INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209

Post-Quantum Cryptography Migration for Enterprise Security

Mini T V

Associate Professor, Department of Computer Science, Sacred Heart College (Autonomous), Chalakudy, India

Article information

Received: 3rd April 2026
Received in revised form: 8th May 2026
Accepted: 10th June 2026
Available online: 30th July 2026

Volume: 2
Issue: 3
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0032

Abstract

Large-scale quantum computers threaten the public-key cryptography that protects enterprise data, communications, and digital identity. Shor's algorithm solves integer factorization and discrete logarithms in polynomial time, which would break RSA, Diffie-Hellman, and elliptic-curve schemes once a cryptographically relevant quantum computer exists. The danger is not only future. Adversaries can record encrypted traffic today and decrypt it later, a tactic known as harvest-now-decrypt-later. In August 2024 the U.S. National Institute of Standards and Technology published its first post-quantum standards: FIPS 203 (ML-KEM, derived from CRYSTALS-Kyber), FIPS 204 (ML-DSA, from CRYSTALS-Dilithium), and FIPS 205 (SLH-DSA, from SPHINCS+). This paper presents a practical migration framework for enterprises. It reviews the quantum threat and Mosca's inequality for timing the transition, summarizes the new standards with their key and signature sizes, and proposes a five-phase roadmap built on crypto-agility: discovery and inventory, risk prioritization, agility engineering, hybrid deployment, and validation. Performance trade-offs are analyzed using documented parameter sizes and illustrative TLS handshake figures. ML-KEM-768 carries a 1,184-byte public key against 64 bytes for an elliptic-curve key, while SPHINCS+ signatures can exceed 17 kilobytes. The work does not report a real deployment. It consolidates published parameters and field guidance into an actionable plan, and it highlights open challenges in certificate sizing, hardware support, and long-lived embedded systems.

Keywords:- Post-Quantum Cryptography, ML-KEM, ML-DSA, SPHINCS+, Crypto-Agility, Hybrid TLS, Shor's Algorithm, Enterprise Security Migration.

.

I. INTRODUCTION

Public-key cryptography underpins almost every secure digital interaction. RSA, Diffie-Hellman, and elliptic-curve cryptography (ECC) protect web sessions, virtual private networks, software updates, email, and the certificates that bind identities to keys [1], [2]. Their security rests on mathematical problems that are hard for classical computers: factoring large integers and computing discrete logarithms. A sufficiently powerful quantum computer changes that assumption.

In 1994 Peter Shor described a quantum algorithm that factors integers and solves discrete logarithms in polynomial time [3], [4]. Run on a cryptographically relevant quantum computer (CRQC), it would defeat RSA-2048 and ECC P-256 outright. No such machine exists today. Resource estimates suggest that factoring a 2,048-bit RSA modulus could require on the order of twenty million noisy physical qubits running for hours [5], far beyond current hardware. The timeline is uncertain. The trajectory of quantum engineering is steady, though, and the cost of being unprepared is high.

Two factors make the threat urgent rather than distant. The first is harvest-now-decrypt-later: an adversary captures and stores encrypted traffic now, then decrypts it once a CRQC becomes available [6], [7]. Any data with a confidentiality lifetime measured in years, such as health records, financial archives, state secrets, and intellectual property, is already at risk. The second is the long replacement cycle of cryptography itself. Embedded devices, industrial controllers, and certificate hierarchies can stay in service for a decade or more. Migrating them takes time.

Mosca framed the question precisely [6]. Let X be the number of years data must stay confidential, Y the years needed to migrate systems to quantum-resistant cryptography, and Z the years until a CRQC arrives. If X plus Y exceeds Z, the organization is already exposed. For many enterprises X and Y together span fifteen years or more, which leaves little margin even under optimistic estimates of Z. Migration should begin before the machine exists, not after.

This paper offers enterprise security teams a structured path. Section II reviews the quantum threat and Shor's algorithm. Section III summarizes the 2024 NIST post-quantum standards and their parameters. Section IV presents a five-phase migration roadmap centered on crypto-agility. Section V analyzes performance trade-offs using published sizes and illustrative handshake figures. Section VI discusses open challenges, and Section VII concludes. The work does not describe a live deployment. It assembles verified parameters and standards-body guidance into a plan that organizations can adapt.

