LATTICE-BASED CRYPTOGRAPHY

ANDREW HAH

ABSTRACT. Lattice-based cryptography is a leading approach for securing data against quantum attacks. This paper explores foundational computational lattice problems and their applications to cryptography, as well as their ring-based extensions, which offer enhanced efficiency.

CONTENTS

1. INTRODUCTION

Imagine a future where quantum computers can effortlessly break the security protocols that currently secure our most sensitive information—online banking, encrypted communication, even government secrets. The cryptographic algorithms we rely on today, such as RSA and elliptic-curve cryptography, could become obsolete overnight, leaving our digital world exposed.

One of the most promising solutions lies in lattice-based cryptography, a field built upon the hardness of certain computational problems in high-dimensional spaces. These problems are not just resistant to quantum attacks, but they also open the door to advanced cryptographic capabilities that were once thought to be impossible. This paper delves into the world of lattice-based cryptography, starting with the foundational problems that make these systems so secure and exploring the sophisticated constructions that have emerged over the past few decades. By understanding how these lattice-based schemes work, we can better appreciate their potential to revolutionize the future of secure communication.

2. Lattices

Definition 2.1. Let

$$\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{m \times n}$$

be n *linearly independent* vectors in $\mathbb{R}^m$. The *lattice* generated by $\mathbf{B}$ is the set

$$\mathcal{L}(\mathbf{B}) = \{\mathbf{B}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^n\} = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}^n \right\} \tag{2.1}$$

The sequence of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is called the *lattice basis*. The integers n and m are called the *rank* and *dimension* of the lattice respectively. When $n = m$, we say $\mathcal{L}(\mathbf{B})$ is *full rank* or *full dimensional*.

Example 2.2. Let us consider a lattice in $\mathbb{R}^2$. Consider basis vectors

$$\mathbf{b}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \mathbf{b}_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Then $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2)$ can be represented graphically as

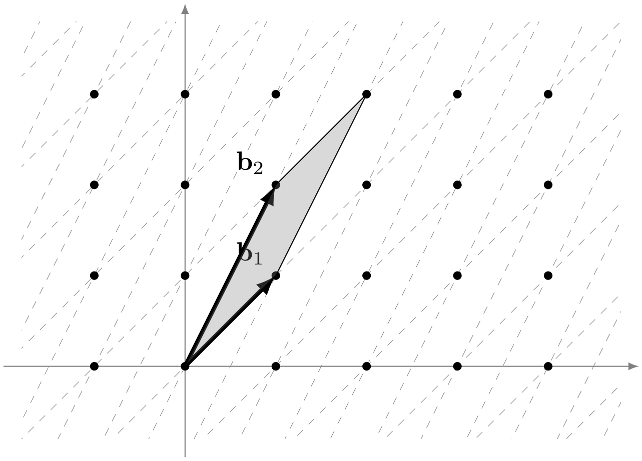

Figure 1. A lattice in $\mathbb{R}^2$.

Definition 2.2. The *minimum distance* of a lattice $\Lambda = \mathcal{L}(\mathbf{B})$ is the minimum distance between any two distinct lattice points

$$\lambda_1(\Lambda) = \min_{\mathbf{x} \neq \mathbf{y} \in \Lambda} ||\mathbf{x} - \mathbf{y}|| = \min_{\mathbf{x} \in \Lambda \setminus \{0\}} ||\mathbf{x}||$$

where $|| \cdot ||$ can be defined with respect to any norm. In this paper we will be using the Euclidean (ℓ_2) norm.

Given a lattice Λ, the problem of finding this minimum distance λ_1 is called the Shortest Vector Problem (SVP).

Definition 2.3 (Shortest Vector Problem (SVP))**.** Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, find the shortest nonzero lattice vector, i.e. find $\mathbf{B}\mathbf{x}$ such that $||\mathbf{B}\mathbf{x}|| \leq ||\mathbf{B}\mathbf{y}||$ for all $\mathbf{y} \in \mathbb{Z}^n \setminus \{0\}$.

Another similar problem is the Closest Vector Problem.

