Proceedings of CONF-CDS 2026 Symposium: Data-Centric AI Security: Securing Models, Learning Agents, and Autonomous Systems
DOI: 10.54254/2755-2721/2026.CH35917

Post-Quantum Cryptography Migration in Internet Protocols: A Review of ML-KEM Hybrid Key Exchange in TLS and SSH

Yan Zhang

Southwest Jiaotong University, Chengdu, China
yan.zhang.ece@outlook.com

Abstract. Internet public-key cryptography faces long-term risk from quantum computers, especially when traffic can be collected now and decrypted later. After the NIST post-quantum cryptography standards, the deployment task has shifted from algorithm selection to protocol migration. This paper reviews ML-KEM hybrid key exchange in TLS and SSH through a standards-first narrative review and protocol comparison. It synthesizes NIST standards, RFCs and IETF drafts, experiments, measurements, and primary deployment reports. The paper develops a deployment-readiness framework with four layers: security continuity, protocol integration, operational observability, and crypto-agility. The analysis shows hybrid key exchange represents the most feasible short-term solution, as it introduces post-quantum confidentiality without discarding existing elliptic curve security guarantees. However, hybrid deployment does not provide full post-quantum security. The review argues that readiness depends on protocol binding, implementation behavior, monitoring, and governance as much as on algorithm strength. Handshake size, middlebox compatibility, implementation safety, telemetry, authentication migration, and organizational crypto-agility determine whether migration can progress without weakening current Internet security.

Keywords: Post-quantum Cryptography, ML-KEM, TLS 1.3, SSH, Hybrid key exchange

1. Introduction

TLS protects web, API, cloud, and application traffic, while SSH protects remote administration and secure file transfer. Both normally rely on classical public-key cryptography for key establishment and authentication. Shor's algorithm showed that factoring and discrete logarithms are vulnerable to a sufficiently capable quantum computer [1]. Migration must therefore start before such a computer exists, because protected data can outlive the assumptions used to secure it [2]. This timing issue makes protocol migration an immediate operational concern, not just a future cryptographic risk. NIST's first post-quantum standards and transition guidance moved the field from algorithm selection toward deployment [3-6]. This paper assesses the practical deployment readiness of ML-KEM hybrid key exchange in TLS 1.3 and SSH, two widely deployed negotiation protocols carrying critical business traffic [7, 8]. Its contribution is a concise synthesis of protocol evidence and a

DOI: 10.54254/2755-2721/2026.CH35917

readiness framework for security, operations, and governance. The paper first reviews the standardization background, then compares TLS and SSH hybrid designs, and finally evaluates deployment challenges and migration strategy.

This paper uses a standards-first narrative review: Sources were selected when they addressed PQC transition risk, NIST standardization, ML-KEM in TLS or SSH, hybrid-exchange evidence, or migration guidance. The corpus contains 24 sources, weighted toward standards, RFCs, versioned Internet-Drafts, experiments, measurements, and primary deployment documentation. Sources on unrelated PQC candidates, theoretical cryptanalysis, non-Internet protocols, or commentary without primary evidence were excluded. The search covered material available through June 2026 and supports readiness synthesis rather than a complete global-adoption measurement. This constraint cannot be ignored: early deployments distribute unevenly worldwide, and public measurement data is far less sufficient than official standard and protocol files. The method therefore favors traceable technical claims over broad market-share claims.

2. Background and standards

The quantum threat includes harvest-now-decrypt-later risk: confidential data can be stored today and decrypted after future cryptanalytic advances. RFC 9794 frames hybrid schemes as combinations of post-quantum and currently trusted traditional mechanisms [9]. This explains why key agreement is often the first Internet migration target: its failure can expose previously captured traffic. Authentication migration is still necessary, but the urgency differs because impersonation usually affects future sessions rather than previously recorded ciphertext.

