Post-Quantum Cryptography

by
Madhav Basur – SE23UCAM009
Sanket Motagi – SE23UCAM020

Contents

1 Introduction

1.1 The Threat of Quantum Computing

Post-Quantum Cryptography (PQC) refers to cryptographic algorithms designed to be secure against quantum computers. The motivation for PQC arises from the fact that ongoing research in quantum computing could threaten existing cryptographic systems, especially public-key cryptosystems based on the Factoring problem (like RSA) and the Discrete Logarithm problem (like Diffie-Hellman and ECC).

A quantum computer is fundamentally different from a classical computer. Instead of bits that represent either 0 or 1, a quantum computer uses **qubits** (quantum bits), which can exist in a superposition of both 0 and 1 states simultaneously. This allows a quantum computer to process many computations at the same time. Moreover, quantum computers utilize **entanglement**, a phenomenon where qubits are strongly correlated with each other, enabling complex parallel processing.

The breakthrough that made quantum computers a significant concern for cryptography was **Shor's Algorithm**, published in 1994. Shor's Algorithm can factor large integers and compute discrete logarithms in **polynomial time** on a quantum computer. This means that public-key systems like RSA, DSA, and ECC — which rely on the difficulty of these problems — would be completely broken if a sufficiently powerful quantum computer could be built.

Currently, practical quantum computers capable of breaking RSA-2048 do not exist. However, researchers estimate that such computers could be developed within the next few decades. Mike Mosca, a leading expert in quantum computing, predicted in 2016 that there was about a 1 in 7 chance of a quantum computer being able to factor RSA-2048 by 2026, and a 50% chance by 2031.

Because it can take many years (or decades) to develop, standardize, and deploy new cryptographic systems across industries and government, there is an urgent need to develop **post-quantum cryptographic algorithms** today. Organizations like **NIST (National Institute of Standards and Technology)** and **NSA** have prioritized research and standardization of PQC, with the goal of transitioning critical systems before quantum computers arrive.

It is important to note that quantum computers will not affect all cryptographic systems equally. For example, **secret-key (symmetric) cryptography** (like AES) is less affected: Grover's Algorithm shows that exhaustive key search can be sped up quadratically (from $O(2^n)$ to $O(2^{n/2})$). Therefore, doubling the key length (e.g., using AES-256 instead of AES-128) can maintain security against quantum attacks.

1.2 Quantum Cryptography vs. Post-Quantum Cryptography

It is important to understand the difference between **quantum cryptography** and **post-quantum cryptography**, because although their names sound similar, they refer to very different concepts.

Quantum cryptography refers to cryptographic techniques that **use principles of quantum mechanics** to achieve security. The most well-known example of quantum cryptography is **Quantum Key Distribution (QKD)**, where two parties use quantum particles (like photons) to create and share a secret key. Quantum cryptography can achieve **unconditional security** based on the laws of physics, meaning that no amount of computational power can break it, as long as the laws of quantum mechanics hold.

In contrast, **post-quantum cryptography (PQC)** refers to cryptographic algorithms that are designed to be secure **against attacks by quantum computers**, but which do **not use quantum mechanics** themselves. PQC algorithms are still implemented on classical (normal) computers and rely on mathematical problems that are believed to be hard even for quantum computers.

Quantum cryptography requires **special hardware and quantum communication channels** (like optical fibers that carry photons without disturbance), while post-quantum cryptography can work on **existing digital infrastructure** (like current computers, networks, and devices).

1.3 Approaches to Post-Quantum Cryptography

Researchers have explored several approaches to building cryptographic systems that remain secure against quantum computers. The main approaches include:

- **Lattice-based cryptography**: uses hard problems related to points in a lattice structure.
- **Code-based cryptography**: based on the difficulty of decoding random linear error-correcting codes.
- **Multivariate cryptography**: relies on solving systems of multivariate quadratic equations.
- **Hash-based signature schemes**: constructs signatures using only hash functions.
- **Isogeny-based cryptography**: based on the difficulty of finding isogenies between elliptic curves.

Each of these approaches is being studied and tested to evaluate their security and practicality for future cryptographic standards.

2 Lattice-based Cryptography

Lattice-based cryptography is one of the most promising approaches to post-quantum cryptography. Its security relies on the hardness of certain mathematical problems in lattices that are believed to be difficult for both classical and quantum computers.

A **lattice** in mathematics is a regular arrangement of points in space. Formally, a lattice is defined as the set of all integer linear combinations of a set of linearly independent vectors called a *basis*. For example, in two dimensions, a lattice can be visualized as a grid formed by repeating translations of two basis vectors.