Definition 2.4 (Closest Vector Problem (CVP))**.** Given a basis $\mathbf{B} \in \mathbb{Z}^{m\times n}$ and a target vector $\mathbf{t} \in \mathbb{Z}^m$, find a lattice vector $\mathbf{Bx}$ closest to the target vector $\mathbf{t}$, i.e. find $\mathbf{x} \in \mathbb{Z}^n$ such that $||\mathbf{Bx} - \mathbf{t}|| \leq ||\mathbf{By} - \mathbf{t}||$ for all $\mathbf{y} \in \mathbb{Z}^n$.

To date, there are no known polynomial time algorithms to solve either of these problems.

3. Cryptography

Cryptography is the science of securing communication and data against unauthorized access. At its core, cryptography provides mechanisms for confidentiality and authentication.

Definition 3.1 (Encryption)**. Encryption** is the process of converting **plaintext** (the original readable message) into **ciphertext** (an encoded message) using an **algorithm** and a **key**. The purpose of encryption is to ensure that only authorized parties, who possess the correct key, can decrypt the ciphertext and recover the original plaintext.

Definition 3.2 (Decryption)**. Decryption** is the reverse process of encryption, where ciphertext is converted back into plaintext using an algorithm and a key. Decryption allows the intended recipient to retrieve the original message from the encrypted data.

Definition 3.3 (Public-Key Cryptography)**. Public-key cryptography**, also known as **asymmetric cryptography**, involves the use of two keys: a public key, which is shared openly, and a private key, which is kept secret. In this system, data encrypted with the public key can only be decrypted with the corresponding private key, and vice versa. This enables secure communication and digital signatures, where the public key is used for encryption or signature verification, and the private key is used for decryption or signing.

4. Lattice-based Cryptography

One of the earliest attempts to apply lattices to cryptography was the Goldreich-Goldwasser-Halevi (GGH) cryptosystem introduced in 1997, which is a public-key encryption scheme based on the hardness of the Closest Vector Problem (CVP).

4.1. Hardness of CVP.

CVP is known to be NP-hard under randomized reductions, and it remains difficult even when approximated within any constant factor. The hardness of CVP can be demonstrated through a reduction from the Shortest Vector Problem (SVP), which is known to be NP-hard.

The reduction works as follows: assume there is an oracle that solves CVP exactly. To solve SVP, consider the following approach. Given a lattice $\mathcal{L}$, let $\mathbf{v}$ be the shortest vector in the lattice. Now, construct a target vector $\mathbf{t}$ that is slightly perturbed from $\mathbf{v}$ by adding a small error term ϵ, i.e., let $\mathbf{t} = \mathbf{v} + \epsilon$ for some small $\epsilon > 0$. The oracle for CVP would return $\mathbf{v}$ as the closest lattice vector to $\mathbf{t}$, because the perturbation ϵ is small enough that $\mathbf{v}$ remains the nearest lattice point. Since $\mathbf{v}$ is the shortest vector in the lattice, the oracle effectively returns this shortest vector. Thus, by using the CVP oracle, we can recover the solution to SVP, showing that solving CVP exactly is at least as hard as solving SVP. This reduction implies that CVP is NP-hard.

4.2. **GGH Cryptosystem.** The GGH cryptosystem leverages the difficulty of finding the closest vector in a lattice to a given point, which we just saw is NP-hard. The security of GGH hinges on the fact that, without knowledge of a special "good" basis of the lattice, the problem of decoding a ciphertext (i.e., finding the original plaintext) reduces to solving CVP, which is computationally infeasible for large lattice dimensions.

In the GGH cryptosystem, the public key is a "bad" basis of a lattice $\mathcal{L}$, consisting of vectors that are relatively long and non-orthogonal, making it difficult to solve CVP using this basis. The difficulty arises because a basis with long and non-orthogonal vectors leads to poor approximations of lattice points when solving CVP. In linear algebra, shorter and more orthogonal basis vectors provide more accurate approximations of distance, which is crucial in problems like CVP where one needs to determine the closest lattice point to a given target. Non-orthogonality introduces significant overlap between basis vectors, making it harder to distinguish the direction of the closest lattice point. Long vectors further exaggerate this effect by spreading the lattice points far apart, increasing the search space.

