PEER-REVIEWED JOURNAL OF COMPUTER SCIENCE (PRJCS)

ISSN Online : Open Access Scholarly Journal

Post-Quantum Cryptography Migration: Kyber And Dilithium in TLS 1.3

V. Sakhtipriyanka

Assistant Professor, Department of Computer Science, Kongunadu Arts and Science College, Coimbatore, India.

Article information

Received: 25th April 2026
Received in revised form: 17th May 2026
Accepted: 30th May 2026
Available online: 13th June 2026

Volume: 1
Issue: 6
DOI: https://doi.org/10.5281/zenodo.20675499

Abstract

A sufficiently capable quantum computer would break the public-key cryptography that secures the modern internet, and data captured today can be stored until that machine arrives. This harvest-now-decrypt-later threat has pushed standards bodies to act well ahead of the hardware. In 2024 NIST published the first finished post-quantum standards: ML-KEM, derived from CRYSTALS-Kyber, for key establishment, and ML-DSA, derived from CRYSTALS-Dilithium, for digital signatures. This paper examines how these algorithms move into the two protocols that carry most secured traffic, TLS 1.3 and IPsec. We describe the hybrid handshake that runs a classical and a post-quantum key-encapsulation mechanism side by side, we measure the cost of the larger keys and signatures these schemes carry, and we discuss the practical friction points such as oversized certificate chains and fragmented handshake records. Benchmarks reported in the literature show that a hybrid X25519 plus Kyber-768 handshake adds only a small latency penalty on a healthy network, while Dilithium signatures inflate the certificate path enough to matter on constrained links. We close with a staged migration plan built around crypto-agility.

Keywords: - Post-Quantum Cryptography, Kyber, Dilithium, ML-KEM, ML-DSA, TLS 1.3, IPsec, Hybrid Key Exchange, Crypto-Agility.

.

I. INTRODUCTION

Almost every secure connection on the internet leans on one of two hard problems: factoring large integers, as in RSA, or computing discrete logarithms over elliptic curves, as in ECDH and ECDSA. Both are easy for a quantum computer running Shor's algorithm [1]. The machines that exist today are far too small to threaten real keys, yet the timeline that matters is not when such a computer is built. It is the moment an adversary begins recording encrypted traffic with the intent of decrypting it later [2]. Anything with a long confidentiality lifetime, medical records, state secrets, financial archives, is already exposed.

Recognising this, the U.S. National Institute of Standards and Technology ran a multi-year public competition to select quantum-resistant primitives [3]. The process concluded in 2024 with three finished standards. FIPS 203 specifies ML-KEM, a key-encapsulation mechanism built on the module-learning-with-errors problem and based directly on CRYSTALS-Kyber [4], [5]. FIPS 204 specifies ML-DSA, a lattice signature scheme based on CRYSTALS-Dilithium [6], [7]. A hash-based signature, SLH-DSA, was standardised alongside them as a conservative backup [8]. These are now the reference algorithms for new deployments.

Standardising an algorithm is the easy part. Fitting it into protocols that were designed around small elliptic-curve keys is harder. A Kyber-768 public key is roughly 1.2 kilobytes against 32 bytes for X25519, and a Dilithium-3 signature runs past 3 kilobytes [5], [7]. Those numbers ripple through handshake records, certificate chains, and packet sizes. This paper looks at the two protocols that carry the bulk of protected traffic, TLS 1.3 and IPsec, and asks a concrete question:

what actually changes when Kyber and Dilithium replace or augment the classical primitives, and how should an operator stage the switch.