ML-KEM is the central KEM standard for this transition. FIPS 203 defines ML-KEM-512, ML-KEM-768, and ML-KEM-1024, while NIST SP 800-227 describes KEMs as shared-secret building blocks [4, 10]. In Internet deployment, ML-KEM-768 is often paired with X25519 because it balances strength, bandwidth, and cost. Table 1 sorts out all relevant standards and protocol mechanisms covered in this paper, and differentiates key-agreement standards from signature standards, which directly influence follow-up certificate transformation.

Table 1. Roles of selected post-quantum standards and protocol mechanisms

Item	Main role	Deployment relevance
ML-KEM, FIPS 203	Key encapsulation for shared-secret establishment	Core KEM for TLS and SSH post-quantum key exchange [4]
ML-DSA, FIPS 204	Digital signatures	Future TLS certificate and software-signing migration path [5]
SLH-DSA, FIPS 205	Stateless hash-based signatures	Conservative signature alternative with larger signatures [6]
X25519MLKEM768	TLS 1.3 hybrid key agreement	Near-term Web deployment candidate combining X25519 and ML-KEM-768 [11]
mlkem768x25519-sha256	SSH hybrid key exchange	OpenSSH-aligned method combining ML-KEM-768 and X25519 [12, 13]

2.1. NIST PQC standards and standardization progress

Digital-signature standards remain relevant because TLS and SSH also authenticate servers and users. ML-DSA and SLH-DSA introduce size and performance trade-offs for certificates, software signing, and infrastructure management [5, 6]. NIST's designation of HQC as an alternative backup

DOI: 10.54254/2755-2721/2026.CH35917

KEM also highlights the value of algorithm diversity, even though ML-KEM remains the primary standard [14]. This paper only centers on key exchange out of research scope constraints, rather than implying that hybrid key exchange delivers complete post-quantum protection.

The IETF documents also require careful interpretation. Several protocol specifications for ML-KEM in TLS and SSH are active Internet-Drafts, which may still be revised before publication. They are used here as evidence of current standardization direction and protocol design choices, not as final RFC requirements. This distinction matters because a deployment-oriented review must separate stable algorithm standards from evolving protocol integration work.

2.2. TLS and SSH migration needs

TLS and SSH are attractive early targets because both negotiate cryptographic algorithms. TLS 1.3 can negotiate new supported groups and key shares [7], while SSH was designed for flexible algorithm negotiation [8]. This does not make migration automatic. New key shares and ciphertexts are larger than classical equivalents, and middleboxes may reject unfamiliar identifiers or larger messages. Thus the core concern lies not in whether protocols can identify new algorithms, but in the stability of negotiation logic, fallback mechanisms, error handling and monitoring modules.

Their risks differ. TLS faces unique challenges brought by public web scenarios: massive heterogeneous clients, widespread CDN nodes, cloud edge services and inconsistent middlebox processing rules. SSH exposes the administration problem of long-lived servers, scripts, and older clients. Comparing the two therefore shows both scale-driven and inertia-driven migration barriers. This comparison is useful because success in a browser or OpenSSH ecosystem does not automatically imply readiness across every dependent system.

3. ML-KEM hybrid key exchange

Hybrid key exchange integrates classical cryptography and post-quantum algorithms into a unified negotiable key agreement scheme. The TLS hybrid design uses concatenated component values and incorporates the resulting secrets into the TLS 1.3 key schedule [15]. The goal is transition robustness: session keys should remain secure if at least one component remains unbroken and the hybrid construction is correctly implemented. This design brings obvious advantages: it lessens reliance on newly proposed post-quantum hardness assumptions and weakens reliance on quantum-vulnerable classical cryptography.

Hybrid exchange avoids a false choice between immediate pure-PQC deployment and delayed migration. It lets operators test post-quantum components inside existing protocol flows while retaining a familiar classical security assumption. The cost is that implementations must combine two mechanisms correctly and must not allow fallback behavior to become a downgrade path.