The private key, on the other hand, is a "good" basis of the same lattice, consisting of shorter and more orthogonal vectors. This structure allows for efficient lattice vector operations, including solving CVP. With a good basis, the shorter and nearly orthogonal vectors enable precise geometric interpretations of distances between lattice points, making it computationally feasible to find the closest vector. This disparity between the public and private keys is what ensures the security of the GGH cryptosystem.

In other words, let B basis of lattice $\mathcal{L}$ be the private key and let U be a unimodular matrix. Then the public key B' is another basis of the lattice $\mathcal{L}$ of the form $B' = UB$.

> **Encryption:** Given a message $m = (m_1, \ldots, m_n)$ and public key B', compute
>
> $$v = m \cdot B'$$
>
> v is also a lattice point, and the ciphertext is then
>
> $$c = v + e$$
>
> **Decryption:** To decrypt the ciphertext, compute
>
> $$c \cdot B^{-1} = (m \cdot B' + e) \cdot B^{-1} = m \cdot U \cdot B \cdot B^{-1} + e \cdot B^{-1} = m \cdot U + e \cdot B^{-1}$$
>
> With a small enough error vector, $e \cdot B^{-1}$ can be ignored, and we then compute
>
> $$m = m \cdot U \cdot U^{-1}$$
>
> to get the original message.

4.3. **Example of the GGH Cryptosystem.** Let $\mathcal{L} \subset \mathbb{R}^2$ be a lattice with basis

$$B = \begin{pmatrix} 7 & 0 \\ 0 & 3 \end{pmatrix} \qquad \text{and thus} \qquad B^{-1} = \begin{pmatrix} \frac{1}{7} & 0 \\ 0 & \frac{1}{3} \end{pmatrix}$$

Also let

$$U = \begin{pmatrix} 2 & 3 \\ 3 & 5 \end{pmatrix} \qquad \text{and} \qquad U^{-1} = \begin{pmatrix} 5 & -3 \\ -3 & 2 \end{pmatrix}$$

This gives

$$B' = UB = \begin{pmatrix} 14 & 9 \\ 21 & 15 \end{pmatrix}$$

Let the message $m = (3, -7)$ and the error vector $e = (1, -1)$. Then the ciphertext is

$$c = m \cdot B' + e = (-104, -79)$$

To decrypt, we compute

$$c \cdot B^{-1} = \left(-\frac{104}{7}, -\frac{79}{3} \right)$$

This is rounded to $(-15, -26)$ [using the Babai rounding technique] and the message is recovered with

$$m = (-15, -26) \cdot U^{-1} = (3, -7)$$

4.4. **Cryptanalysis of GGH.** However, in 1999, Phong Q. Nguyen exposed a significant vulnerability in the Goldreich-Goldwasser-Halevi (GGH) cryptosystem. Nguyen's attack leverages the fact that the GGH cryptosystem produces ciphertexts that are close to lattice points with a predictable distribution of errors. Specifically, the encryption process involves adding a small random error vector to a lattice vector to produce the ciphertext.

Nguyen observed that this predictable error distribution could be exploited using lattice reduction techniques, such as the LLL (Lenstra–Lenstra–Lovász) algorithm. The LLL algorithm is a polynomial-time lattice basis reduction algorithm that takes a "bad" basis of a lattice and outputs a reduced basis with shorter, more orthogonal vectors. This reduced basis is much closer to an ideal "good" basis, making problems like CVP easier to solve. In the context of the GGH cryptosystem, the attacker uses LLL to transform the public key, which is a "bad" basis, into a more orthogonal and shorter basis that approximates the private key. With this approximation, the attacker can decrypt ciphertexts or recover the original lattice basis, thereby breaking the security of the system.

By applying LLL or similar lattice basis reduction algorithms, an attacker could recover the original lattice basis or a sufficiently close approximation. This vulnerability arises because the error term added during encryption in GGH is small, and LLL can exploit this to reveal the underlying lattice structure. As a result, the predictable error distribution and the nature of the public key in GGH allow lattice reduction algorithms like LLL to significantly weaken the system's security.

