Boğaziçi University

Department of Computer Engineering

CMPE 220 – Fall 2025 – Homework 2

Exploring Post-Quantum Cryptography Through Interactive Dialogue

Furkan Ahi

Contents

1 Introduction

Post-quantum cryptography has become one of the most critical research areas in modern cybersecurity. Although large-scale quantum computers capable of breaking today's public-key systems such as RSA and Elliptic Curve Cryptography (ECC) do not yet exist, the rapid progress in quantum hardware and algorithms has made it clear that classical cryptographic schemes will eventually become vulnerable. This creates an urgent need to understand, evaluate, and transition to new cryptographic systems that remain secure even in the presence of powerful quantum adversaries.

For this reason, I chose to focus on **lattice-based cryptography**, specifically the **Module-Learning With Errors (Module-LWE)** structure used in **CRYSTALS-Kyber (ML-KEM)**, the post-quantum key establishment mechanism selected by NIST. Lattice-based schemes represent one of the most promising families in post-quantum cryptography due to their strong theoretical foundations, resistance to quantum attacks, and practical performance across modern computing platforms.

The main motivation behind this study is to understand how the mathematical structure of lattices provides security, why Learning With Errors (LWE) and its variants are considered quantum-resistant, and how these problems evolve into more structured and efficient forms such as Ring-LWE and Module-LWE. In particular, Kyber's use of Module-LWE is notable because it achieves a careful balance between efficiency, conservative security assumptions, and scalable design suitable for real-world deployment. Understanding the reasoning behind NIST's choice of Kyber over other strong candidates such as NTRU or FrodoKEM also provides valuable insight into how modern cryptographic standards are shaped.

Throughout this project, I used a Large Language Model (ChatGPT) as an interactive learning resource, repeatedly asking questions, requesting clarifications, and exploring related concepts such as lattice constructions, LWE hardness, Ring-LWE optimizations, Module-LWE security, Kyber's encapsulation/decapsulation mechanism, and the "Harvest Now, Decrypt Later" threat model. Whenever a term or concept appeared unfamiliar, I asked follow-up questions to ensure deeper understanding and continuity.

This report documents the dialogue with the LLM, followed by a consolidated summary of the concepts I learned and a critical evaluation of the LLM's weaknesses—particularly in areas where explanations may oversimplify, omit technical subtleties, or lack cryptographic nuance.

2 Chat Log with ChatGPT

My Question: Can you explain the idea behind post-quantum cryptography in simple terms?

ChatGPT: Post-quantum cryptography refers to a set of cryptographic algorithms designed to remain secure even if extremely powerful quantum computers become available in the future.

Classical cryptographic systems—such as RSA and Elliptic Curve Cryptography (ECC)—are based on mathematical problems that are hard for classical computers but become easy for quantum computers due to algorithms like Shor's algorithm. This means that once quantum computers reach a certain scale, they could break many of the encryption systems we rely on today.

Post-quantum cryptography aims to develop new algorithms that are resistant to both classical and quantum attacks, typically relying on mathematical structures such as lattices, error-correcting codes, multivariate equations, and hash-based constructions. The key idea is simple: we need encryption methods that will remain secure even in a future where large-scale quantum computers exist.

My Question: How exactly can quantum computers break classical cryptography like RSA and ECC?

ChatGPT: Quantum computers can break classical cryptography because they can solve certain mathematical problems exponentially faster than classical computers. RSA depends on the hardness of factoring large integers, and ECC relies on the difficulty of solving the discrete logarithm problem. Classical computers need astronomical time to solve these problems for current key sizes.

Quantum computers, however, can run Shor's algorithm, which efficiently factors large integers and solves discrete logarithms. Tasks that would take classical computers millions of years could potentially be solved in hours or days on a sufficiently large quantum computer. As a result, attackers could recover private keys and decrypt secure communications.

My Question: Since classical schemes like RSA and ECC will eventually be broken by quantum computers, I want to understand what types of post-quantum approaches exist. What are the main families of post-quantum cryptographic algorithms?

ChatGPT: Post-quantum cryptographic algorithms generally fall into five major families:

- **Lattice-Based Cryptography** – Relies on hard problems such as LWE, Ring-LWE, SIS, and SVP. Most NIST-selected algorithms, including Kyber and Dilithium, belong to this family.
- **Code-Based Cryptography** – Based on the hardness of decoding random linear codes (e.g., Classic McEliece).
- **Hash-Based Cryptography** – Signature schemes built purely from hash functions (XMSS, SPHINCS+).
- **Multivariate Cryptography** – Based on solving systems of multivariate quadratic equations.
- **Isogeny-Based Cryptography** – Uses the difficulty of finding isogenies between elliptic curves.