3.1. ECDHE-MLKEM in TLS 1.3

The TLS ECDHE-MLKEM draft defines X25519MLKEM768, SecP256r1MLKEM768, and SecP384r1MLKEM1024 [11]. X25519MLKEM768 is currently the most widely discussed deployment candidate because X25519 is widely deployed and ML-KEM-768 provides a strong post-quantum component. Its client and server key_exchange values are 1216 and 1120 bytes, respectively, so message size and middlebox behavior remain central deployment issues [11]. This is especially relevant for initial handshake flights, fragmented handshakes, and enterprise inspection products that were tuned for classical TLS behavior.

DOI: 10.54254/2755-2721/2026.CH35917

A related draft registers pure ML-KEM named groups [16], but hybrid groups are generally regarded as a practical near-term deployment compromise. They add post-quantum confidentiality without discarding elliptic-curve assurance. The trade-off is added complexity: two components, larger encodings, and combined failure modes must all be handled correctly. Hybrid deployment is therefore a transition technique, not a final security state.

3.2. Hybrid key exchange in SSH

SSH uses the same hybrid idea in a different message flow. The SSH ML-KEM draft defines mlkem768nistp256-sha256, mlkem1024nistp384-sha384, and mlkem768x25519-sha256 [12]. The client sends post-quantum and ECDH public keys, the server replies with an ML-KEM ciphertext and ECDH public key, and the two shared secrets are combined before SSH key derivation [12].

The draft highlights implementation risk by specifying length checks, fixed-length secret encodings, and abort behavior for failed checks or decapsulation failures [12]. SSH can move quickly inside OpenSSH, but older servers and clients still create a long tail of deployment inertia. SSH thus offers a useful contrast with TLS: its ecosystem is narrower, but operational dependencies can persist for years.

4. Deployment challenges

Prototype and performance studies show that TLS and SSH can support post-quantum or hybrid exchange, but readiness depends on negotiation, key combination, authentication, interoperability, network behavior, and governance [17, 18]. Deployment therefore requires a system-level view, not just algorithmic performance benchmarks. A deployment that passes microbenchmarks can still fail if clients cannot negotiate it reliably or operators lack visibility into why fallback occurs.

4.1. Performance and resource overhead

Performance studies show why performance cannot be summarized by a single primitive benchmark. Time-To-Last-Byte may capture user-visible cost better than handshake time alone [19], and optimized ML-KEM implementations can improve TLS 1.3 handshake throughput [20]. Deployment assessment therefore needs both microbenchmarks and end-to-end measurements. The core evaluation indicator lies in the actual overhead borne by online services under real network conditions including variable RTT, packet loss and high concurrent access. This is especially important for mobile clients and high-volume services, where small handshake changes can scale into visible reliability or capacity effects.

For TLS, larger key shares may increase ClientHello and ServerHello sizes. For SSH, hybrid exchange increases key-exchange payloads but remains within draft packet limits [12]. The practical effect depends on RTT, loss, congestion, certificate-chain size, implementation optimization, and concurrent load [18-20].

4.2. Operational adoption and compatibility

There is early evidence of deployment beyond prototypes. Measurements found OpenSSH and Chrome traffic using PQC mechanisms, but broad ecosystem migration remained limited in 2023-2024 [21]. Chrome's earlier X25519Kyber768 deployment also needed an enterprise policy switch because some appliances were incompatible with unfamiliar or larger handshakes [22].

DOI: 10.54254/2755-2721/2026.CH35917

As of June 2026, Cloudflare documents X25519MLKEM768 as its recommended hybrid TLS key agreement, and OpenSSH reports mlkem768x25519-sha256 as the default in OpenSSH 10.0 [13, 23]. These deployments demonstrate feasibility, but not ecosystem completion. Hybrid KEM exchange is moving first, while authentication, clients, middleboxes, libraries, and managed devices remain uneven. Organizations should therefore read early defaults as signals for pilots and telemetry, not as proof that migration is finished. This distinction is central for policy: default enablement in one major implementation reduces friction but does not eliminate inventory, exception handling, or rollback planning.

