Lattice-Based Key Encapsulation Mechanisms in Post-Quantum Cryptography

Arjun Chandra[*]
Boston University

Recent progress in building stable and scalable quantum computers has opened the door to potential applications in a variety of domains in the near future such as economics, machine learning, drug development and others. In the field of cryptography, however, quantum computers have the potential to break current cryptographic protocols, posing a significant threat to national security and privacy in general. This paper will discuss the foundations for lattice-based protocols which have emerged as a promising solution for quantum-safe cryptographic systems. The learning with errors problem and its connections to lattice theory will be introduced first, followed by a discussion of its applications in key encapsulation mechanisms, and finally the module learning with errors problem will be motivated and related to recent developments in post-quantum cryptography.

I. INTRODUCTION

Quantum computing technology has made rapid progress in recent years toward realizing large-scale quantum computers suitable for practical applications in a variety of domains [1]. One such domain is cryptography, where it has long been believed that quantum computers may be able to break current cryptographic systems such as RSA. This potential has given rise to the field of post-quantum cryptography, which aims to design quantum-resistant cryptographic systems which are secure against attacks by a quantum computer. In August of this year, the Secretary of Commerce approved three Federal Information Processing Standards related to post-quantum cryptography. The first of these standards details a cryptographic scheme for key encapsulation mechanisms (KEMs), while the other two standards outline schemes for digital signatures. Although quantum computers today are not able to implement algorithms such as Order Finding at the scale and stability that would be required to break current cryptographic systems, it is extremely plausible that such quantum computers will be built in the near future. Given the threat of adversaries harvesting sensitive information now and decrypting it later when quantum computers are able to do so, it is important to understand quantum-resistant cryptographic systems. Along these lines, this paper will discuss the theoretical support for KEMs in post-quantum cryptography. There are a variety of quantum-resistant KEM schemes which are discussed in detail in [2]. However, this paper will focus on the KEM scheme recently approved by the Secretary of Commerce, which is characterized as a lattice-based KEM. The learning with errors (LWE) problem and its connections to lattice theory will be introduced first, followed by a discussion of its applications in KEM schemes, and finally the module learning with errors (MLWE) problem will be motivated and related back to the recently approved KEM scheme.

II. LEARNING WITH ERRORS

A. Overview

The learning with errors problem can be easily understood as the task of solving a system of *approximate* linear equations of the form:

$$A \cdot s \approx b \tag{1}$$

This can be equivalently expressed as an exact system of equations by adding an explicit error term:

$$A \cdot s + e = b \tag{2}$$

When formulating an instance of the LWE problem, several considerations are in order as outlined in [3]:

- The number of equations m as well as the size of s denoted by n are parameters to be set.
- Cryptographic applications of LWE involve modular arithmetic so that the $A, s, e, b \in \mathbb{Z}_q$ for some modulus q.
- The error term e is sampled from a discrete probability distribution $\chi \in \mathbb{Z}_q$. One must be cautious in choosing the distribution such that relative to the modulus q the error e is sufficiently large to ensure the problem is non-trivial but not too large that the system of equations is reduced to random noise. In consideration of this, it is often chosen that $\chi \sim \mathcal{N}(0, \alpha^2 q^2)$ so that the standard deviation $\sigma = \alpha q$ where $\alpha > 0$ is $\frac{1}{\text{poly}(n)}$.

One might imagine attempting to solve (1) via Gaussian elimination given that the error in the equations is sufficiently small, but it quickly becomes apparent that small errors in each equation will compound when taking repeated linear combinations of rows in A and the final row echelon form of A will yield a solution that is likely unrelated to the original system of equations.

[*] ac25@bu.edu

B. Connections to Lattice Theory

The LWE problem is equivalent to several classes of problems in lattice theory which are widely believed to be hard and are in fact provably hard under certain conditions [4]. Given n linearly independent basis vectors $\{b_1, b_2, \ldots, b_n\} \in \mathbb{R}^m$, the lattice generated by them is defined as follows:

$$\mathcal{L}(b_1, b_2, \ldots, b_n) = \left\{ \sum_i x_i b_i : x_i \in \mathbb{Z}, 1 \leq i \leq n \right\} \quad (3)$$

Bounded Distance Decoding and the Shortest Vector Problem are two classes of problems which are defined over such a lattice and are related to LWE. There exist many variants of both problems which are described in detail in [5] but their general definitions are provided in the following sections.