These families represent different mathematical foundations believed to resist quantum attacks.

My Question: Since lattice-based cryptography is one of the most important and widely adopted families—and also the foundation of Kyber—I want to explore it in more depth. I understand that a lattice is a grid formed by linear combinations of basis vectors. However, I am curious about how this structure actually provides cryptographic security—specifically, what exactly an attacker is unable to solve.

Could you explain the Shortest Vector Problem (SVP) and its mathematical definition? Also, what is the relationship between SVP and the Learning With Errors (LWE) problem? Specifically, how does the standard LWE equation look in terms of matrix dimensions, and why does the "error" term make it hard to solve?

ChatGPT: The **Shortest Vector Problem (SVP)** asks for the shortest non-zero vector in a lattice. Formally, for a lattice $L \subset \mathbb{R}^n$, SVP seeks

$$\lambda_1(L) = \min_{v \in L \setminus \{0\}} \|v\|.$$

SVP is difficult because high-dimensional lattices contain exponentially many vectors. No efficient algorithms—classical or quantum—are known for solving SVP or even approximating it well in worst-case scenarios.

The **Learning With Errors (LWE)** problem considers noisy linear equations. In matrix form, LWE is

$$b = A \cdot s + e \pmod q,$$

where

- A is a public $n \times n$ matrix,
- s is a secret vector in $\mathbb{Z}_q^n$,
- e is a small error vector in $\mathbb{Z}_q^n$.

Without the error term, the system can be solved using Gaussian elimination. However, the small error term e makes the equations inconsistent, preventing standard linear algebra from recovering s.

Regev proved that solving LWE is at least as hard as solving worst-case lattice problems such as BDD and SVP, giving LWE extremely strong security guarantees.

My Question: Earlier you mentioned that LWE involves expressions of the form $A \cdot s + e$. But in high dimensions such as $n = 500$ or 1000, storing and operating on a random $n \times n$ matrix is expensive, especially for IoT devices. Modern standards such as Kyber rely on Ring-LWE. How does RLWE solve these efficiency problems? What does switching to a "ring" structure change mathematically, and why is polynomial multiplication advantageous?

ChatGPT: Ring-LWE replaces large matrices with polynomials in the ring

$$R_q = \mathbb{Z}_q[x]/(x^n + 1).$$

Instead of storing a full $n \times n$ matrix, RLWE uses a single polynomial of degree n. The equation

$$b = A \cdot s + e$$

becomes

$$b(x) = a(x)s(x) + e(x) \pmod{x^n + 1, q}.$$

This yields two major benefits:

- **Compactness:** Only n coefficients are stored instead of n^2.
- **Speed:** Polynomial multiplication can be accelerated with the **Number Theoretic Transform (NTT)**, reducing complexity from $O(n^2)$ to $O(n \log n)$.

These advantages make RLWE highly suitable for constrained environments.

My Question: That makes sense. So, by using polynomial rings (Ring-LWE), we solve the efficiency and memory issues. Now I want to see how this is applied in the real world. I noticed that CRYSTALS-Kyber (ML-KEM) is based on Module-LWE rather than plain Ring-LWE. Why is this module structure needed? Where does Module-LWE sit between LWE and Ring-LWE, and how does it allow Kyber to scale its security levels?

ChatGPT: Module-LWE generalizes Ring-LWE by working in modules R_q^k, providing a middle ground between the unstructured nature of LWE and the rigid ring structure of RLWE.

Kyber fixes the polynomial degree at $n = 256$. Instead of increasing n for higher security levels, Kyber increases the module rank k:

- Kyber-512: $k = 2$,
- Kyber-768: $k = 3$,
- Kyber-1024: $k = 4$.

This lets Kyber scale security without changing the underlying polynomial degree, maintaining the same efficient NTT operations and simplifying implementation across all security levels.

My Question: What exactly is CRYSTALS-Kyber (ML-KEM), and why is it described as a Key Encapsulation Mechanism (KEM) instead of a standard Public Key Encryption scheme? Could you explain step-by-step how Alice and Bob establish a shared key?

