Post Quantum Cryptography Lattice-Based Algorithms

Dylan Packer
packe089@morris.umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

This paper describes lattice-based post quantum cryptographic algorithms designed to secure digital communication against quantum attacks. Quantum computers pose a threat to classical encryption schemes such as RSA and ECC by efficiently solving problems that were previously considered computationally hard. Lattice-based cryptography relies on mathematical problems such as the *Shortest Vector Problem* and *learning with errors*, which remain resistant to quantum algorithms. Two key encapsulation mechanisms Kyber and Saber, are explored as leading candidates for post quantum encryption. Both schemes support multiple security levels tailored to different deployment needs, from lightweight Internet of Things (IoT) devices to secure government infrastructure. Kyber and Saber demonstrate that quantum-resistant encryption can be both practical and scalable. These algorithms reflect a broader shift toward cryptographic systems that prioritize both theoretical robustness and real-world deployability.

Keywords: Post quantum cryptography, learning with errors, lattices, qubits, shortest vector problem, number theoretic transform

1 Introduction

We trust that our data is protected from those that want to do wrong. When we message people, we don't want people eavesdropping on conversations or worse, bank transactions. The modern methods of cryptography protect that data from current attacks, but what happens when stronger, faster, better computers come along? A sufficiently powerful quantum computer can leverage quantum mechanical effects to solve problems related certain code-breaking problems faster than any classical computers we have today. Developing security protocols against possible quantum attacks before they become a reality is crucial.

Post quantum cryptography (PQC) encompasses a diverse set of algorithms designed to resist attacks from quantum computers. Quantum computers are based on the properties of quantum mechanics. To process information, quantum computers utilize two-level quantum systems, known as quantum bits (qubits) [3]. Instead of using binary bits that represent either 0 or 1, qubits can represent 0, 1, or both at the same time. This property of quantum mechanics is called superposition. Another property that quantum computers use is entanglement. Entanglement links qubits so they are mathematically correlated, allowing the computer to explore many possibilities in parallel and find solutions to complex problems much faster than classical computers. This allows quantum computers to process vast combinations of possibilities simultaneously, making them incredibly powerful. This is called quantum parallelism. This power poses a major threat to modern encryption schemes, but quantum computers today can only handle small numbers and special cases. Many current encryption systems, such as RSA and Elliptic Curve Cryptography, rely on these hard problems that classical computers would take a long time to compute. These problems may become easy to solve using algorithms that take advantage of quantum parallelism, such as Shor's Algorithm [10]. This is why the development of post quantum cryptography is important. In the next section I'll describe the background on Shor's algorithm and the National Institute of Standards and Technology (Section 2). Section 3 discusses some of the current algorithms that are used today. Section 4 describes lattices and hard lattice problems. Section 5 covers the post quantum algorithms that show the most promise in being quantum resistant. Section 6 covers two of the most promising lattice-based PQC finalist. Section 7 will conclude this paper.

2 Background

In the mid-1990s, a mathematical breakthrough by Peter Shor altered the landscape of both quantum computing and digital security. Shor's algorithm is a quantum procedure that, if run on a sufficiently powerful quantum computer, could efficiently break the cryptographic systems that underpin much of today's secure digital communication. Shor's algorithm is the quantum method that transforms the factorization problem, finding the prime factors of a large number that underlay the security of RSA encryption (Section 3). This makes factorization problem, seemingly intractable for classical computers, into one that a quantum computer can solve efficiently solving the factoring problem exponentially faster. While classical computers would take longer than the age of the universe to factor the numbers used in today's encryption, a quantum computer running Shor's algorithm could, in theory, do it in hours or days. Shor's algorithm could render essentially all cryptography deployed today

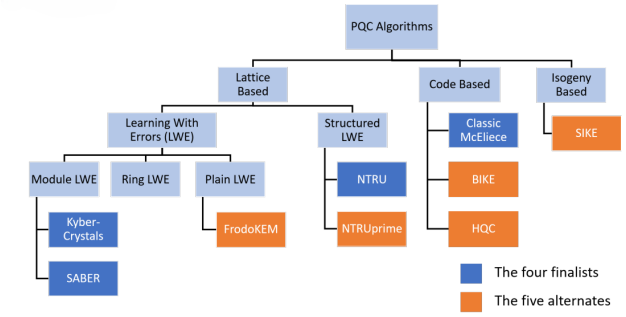

Figure 1. NIST finalist for selection of standard PQC algorithms[6]

insecure. Shor's algorithm is often described as a hybrid algorithm, combining classical and quantum steps. For more information about how Shor's algorithm works refer here [12].