To address some of these shortcomings and build even more secure cryptographic primitives, researchers turned to alternative problems like the Short Integer Solution (SIS) problem. The SIS problem offers a strong connection between average-case and worst-case hardness, making it a powerful tool for developing secure cryptographic functions.

5. The Short Integer Solution (SIS) Problem

The Short Integer Solution (SIS) problem is a foundational problem in lattice-based cryptography, introduced by Ajtai in 1996 [1]. The SIS problem plays a crucial role in constructing various cryptographic primitives, such as collision-resistant hash functions, digital signatures, and identification schemes. At its core, the SIS problem can be seen as a search problem on a particular class of lattices known as q-ary lattices.

Definition 5.1. SIS Problem: Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with uniformly random entries, find a non-zero integer vector $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{Az} = \mathbf{0} \mod q$ and $\|\mathbf{z}\| \leq \beta$ for some norm bound β. Formally, the goal is to solve the equation

$$\mathbf{Az} = \sum_{i=1}^{m} z_i \mathbf{a}_i = \mathbf{0} \mod q,$$

where $\mathbf{a}_i$ are the columns of $\mathbf{A}$ and $\|\mathbf{z}\|$ is the norm of the vector $\mathbf{z}$.

The hardness of the SIS problem is closely related to the worst-case hardness of certain lattice problems, such as the Shortest Vector Problem (SVP) and the Shortest Independent Vectors Problem (SIVP). Specifically, Ajtai demonstrated that solving the average-case SIS problem is at least as hard as approximating these worst-case lattice problems within a certain factor [**?**]. This connection to worst-case hardness provides strong security guarantees, making SIS an attractive foundation for cryptographic constructions.

The reduction from worst-case lattice problems to average-case SIS works as follows: given a lattice $\mathcal{L}$, suppose we are asked to solve a hard instance of the Shortest Independent Vectors Problem (SIVP) in the worst case. Ajtai's reduction shows that if we can solve an average-case instance of SIS efficiently, we can use it to approximate solutions to worst-case instances of SIVP. In this reduction, a random lattice (described by a matrix $\mathbf{A}$) and a vector $\mathbf{v}$ are constructed from the worst-case lattice problem, and solving SIS on this random lattice will yield information about the original hard lattice problem.

To give a simple example, consider a lattice $\mathcal{L}$ generated by a basis $\mathbf{B}$ where we need to approximate the shortest independent vectors. Using Ajtai's reduction, we can create a random instance of SIS by sampling a matrix $\mathbf{A}$, which defines a new lattice closely related to the original one. Solving SIS on $\mathbf{A}$ essentially approximates the shortest vectors in the worst-case lattice problem $\mathcal{L}$. This transformation allows a reduction from worst-case lattice problems, such as SIVP, to the average-case SIS problem, meaning that any efficient algorithm for SIS can be used to solve hard lattice problems in the worst case.

This worst-case to average-case reduction is significant because it ensures that solving SIS in the average case—where the input is generated randomly—is as hard as solving the worst-case instances of lattice problems.

5.1. **Collision-resistant Hash Functions.** One of the most significant applications of SIS is in the construction of collision-resistant hash functions. A hash function built on SIS is secure because finding two different inputs that produce the same output (a collision) would require solving the SIS problem, which is computationally infeasible given the hardness of SIS. This property makes SIS-based hash functions highly desirable in many cryptographic protocols, including digital signatures, message authentication codes, and blockchain technologies.

Definition 5.2 (SIS-Based Hash Function)**.** Let q, n, and m be positive integers such that q is a prime modulus. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a uniformly random matrix. Define the hash function $H : \mathbb{Z}_q^m \to \mathbb{Z}_q^n$ as

$$H(\mathbf{x}) = \mathbf{A}\mathbf{x} \mod q.$$

This hash function maps an input vector $\mathbf{x} \in \mathbb{Z}_q^m$ to a shorter output vector in $\mathbb{Z}_q^n$. The collision-resistance of H relies on the difficulty of finding two distinct vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_q^m$ such that $H(\mathbf{x}) = H(\mathbf{x}')$, which is equivalent to finding a non-zero solution to the equation $\mathbf{A}(\mathbf{x} - \mathbf{x}') = \mathbf{0} \mod q$ with small norm, directly corresponding to an instance of the SIS problem.