ChatGPT: Kyber is a Module-LWE-based **Key Encapsulation Mechanism (KEM)**. It doesn't encrypt arbitrary messages; instead, it encapsulates a randomly generated symmetric key.

KeyGen (Bob):

- Generates a secret s and error e.
- Computes $t = As + e \bmod q$.
- Publishes (A, t); keeps s.

Encapsulation (Alice):

- Samples random r, e_1, e_2.

- Computes
$$u = Ar + e_1, \quad v = tr + e_2 + \text{encode}(k).$$
- Sends ciphertext (u, v).
- Derives shared key $K = \text{KDF}(\text{hash}(k, u, v))$.

Decapsulation (Bob):

- Computes $v' = v - su$.
- Recovers k through Kyber's reconciliation mechanism, which corrects small noise differences.
- Derives the same shared key K.

The reconciliation step ensures both parties derive exactly the same key despite the presence of noise.

My Question: Why did NIST select CRYSTALS-Kyber as the primary post-quantum key establishment algorithm instead of other candidates such as NTRU or FrodoKEM? What factors were decisive?

ChatGPT: Kyber offered the best balance across all major criteria:

- **Performance:** Extremely fast on CPUs and constrained devices.
- **Key Size:** Much smaller than FrodoKEM's very large keys.
- **Security:** Based on conservative Module-LWE assumptions with strong reductions.
- **Implementation Safety:** Simple, constant-time operations.
- **Patent-Free:** Clean, royalty-free intellectual-property status.
- **Scalability:** Security levels increase by adjusting module rank k, not polynomial degree.

This combination made Kyber the most practical and future-proof scheme for global deployment.

My Question: Finally, what exactly is the "Harvest Now, Decrypt Later" attack, and why do NSA and NIST recommend migrating to post-quantum cryptography immediately?

ChatGPT: "Harvest Now, Decrypt Later" is a threat model in which adversaries collect encrypted data today, store it for years, and decrypt it later once quantum computers are powerful enough to break RSA and ECC with Shor's algorithm.

The urgency comes from:

- Many types of data (state secrets, medical data) must remain confidential for decades.
- Storage costs are low, so attackers already archive encrypted traffic.
- Migrating global cryptographic infrastructure takes 5–15 years.
- Once quantum computers arrive, classical public-key cryptography breaks instantly.

Therefore, NSA and NIST emphasize immediate migration to PQC to protect today's sensitive data from future quantum adversaries.

3 Summary of Lessons Learned

Throughout this interactive learning process, I developed a comprehensive understanding of **post-quantum cryptography (PQC)**, the motivations behind it, and the specific mathematical foundations underlying modern standardized schemes such as **CRYSTALS-Kyber (ML-KEM)**. My learning followed a layered progression—from the high-level **quantum threat model** to the underlying **lattice-based hardness assumptions**—and the dialogue allowed me to clarify concepts step by step.

First, I learned why post-quantum cryptography is necessary. Large-scale quantum computers do not yet exist, but algorithms such as **Shor's algorithm** demonstrate that once such machines become available, they will be capable of breaking classical public-key systems like **RSA** and **Elliptic Curve Cryptography** by efficiently solving the integer factorization and discrete logarithm problems. Because cryptographic transitions take many years, and because some data must remain confidential for decades, there is an urgent need to adopt **quantum-resistant schemes** today.

Next, I explored the major families of post-quantum cryptography, including **lattice-based**, **code-based**, **hash-based**, and **isogeny-based** cryptography. Among these, lattice-based cryptography emerged as the most promising category, forming the basis of NIST's selected standards. This motivated a deeper examination of lattice problems, particularly the **Shortest Vector Problem (SVP)**, which is computationally hard even for quantum computers.

Understanding the **Learning With Errors (LWE)** problem was a key turning point. LWE expresses noisy linear equations of the form

$$b = A \cdot s + e \pmod q,$$

where the error vector makes the system resistant to linear algebraic attacks. I learned that the hardness of LWE is tightly connected to worst-case lattice problems such as **SVP**, giving LWE strong theoretical security guarantees.

I then explored how real-world implementations avoid the inefficiencies of using large random matrices. **Ring-LWE** replaces matrices with polynomial arithmetic in structured rings, enabling significant improvements in efficiency due to the **Number Theoretic Transform (NTT)**. Building on this, **Module-LWE** extends RLWE to modules of rank k. This module-based approach allows schemes like Kyber to scale security levels by changing the **module dimension** rather than increasing the polynomial degree—preserving efficient NTT operations.