2.1 NIST

The National Institute of Standards and Technology (NIST) is a government agency under the US Department of Commerce. NIST's main goal is to develop and promote standards in technology. These standards are used by government agencies and many private businesses. When NIST sets a standard, they follow a process. The process starts by identifying the problem. This could be a need for a new standard or a new technology that could cause a security threat, like the possibility of a computer with orders more of computing power than what there is now. Once the problem has been identified, NIST makes an announcement to the public outlining the criteria and invites experts and organizations worldwide to submit proposals, algorithms, or solutions. The next step in the process is evaluating each proposal. NIST puts each proposal through meticulous testing and performance analysis, often going through multiple rounds of testing. The proposals that make it past the testing are then published for community review, where other experts and members of the community review each proposal and give feedback. After receiving feedback, the proposals can be refined and retested, or even discarded. The last step is the final selection, where, after review, NIST selects the best-suited algorithms or solutions. For PQC in 2017, NIST started with 69 candidates. Then, in 2019, in the second round of evaluation, shortened the list down to 26. In 2020, 15 candidates made it to the third round, where only 9 passed [6]. There were four finalists and five alternates as seen in Figure 1. NIST characterized the remaining candidates into to three categories: lattice-based, code-based, and isogeny based algorithms. This paper will focus on lattice-based algorithms that use *module learning with errors*.

3 Current Cryptography

Modern cryptography is the science of protecting digital information using mathematical techniques that ensure confidentiality, integrity, and authenticity. Most systems today rely on a combination of symmetric and asymmetric encryption algorithms.

3.1 Symmetric Encryption

In *symmetric encryption*, the same secret key is used to both encrypt and decrypt data [11]. A secret key is some cipher that someone uses to hide their message and reveal a message. Key encapsulation mechanisms (KEMs), which are cryptographic protocols used to securely establish a shared secret key between two parties over an insecure channel. A widely used example is the Advanced Encryption Standard (AES), which transforms plaintext into ciphertext using a series of substitutions and permutations based on a shared key. AES is designed to operate efficiently on digital hardware, using structured byte-level transformations and matrix operations to scramble data in a way that is both fast and secure, provided the key remains secret from third parties [11].

3.2 Asymmetric Encryption

In contrast, *asymmetric encryption* uses two different keys: a public key for encryption and a private key for decryption. The public key can be shared with anyone, while the private key must be kept secret. A well-known example of this is RSA, which relies on the fact that factoring very large numbers is extremely difficult. To create RSA keys, two large prime numbers P and Q are chosen and multiplied to get $n = P \cdot Q$. From this, a value called Euler's totient is calculated as $\phi(n) = (P-1)(Q-1)$ [11]. Next, a number e is selected as the public exponent, provided that it does not share any factors with $\phi(n)$. The private key is another number, d, which is chosen so that it undoes the effect of e. More precisely, d is defined by this condition. This means that when a message is raised to the power of e (encryption) and then to the power of d (decryption), the original message is recovered. In simple terms, e and d are special exponents that work together so that only the person with the private key can reverse the encryption.

Another asymmetric method is Elliptic Curve Cryptography (ECC), which provides security comparable to RSA but with much smaller key sizes. ECC is built on the algebraic structure of elliptic curves defined over a finite field. Starting with a point P on the elliptic curve and a secret integer k, we compute another point $Q = k \cdot P$. Here, the point P adds itself k times according to the curve's group law. Both P and Q are points lying on the elliptic curve. While it is straightforward to calculate Q from P and k, it is computationally infeasible to reverse the process and recover k from P and Q. This one-way difficulty is known as the Elliptic Curve Discrete Logarithm Problem [5].