Two important problems in lattice theory form the foundation of lattice-based cryptography:

- **Shortest Vector Problem (SVP)**: Given a lattice, find the shortest non-zero vector in the lattice.

- **Closest Vector Problem (CVP)**: Given a lattice and a target point, find the lattice vector closest to the target point.

Both SVP and CVP are known to be computationally hard, especially as the dimension of the lattice increases. Many cryptographic constructions rely on the assumption that solving these problems is infeasible in high dimensions.

Lattice-based cryptographic schemes have several advantages:

- They are believed to be secure against quantum attacks since no efficient quantum algorithms are known for SVP or CVP.
- They allow the construction of advanced cryptographic tools such as fully homomorphic encryption and identity-based encryption.
- They are relatively fast and simple to implement.

A well-known lattice-based cryptosystem is **NTRU**, which we describe in the next section. Another important family of lattice-based schemes are those based on the **Learning With Errors (LWE)** problem, which underlies many modern lattice-based cryptographic protocols.

2.1 NTRU - N^{th} degree Truncated polynomial Ring Units

NTRU is a lattice-based public-key cryptosystem introduced by Hoffstein, Pipher, and Silverman in 1996. It is known for its efficiency and relatively simple structure compared to other public-key systems.

NTRU works with polynomials whose coefficients are integers modulo some numbers. Computations are performed in the ring:

$$R = \mathbb{Z}[x]/(x^N - 1)$$

where N is a positive integer and arithmetic is done modulo both $x^N - 1$ and some integers p and q with $p < q$.

Key generation in NTRU involves selecting two secret polynomials $F(x)$ and $G(x)$ with small coefficients. A polynomial $f(x)$ is defined as:

$$f(x) = 1 + pF(x)$$

and similarly, $g(x) = pG(x)$. The public key is computed as:

$$h(x) = f^{-1}(x) \star g(x) \pmod{q}$$

where $f^{-1}(x)$ is the inverse of $f(x)$ modulo q in the ring R. The public key is the polynomial $h(x)$, while the private key is $f(x)$. Decryption involves computing $a(x) = f(x) \star y(x) \pmod{q}$, where $\star$ denotes convolution multiplication[1].

To encrypt a message polynomial $m(x)$, the sender chooses a random small polynomial $r(x)$ and computes the ciphertext:

$$y(x) = r(x) \star h(x) + m(x) \pmod{q}$$

[1] Convolution multiplication here refers to polynomial multiplication modulo $x^N - 1$. It causes the degrees to wrap around cyclically: each coefficient a_k in the result is given by $a_k = \sum_{i=0}^{N-1} f_i y_{(k-i) \bmod N}$.

Decryption is performed by the receiver using their private key:

$$a(x) = f(x) \star y(x) \pmod q$$

The coefficients of $a(x)$ are then reduced modulo p to recover the original message $m(x)$:

$$m'(x) = a(x) \pmod p$$

Certain conditions on the sizes of N, p, q, and the coefficients of $F(x)$, $G(x)$, $r(x)$, and $m(x)$ are necessary to ensure that decryption succeeds with high probability without errors due to coefficient overflow.

The security of NTRU is related to the hardness of finding short vectors in a certain lattice constructed from the public key. Finding the private key $f(x)$ can be modeled as solving a Shortest Vector Problem (SVP), while recovering the plaintext from a ciphertext can be viewed as solving a Closest Vector Problem (CVP). Although NTRU does not have a formal proof that breaking it is as hard as solving SVP or CVP in the worst case, it has withstood extensive cryptanalysis over many years.

2.2 Learning With Errors (LWE)

The Learning With Errors (LWE) problem is another foundation of lattice-based cryptography. Informally, LWE involves solving a noisy system of linear equations over a finite field.

Given a secret vector $s \in \mathbb{Z}_q^n$, we generate a set of m samples of the form:

$$(a_i, b_i = \langle a_i, s \rangle + e_i \pmod q)$$

where each $a_i \in \mathbb{Z}_q^n$ is chosen uniformly at random, and e_i is an error term drawn from a distribution χ that outputs small integers.

The goal of the LWE problem is: given the pairs (a_i, b_i), recover the secret vector s. The presence of the error term e_i makes this problem difficult. Without the error, solving for s would be a system of linear equations, easily solvable with basic linear algebra. But with the added noise, finding s becomes computationally hard.

The hardness of LWE is backed by reductions showing that solving random instances of LWE is at least as hard as solving certain worst-case lattice problems like the Gap Shortest Vector Problem (GapSVP) in high dimensions.

No efficient quantum algorithms are known to solve LWE. This makes it an attractive basis for constructing post-quantum cryptographic schemes.