Theorem 5.1. *Assuming the hardness of the $SIS_{q,n,m,\beta}$ problem, the hash function H defined above is collision-resistant. Specifically, any probabilistic polynomial-time algorithm that finds a collision in H with non-negligible probability can be used to solve the $SIS_{q,n,m,\beta}$ problem with non-negligible probability.*

Proof. Suppose there exists a probabilistic polynomial-time algorithm $\mathcal{A}$ that, given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, outputs a collision $(\mathbf{x}, \mathbf{x}')$ such that $\mathbf{x} \neq \mathbf{x}'$ and $H(\mathbf{x}) = H(\mathbf{x}')$. Then, consider constructing an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the $\text{SIS}_{q,n,m,\beta}$ problem.

(1) **Input:** $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.
(2) **Execution:** Run $\mathcal{A}$ on $\mathbf{A}$ to obtain a collision $(\mathbf{x}, \mathbf{x}')$.
(3) **Output:** Compute $\mathbf{z} = \mathbf{x} - \mathbf{x}' \mod q$. Since $H(\mathbf{x}) = H(\mathbf{x}')$, we have:

$$\mathbf{A}\mathbf{z} = \mathbf{A}(\mathbf{x} - \mathbf{x}') = \mathbf{0} \mod q.$$

Moreover, if $\|\mathbf{z}\| \leq \beta$, then $\mathbf{z}$ is a solution to the SIS problem.

The norm bound β can be ensured by appropriately choosing the input space for $\mathbf{x}$ and $\mathbf{x}'$. For instance, if $\mathbf{x}$ and $\mathbf{x}'$ are sampled from a discrete Gaussian distribution over $\mathbb{Z}^m$ with a small standard deviation, their difference $\mathbf{z}$ will also have a small norm with high probability.

Therefore, the existence of $\mathcal{A}$ implies the existence of $\mathcal{B}$ that solves $\text{SIS}_{q,n,m,\beta}$ in probabilistic polynomial time, contradicting the assumed hardness of SIS. Hence, H is collision-resistant under the SIS assumption. □

The practical implementation of SIS-based hash functions involves selecting parameters (q, n, m, β) that balance security and efficiency. Typically, m is chosen to be slightly larger than n to ensure a sufficient level of compression and collision-resistance, while β is kept small to maintain the hardness of the underlying SIS problem. These hash functions are not only theoretically sound but also efficient to compute, making them suitable for real-world cryptographic applications such as digital signatures and data integrity verification.

Furthermore, SIS-based hash functions are conjectured to be secure against quantum attacks, positioning them as promising candidates for post-quantum cryptography. Their simple algebraic structure allows for straightforward implementation and analysis, while their security relies on well-studied lattice problems with strong worst-case hardness guarantees.

6. Ring-SIS

Although SIS offers strong security guarantees, the size of the keys and the computational cost associated with SIS-based constructions can be prohibitive, especially for applications requiring high performance or running on resource-constrained devices. To address these challenges, researchers introduced the Ring-SIS problem, a structured variant of SIS that leverages algebraic structures known as rings to achieve more compact representations and faster computations.

The key motivation behind Ring-SIS is to reduce the dimensionality of the problem without compromising its hardness. By encoding lattice vectors as polynomials in a ring, Ring-SIS allows for more efficient operations, including multiplication and addition, which are performed modulo a polynomial with coefficients in a finite field. This ring structure not only reduces the size of the public key and ciphertexts but also enhances the computational efficiency of encryption and decryption processes. Furthermore, Ring-SIS maintains the worst-case to average-case hardness properties of SIS, ensuring that the security of cryptographic schemes based on Ring-SIS remains robust.

Definition 6.1 (Ring-SIS Problem)**.** Let $R = \mathbb{Z}[X]/\langle f(X)\rangle$ be the ring of integer polynomials modulo a monic polynomial $f(X) \in \mathbb{Z}[X]$, and let q be a prime modulus. The Ring-SIS problem asks to find a non-zero polynomial $\mathbf{z}(X) \in R$ such that

$$\mathbf{a}_1(X) \cdot z_1(X) + \mathbf{a}_2(X) \cdot z_2(X) + \cdots + \mathbf{a}_m(X) \cdot z_m(X) = 0 \mod q$$

where $\mathbf{a}_1(X), \mathbf{a}_2(X), \dots, \mathbf{a}_m(X) \in R_q = R/qR$ are uniformly random polynomials, and the norm of the coefficient vector of $\mathbf{z}(X)$ is bounded by a parameter β.