Together, these cryptographic methods and others form the backbone of secure communication on the internet today, from securing websites and emails to protecting financial transactions and personal data. While they are effective against classical computers, their security assumptions are being reevaluated in light of emerging quantum threats.

4 Lattice-Based Algorithms

In this section, we will describe lattices, the hard lattice problems, and their variants. Then I'll will introduce hard problems like the shortest vector problem and learning with errors (LWE). I'll explain the three variants: Plain-LWE, Ring-LWE, and Module-LWE.

4.1 Lattices

Lattices are geometric structures made up of regularly spaced points that extend infinitely in multiple directions. These points form a repeating pattern in space and are generated by linear combinations of a set of basis vectors [9]. Each basis vector originates from a common point, the origin, and points in a specific direction and length. By stacking and combining these vectors with integer coefficients, we construct paths that reach every point in the lattice. Mathematically, a lattice $\mathcal{L}$ in $\mathbb{R}^n$ is defined as: $\mathcal{L} = \left\{\sum_{i=1}^{d} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z}\right\}$ where $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d\}$ are the basis vectors [6]. The number of basis vectors determines the dimension of the lattice. For example, two basis vectors generate a two-dimensional lattice, while d basis vectors generate a d-dimensional lattice. As the dimension increases, the structure becomes exponentially more complex, making certain computational problems on lattices extremely difficult [6].

4.2 Hard Lattice Problems

One hard problem is the Shortest Vector Problem (SVP), which is the computational task of finding, within a lattice generated by basis vectors, the shortest non-zero vector measured by the distance from the origin. This is not simply a matter of measuring distance; in high dimensions, the number of possible combinations of coefficients becomes so intricate that even approximating the shortest vector is computationally hard for any computer, even a quantum computer. Quantum computers threaten traditional cryptographic schemes like RSA and ECC, but lattice problems such as SVP and Learning With Errors remain resistant to known quantum algorithms [6].The difficulty of solving SVP in high-dimensional spaces underpins the security of lattice-based schemes like Kyber and Saber [6] discussed in Section 6. These schemes leverage the complexity of lattice geometry to ensure that even with quantum capabilities, adversaries cannot efficiently recover secret keys or decrypt messages. Thus, the mathematical richness and computational intractability of lattices provide a potential foundation for quantum-resistant cryptographic design.

4.3 Learning With Errors

The learning with errors problem and its variants: Plain-LWE, Ring-LWE, and Module-LWE are central to the hardness assumptions that support lattice-based cryptography. Together with SVP, LWE provides the theoretical foundation for post quantum schemes. Plain-LWE asks us to recover a hidden secret vector from a set of linear equations. However, each equation is noisy: instead of exact values, we are given samples of the form (a, b), where a is a random vector and $b = \langle a, s\rangle + e$. Here, s is the secret vector or secret key we want to recover, and e is a small error term. The presence of this error makes the equations inconsistent, so standard linear algebra techniques fail. Even with many samples, the added noise renders the problem computationally difficult because the perturbations accumulate and obscure the true linear relationships. In high dimensions, every equation is almost correct but slightly distorted, so the solution space becomes exponentially large and indistinguishable from random data. Recovering s therefore requires separating structured signal from random noise, a task conjectured to be as hard as solving worst-case lattice problems such as SVP [6]. This hardness persists even when thousands of samples are available, ensuring that no polynomial-time method can reliably reconstruct the secret vector. Plain-LWE is conceptually straightforward and mathematically clean, which makes it easier to analyze, but schemes based directly on Plain-LWE often require large key sizes and slower operations compared to more structured variants.

Ring-LWE improves efficiency by replacing vectors with polynomials. Instead of working with noisy linear equations over integers, Ring-LWE embeds the problem into polynomial rings [6], where coefficients are taken modulo some integer q. This structure enables fast arithmetic operations, particularly through the Number Theoretic Transform (NTT) [6], the discrete analogue of the Fourier Transform. Just as the Fourier Transform accelerates signal processing, NTT speeds up polynomial multiplication, allowing Ring-LWE schemes to achieve smaller key sizes and faster performance while preserving the hardness of LWE.

