International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Simulating Shor's Algorithm with QuantumRings and the Emerging Threats to RSA Encryption in the Quantum Era

Shafeeq Rahman Thottoli[1], Musfar Muhamed Kozhikkal[2]

[1]Department of Physical Sciences, Physics division, College of Science, Jazan University, Jazan 45142, Kingdom of Saudi Arabia
Email: *sthottoli[at]jazanu.edu.sa*

[2]Institut de Ciències del Cosmos, Universitat de Barcelona, (ICCUB), Spain
Email: *musfarkmohd[at]gmail.com*

Abstract: ***Quantum computing has raised significant concerns regarding the security of classical cryptographic systems, particularly RSA, as it depends on the computational difficulty of factoring large semiprime numbers. In this study, we demonstrate the practical application of Shor's algorithm by successfully factorizing a semiprime integer, up to 30 bits (857830637 = 29167 X 29411) using the QuantumRings simulator. Our work highlights the effectiveness of Shor's algorithm in solving the factorization problem in polynomial time, which will significantly improve over the classical method for a large number. The results suggest that with advances in quantum hardware, such as increasing qubit counts and improving error correction, breaking larger RSA keys (e.g., RSA-2048) may soon become feasible. This seriously threatens current cryptographic systems, emphasizing the need to adopt post-quantum cryptography. Our findings aim to raise awareness among researchers, policymakers, and industry leaders of the importance of preparing for a quantum-safe future.***

Keywords: Quantum computing, Shor's algorithm, RSA encryption, post-quantum cryptography, QuantumRings

1. Introduction

The rapid advancements in quantum computing have led to concerns about the security of modern cryptographic systems, particularly RSA encryption. As a fundamental component of secure communication, RSA depends on the computational difficulty of factoring large semiprime numbers, a challenge that classical computers find hard to solve. Traditional methods such as the number field sieve (NFS) require sub-exponential time for factorization, thereby protecting RSA against classical attacks [1]. However, quantum computing brings about a paradigm shift. In 1994, Peter Shor proposed a quantum algorithm [2] capable of factorizing semiprimes in polynomial time, which presents a significant threat to RSA encryption. Shor's algorithm is a central topic in numerous studies exploring its implementation on Quantum Processing Units (QPUs). Most of these studies suggest an ad hoc quantum circuit for N=15 [3]-[8], implemented across various technologies like photonic systems and superconducting qubits. A compiled version of Shor's algorithm is described in [9]. Although large-scale quantum computers are not yet available, tools like QuantumRings allow researchers to simulate and validate algorithms, bridging the gap between theoretical advances and practical implementation.

In this work, we focus on simulating Shor's algorithm using the QuantumRings simulator [10][11] to factorize semiprimes ranging from 10-bit to 30-bit sizes. We aim to demonstrate the feasibility of quantum factorization and highlight its implications for breaking real-world cryptographic systems like RSA-2048. QuantumRings is an advanced simulator designed to replicate the behavior of quantum hardware. It provides a controlled environment for testing quantum algorithms while avoiding real-world challenges such as decoherence and error accumulation. This study aims to bridge the gap between the theoretical possibilities of quantum computing and its practical implications for cryptanalysis, helping scientists and policy makers recognize and prepare for a future where quantum technology could disrupt existing security systems.

The paper successfully factors a 30-bit semiprime using Shor's algorithm on simulated hardware illustrates the potential of quantum technology to compromise much larger RSA keys as it advances. We quantify the resources required to factorize cryptographically relevant semiprimes, providing valuable insight into when RSA may become vulnerable. This underscores the urgent need to adopt post-quantum cryptographic standards such as lattice-based algorithms before quantum hardware matures.

The remainder of this paper is structured as follows. Section 2 covers Shor's algorithm and its mathematical principles. Section 3 explains our simulation approach using QuantumRings. Section 4 presents our findings and explores the implications for cryptography and the steps needed toward quantum-safe solutions.

2. Theoretical Framework of Shor's Algorithm

Shor's algorithm uses the principles of quantum mechanics to solve the integer factorization problem efficiently. It reduces integer factorization to a period-finding problem using modular arithmetic. In essence, the algorithm exploits the periodicity of functions of the form $f(x) = a^x \bmod N$ to deduce the factors of N.