II. THE QUANTUM THREAT AND THE LATTICE RESPONSE

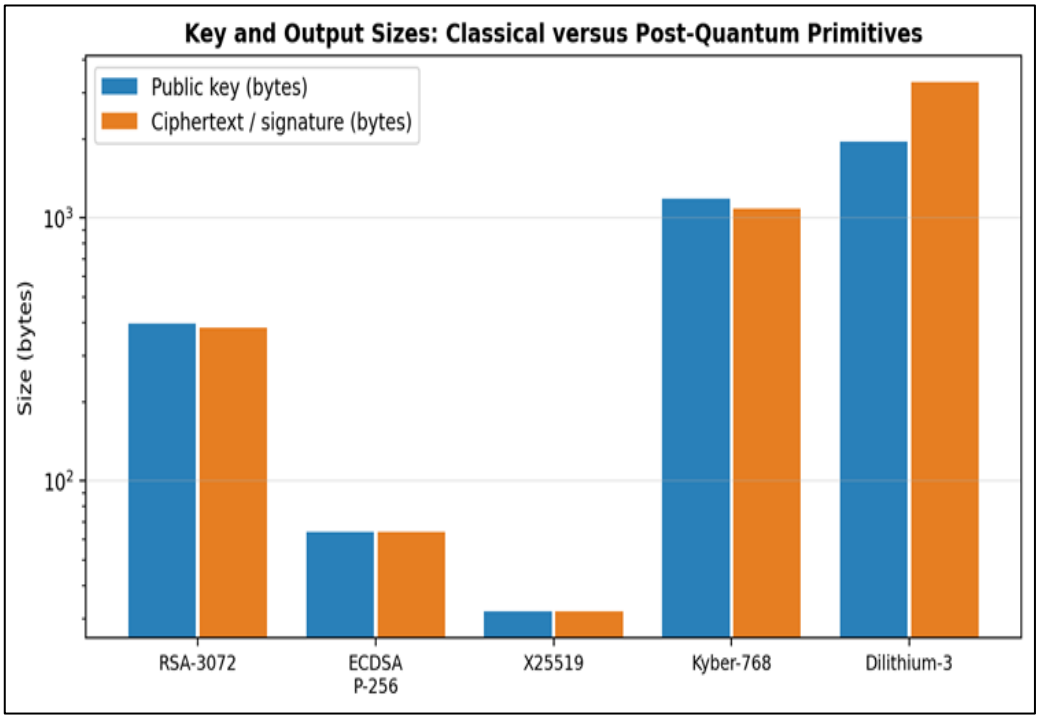
A. What Shor and Grover Break

Shor's algorithm solves factoring and discrete logarithm in polynomial time, which collapses RSA, finite-field Diffie-Hellman, and elliptic-curve cryptography once a large fault-tolerant quantum computer exists [1]. Symmetric cryptography fares better. Grover's search gives only a quadratic speedup against a block cipher, so doubling the key length restores the original security margin [9]. The upshot is narrow but sharp: AES-256 and SHA-384 remain fine, while every widely deployed public-key scheme needs replacing [2], [10].

B. Module Lattices as a Foundation

Both NIST KEM and signature winners rest on structured lattices, specifically the module-learning-with-errors and module-short-integer-solution problems [4], [6]. The appeal is threefold. The problems have resisted decades of analysis, including quantum attacks. The arithmetic is fast because it uses the number-theoretic transform over small polynomial rings. And the parameter sets scale cleanly, so Kyber and Dilithium each ship in three security levels that map onto the AES-128, AES-192, and AES-256 categories [5], [7]. That said, lattice schemes pay for their speed with size, and size is the recurring theme of any migration story.

Fig. 1. Public-key and output sizes for classical and post-quantum primitives. Note the logarithmic axis; Kyber and Dilithium are one to two orders larger than their elliptic-curve counterparts.



III. MIGRATING TLS 1.3

A. Why Hybrid First

Few operators want to bet a connection's confidentiality on a young algorithm alone. The community answer is hybrid key exchange, which combines a classical mechanism such as X25519 with a post-quantum KEM such as Kyber and feeds both shared secrets into the key schedule [11], [12]. The session stays secure as long as either component holds. If a lattice break appears, X25519 still protects against a classical attacker; if a quantum computer arrives, Kyber covers the gap. Google and Cloudflare ran exactly this construction across production traffic years before the standards were final, which gave the industry real telemetry rather than lab estimates [13], [14].