Module-LWE generalizes both Plain-LWE and Ring-LWE by operating over modules of rings [6]. This flexible framework includes the earlier variants as special cases, allowing cryptographic designers to balance efficiency and security while maintaining strong hardness assumptions. Its added generality, however, requires careful design choices to achieve the desired performance.

Finally, Learning With Rounding (LWR) is a close relative of LWE. Instead of injecting random noise, LWR introduces distortion by rounding values to a smaller set of possibilities. Given a random vector a and a secret vector s, we compute $b = \lfloor \langle a, s\rangle \rfloor$, where the inner product is rounded to a restricted range. This rounding step plays the same role as the error term in LWE: it prevents exact recovery of the

secret and makes the problem computationally hard. LWR avoids the need to generate random error terms, simplifying implementations and improving efficiency. Because of this, LWR is often used in practical lattice-based cryptographic schemes, including Saber.

5 PQC Algorithms

Of the finalists NIST selected in the third round, the schemes rely on hard mathematical problems, in particular the shortest vector problem and learning with errors, both of which are resistant to known attacks from quantum computers [8]. A particularly efficient subclass is based on the Module-LWE problem, which generalizes LWE to lattices over polynomial rings. Two leading algorithms in this category are Kyber and Saber. Both are KEM (Section 4), KEM allows one party to encapsulate a random secret into a ciphertext using the recipient's public key, and the recipient can then decapsulate it with their private key to recover the same secret. This shared key can then be used with fast symmetric encryption for secure communication. Kyber uses the NTT 6 to accelerate polynomial multiplication, while Saber employs deterministic rounding techniques from module-learning with rounding (Section 4) to simplify implementation.

As seen in Fig 1, Beyond lattice-based cryptography, NIST has considered code-based schemes such as Classic McEliece [6], which rely on the difficulty of decoding random linear codes. This problem has remained hard for decades [6], even in the face of quantum advances. Although Classic McEliece uses large public keys, it offers extremely fast encryption and decryption, making it suitable for high-throughput applications. The third category, isogeny-based cryptography, leverages the structure of elliptic curves and the difficulty of computing mathematical maps between curves. Each category presents distinct trade-offs in terms of performance, key size, and security assumptions, contributing to a layered and resilient defense against future quantum threats [6]. Of the NIST finalists in Fig.1, this paper focuses on lattice-based schemes.

6 Lattice-Based Cryptography

Both algorithms that we will discuss—Kyber and Saber—are built on the Module-LWE problem, a structured extension of the foundational learning with errors framework 4. The strength of these schemes lies in their reliance on hard lattice problems, particularly the Shortest Vector Problem and LWE, which remain computationally intractable even for quantum adversaries.

Kyber uses fast polynomial multiplication via the NTT (Section 4), while Saber employs deterministic rounding techniques from Module-LWR to avoid sampling noise, simplifying implementation and enhancing side-channel resistance [7]. These innovations reflect a broader trend toward cryptographic primitives that are not only secure but also scalable and practical for real-world deployment. The unpredictability of future mathematical breakthroughs and technological advances makes it imperative to adopt schemes with deep structural hardness—precisely what lattice-based cryptography provides [2].

6.1 Kyber

Kyber is a post quantum key encapsulation mechanism built on the module-LWE problem (Section 4), a structured extension of the foundational LWE framework. Unlike deterministic schemes such as Module-LWR, Kyber introduces randomness through error sampling, which enhances cryptographic hardness by obscuring the secret with carefully controlled noise. The scheme operates over polynomial rings, encoding keys and ciphertexts as polynomials in the ring (Section 3) $\mathbb{Z}_q[X]/(X^n + 1)$ [6]. This notation means we are working with polynomials whose coefficients are integers taken modulo q, and where polynomials are considered equivalent if they differ by a multiple of $X^n + 1$. Shared secrets are derived through modular arithmetic and structured lattice operations, with noise sampled to ensure both security and reliable decryption [6].