In contrast to classical factorization algorithms like the General Number Field Sieve (GNFS) [12], which have sub-exponential time complexity for large N, Shor's algorithm solves the problem in polynomial time. The shift from sub-

exponential to polynomial complexity represents a radical improvement in computational efficiency and fundamentally challenges the security assumptions underlying RSA.

This theoretical framework explains the quantum advantage and lays the groundwork for understanding practical implementation challenges. Factors such as noise, decoherence, and the need for error correction in real quantum systems must be addressed. Although this demonstration was conducted on a simulator, the principles remain valid for quantum hardware. Given the trajectory of quantum technology development, these theoretical constructs are poised to translate into real-world cryptanalytic capabilities in the near future.

The mathematical backbone of Shor's algorithm is rooted in number theory and quantum mechanics. The key steps are as follows:

2.1 Classical Reduction to Period-Finding

Factorizing a semiprime N involves finding a nontrivial divisor, which is one of its two prime factors. Shor's algorithm begins classically by selecting a random integer a ($1 < a < N$) such that gcd(a, N) =1. The goal is to find the smallest positive integer r, known as the period, such that:

$$a^r \equiv 1 \, mod \, N$$

Once the period r is determined, the potential factors of N are given by gcd $(a^{\frac{r}{2}} \pm 1, N)$, provided that r is even and $a^{\frac{r}{2}} \neq -1 \, mod \, N$. If these conditions are not met, a different value of a is selected [2].

2.2 Quantum Period-Finding

The quantum circuit (Figure1) uses two registers:

- **Register 1 (top)**: Stores x in superposition for $x = 0,1, \ldots, 2Q-1$, where Q is a power of 2 close to N^2.
- **Register 2 (bottom)**: Stores $f(x) = a^x \, mod \, N$.

After initializing Register 1 in an equal superposition $\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle$), the function $f(x)$ is computed in Register 2. Measuring Register 2 collapses Register 1 into a superposition of states separated by the period r.

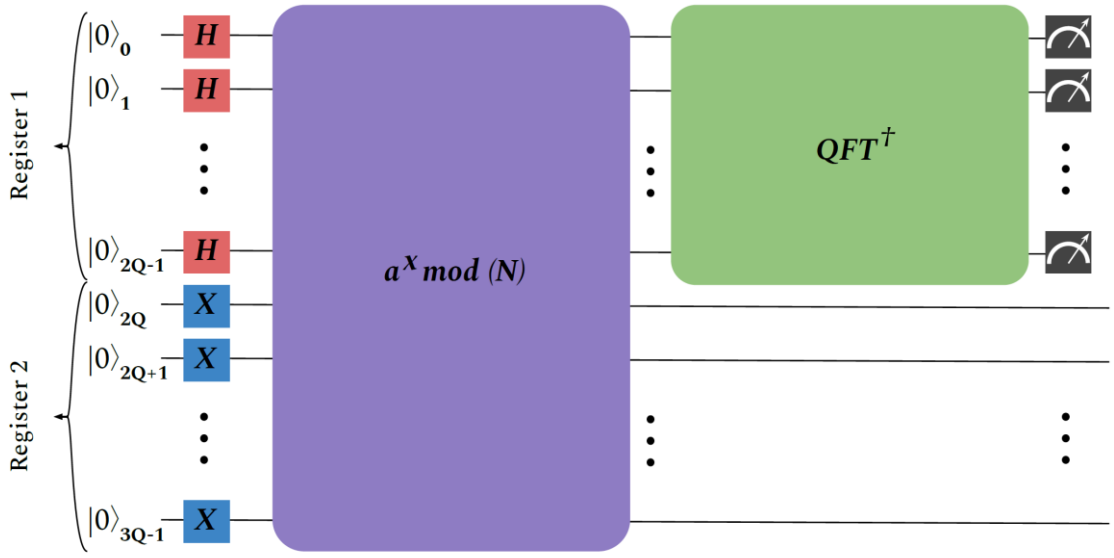

Figure 1: Quantum circuit for Shors algorithm. Register 1 undergoes QFT, while Register 2 computes $a^x \, mod \, N$. Controlled modular multipliers (blue) dominate the circuit.

