Lattice-Based Cryptography and the Learning with Errors Problem

Stephen Harrigan

Abstract

This document serves as an general introduction and overview to lattice-based cryptography. More specifically, cryptosystem based on the learning with errors (LWE) problem introduced by Regev are addressed. This document is designed to be self-contained and is intended to serve as an introduction to the main topics in lattice based cryptography or as a refresher to someone that has seen the contents but may have forgotten some details.

Contents

1 Lattices

Lattices have been an object of study for many centuries in mathematics. They have many interesting properties, with mathematicians such as Gauss and Hermite having written about them. This section serves as an introduction to lattices.

Definition 1.1. Given a set of linearly independent vectors $\mathbf{b}_1,...,\mathbf{b}_m$ in $\mathbb{R}^n$, let $\mathbf{B}=\{\mathbf{b}_1,...,\mathbf{b}_m\}$. Then, a *lattice* L is

$$L = \left\{ \sum_{i=1}^{m} a_i \mathbf{b}_i \middle| a_i \in \mathbb{Z}, \mathbf{b}_i \in \mathbf{B} \right\}. \tag{1}$$

We called the set $\mathbf{B}$ the *basis* of the lattice.

If $m = n$, we call this lattice a *full rank lattice*. For the rest of this text, we will only consider full rank lattices for simplicity, but all the proof can be modified slightly in the case that $m \neq n$.

If $\mathbf{B} \subseteq \mathbb{Z}^n$, then we call the lattice an *integer lattice*.

For those familiar with the notion of a vector space, a lattice can be considered as vector space over integers.

We typically denote the length of the shortest non-zero vector in a lattice $\mathbf{L}$ by $\lambda(\mathbf{L})$.

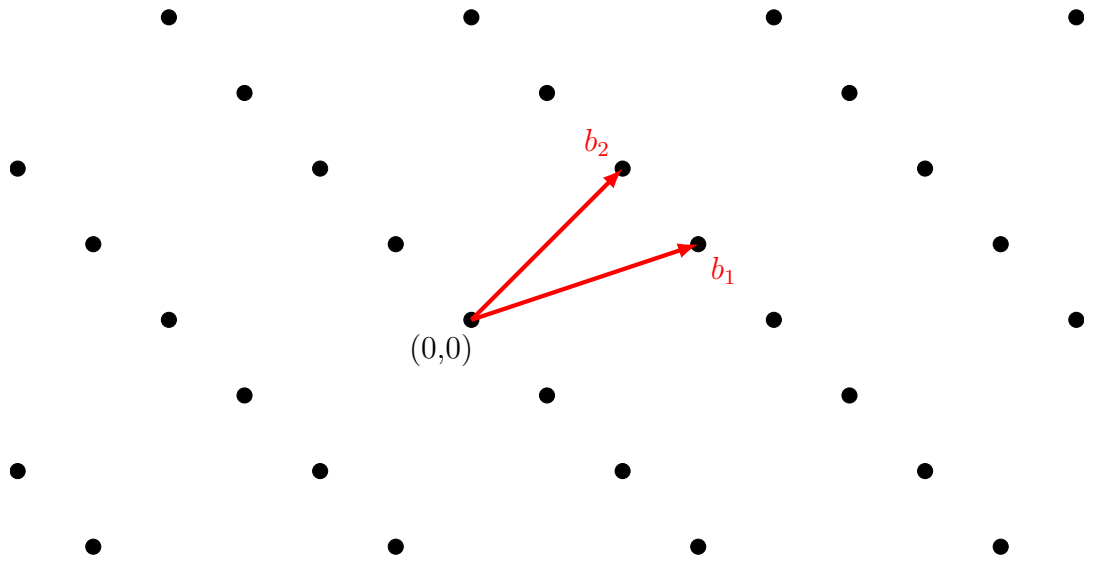

Figure 1: An example of a lattice in $\mathbb{R}^2$ with 2 basis vectors b_1 and b_2 drawn in red.

Lattices have become of interest recently due to their applications in cryptography. In 1994, Shor published a paper showing an efficient quantum algorithm to factor and solve discrete logarithms [6]. This development undermines the security of both RSA and elliptic curve cryptography, the two most popular encryption schemes today, as soon as quantum computers large enough to perform the computations are built. Subsequently, a search began for cryptosystems which are not vulnerable to quantum attacks.

Lattice-based cryptography emerged as a strong candidate to achieve this goal as no known quantum algorithm solves lattice problems efficiently. Furthermore, it is conjectured that no such quantum algorithm exists [5].

1.1 Lattice Problems

Lattice-based cryptography is supported by the claimed intractability of certain problems on lattices, which are generally known as *hard lattice problems*. We now describe some of these hard lattice problems.