1. Bounded Distance Decoding

The Bounded Distance Decoding (BDD) problem aims to find the point v of a lattice $\mathcal{L}$ which is closest to a target point t that is not in $\mathcal{L}$. The formal definition is stated below:

BDD. Given a basis $B = \{b_1, b_2, \ldots, b_n\}$ of an n-dimensional lattice $\mathcal{L}(B)$ and a target point $t \in \mathbb{R}^n$ for which $dist(t, \mathcal{L}) < d$, find the unique $v \in \mathcal{L}$ such that $||t - v|| < d$.

An instance of LWE can be converted into an instance of BDD by letting the column vectors of A form the basis B and letting the target $t = A \cdot s + e$ where the distance upper bound d can be derived from e. The solution for v will then be $v = A \cdot s$.

2. Shortest Vector Problem

The Shortest Vector Problem (SVP) problem aims to find the point v of a lattice $\mathcal{L}$ which is closest to the origin. The formal definition is stated below:

SVP. Given a basis $B = \{b_1, b_2, \ldots, b_n\}$ of a lattice $\mathcal{L}(B)$, find the nonzero shortest vector $v \in \mathcal{L}$.

SVP can be seen as a special case of BDD in which the target point t is chosen to be the origin. Given this, solutions for specific variants of SVP and BDD often overlap [6]. One very intuitive algorithm that can solve certain instances of SVP in polynomial time within an exponential factor is known as the LLL algorithm. The algorithm is based on the idea that it would be much easier to find v if the basis of the lattice were orthogonal, and in fact it is straightforward to prove that any orthogonal basis must include the shortest vector in the lattice:

Proof. Suppose a basis $\{v_1, \ldots, v_n\}$ is orthogonal. The shortest vector in the lattice λ_1 can be written as some linear combination of the basis vectors:

$$\lambda_1 = a_1 v_1 + \cdots + a_n v_n \quad \text{for } a_i \in \mathbb{Z} \quad (4)$$

Since the basis is orthogonal, the norm-squared of λ_1 can be written:

$$\|\lambda_1\|^2 = a_1^2 \|v_1\|^2 + \cdots + a_n^2 \|v_n\|^2 \quad (5)$$

Since $a_i \in \mathbb{Z}$, (5) is minimized for $a_i = \delta_{ij}$ where v_j is the shortest basis vector. Therefore, the shortest vector in the lattice is the shortest basis vector in the orthogonal basis. □

Given this proof, the LLL algorithm aims to generate an orthogonal basis for a given lattice where the norm of the shortest basis vector is minimized. This basis is termed the LLL-reduced basis. In practice, however, not every lattice can be generated by an orthogonal basis, in which case it is shown in [7] that the shortest basis vector b_1 in the LLL-reduced basis is upper bounded by the true shortest vector $\lambda_1(\mathcal{L})$ with an exponential factor in the dimensionality of the lattice n:

$$||b_1|| \leq 2^{\frac{n-1}{2}} \lambda_1(\mathcal{L}) \quad (6)$$

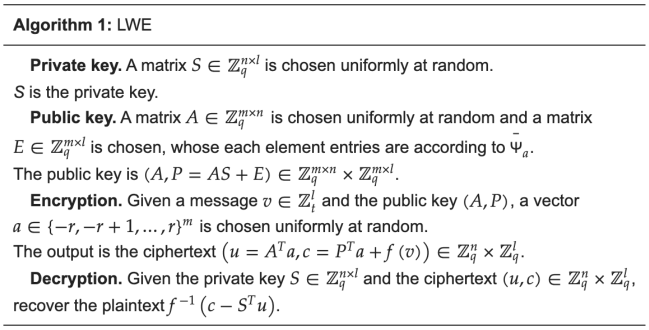
This inequality will be useful later when reasoning about the security of an LWE-based cryptosystem. It is also worth noting that the LLL algorithm is a classical algorithm, and in general quantum algorithms have yet to find significant applications in lattice theory. For this reason, it is thought that lattice-based problems like BDD and SVP are computationally hard even for quantum computers.