2.3 Quantum Fourier Transform (QFT)

A central element in the algorithm is the QFT, which maps the periodic state to a superposition of frequencies, allowing measurement to extract the period r. The efficiency of the QFT is a key factor in enabling the polynomial-time complexity of Shor's algorithm.

The QFT maps the periodic state in Register 1 to the frequency domain, where the period r can be extracted. It transforms a state $|x\rangle$into:

$$QFT|x\rangle = \frac{1}{\sqrt{Q}} \sum_{k=0}^{2Q-1} e^{2\pi i x k/Q} |k\rangle$$

Applying QFT to Register 1 produces peaks at multiples of Q/r, where r is the period. The measurement of the output yields $y \approx \frac{\lambda Q}{r}$, where λ is some integer. Using continued fractions, the period r can be derived from the value y/Q [13].

2.4 Entanglement and Superposition

The algorithm exploits the fundamental quantum phenomena of superposition and entanglement, allowing for the parallel computation of many values and ensuring that the periodic information is preserved during measurement. The quantum system simultaneously computes many values of $f(x)$ by preparing qubits in a superposition of all possible inputs x. The entanglement between qubits ensures the measurement

result in a collapsed state in a manner that preserves the periodic information.

3. Methodology: Simulating Shor's Algorithm

This section describes the implementation of Shor's algorithm in the QuantumRings simulator, including semiprime selection, quantum circuit design, and optimizations for handling large integers. Our experimental setup was based on the QuantumRings simulator [10][11], which is designed to emulate the behaviour of quantum hardware with high fidelity. The simulation environment allowed us to bypass some classical limitations while still adhering to the principles of quantum mechanics.

3.1 Semiprime Selection and Validation

For our experiment, a 30-bit semiprime number was chosen. The selection criteria included the following.

- **Uniform Bit Distribution:** Ensuring that both prime factors have approximately equal bit lengths to maximize computational challenge.
- **Difficulty Level:** The chosen semiprime represents a non-trivial case, but manageable within the simulator's capabilities.

This semiprime can be represented as $N = p \times q$, where p and q are prime numbers of approximately 15 bits each. Semiprimes were generated for bit lengths ranging from 10 to 30 using the GNU Multiple Precision Arithmetic Library (GMP) [14]. The primes p and q were selected to satisfy:

$$|p - q| > 2^{\frac{bit-length}{2}-5}.$$

To ensure resistance against Fermat's factorization method. Each N was validated using the Miller-Rabin primality test [15].

3.2 Quantum Circuit Design

A quantum circuit was designed and implemented to perform the modular exponentiation necessary for the period-finding subroutine. The design focused on optimizing the number of qubits and gates to enhance computational efficiency. The quantum circuit for Shor's algorithm (Figure 1) was implemented in three stages:

1) **Modular Exponentiation:** A sequence of controlled $U_{a^{2^k}}$ gates computes $f(x) = a^x \bmod N$.
2) **Quantum Fourier Transform (QFT):** The QFT was applied to the superposition state produced by the circuit. This step was essential for identifying the periodicity of the modular exponentiation function.
3) **Measurement and post-processing:** The x-register was measured to obtain y, and the period r was extracted using the continued fraction algorithm.