Kyber's design emphasizes performance and scalability, particularly through its use of the NTT to accelerate polynomial multiplication. This optimization enables high-speed execution on general-purpose processors and embedded platforms alike. Kyber also supports constant-time implementations, which are designed to counter timing-based side-channel attacks [8]. Timing-based attacks exploit small differences in how long cryptographic operations take to run, while side-channel attacks more broadly target unintended information leaks including timing, but also power consumption, or electromagnetic signals to infer secret keys. By ensuring that operations take the same amount of time regardless of the input or secret key, Kyber prevents attackers from learning anything useful from these measurements. While lattice-based schemes often involve trade-offs such as larger key sizes or ciphertexts, Kyber achieves a strong balance between efficiency, security, and deployability [6].

Kyber is available in three security levels: Kyber512, Kyber768, and Kyber1024, each suited to different deployment scenarios. Kyber512, the most lightweight variant, is ideal for constrained environments such as mobile devices, smart home hubs, or wearable technology, where fast key exchange and low memory usage are critical. For example, a smartwatch initiating a secure Bluetooth pairing could use Kyber512 to establish a quantum-resistant session key without draining battery or processing resources. Kyber768 offers a middle-ground solution for enterprise-grade applications like VPNs, secure messaging platforms, or cloud-based authentication systems. It provides stronger security guarantees while maintaining efficient performance across diverse hardware [4]. Kyber1024, the highest-security variant, is

designed for long-term confidentiality in high-assurance environments such as government communications, financial infrastructure, or secure firmware distribution. In these contexts, Kyber1024 ensures that even if encrypted data is stored and later attacked by quantum adversaries, the underlying secrets remain protected [4].

Kyber's reliance on Module-LWE provides a well-analyzed foundation with strong theoretical guarantees. Its structured lattice design and efficient arithmetic make it a compelling choice for post-quantum encryption, especially in widely used security protocols such as TLS, SSH, and IPsec. Transport Layer Security (TLS) is the protocol that underpins secure web browsing, ensuring that data exchanged between a browser and a server remains confidential and authenticated. Secure Shell (SSH) is used to establish encrypted remote connections, allowing administrators and developers to safely access and manage servers over insecure networks. Internet Protocol Security (IPsec) provides encryption and authentication at the network layer, protecting data flows across virtual private networks (VPNs) and other IP-based communications. By integrating Kyber into these protocols, future systems can maintain the same trust and functionality they provide today while gaining resilience against quantum adversaries. As noted in Section 6, Kyber exemplifies the shift toward cryptographic primitives that are not only quantum-secure but also engineered for real-world deployment across heterogeneous platforms. Its layered security levels and implementation flexibility position it as a future-proof encryption standard.

6.2 Saber

Saber is a post-quantum key encapsulation mechanism grounded in the Module-LWR problem. Unlike traditional LWE-based schemes that rely on the injection of random noise to obscure the secret, Saber replaces this with a rounding operation that introduces controlled distortion. In practice, this means that when computing inner products between a public vector and a secret vector, the result is rounded to a smaller set of values rather than perturbed with random error [1]. This rounding step plays the same role as noise: it prevents exact recovery of the secret while remaining predictable and efficient to implement. The process begins with one party generating a public key consisting of polynomials with coefficients modulo a fixed integer q. The other party uses this public key to encapsulate a random public key by performing modular arithmetic operations in the polynomial ring and applying the rounding function. The encapsulated ciphertext is then sent back, and the original party can decapsulate it using their private key, recovering the same shared secret. Because the rounding operation is deterministic, Saber avoids the need for complex error sampling routines, thereby reducing computational overhead and improving consistency across platforms. The scheme operates over polynomial rings, where keys and ciphertexts are represented as collections of polynomials with coefficients taken modulo q. Shared secrets are derived through these modular arithmetic operations, which preserve the algebraic structure while ensuring cryptographic hardness. This design makes Saber particularly efficient for constrained environments while still maintaining strong post-quantum security guarantees.

