with some simple linear calculation, one can see that each $f_i^{(j)}$ has the form $f_i^{(j)}=\langle b_i,y_j\rangle-\sigma_j(c)m_i$, i.e., "$m_i$ hidden by some y_j" where y_j is unknown to the verifier. And of course the paper proves that it does improve the soundness error rate even though those $\sigma^j(c)$'s are not independently selected challenges c's.

4.3 Other ZKP Studies

Originated from $As = t$, in this section we have seen the ZKP for $A\bar{s} = \bar{c}t$ and $m_1 m_2 = m_3$. In this series there are also other relation ZKPs such as proving $m_1 + m_2 = m_3$, or $Lm = u$ where L is a matrix, L, u are known, and m is unknown, or $Lm = n$ where m, n are both unknown. There are also range proofs like proving m's coefficients are all in $\{-1, 0, 1\}$, i.e. $\|m\|_\infty = 1$. The above are all about polynomials, there are also proofs of integer relations (addition, multiplication). And finally, they induced the exact ZKP: the knowledge of the short secret s such that $As = t$ [27].

4.4 Research Opportunities

4.4.1 More Algebra Structure and Number Theory

Ever since RLWE was introduced, this area seems to have good potential to grow with algebraic number theory study. On the one hand, the existing number theory tools can be applied to get useful properties in RLWE - for example, polynomial splitness on cyclotomic fields. On the other hand, the demands from Cryptography in RLWE can give rise to new research opportunities in number theory - not only just for new concepts like Galois group over rings as we just seen, but it could also make some types of questions/structures more meaningful in number theory (e.g. invertibility of elements in a polynomial ring).

The researches mentioned in this section are mostly published within 5 years. To me it seems like that theses are very few of the researches that start to take advantage of algebra structures. There are still many what-if's - what if we have more understanding about the splitness, on certain types of irreducible polynomials (can be cyclotomic polynomials or others) under some conditions; what if we leverage more properties of ring of integers (or algebraic integers), dual lattice, and trace, or discriminant. Those maybe applicable to creating, enhancing, or attacking the protocols.

And once an algebra property was observed and used, like in [22], the property can benefit other protocols too [23]. The 2018 paper has already been cited by 35 other papers as of now.

4.4.2 Other Opportunities

All the ZKPs in this research series stem from the same approximate ZKP and a commitment scheme [28]. Therefore if the commitment scheme gets improved then all the ZKPs might get improved. Alternatively, one can improve any individual ZKP too. These ZKPs can also be used to generate new blind signatures/group signatures/e-votings or other protocols.

5 Some Open Reduction Problems

There are a variety of hardness reduction problems related to RLWE, i.e., when solving a problem induces the solution of another problem. Some of them have been resolved as mentioned in Section 2 and some are still open. These problems have many kinds of variations, e.g., SVP can also be approx-RSVP, and the reduction type can be quantum or classical, meaning the hardness relation is true when using quantum algorithms or in general. For example, sRLWE is quantumly harder than or equivalent to approx-RSVP. The existence of its classical reduction is still an open question. And the other direction - whether approx-RSVP is quantumly harder than sRLWE - is unknown too. The ring type can add some variety too. For example, sRLWE is equivalent to dRLWE in cyclotomic ring, and the same case on other rings are not fully clarified.

6 Reflection on this Literature Search Project

The journey of this literature search is definitely a challenging but rewarding exercise. The most challenging part is to put together something from scratch with literally zero knowledge about the research trends in recent years.

When reading these papers I had to keep reminding myself to take a broader approach in order to have a better understanding of the current lay of the land instead of getting too deep and taking too much time in details. I repeatedly found myself in this circle of pattern: get stuck on a proof or an argument for hours, figure it out eventually and feel satisfied, and then forget about it in a few days.

During the literature search, I had many questions, and I think it would be helpful to discuss with others who are actively engaged in research in these areas. I was able to resolve some of my questions with additional learning and readings. However, for some other questions, I had to make an informed guess. This is particularly true when I had high-level, big picture questions.

It is also rewarding to be able to have a grasp of the current literature and write up this document, though most of the thoughts and comments in this document are based on my personal perspective or inference and can be incorrect or naive. It is good to see the growing applications of abstract algebra and number theory concepts in the real world. There seems to be some opportunities for pure math students to contribute to real world questions/situations.