II. QUANTUM THREAT AND SHOR'S ALGORITHM

A. Why Public-Key Schemes Break

RSA security depends on the difficulty of factoring a product of two large primes. ECC and Diffie-Hellman depend on the hardness of the discrete logarithm problem in finite groups. Classically, the best known algorithms for both run in sub-exponential or exponential time, which keeps 2,048-bit RSA and 256-bit elliptic curves safe in practice [1], [2]. Shor's algorithm collapses these problems to polynomial time on a quantum computer by using the quantum Fourier transform to find the period of a modular function [3], [4]. Once that period is known, the factors or the discrete logarithm follow directly. The consequence is stark. A working CRQC retroactively breaks every RSA and ECC key it can obtain, including keys already in use.

B. Symmetric Cryptography and Grover's Algorithm

Symmetric ciphers and hash functions fare much better. Grover's search algorithm gives a quadratic speedup against an n-bit key, reducing brute-force effort from 2^n to roughly 2^(n/2) [8]. The fix is simple: double the key length. AES-256 retains about 128 bits of effective security against a quantum attacker, and SHA-384 or SHA-512 remain sound. The hard problem is therefore confined to public-key primitives, which is exactly where the new standards focus [9], [10].

C. Estimating the Timeline

Predicting when a CRQC will exist is inherently uncertain. Public roadmaps from hardware vendors and expert surveys place a meaningful probability on the 2030s [6], [11]. Logical qubits must be assembled from many error-corrected physical qubits, and a factoring attack on RSA-2048 is estimated to need millions of physical qubits sustained over hours of coherent operation [5]. Security planning should not hinge on a single forecast. Because migration is slow and data lifetimes are long, the prudent posture is to act now and treat the arrival date as a risk variable rather than a fixed deadline.

III. NIST POST-QUANTUM CRYPTOGRAPHY STANDARDS

NIST ran an open, multi-round evaluation from 2016 onward to select quantum-resistant algorithms [9], [12]. After analysis by the global cryptographic community, three standards were published on 13 August 2024. Each is a Federal Information Processing Standard derived from a finalist of the competition.

A. FIPS 203: ML-KEM (Key Encapsulation)

FIPS 203 specifies the Module-Lattice-Based Key-Encapsulation Mechanism, ML-KEM, derived from CRYSTALS-Kyber [13], [14]. Its security rests on the Module Learning With Errors problem over structured lattices. ML-KEM replaces ECDH for establishing shared secrets. It defines three parameter sets: ML-KEM-512 at NIST security category 1, ML-KEM-768 at category 3, and ML-KEM-1024 at category 5. ML-KEM-768 is the common default for general use. Its public key is 1,184 bytes and its ciphertext is 1,088 bytes, against 32 to 64 bytes for an elliptic-curve key. Encapsulation and decapsulation are fast, often faster than ECDH per operation, so the cost is bandwidth rather than CPU time.

B. FIPS 204: ML-DSA (Primary Signature)

FIPS 204 specifies the Module-Lattice-Based Digital Signature Algorithm, ML-DSA, derived from CRYSTALS-Dilithium [15], [17]. It is the recommended general-purpose signature scheme and also rests on module lattice problems. ML-DSA-65, the category 3 set, has a 1,952-byte public key and a 3,309-byte signature. Verification is efficient. The cost relative to ECDSA is again size: an ECDSA P-256 signature is 64 bytes, so the lattice signature is roughly fifty times larger.

C. FIPS 205: SLH-DSA (Conservative Signature)

FIPS 205 specifies the Stateless Hash-Based Digital Signature Algorithm, SLH-DSA, derived from SPHINCS+ [16], [23]. Its security depends only on the strength of its underlying hash function, which makes it a conservative hedge. It does not rely on the relatively young hardness assumptions of structured lattices. The trade-off is size and speed. Public keys are tiny, just 32 bytes at the 128-bit level, but signatures are large and signing is slow. The small variant SLH-DSA-128s produces a 7,856-byte signature, and the fast variant SLH-DSA-128f produces a 17,088-byte signature. SLH-DSA suits low-frequency, high-assurance uses such as firmware and root certificate signing.