One of Saber's notable strengths is its efficiency in constrained environments, such as embedded systems or low-power devices [1]. By avoiding probabilistic noise sampling, Saber benefits from predictable behavior and reduced memory requirements, making it well-suited for real-world deployment. However, this deterministic design also precludes the use of NTT, which limits certain performance optimizations available to schemes like Kyber. Despite this trade-off, Saber achieves competitive speed and compact key sizes, striking a balance between simplicity and security [6].

Saber is offered in three security levels: LightSaber, Saber, and FireSaber, each tailored to different deployment needs. LightSaber, the most lightweight variant, is ideal for ultra-constrained environments such as smart cards, RFID tags, or battery-powered Internet of Things (IoT) sensors [1]. For example, a smart agriculture sensor using LightSaber could securely transmit soil moisture data to a central hub with minimal energy consumption and memory overhead. Saber, the middle-tier variant, provides a balanced trade-off between security and performance, making it suitable for general-purpose embedded systems like industrial controllers or automotive Electronic Control Units [6]. These devices often require robust encryption without sacrificing responsiveness or resource efficiency. FireSaber, the highest-security variant, is designed for applications demanding maximum post quantum resilience, such as firmware updates in critical infrastructure or secure boot processes in defense-grade hardware. In such contexts, FireSaber can ensure that even highly sensitive cryptographic exchanges remain secure against quantum-capable adversaries [6].

From a security standpoint, Module-LWR is considered a strong foundation, though it has undergone less extensive cryptanalysis than LWE, leaving a slightly narrower margin of confidence in long-term resilience. Nevertheless, Saber's design reflects a deliberate prioritization of implementation practicality without compromising theoretical robustness. Saber exemplifies the broader trend in post quantum cryptography toward schemes that are not only resistant to quantum attacks but also optimized for deployment across diverse hardware and software environments. Saber's potential to serve as a future post quantum encryption standards, particularly in applications where deterministic behavior and lightweight performance are critical.

7 Conclusion

As quantum computing advances toward practical implementation, the urgency to develop cryptographic systems that can withstand quantum attacks becomes paramount. This paper has explored the foundations and motivations behind post quantum cryptography, with a particular emphasis on lattice-based algorithms, currently the most promising and widely studied class of quantum-resistant schemes [6]. Classical cryptographic systems such as RSA and ECC rely on number-theoretic problems that are vulnerable to quantum algorithms like Shor's. In contrast, lattice-based cryptography leverages the hardness of problems like the SVP and LWE, which remain intractable even for quantum computers.

Kyber and Saber, two NIST-selected lattice-based finalists, exemplify the practical and theoretical strengths of Module-LWE and Module-LWR frameworks. Kyber's use of the Number Theoretic Transform enables efficient polynomial operations, making it highly performant across a range of platforms—from embedded systems to cloud infrastructure. Saber's deterministic rounding avoids sampling noise and simplifies implementation, offering predictable behavior and reduced memory usage in constrained environments. Both schemes support multiple security levels, allowing developers to tailor cryptographic strength to specific use cases—from lightweight IoT sensors to high-assurance government communications.

Looking ahead, the adoption of PQC will require not only technical innovation but also coordinated efforts across industry, academia, and government. Migration strategies must account for legacy systems, interoperability challenges, and the need for hybrid cryptographic models during the transition period. Continued cryptanalysis will be essential to validate the long-term resilience of these schemes, especially as quantum hardware evolves and new attack vectors emerge. Additionally, hardware acceleration, side-channel resistance, and formal verification will play critical roles in ensuring robust and trustworthy implementations.

Ultimately, the transition to PQC is not merely a technical upgrade; it is a strategic imperative for safeguarding digital infrastructure in the quantum era. By grounding security in deep mathematical hardness rather than computational assumptions, lattice-based cryptography provides a resilient foundation for future-proof encryption. Kyber and Saber are not just theoretical constructs; they are practical algorithms, offering scalable, secure, and efficient solutions for the post quantum world. Their success will depend on continued research, thoughtful deployment, and a collective commitment to building cryptographic systems that can endure the challenges of tomorrow.