The transition from SIS to Ring-SIS brings several advantages. First, the use of a ring structure allows the dimensionality of the underlying lattice problem to be compressed from m to n, where n is the degree of the polynomial $f(X)$. This compression leads to more compact keys and faster arithmetic operations, which are critical in applications like homomorphic encryption, where performance is a key consideration. Additionally, Ring-SIS can be seen as a generalization of SIS, where the ring R introduces algebraic relationships that are absent in the standard SIS problem, enabling the construction of more advanced cryptographic primitives.

Theorem 6.1. *Assuming the hardness of the Ring-SIS$_{q,f(X),m,\beta}$ problem, cryptographic schemes based on Ring-SIS are secure. The problem remains hard under certain parameter choices, where $f(X)$ is typically chosen to be a cyclotomic polynomial or another carefully selected polynomial that ensures the worst-case hardness reduction from lattice problems in ideal lattices to Ring-SIS.*

The security of Ring-SIS relies on the reduction from the worst-case hardness of certain lattice problems in ideal lattices to the average-case hardness of the Ring-SIS problem. Specifically, it can be shown that solving Ring-SIS in the average case is at least as hard as approximating the shortest vector in an ideal lattice to within a certain factor, which is determined by the choice of the polynomial $f(X)$ and the modulus q. The ring structure ensures that lattice vectors in the underlying ideal lattice have algebraic symmetries, which preserve the hardness of lattice problems when transformed into the Ring-SIS problem. This reduction guarantees that any efficient algorithm solving Ring-SIS would also solve the corresponding lattice problem in ideal lattices, thus securing the cryptographic scheme.

The Ring-SIS problem has been widely adopted in the design of various cryptographic protocols, including key exchange, digital signatures, and homomorphic encryption. Its efficiency gains over traditional SIS make it an attractive choice for real-world applications, particularly in scenarios where performance and key size are critical. Moreover, the structured nature of Ring-SIS, while providing these benefits, does not compromise security, as the underlying hardness assumptions remain firmly grounded in worst-case lattice problems.

7. The Learning With Errors (LWE) Problem

Building upon the foundations laid by SIS and Ring-SIS, another fundamental problem in lattice-based cryptography is the Learning With Errors (LWE) problem. Introduced by Regev in 2005, LWE has become a central component in the construction of various cryptographic primitives, including public-key encryption schemes, pseudorandom functions, and fully homomorphic encryption. The LWE problem generalizes the concept of learning from noisy linear equations and has strong connections to worst-case lattice problems, making it a robust foundation for cryptographic security.

Definition 7.1 (LWE Problem)**.** Let q be a prime modulus, $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a uniformly random matrix, $\mathbf{s} \in \mathbb{Z}_q^n$ be a secret vector, and $\mathbf{e} \in \mathbb{Z}_q^m$ be an error vector sampled from a discrete Gaussian distribution or another bounded noise distribution. The LWE problem asks to distinguish between the following two distributions:

- The distribution of pairs $(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e})$.
- The uniform distribution over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$.

The security of cryptographic schemes based on LWE stems from the difficulty of recovering the secret vector $\mathbf{s}$ given only noisy linear combinations of its components. In practice, the noise vector $\mathbf{e}$ ensures that even if the matrix $\mathbf{A}$ and the product $\mathbf{A}^T\mathbf{s}$ are known, the presence of errors prevents an adversary from accurately determining $\mathbf{s}$. The hardness of LWE has been shown to reduce to the worst-case hardness of approximating certain lattice problems, such as the Shortest Vector Problem (SVP), within a polynomial factor.

The LWE problem has led to the development of a wide array of cryptographic schemes that are not only secure against classical adversaries but also conjectured to be secure against quantum attacks. As with SIS, one of the key strengths of

LWE-based constructions is their provable security based on well-understood lattice problems.