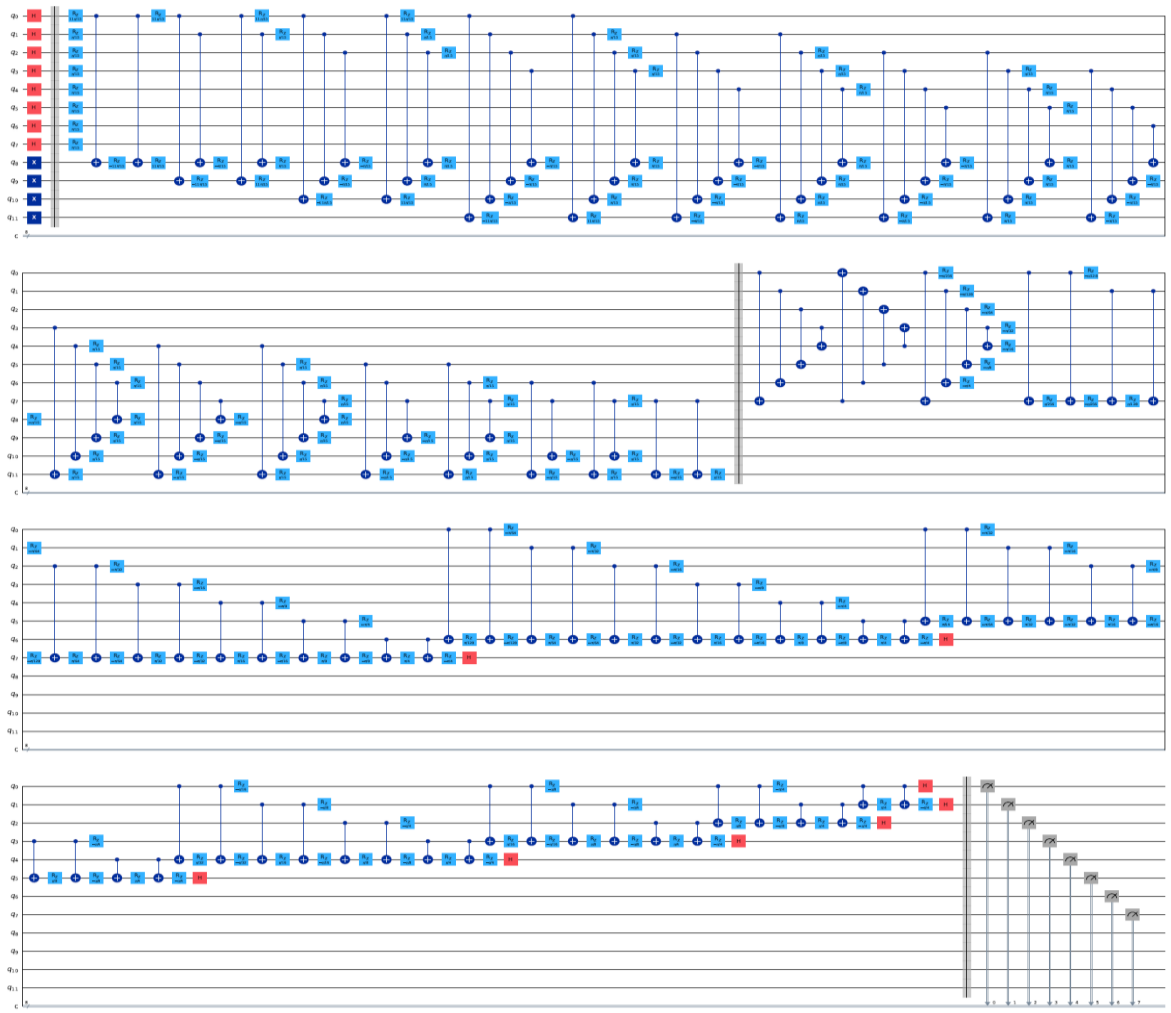

Figure 2: Quantum circuit for factorizing a 4-bit semiprime number 15 with base 11.

An example of the Quantum circuit for factorizing a 4-bit semiprime number 15 with base 11 is given in Figure 2. Optimization techniques were employed to reduce circuit depth and minimize resource usage. The simulator provided detailed reports on the usage of the qubits.

4. Results and Analysis

Detailed logs from QuantumRings indicated efficient use of qubits are given in the Table 1. These results indicate that Shor's algorithm scales well within the simulated environment.

Table 1: Quantum factorization results and resource usage for factorizing semiprimes up to 30 bits

Bit Length	Semiprime (N)	Base (a)	Factor 1	Factor 2	Number of Qubits
8	143	60	11	13	25
10	899	428	31	29	31
12	3127	251	59	53	37
14	11009	1735	109	101	43
16	47053	261	223	211	49
18	167659	78957	431	389	55
20	744647	12371	907	821	61
22	3036893	1285503	1709	1777	67
24	11426971	1873261	3191	3581	73
26	58949987	1194295	8039	7333	79
28	208241207	100906083	15727	13241	85
30	857830637	317637464	29167	29411	91

We successfully factorized semiprime integers ranging from 10-bit to 30-bit using Shor's algorithm on the QuantumRings simulator. Our results indicate that the algorithm, when implemented in a high-fidelity simulator, can reliably factorize numbers that are intractable for classical methods within polynomial-time complexity.