Definition 1.2. *Shortest Vector Problem (SVP):* Given a basis $\mathbf{B}$ for a lattice $\mathbf{L}$, find the shortest non-zero vector in $\mathbf{L}$. *(See Figure 2 for example)*

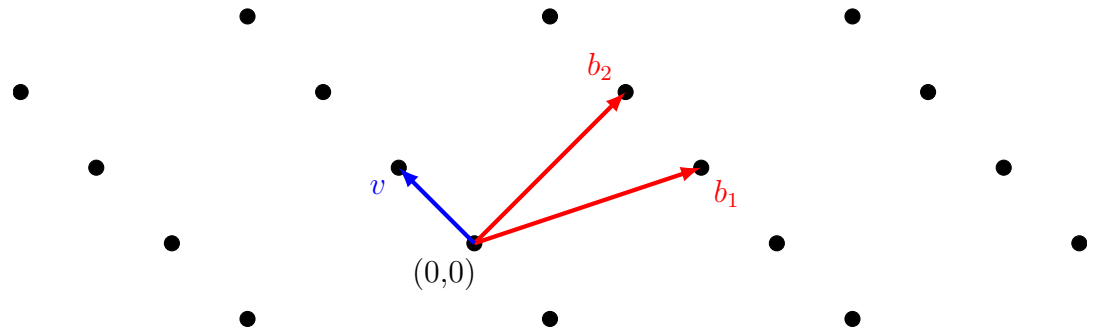

Figure 2: An example SVP in a lattice in $\mathbb{R}^2$ with 2 basis vectors b_1 and b_2 drawn in red. The shortest vector v is drawn in blue.

Definition 1.3. *Closest Vector Problem (CVP):* Given a basis $\mathbf{B}$ for a lattice $\mathbf{L} \subset \mathbb{R}^n$ and a point $p \in \mathbb{R}^n$, find the closest vector in $\mathbf{L}$ to p. *(See Figure 3 for example)*

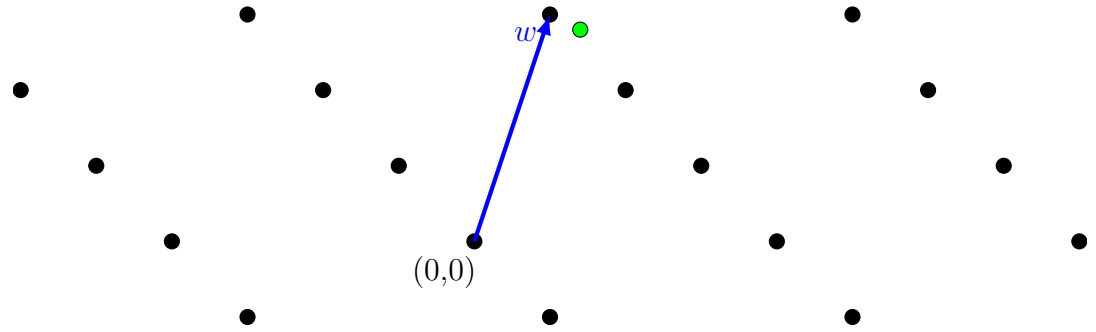

Figure 3: An example CVP in a lattice in $\mathbb{R}^2$. The closest vector to the point p, in green, is the vector w, drawn in blue.

The following problems are variations or generalizations of the problems above with important cryptographic implications.

Definition 1.4. *Shortest Independent Vector Problem (SIVP):* Given a basis $\mathbf{B}$ for a lattice $\mathbf{L}$, find the m shortest linearly independent vectors in $\mathbf{L}$ for some $m \leq n$.

Remark 1.5. Notice that SIVP is a generalization of SVP. By setting $m = 1$, we obtain SVP.

Definition 1.6. *GapSVP:* Let $\beta > 1$ be a real number. Then, given a basis $\mathbf{B}$ for a lattice $\mathbf{L}$, return true if $\lambda(\mathbf{L}) \leq 1$ or return false if $\lambda(\mathbf{L}) > \beta$. The algorithm is allowed to return anything if $1 < \lambda \leq \beta$.

There are also approximate variants of CVP and SVP.

Definition 1.7. *γ-approximation Closest Vector Problem (CVP_γ)*: Given a basis $\mathbf{B}$ for a lattice $\mathbf{L}$, a point p and some $\gamma \geq 1$, find a point within $\gamma \cdot \lambda(\mathbf{L})$ of p.

Definition 1.8. *γ-approximation Shortest Vector Problem (SVP_γ)*: Given a basis $\mathbf{B}$ for a lattice $\mathbf{L}$ and some $\gamma \geq 1$, find a vector with length at most $\gamma \cdot \lambda(\mathbf{L})$.

1.2 Ideal Lattices

In this subsection, we will address the topic of ideal lattices. Although ideal lattices are not very extensively used in the remaing part of this document, they do appear frequently in the literature of lattice based cryptography, including being a key component of Gentry's fully homomorphic encryption [2]. Although they were first used implicitly in [3], they were first defined explicitly in [4].

The idea behind ideal lattices is related to the notion of an ideal in a ring. We need to make a connection between lattices and rings. We will then know why the name "ideal lattice" is appropriate.

Definition 1.9. Let L be an integer lattice in n dimensions let f be a monic polynomial of degree n. Then, define the embedding $\phi : L \to \mathbb{Z}[x]/\langle f \rangle$ from the lattice to the ring of integer polynomials modulo f by

$$\phi(a_0, a_1, \ldots, a_n) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}.$$

Note that this is a homomorphism of additive abelian groups.