To illustrate how LWE-based encryption works, let's walk through a simple numeric example.

Let $q = 11$, $n = 3$, and $m = 4$. The secret vector $\mathbf{s} \in \mathbb{Z}_q^3$ is chosen as:

$$\mathbf{s} = \begin{pmatrix} 3 \\ 7 \\ 2 \end{pmatrix}.$$

We choose a random matrix $\mathbf{A} \in \mathbb{Z}_q^{3\times 4}$:

$$\mathbf{A} = \begin{pmatrix} 1 & 5 & 3 & 9 \\ 2 & 6 & 4 & 7 \\ 8 & 10 & 1 & 0 \end{pmatrix}.$$

Next, we generate a small error vector $\mathbf{e} \in \mathbb{Z}_q^4$:

$$\mathbf{e} = \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \end{pmatrix}.$$

To generate the ciphertext, we compute $\mathbf{A}^T\mathbf{s}$ and add the error vector $\mathbf{e}$:

$$\mathbf{A}^T\mathbf{s} = \begin{pmatrix} 1\cdot 3 + 2\cdot 7 + 8\cdot 2 \\ 5\cdot 3 + 6\cdot 7 + 10\cdot 2 \\ 3\cdot 3 + 4\cdot 7 + 1\cdot 2 \\ 9\cdot 3 + 7\cdot 7 + 0\cdot 2 \end{pmatrix} = \begin{pmatrix} 3 + 14 + 16 \\ 15 + 42 + 20 \\ 9 + 28 + 2 \\ 27 + 49 + 0 \end{pmatrix} = \begin{pmatrix} 33 \\ 77 \\ 39 \\ 76 \end{pmatrix} \mod 11 = \begin{pmatrix} 0 \\ 0 \\ 6 \\ 10 \end{pmatrix}.$$

Now, adding the error vector $\mathbf{e}$:

$$\mathbf{A}^T\mathbf{s} + \mathbf{e} = \begin{pmatrix} 0 + 1 \\ 0 + 0 \\ 6 + 2 \\ 10 + 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 8 \\ 0 \end{pmatrix} \mod 11.$$

The ciphertext is the pair $(\mathbf{A}, \mathbf{b})$ where $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e} = (1, 0, 8, 0)$. An adversary attempting to recover $\mathbf{s}$ from $\mathbf{A}$ and $\mathbf{b}$ would be faced with the challenge of solving an instance of the LWE problem, which is hard due to the noise introduced by the error vector $\mathbf{e}$.

8. The Ring-LWE Problem

While LWE has proven to be a versatile and secure foundation for cryptography, the computational efficiency of LWE-based schemes can still be improved, particularly in applications requiring high performance or operating on constrained devices. To this end, the Ring-LWE problem was introduced as a natural extension of LWE, incorporating the algebraic structure of rings to achieve significant improvements in efficiency and key size, similar to the transition from SIS to Ring-SIS.

Definition 8.1 (Ring-LWE Problem)**.** Let $R = \mathbb{Z}[X]/\langle f(X)\rangle$ be a ring of integer polynomials modulo a monic polynomial $f(X)$, and let q be a prime modulus. The Ring-LWE problem is defined as follows: given a uniformly random polynomial $\mathbf{a}(X) \in R_q$ and a secret polynomial $\mathbf{s}(X) \in R_q$, along with a noise polynomial $\mathbf{e}(X)$ whose coefficients are sampled from a discrete Gaussian or bounded noise distribution, the goal is to distinguish between the following two distributions:

- The distribution of pairs $(\mathbf{a}(X), \mathbf{a}(X) \cdot \mathbf{s}(X) + \mathbf{e}(X))$.
- The uniform distribution over $R_q \times R_q$.

Ring-LWE preserves the security guarantees of LWE while introducing significant performance benefits. The ring structure allows for more compact representations of the underlying data and supports efficient polynomial arithmetic, which reduces the computational complexity of encryption and decryption operations. Furthermore, the algebraic properties of the ring enable the construction of advanced cryptographic protocols, such as key exchange and fully homomorphic encryption, with improved efficiency.