References

[1] Alison Lin. Lattice cryptography: A review of current literature. https://github.com/eggburg/Lattice_Crypto_LitReview, 2021.

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

[3] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, 60(6):1–35, 2013.

[4] Amber Sprenkels. The number theoretic transform in kyber and dilithium. https://electricdusk.com/ntt.html, 2020. Accessed: 2024-06-16.

[5] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[6] W Morven Gentleman and Gordon Sande. Fast fourier transforms: for fun and profit. In *Proceedings of the November 7-10, 1966, fall joint computer conference*, pages 563–578, 1966.

[7] Daan Sprenkels. The kyber/dilithium ntt. https://dsprenkels.com/ntt.html. Accessed: 2021-07-23.

[8] János Folláth. Gaussian sampling in lattice based cryptography. *Tatra Mountains Mathematical Publications*, 60(1):1–23, 2014.

[9] Carlos Aguilar-Melchor, Joris Barrier, Serge Guelton, Adrien Guinet, Marc-Olivier Killijian, and Tancrede Lepoint. Nfllib: Ntt-based fast lattice library. In *Cryptographers' Track at the RSA Conference*, pages 341–356. Springer, 2016.

[10] David Harvey. Faster arithmetic for number-theoretic transforms. *Journal of Symbolic Computation*, 60:113–119, 2014.

[11] Donald Donglong Chen, Nele Mentens, Frederik Vercauteren, Sujoy Sinha Roy, Ray CC Cheung, Derek Pao, and Ingrid Verbauwhede. High-speed polynomial multiplication architecture for ring-lwe and she cryptosystems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(1):157–166, 2014.

[12] Johannes Buchmann, Daniel Cabarcas, Florian Göpfert, Andreas Hülsing, and Patrick Weiden. Discrete ziggurat: A time-memory trade-off for sampling from a gaussian distribution over the integers. In *International Conference on Selected Areas in Cryptography*, pages 402–417. Springer, 2013.

[13] Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. High precision discrete gaussian sampling on fpgas. In *International Conference on Selected Areas in Cryptography*, pages 383–401. Springer, 2013.

[14] Ruan De Clercq, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. Efficient software implementation of ring-lwe encryption. In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 339–344. IEEE, 2015.

[15] Donald E Knuth, KNUTH DE, and YAO AC. The complexity of nonuniform random number generation. 1976.

[16] Sujoy Sinha Roy, Frederik Vercauteren, Nele Mentens, Donald Donglong Chen, and Ingrid Verbauwhede. Compact ring-lwe cryptoprocessor. In *International workshop on cryptographic hardware and embedded systems*, pages 371–391. Springer, 2014.

[17] Paul Barrett. Implementing the rivest shamir and adleman public key encryption algorithm on a standard digital signal processor. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 311–323. Springer, 1986.

[18] Peter L Montgomery. Modular multiplication without trial division. *Mathematics of computation*, 44(170):519–521, 1985.

[19] Shai Halevi and Victor Shoup. Design and implementation of a homomorphic-encryption library. *IBM Research (Manuscript)*, 6(12-15):8–36, 2013.

[20] Patrick Longa and Michael Naehrig. Speeding up the number theoretic transform for faster ideal lattice-based cryptography. In *International Conference on Cryptology and Network Security*, pages 124–139. Springer, 2016.

[21] Po-Chun Kuo, Wen-Ding Li, Yu-Wei Chen, Yuan-Che Hsu, Bo-Yuan Peng, Chen-Mou Cheng, and Bo-Yin Yang. High performance post-quantum key exchange on fpgas. *IACR Cryptology ePrint Archive*, 690, 2017.

[22] Vadim Lyubashevsky and Gregor Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 204–224. Springer, 2018.

[23] Rafaël Del Pino, Vadim Lyubashevsky, and Gregor Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 574–591, 2018.

[24] Gregor Seiler. New techniques for practical lattice-based zero-knowledge. https://simons.berkeley.edu/talks/new-techniques-practical-lattice-based-zero-knowledge, 2020. Accessed: 2021-07-23.

[25] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Practical lattice-based zero-knowledge proofs for integer relations. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1051–1070, 2020.

[26] Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. Practical product proofs for lattice commitments. In *Annual International Cryptology Conference*, pages 470–499. Springer, 2020.

[27] Muhammed F Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 259–288. Springer, 2020.

[28] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. In *International Conference on Security and Cryptography for Networks*, pages 368–385. Springer, 2018.