By defining this embedding, we can think of these lattices in terms of rings, which are well studied and often have more underlying algebraic structure. One important structure in rings is that of an ideal, and in fact there is a very general connection between ideals in these quotient polynomials.

Theorem 1.10. (from [4]) *Suppose we have a monic, irreducible integer polynomial of degree n and an ideal $I \in \mathbb{Z}[x]/\langle f \rangle$. Then I is isomorphic to a full-rank lattice in $\mathbb{Z}^n$.*

Note that the converse is not true.

Example 1.11. Suppose we take $f = x^n - 1$, which is not irreducible over the integers. However, it is isomorphic to a lattice in $\mathbb{Z}^n$. This modulus is in fact used in *cyclic lattices*, which is used in the construction of the NTRU cryptosystem implicitly [3]. Thus we have a full rank lattice where f is not irreducible.

Example 1.12. The lattice in $\mathbb{Z}^2$ generated by (2,0) and (0,1) (or equivalently in polynomial form by $2 \cdot x$ and 1) is not an ideal lattice because any ideal containing 1 will also contain $1 \cdot x$ but (1,0) is not in this lattice. Thus we have a full rank lattice in $\mathbb{Z}^2$, but it cannot correspond to an ideal in any polynomial ring.

Definition 1.13. An *ideal lattice* is an integer lattice L such that $\phi(L) = \{g \bmod f \mid g \in I\}$ for some monic polynomial f of degree n and an ideal $I \subseteq \mathbb{Z}[x]/\langle f \rangle$.

Note that not all lattices can be represented in this way, with example 1.12 being a particular lattice. Thus, the set of ideal lattices is a proper subset of all lattices.

When is I' a full rank lattice? Now that we have this definition, a useful trick would be to be able to generate these lattices. The procedure below does exactly this.

Lemma 1.14. *If v is an element of $\mathbb{Z}[x]/\langle f(x)\rangle$ where f is an irreducible monic polynomial of degree n, then $v, vx, ..., vx^{n-1}$ are linearly independent.*

Proof. Let $v \in \mathbb{Z}[x]/\langle f(x)\rangle$. Therefore, there will be a representative of v such that $\deg(v) < n$. Now suppose, towards a contradiction, that the set was linearly dependent. So, there exists $a_0, a_1, ..., a_{n-1}$ such that

$$a_0 v + a_1 vx + a_2 vx^2 + ... + a_{n-1} vx^{n-1} \equiv 0 \pmod{f}$$

$$v(a_0 + a_1 x + a_2 x^2 + ... + a_{n-1} x^{n-1}) \equiv 0 \pmod{f}$$

For simplicity, let w=$a_0 + ... + a_{n-1}x^{n-1}$. Since f is irreducible, this implies that either $f|v$ or $f|w$. However, this is a contradiction since v and w are both of degree less than n and f is irreducible. ■

Remark that $v, vx, ..., vx^{n-1}$ will by definition span the ideal generated by v in $\mathbb{Z}[x]/\langle f\rangle$, so this is the smallest ideal containing v; this makes it the most general construction possible. Note that it could happen that the ideal might be all of $\mathbb{Z}[x]/\langle f\rangle$, thus the ideal lattice generated is $\mathbb{Z}^n$.

This lemma gives us a good way to generate ideal lattices, even though an arbitrary lattice is not that likely to be ideal. Let us construct an ideal lattice using this technique.

Example 1.15. Suppose that $f = x^2 + 1$, which is irreducible over the integers. So, the quotient ring we are living in will be $\mathbb{Z}[x]/\langle x^2 + 1\rangle$. Now, let us take $x \in \mathbb{Z}[x]/\langle x^2 + 1\rangle$. Then, from 1.14, we know that x and $x^2 \equiv -1$ are linearly independent. Thus, the lattice generated by (1,0) and (0,-1) is an ideal lattice. Since this is in fact $\mathbb{Z}^2$, which we know is ideal, the method from the lemma has produced an ideal lattice, albeit not the most interesting example.

2 Learning with Errors

In this section, we introduce the Learning with Errors problem (also known as LWE) and then give a more precise statement of the problem.

Introduction Suppose that there exists a "secret" vector $\mathbf{s}$=$(s_1, s_2, \ldots, s_n) \in \mathbb{Z}^n$ with the coefficients as integers. Now suppose that we have a bunch of linear equations in $\mathbf{s}$, where the coefficients are known. That is, something of the form

$$\begin{aligned} a_{1,1}s_1 + a_{1,2}s_2 + \cdots + a_{1,n}s_n &= a \\ a_{2,1}s_1 + a_{2,2}s_2 + \cdots + a_{2,n}s_n &= b \\ &\vdots \\ a_{m,1}s_1 + a_{m,2}s_2 + \cdots + a_{m,n}s_n &= m \end{aligned}$$

The problem is to determine what $\mathbf{s}$ is.

In a situation like this, determining $\mathbf{s}$ is easy provided that enough equations appear. If $m \geq n$, then a simple row reduction can easily provide a solution in polynomial time to the problem.