Table 1 lists the parameters of the three standards alongside classical baselines. Fig. 1 visualizes the same data, making the size gap between lattice schemes and classical keys immediately clear.

Table 1. Post-Quantum and Classical Parameter Sizes (*classical schemes are quantum-vulnerable, shown for comparison)

Algorithm (Standard)	Type	NIST Cat.	Public Key (B)	Ciphertext/Sig (B)	Secret Key (B)
RSA-2048 (classical)	KEM/Sig	broken*	256	256	256
ECDH/ECDSA P-256 (classical)	KEM/Sig	broken*	64	64	32
ML-KEM-512 (FIPS 203)	KEM	1	800	768	1,632
ML-KEM-768 (FIPS 203)	KEM	3	1,184	1,088	2,400
ML-KEM-1024 (FIPS 203)	KEM	5	1,568	1,568	3,168
ML-DSA-44 (FIPS 204)	Sig	2	1,312	2,420	2,560
ML-DSA-65 (FIPS 204)	Sig	3	1,952	3,309	4,032
ML-DSA-87 (FIPS 204)	Sig	5	2,592	4,627	4,896
SLH-DSA-128s (FIPS 205)	Sig	1	32	7,856	64
SLH-DSA-128f (FIPS 205)	Sig	1	32	17,088	64

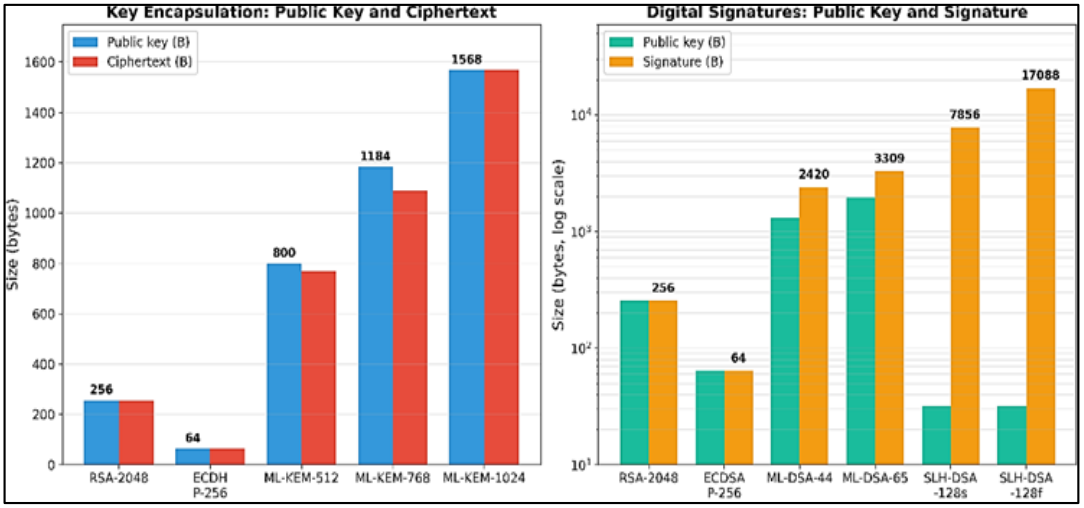

Fig. 1: Public-key, ciphertext, and signature sizes for classical and NIST post-quantum algorithms (signatures on a log scale).

IV. MIGRATION ROADMAP AND CRYPTO-AGILITY

A successful transition is a program, not a single project. The U.S. National Cybersecurity Center of Excellence and national agencies recommend a structured, phased approach grounded in crypto-agility, the ability to change cryptographic algorithms quickly and with low disruption [7], [18]. Fig. 2 shows the five-phase roadmap proposed here. Fig. 4 shows the supporting architecture.