III. APPLICATIONS IN CRYPTOGRAPHY

In order for a theoretical problem to be established as a practical cryptosystem, it must be (1) a trapdoor function and (2) computationally feasible and efficient to implement at scale. A trapdoor function is one that is easy to compute in one direction but difficult to compute in the inverse direction without special information. The previous section addressed this criterion by connecting LWE to hard problems in lattice theory; however, the second criterion will be discussed later. For now, on the assumption that LWE is computationally feasible to implement, an LWE-based KEM scheme will be introduced.

A. LWE-KEM Scheme

The psuedocode in FIG. 1 illustrates how LWE could be used to design an asymmetric cryptosystem in which two parties communicate using a public key for encryption and a private key for decryption. The private key



Algorithm 1: LWE

Private key. A matrix $S \in \mathbb{Z}_q^{n\times l}$ is chosen uniformly at random.
S is the private key.
Public key. A matrix $A \in \mathbb{Z}_q^{m\times n}$ is chosen uniformly at random and a matrix
$E \in \mathbb{Z}_q^{m\times l}$ is chosen, whose each element entries are according to $\bar{\Psi}_a$.
The public key is $(A, P = AS + E) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^{m\times l}$.
Encryption. Given a message $v \in \mathbb{Z}_t^l$ and the public key (A, P), a vector
$a \in \{-r, -r+1, \dots, r\}^m$ is chosen uniformly at random.
The output is the ciphertext $\left(u = A^T a, c = P^T a + f(v)\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^l$.
Decryption. Given the private key $S \in \mathbb{Z}_q^{n\times l}$ and the ciphertext $(u, c) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^l$,
recover the plaintext $f^{-1}\left(c - S^T u\right)$.

FIG. 1. LWE Cryptosystem [8]

S is generalized to be a matrix $S \in \mathbb{Z}_q^{n\times l}$ and the public key is given as $(A, P = A \cdot S + E)$ where the elements of A and S are chosen uniformly at random and E is sampled from an appropriate distribution Ψ_a as described in Section II. The encryption and decryption procedures are specified so that decrypting the ciphertext recovers the message v:

$$\begin{aligned}
f^{-1}\left(c - S^T u\right) &= f^{-1}\left((P^T a + f(v)) - S^T(A^T a)\right) \\
&= f^{-1}\left(((AS+E)^T a + f(v)) - S^T(A^T a)\right) \\
&= f^{-1}\left(S^T A^T a + E^T a + f(v) - S^T A^T a\right) \\
&= f^{-1}\left(E^T a + f(v)\right) \\
&\approx v.
\end{aligned}$$

While this is useful, KEMs are used in *symmetric* cryptosystems in which two parties establish a shared secret key K used for both encryption and decryption in a single communication session. The role of the KEM is to allow the sender to generate the shared secret key K and transmit it securely to the receiver to initiate the session. This is done using the following three methods:

1. **Gen**(): Takes no inputs and returns the public key and private key pair (pk, sk).

2. **Encap**(pk): Takes the public key pk as input and returns the shared secret key K along with its encapsulation c.

3. **Decap**(sk, c): Takes the private key sk and the encapsulated shared secret key c as input and returns the shared secret key K.

All three of the above methods must be fully specified in order to define a valid KEM scheme. In the case of an LWE-based KEM scheme, these methods can be defined by adapting the pseudocode in FIG. 1 and invoking additional methods:

1. **Gen**(): Generate the public key pk by invoking **LWE.Public key** and the private key sk by invoking **LWE.Private Key**.

2. **Encap**(pk): Generate the shared secret key K by choosing $r \in \mathbb{Z}_q$ uniformly at random and invoking a key derivation function KDF so that $K = KDF(r)$. Generate the encapsulation c by invoking **LWE.Encryption** with the public key pk and the message $v = K$.

3. **Decap**(sk, c): Recover the shared secret key K by invoking **LWE.Decryption** with the private key sk and the encapsulation c.