4.1 Limitations and Challenges

Despite the success of the simulation, several challenges remain:

- **Scaling to Larger Qubits:** Achieving practical implementation on hardware with more than 1,000 qubits remains a major challenge due to current limitations in qubit coherence and interconnectivity.
- **Error Rates in Physical Systems:** The simulated environment can only approximate idealized error correction, whereas real quantum devices require substantial advances in error mitigation.

These challenges underscore that, while the simulation shows promise, transitioning from simulation to real quantum hardware will require overcoming significant technical obstacles.

Extrapolating from our results, factorizing RSA-2048 would require 6,145 logical qubits and tens of millions of quantum gates.

4.2 Implications for RSA and Post-Quantum Cryptography

Our results demonstrate that Shor's algorithm, when scaled to a sufficient number of qubits, can efficiently break RSA encryption. Factorizing a 30-bit semiprime on the QuantumRings simulator required 91 qubits. Extrapolating this to RSA-2048 (a 2048-bit modulus), we estimate that approximately 6,145 logical qubits and tens of millions of quantum gates would be required. Assuming a 10% annual improvement in qubit quality [16] since current quantum hardware is constrained by decoherence and noise, rapid advancements in error correction (e.g., surface codes [17]), breaking RSA-2048 could become computationally feasible within the next 5–10 years.

The successful demonstration of Shor's algorithm on a 30-bit semiprime highlights the potential risks to RSA-based cryptography. Our findings suggest that further advancements in quantum computing could eventually make breaking RSA-2048 feasible. Such a breakthrough would have serious implications for financial systems, government communications, and data privacy.

Our simulation shows that factoring larger keys will require not just more qubits but also major improvements in error correction and qubit accuracy. Projections indicate that with advancements in quantum hardware over the next few years, quantum computers could reach the capability needed to break RSA-2048. This highlights the urgent need for research and development in post-quantum cryptography.

Efforts to standardize quantum-resistant algorithms are well underway, primarily led by the National Institute of Standards and Technology (NIST) in the United States. As part of its Post-Quantum Cryptography (PQC) standardization project, NIST has evaluated numerous candidates and has selected several algorithms for standardization. Notably, lattice-based cryptography has emerged as a leading approach due to its strong security foundations and efficient implementation potential. Algorithms such as CRYSTALS-Kyber (for key encapsulation) and CRYSTALS-Dilithium (for digital signatures) were formally selected by NIST in 2022 for standardization [18], [19]. Other approaches, including hash-based, multivariate, and code-based schemes, are also being considered as viable alternatives to classical cryptographic systems like RSA and elliptic-curve cryptography (ECC), which are vulnerable to attacks from quantum computers.

Despite these advancements, the global adoption of PQC remains sluggish. This lag is primarily due to challenges such as compatibility with legacy systems, lack of standardization across industries, limited awareness and technical expertise among small and medium-sized enterprises (SMEs), and underinvestment in PQC research and infrastructure,

particularly in developing countries [20][21]. Additionally, there is uncertainty around how soon large-scale quantum computers capable of breaking current cryptographic standards will become operational, which can lead to complacency in the transition process.

Governments and industries must prioritize the transition to post-quantum cryptography (PQC), invest in quantum-safe infrastructure, and establish strict compliance deadlines. Delaying this shift could result in severe security breaches as quantum technology continues to advance.

5. Conclusion and Call to Action

In this study, we successfully factorized semiprimes up to 30 bits using Shor's algorithm on the QuantumRings simulator, demonstrating its feasibility with near-term quantum resources. Our results underscore the significant computational advantage that quantum algorithms have over classical methods, posing a growing threat to current cryptographic standards, particularly RSA.

The implications of this work are profound. As quantum computing technology matures, the likelihood of breaching RSA-2048 and similar cryptographic schemes becomes increasingly plausible. Therefore, researchers, industry leaders, and policymakers must expedite their efforts to develop and implement post-quantum cryptographic algorithms.