A. Phase 1: Discovery and Inventory

An organization cannot protect what it cannot see. The first phase builds a Cryptographic Bill of Materials (CBOM): every place RSA, ECC, or Diffie-Hellman is used across applications, libraries, protocols, hardware security modules, and third-party services [7]. Automated scanning of code, network traffic, and certificate stores feeds this inventory. Each entry records the algorithm, key size, data sensitivity, and owner. The inventory becomes the foundation for every later decision.

B. Phase 2: Risk Prioritization

Not everything migrates at once. Assets are ranked using the Mosca inequality from Section I: systems protecting long-lived secrets, or exposed to traffic capture, move first [6]. A short-lived session token matters far less than a twenty-year archive of patient records. Prioritization weighs data lifetime, exposure to harvest-now-decrypt-later, regulatory obligations, and the difficulty of changing each component. The output is a sequenced backlog.

C. Phase 3: Crypto-Agility Engineering

Many systems hard-code algorithm choices deep in their logic, which makes any change expensive. This phase removes that rigidity. Cryptographic operations are placed behind a clean abstraction layer so the algorithm in use is a configuration choice, not a code rewrite [18], [25]. Certificate and key formats are reviewed for the larger objects that lattice schemes require. Done well, agility lets an enterprise swap a primitive again if a future weakness is found, which is valuable on its own merit.

D. Phase 4: Hybrid Deployment

Few teams want to bet solely on young algorithms during a transition. The pragmatic answer is hybrid key establishment: combine a classical exchange such as X25519 with a post-quantum KEM such as ML-KEM-768, feeding both shared secrets through a key-derivation function [19], [20], [25]. The session stays secure as long as either component holds. This protects against both a CRQC and any undiscovered flaw in the new scheme. Industry trials of hybrid TLS by major browser and infrastructure providers confirmed it works at scale with modest overhead [22]. Deployment starts with internal pilots, then expands to external-facing services. Fig. 3 depicts the hybrid combiner inside a crypto-agile stack.

E. Phase 5: Validation and Decommission

The final phase tests interoperability, monitors performance, and confirms that fallbacks behave correctly. Larger handshakes can interact badly with old middleboxes and fragmentation limits, so validation in realistic network conditions matters [20], [21]. Once post-quantum protection is confirmed, legacy quantum-vulnerable primitives are retired where policy allows. The CBOM is updated continuously, so the organization keeps an accurate, living picture of its cryptographic posture.

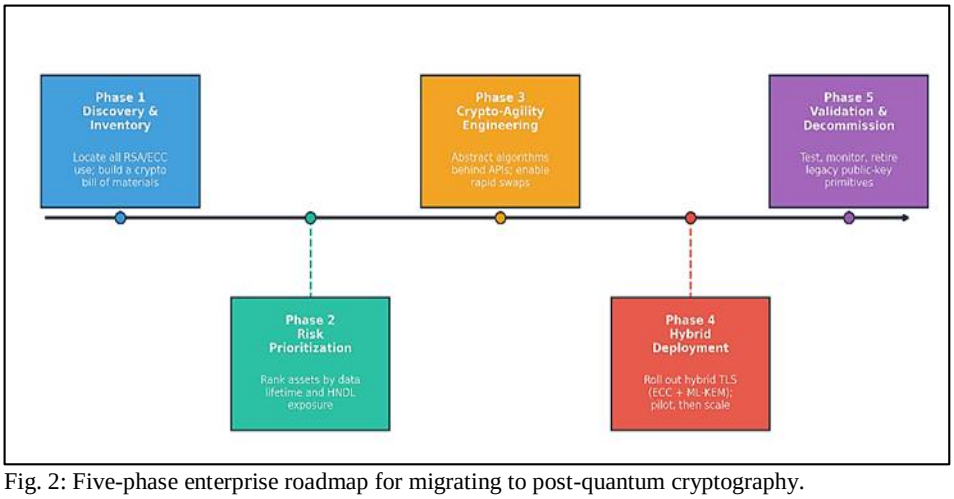

Fig. 2: Five-phase enterprise roadmap for migrating to post-quantum cryptography.