B. Where Kyber Sits in the Handshake

TLS 1.3 already negotiates key material in the first round trip through the key_share extension [15]. A hybrid group slots in naturally: the client sends an X25519 share concatenated with a Kyber encapsulation key, and the server replies with its X25519 share alongside the Kyber ciphertext. Both sides then derive a combined secret. Figure 2 traces the exchange. Crucially the round-trip structure does not change, so the latency cost is dominated by the extra bytes on the wire rather than by added network round trips [12], [16].

Fig. 2. A hybrid TLS 1.3 handshake. The classical and post-quantum key shares travel together, and Dilithium provides authentication through the certificate and Certificate Verify messages.

C. The Cost of Authentication

Key exchange is the gentle half of the problem. Authentication is rougher. A server certificate signed with Dilithium-3, presenting a Dilithium public key, can enlarge the certificate message severalfold once the full chain is counted [16], [17]. Measurements on real handshakes show that the KEM swap is nearly free on a well-provisioned path, whereas the signature growth is what stretches the handshake and can trigger extra TCP segments under a small initial congestion window [16], [18]. Table 1 summarises representative figures drawn from published benchmarks.

Table 1: Representative TLS 1.3 Handshake Cost (from published benchmarks

Configuration	Handshake Bytes	Extra RTTs	Median Latency Overhead
X25519 + ECDSA (classical)	~1.5 KB	0	Baseline
X25519 + Kyber-768 (hybrid KEM)	~3.8 KB	0	Low (a few %)
Hybrid KEM + Dilithium-3 auth	~10–13 KB	0–1	Moderate on slow links
Pure PQ (Kyber + Dilithium)	~10–12 KB	0–1	Moderate on slow links

IV. MIGRATING IPSEC

A. IKEv2 Key Exchange

IPsec establishes its keys through the Internet Key Exchange protocol, IKEv2 [19]. The same hybrid idea applies, but the mechanics differ because IKE runs over UDP and an oversized message must be fragmented at the IKE layer rather than relying on TCP. An extension lets IKEv2 carry one or more additional post-quantum key exchanges and mix every shared secret into the final keying material [20]. A Kyber payload therefore rides alongside the traditional Diffie-Hellman group, and the negotiation tolerates the larger values provided both peers support IKE fragmentation.

B. Tunnels Versus Handshakes

IPsec changes the cost calculus in one helpful way. A site-to-site tunnel performs the expensive asymmetric work only when the security association is created or rekeyed, not on every packet, since bulk traffic is protected by a symmetric cipher such as AES-GCM [10]. The post-quantum penalty is paid at setup and then amortised across the life of the tunnel. For long-lived VPNs this makes the migration far less sensitive to Dilithium's signature size than the per-connection world of web TLS, where fresh handshakes happen constantly.

V. A STAGED MIGRATION PLAN

A. Inventory and Crypto-Agility

No migration succeeds without first knowing what is deployed. The opening step is a cryptographic inventory that records where keys live, which algorithms protect them, and how long each protected item must stay confidential [2], [21]. Systems whose secrets must survive for a decade or more move to the front of the queue. Alongside the inventory, software needs crypto-agility, meaning algorithms are negotiated and swappable rather than wired in, so the next change does not demand another decade of effort [3].

B. Sequencing the Rollout

A pragmatic order has emerged from early deployments. Hybrid key exchange goes first, because it is low-risk and protects confidentiality immediately against the harvest-now threat [13], [14]. Post-quantum authentication follows once certificate authorities issue Dilithium credentials and the larger chains are shown to behave on real networks [17], [18]. Table II lays out the phases. Throughout, classical algorithms remain in the hybrid construction until confidence in the lattice schemes is broad, which is the whole point of hybrids.