Future work should focus on scaling quantum simulations to larger qubit counts, improving error correction techniques, and bridging the gap between theoretical algorithms and practical, deployable quantum systems. We urge the scientific community to heed these warnings and prepare for a quantum-safe future.

Accelerating the shift to PQC is not just a technical imperative but also a policy and educational one. Governments, academic institutions, and industry leaders must collaborate to increase funding for PQC research, create tools for easier migration, and raise awareness about the quantum threat. The proactive adoption of PQC is the only defence against this existential risk.

References

[1] C. Pomerance, "The quadratic sieve factoring algorithm. Advances in Cryptology," In: Beth, T., Cot, N., Ingemarsson, I. (eds) Advances in Cryptology. pp. EUROCRYPT 1984. Lecture Notes in Computer Science, vol 209. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-39757-4_17

[2] P.W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124-134, 1994.

[3] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," Nature, vol. 414, pp. 883–887, 2001. arXiv: quant-h/0112176.

[4] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White, "Experimental Demonstration of a Compiled Version of Shor's Algorithm with Quantum Entanglement," Physical Review Letters, vol. 99, pp. 250505, 2007.

[5] C.- A. Politi, J. C. F. Matthews, and J. L. O'Brien, "Shor's Quantum Factoring Algorithm on a Photonic Chip," Science, vol. 325, pp. 1221–1221, 2009.

[6] E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O'Malley, D. Sank, A. Vainsencher, J. Wenner, T. White, Y. Yin, A. N. Cleland, and J. M. Martinis, "Computing prime factors with a Josephson phase qubit quantum processor," Nature Physics, vol. 8, pp. 719–723, Oct. 2012. arXiv: 1202.5707.

[7] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, "Realization of a scalable Shor algorithm," Science, vol. 351, pp. 1068–1070, Mar. 2016.

[8] Section: Brevia.Y. Lu, D. E. Browne, T. Yang, and J.-W. Pan, "Demonstration of a Compiled Version of Shor's Quantum Factoring Algorithm Using Photonic Qubits," Physical Review Letters, vol. 99, p.250504, 2007.

[9] J. A. Smolin, G. Smith, and A. Vargo, "Pretending to factor large numbers on a quantum computer," Nature, vol. 499, pp. 163–165, July 2013. arXiv: 1301.7007.

[10] "Welcome to Quantum Rings SDK Documentation! — Quantum Rings SDK." 2024. Quantumrings.com. 2024. https://portal.quantumrings.com/doc/.

[11] V. Kasirajan, T.Battelle, and B.Wold. "Empowering Large Scale Quantum Circuit Development: Effective Simulation of Sycamore Circuits." arXiv preprint arXiv:2411.12131 (2024).

[12] T. Kleinjung, et al., "Factorization of a 768-Bit RSA Modulus," Advances in Cryptology – CRYPTO 2010, 2010, pp. 333-350.

[13] Nielsen, M.A. and Chuang, I.L. (2010) Quantum Computation and Quantum Information, Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511976667

[14] Granlund, T. (2023). GNU Multiple Precision Arithmetic Library. https://gmplib.org/

[15] G.L. Miller, "Riemann's hypothesis and tests for primality", Journal of Computer and System Sciences, 13(3), pp. 300–317, 1976.

[16] IBM Quantum Roadmap. Quantum Computing. https://www.ibm.com/roadmaps/quantum/

[17] A. G. Fowler et al., "Surface codes: Towards practical large-scale quantum computation", Physical Review A, 86 (3), pp. 032324, 2012.

[18] National Institute of Standards and Technology (NIST). Post-Quantum Cryptography Standardization, https://csrc.nist.gov/projects/post-quantum-cryptography

[19] Alagic, G., et al. (2022). Status Report on the Third Round of the NIST PQC Standardization Process. NIST.

[20] European Union Agency for Cybersecurity (ENISA). (2021). *Post-Quantum Cryptography: Current State and Quantum Mitigation Strategies*.

[21] Bernstein, D. J., et al. (2017). *Post-Quantum Cryptography: Roadmap and Challenges*. Communications of the ACM.