8 Acknowledgments

I would like to thank my advisor, KK Lamberty and Elena Machkasova for all the help with the research and development of this paper.

References

[1] Yajing Chang, Yingjian Yan, Chunsheng Zhu, and Yanjiang Liu. 2023. A High-performance Masking Design Approach for Saber against High-order Side-channel Attack. *ACM Trans. Des. Autom. Electron. Syst.* 28, 6, Article 91 (Oct. 2023), 19 pages. doi:10.1145/3611670

[2] Yihang Cheng, Yansong Feng, and Yanbin Pan. 2024. Embedding Integer Lattices as Ideals into Polynomial Rings. In *Proceedings of the 2024 International Symposium on Symbolic and Algebraic Computation* (Raleigh, NC, USA) *(ISSAC '24)*. Association for Computing Machinery, New York, NY, USA, 170–179. doi:10.1145/3666000.3669688

[3] Pierre-Richard Dahoo, Philippe Pougnet, and Abdelkhalak El Hami. 2021. *Quantum Optics and Quantum Computers*. 135–184. doi:10.1002/9781119818984.ch3

[4] Giovanni Di Crescenzo, Matluba Khodjaeva, Dilan D. Morales Caro, and Delaram Kahrobaei. 2024. Single-Server Delegation of NTT with Application to Crystals-Kyber. In *Proceedings of the 2024 on Cloud Computing Security Workshop* (Salt Lake City, UT, USA) *(CCSW '24)*. Association for Computing Machinery, New York, NY, USA, 29–42. doi:10.1145/3689938.3694777

[5] Rares Ifrim, Dumitrel Loghin, and Decebal Popescu. 2024. A Systematic Review of Fast, Scalable, and Efficient Hardware Implementations of Elliptic Curve Cryptography for Blockchain. *ACM Trans. Reconfigurable Technol. Syst.* 17, 4, Article 62 (Nov. 2024), 33 pages. doi:10.1145/3696422

[6] Manoj Kumar and Pratap Pattnaik. 2020. Post Quantum Cryptography(PQC) - An overview: (Invited Paper). In *2020 IEEE High Performance Extreme Computing Conference (HPEC)*. 1–9. doi:10.1109/HPEC43674.2020.9286147

[7] Suparna Kundu. 2025. Towards Solving Real-world Problems of Post-quantum Cryptography. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security* (Taipei, Taiwan) *(CCS '25)*. Association for Computing Machinery, New York, NY, USA, 4878–4880. doi:10.1145/3719027.3765575

[8] Catinca Mujdei, Lennert Wouters, Angshuman Karmakar, Arthur Beckers, Jose Maria Bermudo Mera, and Ingrid Verbauwhede. 2024. Side-channel Analysis of Lattice-based Post-quantum Cryptography: Exploiting Polynomial Multiplication. *ACM Trans. Embed. Comput. Syst.* 23, 2, Article 27 (March 2024), 23 pages. doi:10.1145/3569420

[9] Richard C. Penney. 2008. *Linear Algebra: Ideas and Applications* (3 ed.). Wiley-Interscience, USA.

[10] Peter W. Shor. 1999. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Rev.* 41, 2 (1999), 303–332. arXiv:https://doi.org/10.1137/S0036144598347011 doi:10.1137/S0036144598347011

[11] Pankaj Singh, Ashutosh Srivastava, Divya Srivastava, Shivam, Madhushi Verma, and Vaishnavi Srivastava. 2025. An Overview of Quantum Cryptography Evolution From Classical Cryptography. In *2025 International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI)*. 1–6. doi:10.1109/IC3ECSBHI63591.2025.10990566

[12] Dewang Sun, Naifeng Zhang, and Franz Franchetti. 2023. Optimization and Performance Analysis of Shor's Algorithm in Qiskit. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*. 1–7. doi:10.1109/HPEC58863.2023.10363522