However, let us change the situation slightly and see what happens. Suppose now instead we have the same set up but this time, the linear equations are only "approximately" correct. That is, something that looks like

$$\begin{aligned} a_{1,1}s_1 + a_{1,2}s_2 + \cdots + a_{1,n}s_n &\approx a \\ a_{2,1}s_1 + a_{2,2}s_2 + \cdots + a_{2,n}s_n &\approx b \\ &\vdots \\ a_{m,1}s_1 + a_{m,2}s_2 + \cdots + a_{m,n}s_n &\approx m \end{aligned}$$

where "$\approx$" simply means that the value is close to the real answer to within a certain error.

In this setting, the problem becomes much more difficult. The simple row reduction trick from earlier will not work because as we multiply and add rows together, the errors in each different equation will compound, causing the final row reduced state to be of no real value as the answer could be a far removed from the actual value; alternatively, it could be inconsistent.

Another important fact worth mentioning is that solving the linear system and then rounding to the closest integer is not necessarily the solution, or even close to it. An example can easily illustrate this fact.

Example 2.1. Suppose that $\mathbf{s} = (3, 7)$ and the $e_1 = e_2 = -1$. Then we get the following linear system:

$$\begin{aligned} 5s_1 + 3s_2 &\approx 35 \\ 4s_1 + 2s_2 &\approx 27 \end{aligned}$$

If we write this as a matrix and row reduce, we get that $\mathbf{s} = (\frac{11}{2}, \frac{5}{2})$ which rounds to $\mathbf{s} = (6, 3)$. This is not even that close to what we started with. This is a similar to what occurs when one tries to solve CVP.

In fact, it has been shown that possessing an algorithm that solves the LWE problem implies a solution to certain hard lattice problems, which are believed to be difficult. This, and the fact that no one has yet found a quantum algorithm to solve these lattice problems, are what motivated the use of LWE based encryption schemes in the first place.

Now, we give a precise statement for Learning with Errors (This problem is also known as Search-LWE).

2.1 The Learning with Errors Problem

Definition 2.2. Fix $n \geq 1$, $q \geq 2$ and an "error" probability distribution χ on $\mathbb{Z}_q$. Let $\mathbf{s}$ be an vector with n coefficients in $\mathbb{Z}_q$. Let $A_{s,\chi}$ on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ be the probability distribution obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \in \mathbb{Z}_q$ according to χ and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$, where additions are performed in $\mathbb{Z}_q$.

We say an algorithm solves LWE with modulus q and error distribution χ if for any $\mathbf{s} \in \mathbb{Z}_q^n$, given enough samples from $A_{s,\chi}$ it outputs $\mathbf{s}$ (with high probability).

There is another version of the problem, known as decision-LWE that is as follows.

Definition 2.3. Suppose we have a way of generating samples from $A_{s,\chi}$ as above and also generating random uniformly distributed samples of (**a**,b) from $\mathbb{Z}_q^n \times \mathbb{Z}_q$. We call this uniform distribution U. The decision-LWE problem is to determine after a polynomial number of samples whether the samples are coming from $A_{s,\chi}$ or U.

In other words, if someone one asks you for **s** given a certain amount of samples, are you able to call their bluff if they are giving you samples from the wrong distribution?

Interestingly, there exist a reduction from Search-LWE to Decision-LWE.

2.2 Equivalence of Search-LWE and Decision-LWE

Lemma 2.4. *Suppose there is a distinguisher that can solve decision-LWE. Then if the modulus p is prime, there exists an efficient algorithm to solve Search-LWE.*

Proof. Let us find s_1. The procedure is similar for finding all other coordinates of **s**.
For each $k \in \mathbb{Z}_p$, define the transformation f_k by

$$f_k : (\mathbf{a}, b) \longmapsto (\mathbf{a} + (l, 0, ..., 0), b + l \cdot k) \tag{2}$$

where $l \in \mathbb{Z}_p$ is chosen uniformly at random.

If (**a**,b)$\in A_{s,\chi}$, then when $s_1 = k$, then $f_k(\mathbf{a}, b) = (\mathbf{a} + (l, 0, ..., 0), b + l \cdot k)$ is again a sample from $\in A_{s,\chi}$. It remains to show that if $s_1 \neq k$, then $f_k(\mathbf{a}, b) \in U$.

For this step, p must be prime. Then for any given k, k is a generator for the group $\mathbb{Z}_p$ under addition since gcd(k,p)=1. Thus, for each possible value of l, $k \cdot l$ is a different value in $\mathbb{Z}_p$. Since l is chosen uniformly at random, the resulting $b + k \cdot l$ are also distributed uniformly at random. Thus, for all $k \neq s_1$, $f_k \in U$.

Since the distinguisher can solve decision-LWE, given enough samples of f_k, they can decide if $f_k \in U$ (in which case $k \neq s_1$) or $f_k \in A_s\chi$ (in which case $k = s_1$).

Therefore, in at most p steps, we can determine the value of s_1.