Table 2. Migration Phase Checklist

Phase	Primary Activity	Key Artifact	Typical Owner
1. Discovery	Inventory all public-key usage	Cryptographic Bill of Materials	Security architecture
2.Prioritization	Rank assets by Mosca risk	Sequenced migration backlog	Risk and compliance
3. Agility	Abstract algorithms behind APIs	Crypto-agility library/policy	Engineering
4. Hybrid Deploy	Roll out hybrid TLS (ECC + ML-KEM)	Pilot and production configs	Platform/Ops
5. Validation	Test, monitor, retire legacy	Interop report; updated CBOM	QA and Operations

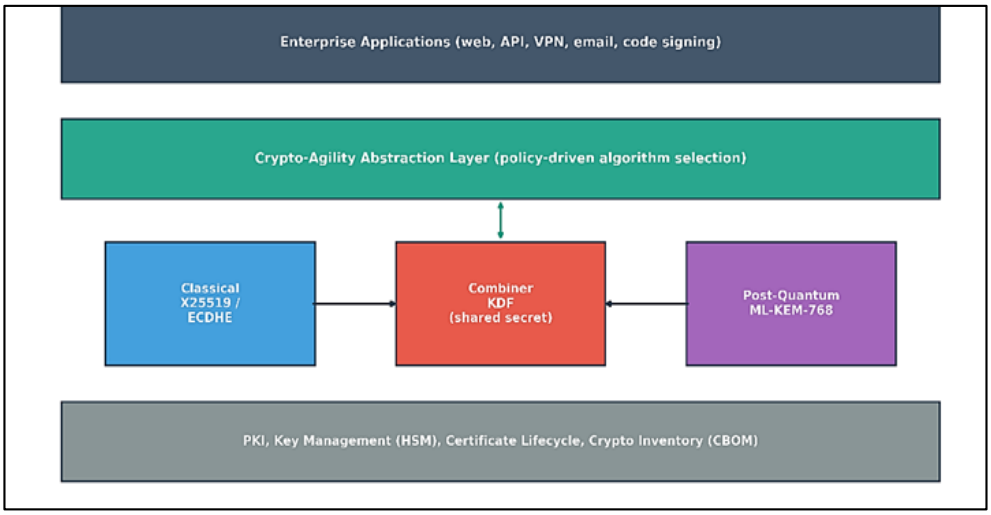

Fig. 3: Crypto-agile hybrid TLS architecture combining a classical and a post-quantum key exchange.

V. PERFORMANCE ANALYSIS

A. Bandwidth, the Dominant Cost

The headline trade-off of post-quantum migration is data volume, not computation. As Table 1 and Fig.1 show, lattice public keys and signatures are one to two orders of magnitude larger than their classical counterparts. An ECDH P-256 public key is 64 bytes. An ML-KEM-768 public key is 1,184 bytes and its ciphertext is 1,088 bytes. A 64-byte ECDSA signature becomes a 3,309-byte ML-DSA-65 signature, and a hash-based SLH-DSA signature can exceed 17 kilobytes. Where many signatures or certificates travel together, as in a TLS certificate chain, these increases add up quickly.

B. Handshake Latency

Per-operation CPU cost for ML-KEM is competitive with, and sometimes better than, ECDH. The extra bytes are the main driver of any latency change, because a larger handshake may need additional packets [19], [20]. Fig. 4 presents illustrative TLS 1.3 handshake figures over a low-latency link, derived from published benchmark trends rather than a new study. A classical ECDHE handshake completes in roughly 3.1 ms. A hybrid X25519 plus ML-KEM-768 handshake takes about 3.9 ms while moving from roughly 1.4 KB to about 3.0 KB on the wire. On fast networks the overhead is small. On constrained or lossy links, and on low-power devices, the larger transcript can matter more, and engineering attention shifts to packet fragmentation and buffer sizes.