Two parties that wish to communicate securely via a symmetric cryptosystem can invoke the KEM as described above to allow both parties to obtain a shared secret key. Once this is done, the two parties can then use the shared secret key for both encryption and decryption according to a typical symmetric key algorithm such as the Advanced Encryption Standard (AES).

IV. MODULE LEARNING WITH ERRORS

A. Motivation & Overview

In Section III, it was mentioned that a theoretical problem must be computationally feasible and efficient to implement in order to be utilized as a practical crypotsystem. It turns out that LWE alone does not satisfy this criterion because real-world cryptosystems typically require large key sizes for security — RSA for example uses 2048 bits. Equation 6 provides some reasoning for why LWE would also require large key sizes for security as the result of the LLL algorithm in solving SVP is exponentially worse in the size of the private key n. However, the size of the matrix A will also increase for larger n since $A \in \mathbb{Z}_q^{n\times l}$, and matrix multiplication with large matrices is computationally expensive to implement at scale. In order to address this limitation, the module learning with

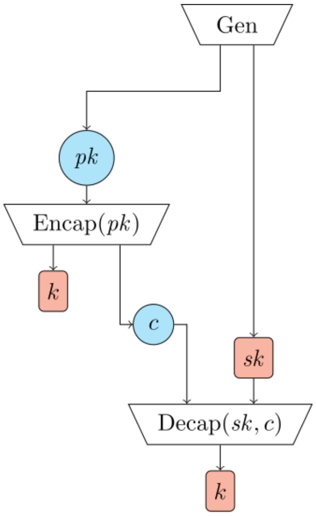


FIG. 2. KEM Schematic

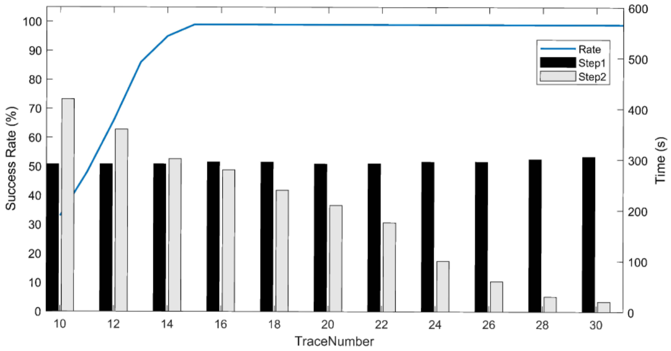


FIG. 3. Relationship of Success Rate & Time to Number of Power Traces for Two-Step Attack [10]

errors problem (MLWE) introduces an extension of LWE based on modules in abstract algebra. This extension allows for smaller key sizes without compromising security and makes MLWE ideal for cryptosystems. MLWE cryptosystems are also thought to be robust to attacks by quantum computers and the recently approved Module-Lattice-Based Key-Encapsulation Mechanism Standard [9] is based on MLWE as its name suggests. This new KEM standard is one of the first three finalized post-quantum encryption standards and is the result of an eight year effort organized by the National Institute of Standards and Technology (NIST).

B. Practical Considerations

The theoretical security of the Module-Lattice-Based Key-Encapsulation Mechanism Standard has been rigorously tested; however, there are also many important practical details which must be considered to ensure that the standard remains secure once it is deployed. One particularly important consideration is side-channel attacks, which aim to break a cryptosystem by exploiting additional information about the system that can obtained due to flaws in its implementation. Recent work in post-quantum cryptography has begun to test the new KEM standard's resilience to various types of side-channel attacks to provide guidelines for secure implementation. One such work demonstrates the experimental success of a two-step attack which utilizes correlation power analysis (CPA) to recover the private key sk in around 9 minutes using a 16-core machine — these results are summarized in FIG. 3. The experimental CPA attacks are carried out under the assumption that the attacker can run the KEM decryption algorithm on a device with arbitrary ciphertext c and can capture the EM radiation of the device to record its power consumption. This additional information is what allows the attacker to break the KEM. Formally, in the first step of the attack, the attacker defines a computation during the decryption algorithm for which they will measure the power consumption. The power consumption is then measured for D different ciphertexts each at T different times during the computation and stored in $\mathbf{T}_{D\times T}$. The attacker must then formulate K guesses for the private key and compute $\mathbf{H}_{D\times K}$ which represents the hypothesized power consumption for every guessed key and ciphertext pair. Finally, the Pearson correlation coefficient (Equation 7) is computed for each column in $\mathbf{H}$ with each column in $\mathbf{T}$ to capture the correlation between the power consumption for each guessed key and the true power consumption for each ciphertext across all measurements in time. The resulting coefficients are stored in $\mathbf{P}_{K\times T}$:

$$\mathbf{P}_{ij} = \frac{\sum_{d=1}^{D}\left(H_{d,i} - \bar{H}_i\right)\left(T_{d,j} - \bar{T}_j\right)}{\sqrt{\sum_{d=1}^{D}\left(H_{d,i} - \bar{H}_i\right)^2}\sqrt{\sum_{d=1}^{D}\left(T_{d,j} - \bar{T}_j\right)^2}} \tag{7}$$

The largest correlation in $|P|$ can then be used to obtain a partial estimate of the private key. In order to obtain the full estimate of the private key, this process is repeated $38 \sim 60$ times in [10]. Step 2 of the attack then utilizes the estimated private key from step 1 in order to recover the exact private key. The success rate of this two-step attack is shown in FIG. 3 as a function of the number of power traces (D) measured, and notably the attack is always successful when at least 15 power traces are used in step 1. FIG. 3 also shows the time taken for both steps in the attack and highlights that step 2 is much faster in recovering the exact private key when a larger number of power traces are measured – less than 100s for at least 24 power traces – since this will produce a more accurate estimated private key in step 1. Ultimately, these results demonstrate that side-channel attacks such as CPA are capable of breaking the recently approved KEM scheme, emphasizing the need for secure implementation.

V. CONCLUSION

Looking ahead, the development and implementation of quantum-resistant cryptographic systems remains an essential and active area of research within the cryptography community. The three recently approved Federal Information Processing Standards for post-quantum cryptography signify a crucial step toward preparing for a future where quantum computers can be used by malicious adversaries to threaten individual and national security, and it is important that current products and encryption systems begin to incorporate these new standards as soon as possible. Much work remains to be done in this regard in ensuring that the adoption of the new post-quantum cryptographic standards is both rapid and widespread. NIST is also currently evaluating two other candidate algorithms that have the potential to serve as backup standards in case of unforeseen developments in quantum computing technology. These ongoing efforts to address urgent challenges in cryptography are essential in order to continue making meaningful progress toward a secure and quantum-resilient future.

[1] S. M. L. Pfaendler, K. Konson, and F. Greinert, Advancements in quantum computing—viewpoint: Building adoption and competency in industry, Datenbank Spektrum **24**, 5 (2024).

[2] M. Harmalkar, K. Jain, and P. Krishnan, A survey of post quantum key encapsulation mechanism, in *2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)* (2024) pp. 141–149.

[3] O. Regev, The learning with errors problem (invited survey), in *2010 IEEE 25th Annual Conference on Computational Complexity* (2010) pp. 191–204.

[4] H. Bennett and C. Peikert, Hardness of bounded distance decoding on lattices in 120001p norms, CoRR **abs/2003.07903** (2020), 2003.07903.

[5] V. Lyubashevsky and D. Micciancio, On bounded distance decoding, unique shortest vectors, and the minimum distance problem, in *Advances in Cryptology - CRYPTO 2009*, edited by S. Halevi (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009) pp. 577–594.

[6] R. Allen, R. E. Berker, S. Casacuberta, and M. Gul, Quantum and classical algorithms for bounded distance decoding (2022), arXiv:2203.05019 [cs.CC].

[7] X. Deng, An introduction to lenstra-lenstra-lovasz lattice basis reduction algorithm (2016).

[8] M. E. Sabani, I. K. Savvas, and G. Garani, Learning with errors: A lattice-based keystone of post-quantum cryptography, Signals **5**, 216 (2024).

[9] NIST, Module-lattice-based key-encapsulation mechanism standard, `https://doi.org/10.6028/nist.fips.203` (2024).

[10] K. Wang, D. Xu, and J. Tian, An improved two-step attack on kyber (2024), arXiv:2407.06942 [cs.CR].