Table 2: Phased Post-Quantum Migration

Phase	Action	Primary Risk Addressed
Discover	Cryptographic inventory, label long-lived data	Unknown exposure
Agility	Make algorithms negotiable and swappable	Future lock-in
Hybrid KEM	Deploy X25519 + Kyber in TLS and IKEv2	Harvest-now-decrypt-later
PQ Auth	Roll out Dilithium certificates	Quantum forgery of signatures
Review	Retire classical where mandated, monitor	Residual classical reliance

VI. CONCLUSION

The post-quantum transition is no longer a research topic but an engineering programme with finished standards behind it. Kyber and Dilithium, now ML-KEM and ML-DSA, give protocol designers concrete tools, and TLS 1.3 and IPsec both have clear paths to adopt them. The honest summary is that key exchange is close to a solved problem: a hybrid X25519 plus Kyber handshake costs little on a healthy network and protects data now. Authentication is the harder edge, because Dilithium's signatures stretch certificate chains in ways that show up on slow or lossy links. A staged plan that starts from a cryptographic inventory, insists on crypto-agility, deploys hybrid key exchange early, and adds post-quantum signatures once the ecosystem is ready offers a route that manages risk without waiting for the threat to become urgent.

REFERENCES

[1] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” IEEE Trans. Inf. Theory, vol. 26, no. 5, pp. 1484–1509, 1997.

[2] 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.

[3] G. Alagic et al., “Status report on the third round of the NIST post-quantum cryptography standardization process,” NIST Interagency/Internal Rep. (NISTIR) 8413, Gaithersburg, MD, USA, Jul. 2022.

[4] 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., Apr. 2018, pp. 353–367.

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

[6] 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, 2018.

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

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

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

[10] L. Chen et al., “Report on post-quantum cryptography,” NIST Interagency/Internal Rep. (NISTIR) 8105, Gaithersburg, MD, USA, Apr. 2016.

[11] D. Stebila, S. Fluhrer, and S. Gueron, “Hybrid key exchange in TLS 1.3,” IETF Internet-Draft, draft-ietf-tls-hybrid-design, 2023.

[12] E. Crockett, C. Paquin, and D. Stebila, “Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH,” in Proc. 2nd NIST Post-Quantum Cryptography Standardization Conf., Santa Barbara, CA, USA, Aug. 2019.

[13] D. Stebila and M. Mosca, “Post-quantum key exchange for the Internet and the Open Quantum Safe project,” in *Selected Areas in Cryptography (SAC)*, Cham, Switzerland: Springer, 2016, pp. 14–37.

[14] K. Kwiatkowski and L. Valenta, “The TLS post-quantum experiment,” Cloudflare Research, 2019.

[15] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3,” IETF RFC 8446, Aug. 2018.

[16] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, “Post-quantum authentication in TLS 1.3: A performance study,” in Proc. Network Distrib. Syst. Security Symp. (NDSS), San Diego, CA, USA, Feb. 2020.

[17] P. Kampanakis, P. Panburana, E. Daw, and D. Van Geest, “The viability of post-quantum X.509 certificates,” IACR Cryptol. ePrint Archive, Rep. 2018/063, 2018.

[18] C. Paquin, D. Stebila, and G. Tamvada, “Benchmarking post-quantum cryptography in TLS,” in *Post-Quantum Cryptography (PQCrypto)*, Cham, Switzerland: Springer, 2020, pp. 72–91.

[19] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen, “Internet Key Exchange Protocol Version 2 (IKEv2),” IETF RFC 7296, Oct. 2014.

[20] C. Tjhai et al., “Multiple key exchanges in IKEv2,” IETF RFC 9370, Mar. 2023.

[21] W. Barker, W. Polk, and M. Souppaya, “Getting ready for post-quantum cryptography,” NIST Cybersecurity White Paper, Gaithersburg, MD, USA, 2021.