Communication

A Readiness and Maturity Framework for Post-Quantum TLS Adoption

Francis Kagai *

Posted Date: 2 June 2026

doi: 10.20944/preprints202606.0067.v1

Keywords: post-quantum cryptography; Transport Layer Security; technology adoption; operating systems; harvest-now decrypt-later

 doi:10.20944/preprints202606.0067.v1

Communication

A Readiness and Maturity Framework for Post-Quantum TLS Adoption

Francis Kagai

Swinburne University of Technology, Melbourne, Australia; fkagai@swin.edu.au

Abstract

Transport Layer Security (TLS) protects most Internet traffic, yet migration to post-quantum TLS (PQ-TLS) depends on more than publishing a new algorithm standard. Despite rapid standardization of the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) and hybrid TLS groups, practitioners lack a structured way to assess whether deployment ecosystems are ready to adopt PQ-TLS. This letter proposes a six-dimensional readiness and adoption framework and a reproducible maturity rubric (levels 0–4). A case study of Windows, Linux, macOS, and Android applies the rubric to cited vendor documentation. The main finding is that cryptographic and protocol readiness are uniformly high across the case study, while platform and application readiness show the widest gap and act as the primary adoption bottlenecks for software that inherits OS TLS APIs. The study reports documented transition signals, not live PQ-TLS negotiation rates.

Keywords: post-quantum cryptography; Transport Layer Security; technology adoption; operating systems; harvest-now decrypt-later

1. Introduction

Despite rapid standardization of ML-KEM in Federal Information Processing Standards (FIPS) publication 203 [1] and growing deployment of hybrid key-exchange groups such as X25519MLKEM768 (which pairs the X25519 elliptic curve with ML-KEM at the 768-bit parameter set) [2,3], there is no compact, reproducible method for assessing whether real deployment ecosystems are ready to adopt post-quantum TLS (PQ-TLS). This letter addresses that gap.

PQ-TLS protects session keys against future cryptanalytic advances while much of the Internet still negotiates classical key exchange under TLS version 1.3 (TLS 1.3) [4]. The threat is not only hypothetical. An adversary can record ciphertext today and attempt to decrypt it later if session keys depended on mathematics that a future quantum computer could break. Security literature describes this pattern as harvest-now, decrypt-later (HNDL) [5]. Migration is therefore a layered adoption problem spanning standards, protocol behavior, operating system (OS) APIs, applications, operator practice, and vendor deployment reporting.

Prior work measures handshake cost [6], builds laboratory testbeds [7], and studies domain-specific stacks [8–10], but does not provide a cross-layer maturity method that others can reapply when vendor guidance changes. This letter contributes a six-dimensional PQ-TLS readiness and adoption framework, a level 0–4 scoring rubric applied consistently across all six dimensions, and an OS case study that extracts a single cross-layer finding from cited vendor documentation.

2. Background

During a TLS 1.3 handshake, the client and server advertise allowed key-exchange groups in the opening ClientHello message; the negotiated group determines how session keys are derived [4]. Deployments typically retain classical groups during transition so a session remains protected if either the classical or post-quantum component withstands attack [3]. Many programs inherit TLS from OS networking APIs (Schannel on Windows, OpenSSL on Linux, Network framework and URLSession on

 doi:10.20944/preprints202606.0067.v1

Apple platforms, Conscrypt on Android [10]) rather than shipping a private TLS library. Browser and cloud services can upgrade on their own release schedules [2], but enterprise agents and mobile apps that call platform TLS move with what vendors document and enable on that path.

3. PQ-TLS Readiness and Adoption Framework

The six dimensions were selected to represent the principal deployment layers through which PQ-TLS capabilities propagate from standards to operational Internet deployment. Cryptographic readiness covers standardized post-quantum algorithms and hybrid groups. Protocol readiness covers TLS 1.3 negotiation, compatibility, and fallback when hybrid groups are offered. Platform readiness covers hybrid support in OS TLS stacks such as Schannel, OpenSSL, Network framework, and Conscrypt. Application readiness covers whether browsers, servers, and apps can use a documented PQ-TLS path in production. Operational readiness covers configuration, testing, monitoring, and migration without breaking dependents. Ecosystem readiness covers vendor guidance, certificate authority practice, and public deployment reporting. Each dimension asks whether that layer is documented, deployable, or observable in practice. Table 1 summarizes the core question and an example transition signal for each dimension.

Table 1. PQ-TLS readiness and adoption framework.

Dim.	Core question	Example signal
A. Crypto	Are PQ algorithms and hybrid groups standardized and available?	ML-KEM in FIPS 203 [1].
B. Protocol	Does TLS 1.3 negotiation support hybrid groups with safe fallback?	Hybrid groups in ClientHello [4,11].
C. Platform	Does the OS TLS stack document a hybrid path on the default API?	Schannel, OpenSSL+OQS, Network framework, Conscrypt [12–14].
D. Application	Can production apps use an available PQ-TLS path?	Chrome defaults; Conscrypt apps [2,10].
E. Operational	Can operators configure, test, and migrate safely?	Policy controls, provider install [11,13].
F. Ecosystem	Do vendors and standards bodies expose PQ-relevant guidance or deployment reporting?	Vendor PQ roadmaps; browser deployment reports [2,12].