Continuing like this for each coordinate, adding l in the next coordinate, we can recover all the coefficients in **s**. ■

Lemma 2.5. *Suppose that there exists an efficient algorithm to solve Search-LWE. Then there exists an efficient algorithm to solve Decision-LWE.*

Proof. Suppose we are given a sample (**a**,b). Then using the search algorithm, determine the candidate for **s'**. Then, subtract $\langle \mathbf{a}, \mathbf{s'} \rangle$ from **b**. If the resulting coordinates of the difference are distributed according to the error distribution, return true. If the resulting coordinates are from uniform distribution, return false. ■

Corollary 2.6. *The Search and Decision version of LWE are equivalent.*

Proof. This follows immediately from Lemma 2.4 and Lemma 2.5. ■

2.3 Reduction from LWE to CVP

One of the key features that those who support LWE-based cryptosystems point out is that there exists a reduction from the LWE problem to certain hard lattice problems. although the exact complexity of these approximate solutions to lattice problems is currently unknown for many cases. However, since no one has found any good classical or quantum algorithms to solve these lattice problems despite a fairly substantial research effort to find some, this gives cautious optimism that the problem is hard in both classical and quantum settings.

Theorem 2.7. (Informal) *Solving n-dimensional LWE with* poly(n) *modulus implies an equally efficient solution to a worst-case lattice problem in dimension* $\sqrt{n}$.

Here, "poly(n) modulus" means that the modulus p is bounded by some polynomial function in n.

A slightly different theorem to the one presented here appeared in [5] but was a quantum reduction as one of the steps in the reduction was quantum. However, in [1] the authors show that this reduction exists in the classical setting.

The whole proof is quite technical, but curious reader are encouraged to consult [1] and [5] for the full details. This section meant to show that this reduction exists, and as a result justifies the construction of a cryptosystem based on the LWE problem.

3 LWE Cryptosystem

Now that we have a certain hardness guarantee for solving the LWE problem, we can construct a cryptosystem based on the LWE problem. Here is a simple encryption scheme based on LWE in the setting of modular arithmetic, developped by Regev in [5].

3.1 Algorithm

Set Up and Key Generation Generate m vectors $\mathbf{a}_i$ randomly from $\mathbb{Z}_p^n$. Then generate m "error terms" e_i according to an certain error distribution χ on $\mathbb{Z}_p$. Then for $i \in \{1, ..., m\}$, output

$$(\mathbf{a}_i, b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i).$$

The public key is $\big\{(\mathbf{a}_i, b_i) | i \in \{1, ..., m\}\big\}$.

Encryption The encryption acts on one bit, either 0 or 1. To encrypt, take a random subset S of {1,2,...,m}. Then, compute ($\sum_{i \in S}$ $\mathbf{a}_i$, $\sum_{i \in S}$ b_i) if the bit is 0 or ($\sum_{i \in S}$ $\mathbf{a}_i$, $\lfloor \frac{p}{2} \rfloor + \sum_{i \in S}$ b_i) if the bit is 1.

Decryption To decrypt, compute $b - \langle \mathbf{s}, \mathbf{a} \rangle$. If the result is closer to 0 than $\lfloor \frac{p}{2} \rfloor$, return 0. Otherwise, return 1.

3.2 Example

Here is an example of the encryption scheme at work.

Set Up and Key Generation Consider an example in $\mathbb{Z}_7^4$. Let the private key $\mathbf{s}$ be [3,4,0,6]. Then, let m=3. So generate 3 vectors and 3 error terms:

a_1=[1,6,6,2], e_1=0
a_2=[6,0,5,3], e_2=-1
a_3=[2,5,4,1], e_3=1

Thus, the public key is

$$\Big\{\Big([1,6,6,2],4\Big), \Big([6,0,5,3],0\Big), \Big([2,5,4,1],5\Big)\Big\}.$$

Encryption Suppose we want to encrypt the bit 1. Take the subset S={1,3}. So

$$\left(\sum_{i\in S}\mathbf{a}_i, \lfloor\frac{p}{2}\rfloor + \sum_{i\in S} b_i\right) = \left([1,6,6,2]+[2,5,4,1], \lfloor\frac{7}{2}\rfloor + 4 + 5\right) = \left([3,4,3,3], 5\right)$$

Decryption To decrypt, simply compute

$$b - \langle \mathbf{s}, \mathbf{a}\rangle = 5 - \langle [3,4,0,6], [3,4,3,3]\rangle = 5 - 1 = 4$$

Since 4 is closer to 3 (which is the floor of $\frac{7}{2}$) than 0, output 1 as the encrypted bit. Since this is what we started with, the encryption scheme worked correctly.

3.3 Drawbacks

Although the encryption scheme outlined above is secure, it will not be implemented in any sort of volume capacity. The main reason why is due to the inefficiency of the encryption. For one, when passing from plaintext to ciphertext, the message gets amplified by n. Since the encryption acts on one bit and the message sent is an n-dimensional vector, it is very inefficient for sending long messages, especially if n is large. So, we need an alternative encryption scheme that avoids this problem.

4 Learning with Errors over Rings

Although the cryptosystem based on LWE is secure, the main drawback is that it is very inefficient due to the amplification from plaintext to ciphertext. We will define some machinery which will be useful further on to extend LWE into the more general setting of rings.