4.3. Implementation security and governance

Successful deployment depends on performance and interoperability, but also on implementation security and long-term governance. Implementation security requires correct randomness, strict length validation, safe decapsulation failure handling, side-channel resistance, and downgrade protection [10, 12]. Table 2 summarizes practical mitigation approaches.

Table 2. Deployment challenges and mitigation approaches

Challenge	Protocol impact	Mitigation approach
Handshake size	Larger TLS key shares and SSH exchange payloads	Measure handshake and application latency under realistic RTT and loss [18, 19]
Middlebox compatibility	Failure on unfamiliar TLS groups or larger messages	Pilot deployment, fallback telemetry, and enterprise controls [22, 23]
Implementation security	Randomness, decapsulation, encoding, and timing risks	Use vetted libraries, fixed-length encodings, and side-channel-resistant code [10, 12]
Operational governance	Incomplete migration across endpoints and dependencies	Build cryptographic inventories and staged replacement roadmaps [24]

It is necessary to distinguish key exchange from authentication mechanisms strictly. A hybrid KEM group protects session-key confidentiality, but classical certificate or host-key signatures may remain quantum-vulnerable. Thus, KEM and signature algorithm migration should be designed as a coordinated project, even with asynchronous deployment progress. Crypto-agility becomes a security property because systems must retire weak or incompatible choices without application redesign. This includes not only algorithm lists, but also monitoring, configuration control, library updates, and incident response when a selected algorithm or implementation must be replaced.

5. Phased migration strategy

A phased strategy should treat hybrid key exchange as both protection and operational learning. It reduces long-term confidentiality risk while preserving elliptic-curve assurance, giving operators time to measure incompatibilities and wait for protocol guidance to mature.

5.1. Hybrid migration path

TLS and SSH are suitable for this path because both already negotiate key exchange methods. X25519MLKEM768 and mlkem768x25519-sha256 allow deployment before pure PQC exchange and PQC authentication are universal [11, 12]. The governance risk is false completion. A realistic sequence is opt in testing, limited default deployment, monitored fallback, and eventual removal of classical, only paths. The final step should depend on measured client support.

DOI: 10.54254/2755-2721/2026.CH35917

5.2. Organizational migration framework

Enterprise-level PQC migration starts with a comprehensive cryptographic asset inventory, covering TLS endpoints, SSH servers, client cryptographic libraries, load balancers, network proxies, HSM devices, certificate services and embedded systems. Controlled function enablement must be accompanied by continuous telemetry monitoring, including operational failures, handshake latency, packet fragmentation, downgrade events and client algorithm compatibility status. The NCCoE project similarly emphasizes discovery and staged replacement planning [24].

The proposed deployment-readiness framework serves as a migration access gate rather than a simple evaluation scorecard: any migration scenario is deemed immature if operators lack full visibility into its operational layers. It asks whether migration preserves security continuity, integrates cleanly, can be observed, and can be changed as standards or risks evolve. Table 3 converts these criteria into operational indicators. This is the paper's main synthesis: the same hybrid mechanism can be cryptographically sound but operationally unready if failures, fallbacks, or upgrade paths are not measurable. The framework also helps separate protocol readiness from organizational readiness, which are often collapsed in deployment discussions.

Table 3. Operational deployment-readiness indicators for ML-KEM hybrid migration

Readiness layer	Operational indicators	Current risk
Security continuity	Transcript binding; downgrade resistance; remaining classical authentication	Hybrid key exchange is strong, but authentication is not fully post-quantum [9, 12, 15]
Protocol integration	Fixed-length encoding; explicit failure handling; draft or RFC maturity	TLS and SSH mechanisms are specified, but several items remain draft-dependent [11, 12, 16]
Operational observability	Negotiated group logs; fallback rate; failure rate; latency impact	Evidence is emerging from browsers, CDNs, OpenSSH, and measurement studies [13, 21-23]
Crypto-agility	Asset inventory; versioned configuration; upgrade path; retirement process	Migration requires organizational process beyond enabling a protocol option [10, 24]

Future work should extend to PQC signatures in certificate chains, host-key authentication, software signing, and update infrastructure. HQC should also be monitored as a code-based backup if lattice assumptions or deployment constraints change [14]. It is also neccessary to investigate end-to-end service scenarios covering TLS, SSH, VPN, API communication, certificate verification and device management channels. This broader view matters because real services often depend on several cryptographic layers, and weakness in one layer can limit the value of migration elsewhere.