The transition from LWE to Ring-LWE is analogous to the transition from SIS to Ring-SIS, where the use of structured lattices (ideal lattices) enables a more efficient design without compromising security. The hardness of Ring-LWE is similarly based on the worst-case hardness of approximating lattice problems in ideal lattices, ensuring that cryptographic schemes built on Ring-LWE remain secure under rigorous assumptions.

In summary, Ring-LWE provides a powerful and efficient foundation for modern cryptographic schemes, combining the robustness of LWE with the computational benefits of ring structures. This makes it a critical tool in the development of scalable, secure, and quantum-resistant cryptographic systems.

9. Recent Developments and Open Problems

Lattice-based cryptography has emerged as one of the most promising areas of research in the field of cryptography, particularly due to its strong security foundations and resistance to quantum attacks. Over the past few decades, significant advancements have been made in developing cryptographic primitives based on hard lattice problems such as SIS, LWE, Ring-SIS, and Ring-LWE. These advancements have led to practical implementations of secure cryptographic schemes, including public-key encryption, digital signatures, and key exchange protocols. As we have discussed, the evolution from SIS to Ring-SIS and from LWE to Ring-LWE has brought about substantial improvements in efficiency and scalability, making lattice-based cryptography a viable option for real-world applications.

One of the most groundbreaking developments in lattice-based cryptography has been the construction of fully homomorphic encryption (FHE) schemes. FHE allows for arbitrary computations on encrypted data without decrypting it, a property that has profound implications for privacy-preserving computation and secure cloud computing. Early FHE schemes were limited in their capabilities, often requiring a bootstrapping process to manage noise growth during computation. However, recent work has focused on developing unbounded FHE schemes, which aim to perform an unlimited number of operations on ciphertexts without the need for frequent bootstrapping. Achieving truly unbounded FHE remains an open problem, as researchers continue to seek more efficient and practical constructions that can handle complex computations while maintaining security and performance.

Another exciting frontier in lattice-based cryptography is the development of unbounded attribute-based encryption (ABE) schemes. ABE allows for fine-grained access control over encrypted data, where decryption is possible only if the attributes of the ciphertext and the decryption key satisfy certain conditions. While bounded ABE schemes, where the number of attributes is fixed, have been successfully constructed using lattice-based techniques, the challenge lies in creating unbounded ABE schemes that can support an arbitrary number of attributes. Such schemes would provide greater flexibility and scalability in applications like secure data sharing and access control in cloud environments. Despite significant progress, the design of unbounded ABE schemes with practical efficiency and strong security guarantees remains an open area of research.

In addition to these open problems, lattice-based cryptography continues to face challenges in optimizing efficiency, particularly in reducing key sizes and computational overhead for practical deployment. The interplay between security and performance is a delicate balance that researchers are striving to improve. Furthermore, while lattice-based schemes are believed to be resistant to quantum attacks, ongoing research is essential to rigorously analyze and verify these assumptions against potential quantum adversaries.

Acknowledgments

Many thanks to my mentor Jessica Cao who inspired this paper and helped me learn so much this summer.

References

[1] Ajtai, M. (1996). "Generating hard instances of lattice problems (extended abstract)." Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing (STOC '96).

[2] Goldreich, O., Goldwasser, S., & Halevi, S. (1997). "Public-key cryptosystems from lattice reduction problems." Advances in Cryptology — CRYPTO' 97, Lecture Notes in Computer Science, vol 1294. Springer, Berlin, Heidelberg.

[3] Micciancio, D., & Goldwasser, S. (2002). Complexity of Lattice Problems: A Cryptographic Perspective. Springer Science & Business Media.

[4] Nguyen, P. Q. (1999). "Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto'97." Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '99), Lecture Notes in Computer Science, vol 1666. Springer, Berlin, Heidelberg.

[5] Peikert, C. (2016). A Decade of Lattice Cryptography. Foundations and Trends in Theoretical Computer Science, 10(4), 283-424. Now Publishers.

[6] Peikert, C. (2022). Lattices in Cryptography Lecture Notes. GitHub repository. Retrieved from https://github.com/cpeikert/LatticesInCryptography

[7] Regev, O. (2005). "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography." Journal of the ACM (JACM), 56(6), Article 34.