Definition 4.1. Let K be a number field with ring of integers $R = \mathcal{O}_K$ and let $q \geq 2$ be an integer modulus. For any fractional ideal J in K, let $J_q = J/qJ$. Recall that the fractional ideal $R^\vee$ is the dual (or "codifferent") fractional ideal of R and let $\mathbb{T} = K_\mathbb{R}/R^\vee$.

Example 4.2. For $\mathbb{Z}$, which is related to LWE, we apply the following to this definition: $K = \mathbb{Q}, R = \mathbb{Z}, J = \mathbb{Z}, J_q = \mathbb{Z}/q\mathbb{Z}, \mathbb{T} = \mathbb{R}/\mathbb{Z}, R^\vee = \mathbb{Z}$, and $R_q^\vee = (R^\vee)_q = \mathbb{Z}/q\mathbb{Z}$.

Example 4.3. We show how a quotient polynomial rings is related to Definition 4.1. Take n to be a power of 2 and define the following quotient ring $K = \mathbb{Q}[x]/\langle x^n + 1\rangle$. For any given quotient ring, K need not be a field. In this case however, it is a field since $x^n + 1$ is irreducible over $\mathbb{R}$. By a theorem, we know that its ring of integers will simply be $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$. Take the fractional ideal $J = R$. Then, for any prime $q \in \mathbb{Z}$, take $J_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$. We also note that $R^\vee = R$ and that $K_R = \mathbb{R}[x]/\langle x^n + 1\rangle$. Then $\mathbb{T} = \mathbb{T}[x]/\langle x^n + 1\rangle$, which corresponds to polynomials of degree less than n with real coefficients in [0,1).

This description is meant for more generality than what is actually required in the cryptosystem, as real numbers are hardly ever useful in practical application due to the finite memory in computers.

We now extend the notion of Search-LWE to a more general ring setting by giving a precise statement for Ring Learning with Errors. This problem is also known as Search Ring-LWE.

Definition 4.4. Let $s \in (R^\vee)_q$ be the "secret" and let ψ be an error distribution over $K_\mathbb{R}$. The use the straightforward embedding ϕ of R_q into $K_\mathbb{R}$ where we map a number in R_q to the same number in $K_\mathbb{R}$. Then a sample of $A_{s,\phi}$ on $K_\mathbb{R} \times \mathbb{T}$ is generated by choosing $a \leftarrow R_q$ uniformly at random, choosing $e \leftarrow \psi$, which are between 0 and $q-1$ and computing $(a,\, b = (a \cdot s)/q + e \mod R^\vee)$, where division by q means that the coefficients are divided by q.

We say an algorithm solves Search Ring-LWE if given enough samples from $A_{s,\phi}$, it can recover $\mathbf{s}$ with high probability.

We now define the associated decision problem, known as Decision Ring-LWE.

Definition 4.5. Suppose we have a way of generating samples from $A_{s,\phi}$ and we also have another set of generating uniformly distributed random samples from $K_\mathbb{R} \times \mathbb{T}$, denoted U. We say an algorithm can solve Decision Ring-LWE if after a certain number of samples it can distinguish whether the samples are coming from $A_{s,\phi}$ or U.

Similar to the LWE problem, the search and decision version of Ring-LWE are equivalent.

5 Ring-LWE Cryptosystem

The main advantage that Ring-LWE possesses over standard LWE is much greater efficiency without sacrificing security. Here is an a outline of a simple encryption scheme over the ring of polynomials modulo an ideal with the coefficients living in $\mathbb{Z}_p$.

5.1 Algorithm

Set Up and Key Generation Let $R = \mathbb{Z}_p[x]/\langle x^n + 1\rangle$, where $n = 2^m$ for some $m, p \in \mathbb{Z}$. Let $a \in R$, and let $s, e \in R$ be 2 "small" elements. s is the private key. Next, generate $(a, b = a \cdot s + e)$. This tuple is the public key.

Encryption Suppose the sender wants to encrypt an n-bit binary message. To encrypt, we must map the binary message as the coefficients of the polynomial in R. That is, if $z = (a_{n-1}, a_{n-2}, ..., a_0)$, with each $a_i \in \{0, 1\}$ then the corresponding polynomial z will be $a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + ... + a_1 x + a_0$.

Then, to encrypt, we need 3 additional "small" elements $r, e_1, e_2 \in R$. We then send $(u, v) \in R^2$ to the intended recipient, where

$$u = a \cdot r + e_1$$
$$v = b \cdot r + e_2 + \lfloor p/2 \rfloor \cdot z.$$

Decryption To decrypt, compute $v - u \cdot s = (r \cdot e - s \cdot e_1 + e_2) + \lfloor p/2 \rfloor \cdot z$. For a suitable choice of parameters, the magnitude of each coefficient of $(r \cdot e - s \cdot e_1 + e_2)$ will be less than $p/4$. Thus, we can extract the original message by 0 or 1 if the respective coefficient is closer to 0 or $\lfloor p/2 \rfloor$ respectively.

5.2 Example

Here is an example of the above algorithm to illustrate it.