6. Conclusion

Post-quantum migration for core Internet protocols has progressed from theoretical planning to practical deployment stages. ML-KEM in FIPS 203 gives designers and operators a standardized KEM for post-quantum key establishment. TLS and SSH are suitable early targets because both negotiate cryptographic algorithms and protect high-value communication. Existing empirical evidence confirms hybrid key exchange as the most feasible short-term migration solution.

The main conclusion is that deployment is broader than the primitive. Migration effectiveness is determined by multiple practical factors, including packet encoding rules, handshake payload size, endpoint compatibility, middlebox behavior, secure implementation, real-time monitoring and

DOI: 10.54254/2755-2721/2026.CH35917

systematic governance. Hybrid ML-KEM exchange reduces long-term confidentiality risk but does not complete authentication migration and can increase implementation complexity. The readiness framework evaluates this gap through security continuity, protocol integration, observability, and crypto-agility. Although several protocol documents remain Internet-Drafts and public measurements are limited, the evidence supports measurable hybrid deployment followed by authentication and certificate migration. This phased migration strategy is prudently designed. It outperforms two extreme strategies: indefinite waiting for full PQC standard maturity and premature completion declaration based only on single hybrid algorithm enablement. For TLS and SSH, the near-term objective should be measurable transition without weakening classical security during the transition period.

This study further clarifies the necessity of rational evaluation for hybrid deployment schemes. It is a practical and defensible transition path, but not a complete post-quantum state. If organizations enable hybrid key exchange without asset inventories, negotiated group telemetry, fallback monitoring, and upgrade procedures, they may gain a new cryptographic component without gaining real migration control. The proposed readiness framework therefore treats security continuity, protocol integration, operational observability, and crypto-agility as joint requirements. This analytical framework avoids two typical migration misconceptions: passive delay awaiting full PQC maturity, and overestimating the security capability of a single default hybrid algorithm. The most sustainable path is staged adoption backed by measurement, followed by signature, certificate, and multi-protocol migration as evidence improves.

References

[1] Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 124–134. https://doi.org/10.1109/SFCS.1994.365700
[2] Mosca, M., & Piani, M. (2024). Quantum threat timeline report 2024. *Global Risk Institute*. https://globalriskinstitute.org/publication/2024-quantum-threat-timeline-report/
[3] Moody, D., Perlner, R., Regenscheid, A., Robinson, A., & Cooper, D. (2024). Transition to post-quantum cryptography standards. *NIST IR 8547 Initial Public Draft*. https://doi.org/10.6028/NIST.IR.8547.ipd
[4] National Institute of Standards and Technology. (2024). Module-lattice-based key-encapsulation mechanism standard. *FIPS 203*. https://doi.org/10.6028/NIST.FIPS.203
[5] National Institute of Standards and Technology. (2024). Module-lattice-based digital signature standard. *FIPS 204*. https://doi.org/10.6028/NIST.FIPS.204
[6] National Institute of Standards and Technology. (2024). Stateless hash-based digital signature standard. *FIPS 205*. https://doi.org/10.6028/NIST.FIPS.205
[7] Rescorla, E. (2018). The transport layer security (TLS) protocol version 1.3. RFC 8446. https://www.rfc-editor.org/rfc/rfc8446.html
[8] Ylonen, T., & Lonvick, C. (2006). The secure shell (SSH) transport layer protocol. RFC 4253. https://www.rfc-editor.org/rfc/rfc4253.html
[9] Driscoll, F., Parsons, M., & Hale, B. (2025). Terminology for post-quantum traditional hybrid schemes. RFC 9794. https://www.rfc-editor.org/rfc/rfc9794.html
[10] Alagic, G., Barker, E., Chen, L., Moody, D., Robinson, A., Silberg, H., & Waller, N. (2025). Recommendations for key-encapsulation mechanisms. NIST SP 800-227. https://doi.org/10.6028/NIST.SP.800-227
[11] Kwiatkowski, K., Kampanakis, P., Westerbaan, B. E., & Stebila, D. (2026). Post-quantum hybrid ECDHE-MLKEM key agreement for TLSv1.3. IETF Internet-Draft, draft-ietf-tls-ecdhe-mlkem-05. https://datatracker.ietf.org/doc/html/draft-ietf-tls-ecdhe-mlkem-05
[12] Kampanakis, P., Stebila, D., & Hansen, T. (2026). PQ/T hybrid key exchange with ML-KEM in SSH. IETF Internet-Draft, draft-ietf-sshm-mlkem-hybrid-kex-10. https://datatracker.ietf.org/doc/html/draft-ietf-sshm-mlkem-hybrid-kex-10
[13] OpenSSH. (n.d.). Post-quantum cryptography. *OpenSSH*. https://www.openssh.org/pq.html Accessed June 21, 2026.