I gained a detailed understanding of **CRYSTALS-Kyber**, its design as a **Key Encapsulation Mechanism (KEM)**, and its use of Module-LWE. I learned how Kyber's encapsulation/decapsulation process works, how noise terms are handled, and why Kyber encapsulates random symmetric keys instead of encrypting arbitrary messages. I also noted that, unlike RSA, lattice schemes have a **probabilistic** nature, where a decryption failure is theoretically possible but negligibly rare.

Another important insight was why **NIST** selected Kyber over candidates like **FrodoKEM** or **NTRU**. The decisive factors included Kyber's strong balance of **security**, **performance**, **compact key sizes**, and **patent-free** status. Finally, I learned the importance of the **"Harvest Now, Decrypt Later"** threat model, which compels organizations to transition to PQC now to protect present-day communications from future quantum decryption.

Overall, this learning process gave me a clear, structured understanding of post-quantum cryptography—from fundamental motivations to real-world standardization decisions.

4 Discussion on Weak Points of ChatGPT

Although ChatGPT was highly effective in guiding my understanding of post-quantum cryptography, it exhibited several limitations throughout the learning process. These weaknesses reflect the fundamental challenges of **Large Language Models (LLMs)** when dealing with highly technical, mathematically dense, and security-critical topics.

1. **Tendency to Oversimplify Mathematical Rigor.** While the model correctly described concepts like LWE, it often simplified the distinction between *mathematical proof* and *computational assumption*. For instance, it presented the hardness of lattice problems as a definitive fact, whereas in cryptography these are technically **conjectures** (unlike the existence of prime numbers). It also struggled to detail the structure of security reductions (e.g., how LWE reduces to worst-case SVP) without explicit prompting, treating them as narrative descriptions rather than formal mathematical objects.

2. **Inconsistent Precision in Notation and Parameters.** Cryptography requires explicit definitions. During our discussion on LWE, the model initially provided the equation $b = A \cdot s + e$ without specifying the dimensions of the matrices or the probability distribution of the error term e (which typically follows a centered binomial distribution in Kyber). It assumed implicit context. For a student trying to implement a simulation, such omissions of concrete parameters are critical and required follow-up questions to clarify.

3. **Blindness to Implementation Pitfalls (Side-Channels).** The model focused heavily on the theoretical algebra of Kyber but initially failed to mention practical implementation vulnerabilities, such as side-channel attacks or decryption failure probabilities. In real-world cryptography, a mathematically sound scheme can be broken if the implementation leaks data via power consumption or timing. The LLM did not proactively raise these engineering risks, limiting its scope to abstract theory.

4. **Dependence on User Guidance (Prompt Sensitivity).** I observed that the model does not autonomously expand on deep technical layers unless guided by precise questions. Its effectiveness relied heavily on my ability to identify ambiguities (e.g., asking specifically about the module structure or NIST's selection criteria). This suggests that while LLMs are powerful tools for retrieval and explanation, they cannot replace structured study, as they mirror the user's level of inquiry rather than leading the pedagogical process.

5 Conclusion

This project provided a structured and incremental exploration of post-quantum cryptography, focusing on the mathematical foundations and practical motivations driving the transition from classical public-key systems to quantum-resistant designs. By engaging in a step-by-step dialogue with ChatGPT, I was able to clarify essential concepts—from the quantum threat model and the weaknesses of RSA/ECC, to the core hardness assumptions behind lattice-based cryptography such as SVP and LWE.

Through this interactive learning process, I gained a deep understanding of how modern schemes like CRYSTALS-Kyber use Module-LWE to achieve an effective balance of security, performance, and deployability. I also learned why Kyber was selected by NIST as the primary post-quantum key establishment mechanism and how the "Harvest Now, Decrypt Later" threat model makes immediate migration urgent.

At the same time, evaluating ChatGPT's responses revealed important limitations of LLMs, particularly in areas requiring mathematical rigor, explicit notation, and awareness of real-world implementation risks. These observations highlight that LLMs are valuable companions for conceptual understanding, but not substitutes for formal study, peer-reviewed sources, or security-critical analysis.

Overall, this project strengthened both my technical understanding of post-quantum cryptography and my ability to critically assess AI-assisted learning tools. The combination of theoretical insight, practical context, and reflective evaluation allowed me to approach the topic with both depth and perspective.