Set Up and Key Generation Let $R = \mathbb{Z}_{71}[x]/\langle x^8 + 1\rangle$ (notice $8 = 2^3$). Let $a = 19x^7 + 10x^6 + 7x^5 + 18x^4 + 24x^3 + 24x^2 + 31x$, $s = -x^7 + x^6 + 2x^5 - x^4 - x^3 + 2x^2 + x + 1$ and $e = x^7 - x^6 + x^5 - x^3 - x^2 + 2$. Now s is the private key. The public key is $(a,\ b)$, where $b = a \cdot s + e = 10x^7 + 14x^6 + 11x^5 + 18x^4 + 48x^3 + 4x^2 + 45x + 18$.

Encryption Suppose the sender wants to encrypt the message (1,0,1,1,0,0,1,0). We will then map this to the polynomial $x^7 + x^5 + x^4 + x$. Now, we generate $r = x^7 + 3x^4 - 2x + 1$, $e_1 = x^6 + x^3 - 2x$ and $e_2 = -x^7 + 2x^5 - x^4 - x^2$. So,

$$\begin{aligned} u &= a \cdot r + e_1 \\ &= 50x^6 + 54x^5 + 34x^4 + 44x^3 + 50x^2 + 55x + 24 \end{aligned}$$

and

$$\begin{aligned} v &= b \cdot r + e_2 + \lfloor p/2 \rfloor \cdot z \\ &= 36x^7 + 67x^6 + 61x^5 + 63x^3 + 36x^2 + 10x + 10 \end{aligned}$$

Decryption To decrypt, we simply need to compute

$$v - u \cdot s = 38x^7 + 68x^6 + 34x^5 + 39x^4 + 3x^3 + 2x^2 + 31x + 10$$

Then, we look to see which of the coefficients are closer to 30 than 0 modulo 71. We can see that the 1st, 3rd, 4th and 7th coefficients are indeed closer to 30 than 0 modulo and assign them 1's. The remaining terms are assigned 0's. Thus we obtain (1,0,1,1,0,0,1,0), which is exactly what we began with and thus the decryption was successful.

5.3 How Small is "Small"?

In the previous paragraphs, we stated that certain error term must be "small" without stating any explicit bounds of the coefficients. In this subsection, we will present a function for computing the coefficient for a specific term and use it to develop a bound on how "large" any coefficient will be.

Formula for a specific coefficient Suppose we are living in $R = \mathbb{Z}_p[x]/\langle x^n + 1\rangle$. Notice that in this ring, $x^n \equiv -1$.

Normally, in a polynomial, when we multiply the terms, the coefficient in front of the kth power is simply

$$\sum_{i+j=k} a_i b_j \tag{3}$$

if we are multiplying two polynomials for which the coefficients of the ith power are a_i and b_i respectively. However, in this quotient ring, things are slightly different. For example,

$$x^{k+n} = (x^k)(x^n) = (x^k)(-1) = -x^k. \tag{4}$$

Thus, we must consider the coefficients where $i + j$ is *congruent* to k, rather than equal.

Formally speaking, since there are a large (i.e. infinite) number of numbers congruent to k modulo n, it may seem difficult to count up all the possible ways the product can sum up too k.

However, conveniently, since the largest power in any product in the same ring is x^{n-1} (since it can be reduced by the same process as equation 4), the largest possible power in a multiplication is x^{2n-2}. Thus, we only need to concern ourselves with 2 scenarios:

(a) $i + j < n$

(b) $i + j \geq n$

For (a), the result is relatively straightforward. Since the sums are always less than n, we don't need to worry about any negatives popping up. Thus, the formula is simply

$$c_k = \sum_{i+j=k} a_i b_j \tag{5}$$

as before.

For (b), the situation becomes slightly trickier. Since the sum is above n but below $2n$, we will have to take the negative sign out in front. Furthermore, since we are only adding n once, the sum will be $i + j = k + n$. Thus, the formula should be

$$d_k = -\left(\sum_{i+j=k+n} a_i b_j \right) \tag{6}$$

Combining equation 5 and 6 together yield the final formula

$$e_k = \sum_{i=0}^{k} a_i b_{k-i} - \left(\sum_{i=k+1}^{k+n} a_i b_{k+n-i} \right) \tag{7}$$

Largest Possible Coefficient Now suppose that we wanted to make a specific package to implement the cryptographic scheme above. Inside we want a set $\{-l,-l+1,\ldots,l-1,l\}$ to act as the coefficients of the polynomials generated such that no matter which two polynomials we multiply together in the ring, the largest coefficient will always be less than some m. What is the largest possible value for l?

Theorem 5.1. *Suppose that we have two polynomials $a, b \in \mathbb{Z}[x]/\langle x^n + 1\rangle$ with coefficients from the set $\Phi = \{-l, -l+1, \ldots, l-1, l\}$. Then the magnitude of all of the coefficient in $a \times b$ are less than or equal to nl^2. Moreover, the bound is tight.*

Proof. Let us try and maximize the absolute value of the *kth* power, for some arbitrary k.

For the powers less than k, the part corresponding to the first sum in (7), we want the maximal absolute value for all the coefficients in the polynomials being multiplied to add as much as possible to the sum. Thus, we should take each coefficient in the 2 polynomials being multiplied to be $|l|$. Then, the maximal contribution of this sum to the absolute value would be $(k+1)l^2$, since each pair will have a product of l^2 and there are $k+1$ such pairs.