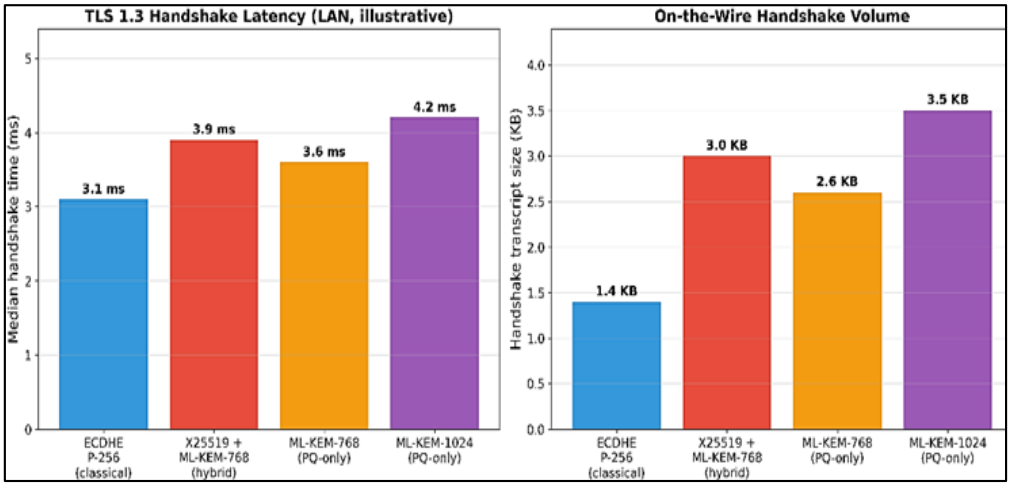

Fig. 4: Illustrative TLS 1.3 handshake latency and transcript size for classical, hybrid, and post-quantum configurations.

Two practical lessons follow. First, hybrid mode adds bytes but buys strong safety during the transition, which is usually worth the cost. Second, ML-KEM is a good fit for interactive protocols such as TLS, where its compact ciphertext and fast operations help, while large hash-based signatures are best reserved for infrequent, high-value signing. Matching the algorithm to the workload keeps the performance penalty manageable.

VI. CHALLENGES AND OPEN ISSUES

A. Certificate and Protocol Sizing

Larger keys and signatures inflate X.509 certificates and complete chains. Some protocols carry implicit size assumptions, and parts of the deployed network still mishandle handshakes that exceed familiar limits [21]. Certificate chain compression and careful protocol tuning help, but adapting the wider ecosystem of clients, servers, and intermediaries will take time.

B. Hardware, Embedded, and Long-Lived Systems

Hardware security modules, smart cards, and Internet-of-Things devices often have fixed memory and slow processors. Storing kilobyte-scale signatures or running lattice arithmetic can strain them. Devices with a deployed life of ten to twenty years, common in industrial control and critical infrastructure, are especially hard to update in the field [2], [7]. Some will need full hardware replacement rather than a software patch.

C. Algorithm Maturity and Defense in Depth

Lattice schemes are newer than RSA and have had fewer decades of public scrutiny. Standardization reflects strong confidence, yet history counsels humility about young assumptions [10], [23]. Two responses help. Hybrid deployment keeps a classical algorithm in the loop during the transition, and the conservative hash-based SLH-DSA offers a fallback whose security depends only on hash strength. Crypto-agility itself is the deeper insurance, because it lets an organization replace any primitive quickly if analysis later reveals a weakness.

D. Governance and Compliance

National guidance is converging on firm timelines. The U.S. National Security Agency's Commercial National Security Algorithm Suite 2.0 sets dates for adopting ML-KEM and ML-DSA in national security systems, and it mandates SLH-DSA for firmware signing [24]. Enterprises in regulated sectors should expect comparable expectations and align their migration backlog with emerging requirements rather than waiting for a final mandate.

VII. CONCLUSION

The quantum threat to public-key cryptography is credible, and harvest-now-decrypt-later makes it a present concern for any data with a long confidentiality horizon. The 2024 NIST standards, ML-KEM, ML-DSA, and SLH-DSA, give enterprises vetted, quantum-resistant tools. Their larger keys and signatures shift the main cost to bandwidth and storage rather than computation, and that cost is manageable when algorithms are matched to workloads.