DOI: 10.54254/2755-2721/2026.CH35917

[14] National Institute of Standards and Technology. (2025). NIST selects HQC as fifth algorithm for post-quantum encryption. *NIST*. https://www.nist.gov/news-events/news/2025/03/nist-selects-hqc-fifth-algorithm-post-quantum-encryption Accessed June 18, 2026.
[15] Stebila, D., Fluhrer, S., & Gueron, S. (2025). Hybrid key exchange in TLS 1.3. *IETF Internet-Draft, draft-ietf-tls-hybrid-design-16*. https://datatracker.ietf.org/doc/html/draft-ietf-tls-hybrid-design-16
[16] Connolly, D. (2026). ML-KEM post-quantum key agreement for TLS 1.3. *IETF Internet-Draft, draft-ietf-tls-mlkem-08*. https://datatracker.ietf.org/doc/html/draft-ietf-tls-mlkem-08
[17] Crockett, E., Paquin, C., & Stebila, D. (2019). Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH. *Cryptology ePrint Archive, Paper 2019/858*. https://eprint.iacr.org/2019/858
[18] Sikeridis, D., Kampanakis, P., & Devetsikiotis, M. (2020). Assessing the overhead of post-quantum cryptography in TLS 1.3 and SSH. *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies*, 149–156. https://doi.org/10.1145/3386367.3431305
[19] Kampanakis, P., & Childs-Klein, W. (2024). The impact of data-heavy, post-quantum TLS 1.3 on the time-to-last-byte of web connections. *Workshop on Measurements, Attacks, and Defenses for the Web*. https://doi.org/10.14722/madweb.2024.23010
[20] Zheng, J., Zhu, H., Dong, Y., Song, Z., Zhang, Z., Yang, Y., & Zhao, Y. (2024). Faster post-quantum TLS 1.3 based on ML-KEM: Implementation and assessment. arXiv:2404.13544. https://arxiv.org/abs/2404.13544
[21] Sowa, J., Hoang, B., Yeluru, A., Qie, S., Nikolich, A., Iyer, R. K., & Cao, P. (2024). Post-quantum cryptography (PQC) network instrument: Measuring PQC adoption rates and identifying migration pathways. *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 1835-1846. https://doi.org/10.1109/QCE60285.2024.00213
[22] O'Brien, D. (2023). Protecting Chrome traffic with hybrid Kyber KEM. *Chromium Blog*. https://blog.chromium.org/2023/08/protecting-chrome-traffic-with-hybrid.html
[23] Cloudflare. (2026). Post-quantum cryptography. *Cloudflare SSL/TLS Documentation*. https://developers.cloudflare.com/ssl/post-quantum-cryptography/
[24] National Cybersecurity Center of Excellence. (n.d.). Migration to post-quantum cryptography. https://www.nccoe.nist.gov/applied-cryptography/migration-to-pqc