For the second sum in (7), we essentially want the same thing. Thus, this will contribute at most $(n-k-1)l^2$ to the absolute value, since there are $n-k-1$ pairs each contributing l^2.

If the signs of both sums are the same sign, then the sum of the absolute value is the absolute value of the sums. Thus, the largest possible value a coefficient could take in this ring is

$$(k+1)l^2 + (n-k-1)l^2 = nl^2$$

where n is the degree of the highest power in the modulus. Since we attain this maximal value, the bound is therefore tight. ■

Now, we have an easy way to say how big the coefficients are allowed to be to avoid a decryption failure.

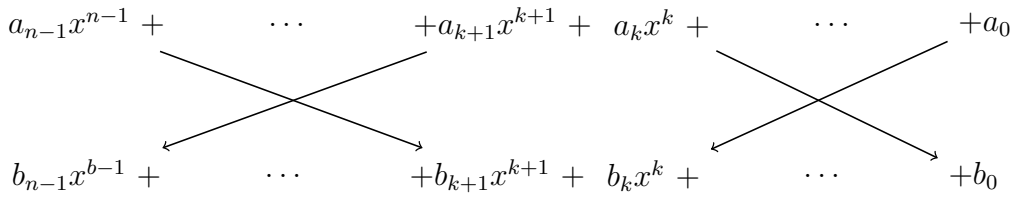

Figure 4: Example of all the pairs which contribute to the coefficient of x^k. The multiples of the coefficients on the left will have an extra negative sign multiplied in whereas the ones on the right will not.

Corollary 5.2. *Suppose we want all of the coefficients in a multiplication to be less than m. Then, the largest possible value of $|l|$ is $\lfloor\sqrt{\frac{m}{n}}\rfloor$.*

Proof. If $|l| = \left\lfloor\sqrt{\frac{m}{n}}\right\rfloor$, then

$$nl^2 = n\left(\left\lfloor\sqrt{\frac{m}{n}}\right\rfloor\right)^2 \leq n\left(\sqrt{\frac{m}{n}}\right)^2 = n\left(\frac{m}{n}\right) = m$$

But, if $|l| \geq \left\lfloor\sqrt{\frac{m}{n}}\right\rfloor$, then

$$nl^2 = n\left(\left\lfloor\sqrt{\frac{m}{n}}\right\rfloor + 1\right)^2 \geq n\left(\sqrt{\frac{m}{n}}\right)^2 = n\left(\frac{m}{n}\right) = m$$

Thus the number is maximal. ■

6 Comparison of LWE and Ring-LWE

The Ring-LWE problem is a slightly more general version of the LWE problem, although it the generalization is not direct.

We can see this relationship by taking a related example, namely:

1. $K = \mathbb{Q}$ and therefore $R = \mathcal{O}_K = \mathbb{Z}$
2. Since $\mathbb{Z}$ is self-dual, $R^\vee = R = \mathbb{Z}$

Then, $K_\mathbb{R} = K \otimes_\mathbb{Q} \mathbb{R} = \mathbb{R}$ since $a \otimes_\mathbb{Q} b = 1 \otimes_\mathbb{Q} ab = ab$ because $\mathbb{R}$ is a field extension of $\mathbb{Q}$ and thus the multiplication acts in the "ordinary" fashion. Then, $\mathbb{T} = K_\mathbb{R}/R^\vee$=[0,1).

Looking at this specific instantiation of Ring-LWE, this would correspond to the multiplication of two numbers $a, s \in \mathbb{Z}$, and adding to it a number $e \in$[0,1). This corresponds to the case in LWE with $n = m$=1 and an error distribution over the interval [0,1).

However, when we generalize further, the LWE cryptosystem can take m tuples of size $(n, 1)$ for any $m \in \mathbb{N}$ whereas the Ring-LWE problem requires just one tuple of size (n,n). Thus, the only case in which the 2 problems coincide is specifically when $m = n = 1$, which is exactly what is described above.

References

[1] Z. Brakerski, A. Langlois, C. Peikert, O. Regev and D. Stehl, (2013). *Classical Hardness of Learning with Errors* Proc. of the 45th Annual ACM Symp. on Theory of Computing (STOC), 575-584.

[2] C. Gentry, (2009). *Fully Homomorphic Encryption Using Ideal Lattices* Proc. of the 41st Annual ACM Symp. on Theory of Computing (STOC), 169-178.

[3] J. Hoffstein, J. Pipher and J. H. Silverman, (1998). *NTRU: A ring-based public key cryptosystem* Proc. of the 3rd Algorithmic Number Theory Symp., 267-288.

[4] V. Lyubashevsky and D. Micciancio, (2006). *Generalized Compact Knapsacks Are Collision Resistant* Proc. of the 33rd International Colloquium on Automata, Languages and Programming., 144-155.

[5] O. Regev, (2005). *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*, Proc. of the 37th Annual ACM Symp. on Theory of Computing (STOC), 84-93.

[6] P. W. Shor, (1997). *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM J. Comput., 26(5), 1484-1509.