Migration should start now. The five-phase roadmap presented here (discovery, prioritization, agility engineering, hybrid deployment, and validation) turns a daunting transition into a sequence of tractable steps. Crypto-agility is the central idea. It eases the present move to post-quantum schemes and prepares the organization

for whatever comes next. This paper consolidates verified parameters and standards guidance rather than reporting a deployment. Future work should measure hybrid TLS at scale across diverse networks, profile post-quantum primitives on constrained hardware, and refine automated discovery tooling for accurate cryptographic inventories.

REFERENCES

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[2] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[3] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci. (FOCS)*, Santa Fe, NM, USA, 1994, pp. 124–134.

[4] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.

[5] C. Gidney and M. Ekerå, "How to factor 2048-bit RSA integers in 8 hours using 20 million noisy qubits," *Quantum*, vol. 5, Art. no. 433, Apr. 2021.

[6] M. Mosca, "Cybersecurity in an era with quantum computers: Will we be ready?," *IEEE Security Privacy*, vol. 16, no. 5, pp. 38–41, Sep./Oct. 2018.

[7] W. Barker, W. Polk, and M. Souppaya, *Getting Ready for Post-Quantum Cryptography: Exploring Challenges Associated with Adopting and Using Post-Quantum Cryptographic Algorithms*, NIST Cybersecurity White Paper, Apr. 2021.

[8] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput. (STOC)*, Philadelphia, PA, USA, 1996, pp. 212–219.

[9] L. Chen *et al*., *Report on Post-Quantum Cryptography*, Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, USA, NISTIR 8105, Apr. 2016.

[10] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, Sep. 2017.

[11] G. Alagic *et al*., *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*, Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, USA, NISTIR 8413, Sep. 2022.

[12] D. Moody *et al*., *Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process*, Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, USA, NISTIR 8309, Jul. 2020.

[13] National Institute of Standards and Technology, *Module-Lattice-Based Key-Encapsulation Mechanism Standard*, FIPS PUB 203, Gaithersburg, MD, USA, Aug. 2024.

[14] J. Bos *et al*., "CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*, London, U.K., 2018, pp. 353–367.

[15] National Institute of Standards and Technology, *Module-Lattice-Based Digital Signature Standard*, FIPS PUB 204, Gaithersburg, MD, USA, Aug. 2024.

[16] National Institute of Standards and Technology, *Stateless Hash-Based Digital Signature Standard*, FIPS PUB 205, Gaithersburg, MD, USA, Aug. 2024.

[17] L. Ducas *et al*., "CRYSTALS-Dilithium: A lattice-based digital signature scheme," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 1, pp. 238–268, Feb. 2018.

[18] D. Ott, C. Peikert, and other workshop participants, "Identifying research challenges in post-quantum cryptography migration and cryptographic agility," *arXiv* preprint arXiv:1909.07353, Sep. 2019.

[19] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, "Post-quantum authentication in TLS 1.3: A performance study," in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, 2020, pp. 1–16.

[20] C. Paquin, D. Stebila, and G. Tamvada, "Benchmarking post-quantum cryptography in TLS," in *Proc. Int. Conf. Post-Quantum Cryptography (PQCrypto)*, Paris, France, 2020, pp. 72–91.

[21] P. Kampanakis, P. Panburana, E. Daw, and D. Van Geest, "The viability of post-quantum X.509 certificates," *IACR Cryptol. ePrint Arch.*, Paper 2018/063, 2018.

[22] K. Kwiatkowski and L. Valenta, "The TLS post-quantum experiment," *The Cloudflare Blog*, Oct. 2019.

[23] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The SPHINCS+ signature framework," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, Toronto, ON, Canada, 2019, pp. 2129–2146.

[24] National Security Agency, *Announcing the Commercial National Security Algorithm Suite 2.0 (CNSA 2.0)*, Cybersecurity Advisory, Sep. 2022.

[25] D. Stebila and M. Mosca, "Post-quantum key exchange for the Internet and the Open Quantum Safe project," in *Proc. Int. Conf. Sel. Areas Cryptogr. (SAC)*, St. John's, NL, Canada, 2016, pp. 14–37.