2.2.1 The Regev Cryptosystem

The Regev Cryptosystem is a public-key encryption scheme based on the LWE problem. It demonstrates how LWE can be used to build a secure cryptosystem.

Key generation works as follows. Choose a secret vector:

$$s \in \mathbb{Z}_q^n$$

and generate m random samples:

$$a_i \in \mathbb{Z}_q^n$$

$$b_i = \langle a_i, s \rangle + e_i \pmod{q}$$

where each e_i is drawn from the error distribution χ. The public key is the collection of pairs (a_i, b_i), and the private key is s.

To encrypt a single bit $x \in \{0, 1\}$, a random subset S of the public key indices is chosen. The ciphertext is computed as:

$$u = \sum_{i \in S} a_i \pmod{q}$$

$$v = \sum_{i \in S} b_i + \left\lfloor \frac{q}{2} \right\rfloor x \pmod{q}$$

To decrypt, compute:

$$d = v - \langle u, s \rangle \pmod{q}$$

If d is closer to 0 than to $q/2$, then the decrypted bit is 0; otherwise, it is 1.

Correct decryption requires that the combined error from the selected samples is small enough so that d remains closer to 0 or $q/2$ rather than crossing over the midpoint.

The Regev Cryptosystem is mainly a theoretical construction that demonstrates LWE's cryptographic usefulness. It is not practical for large messages due to inefficiency and large key sizes, but more efficient variants based on LWE and related problems have been developed.

3 Code-based Cryptography

Code-based cryptography is based on the difficulty of decoding general linear error-correcting codes, a problem known to be NP-hard. The most well-known code-based cryptosystem is the McEliece cryptosystem, which uses Goppa codes with hidden structure. It remains secure even against quantum adversaries, though its large key sizes make it less practical.

4 Multivariate Cryptography

Multivariate cryptography is built on the hardness of solving systems of multivariate quadratic equations over finite fields. These schemes are often used for digital signatures. Notable examples include the Hidden Field Equations (HFE) scheme and the Oil and Vinegar signature scheme. While efficient, many early variants have been broken, and ongoing research continues to refine secure constructions like UOV.

5 Hash-based Signature Schemes

Hash-based signature schemes rely only on the security of hash functions and are unaffected by quantum computers. They use one-time or few-time keys and extend their usability using Merkle trees. Though signature sizes can be large, schemes like the Winternitz and SPHINCS+ families offer practical solutions for post-quantum digital signatures.

Comparison of PQC Approaches

Approach	**Security Assumption**	**Efficiency**	**Remarks**
Lattice-based	Hardness of SVP/CVP or LWE	Very efficient (fast operations, small key sizes in ring variants)	Basis for many NIST finalists; supports encryption, key exchange, signatures
Code-based	Hardness of decoding random linear codes	Efficient decryption but large key sizes	McEliece is very mature; key size is major drawback
Multivariate	Solving systems of quadratic equations (MQ problem)	Fast signing, moderate verification; small keys	Mostly used for digital signatures; UOV is a strong candidate
Hash-based	Security of hash functions (e.g., pre-image resistance)	Moderate (Winternitz) to slow (Lamport); large signatures	Strong security, stateless or stateful options; Merkle trees extend usage
Isogeny-based	Hardness of finding isogenies between elliptic curves	Compact keys; slower operations	Youngest field; promising but under more active research

Table 1: Comparison of Post-Quantum Cryptographic Approaches

6 Conclusion

The rapid progress in quantum computing poses a serious threat to existing cryptographic systems, particularly those relying on factoring and discrete logarithms. Post-quantum cryptography (PQC) aims to develop new cryptographic schemes that remain secure in the presence of quantum adversaries.

While practical quantum computers capable of breaking current systems may still be years away, the transition to quantum-resistant algorithms requires significant time for research, standardization, and deployment. Recognizing this urgency, organizations like NIST and NSA have prioritized PQC, with multiple candidates already in advanced stages of evaluation.

Ongoing efforts in the cryptographic community continue to refine the design, analysis, and implementation of post-quantum algorithms to ensure secure and practical systems for the future.

References

[1] Douglas R. Stinson, Maura B. Paterson, *Cryptography: Theory and Practice*, CRC Press.

[2] S. I. Gass, *Linear Programming: Methods and Applications*, Dover Publications.

[3] Shor, P. W. Algorithms for quantum computation: discrete logarithms and factoring. In *Proc. 35th Ann. Symp. on Foundations of Computer Science (FOCS '94)* 124–134 (IEEE, 1994)

[4] Bernstein, D., Lange, T. Post-quantum cryptography. *Nature* 549, 188–194 (2017). https://doi.org/10.1038/nature2346