Weakness in any layer can block end-to-end PQ-TLS even when another layer is advanced. In practice, lower layers constrain higher ones: standardized algorithms must exist before protocol negotiation can advertise hybrid groups, platform and application paths must expose those groups before operators can deploy them, and ecosystem tools then help track progress at scale. Figure 1 shows this dependency chain and marks the platform and application layers where the case study finds the widest maturity gap.

 doi:10.20944/preprints202606.0067.v1

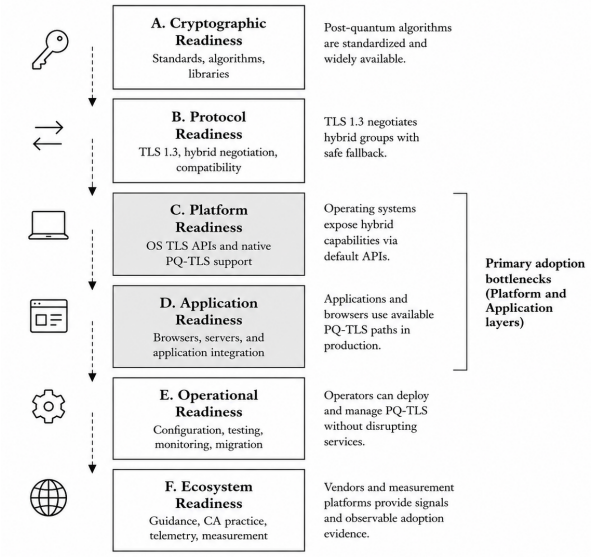

Figure 1. PQ-TLS readiness and adoption framework. Successful post-quantum TLS migration requires coordinated progress across six interdependent layers spanning cryptographic standards, protocol support, platform capabilities, application integration, operational deployment, and ecosystem observability. The case study demonstrates that platform and application readiness remain the principal bottlenecks despite mature cryptographic and protocol foundations.

3.1. Maturity Rubric and Scoring Rules

We score each dimension independently using the same ordinal rubric: level 0 means no documented support; level 1 means experimental or external-library support only; level 2 means opt-in or platform-adjacent support; level 3 means documented native support on the relevant API or stack; level 4 means documented native support enabled by default on current vendor releases. Scores reflect cited vendor or standards documentation only, not measured negotiation on the live Internet.

For dimensions A and B, all four OS families inherit the same ecosystem signals in this case study because ML-KEM is standardized [1], TLS 1.3 defines the negotiation framework [4], and hybrid group behavior is documented in browser and enterprise TLS guidance [2,11]. Dimensions C through F are scored per OS using stack-specific documentation from Microsoft [12], the Open Quantum Safe (OQS) project [13], Apple [11,14], Google [2,15], and Mankowski et al. [10]. We report the bottleneck score as the minimum across the six dimensions for each OS row. We do not average dimensions with equal weights because layers are sequential dependencies rather than interchangeable attributes.

4. Case Study: Major Operating Systems

We applied the rubric to Windows 11, Linux, macOS 26, and Android using the vendor and standards sources cited above. Table 2 reports the resulting maturity scores for all six dimensions and the bottleneck score (minimum across dimensions) for each platform.

Table 2. Documented maturity matrix (levels 0–4) from cited vendor and standards sources.

Platform	A	B	C	D	E	F	Min
Windows 11	4	4	3	3	3	3	3
Linux	4	4	2	3	3	3	2
macOS 26	4	4	4	4	3	3	3
Android	4	4	0	2	2	2	0

cryptographic, protocol, platform, application, operational, ecosystem. Min: bottleneck score (minimum across dimensions).

Cryptographic and protocol scores are 4 for every row because the cited standards and protocol guidance apply ecosystem-wide. Platform scores range from 0 on Android (Conscrypt application programming interface without documented hybrid groups [10,15]) to 4 on macOS 26 (Network

 doi:10.20944/preprints202606.0067.v1

framework and URLSession with hybrid TLS enabled by default [11,14]). Windows 11 receives platform level 3 because Schannel documents X25519MLKEM768 on the native path but not as universally default-enabled in the Microsoft material we cite [12]. Linux receives platform level 2 because hybrid TLS requires installing and configuring the OQS provider for OpenSSL 3 [13].

Application scores reflect whether typical production software can reach PQ-TLS on documented paths. macOS 26 scores 4 because Apple documents default hybrid behavior for apps that use the cited networking APIs. Windows and Linux score 3 because PQ-TLS is documented for Schannel and for OpenSSL with OQS, respectively, but not for every application category in the sources. Android scores 2 because hybrid TLS is documented for the shipped Chrome and WebView stacks [2] while many Conscrypt-based applications lack a documented platform path [10].

4.1. Evidence Mapping

Operational and ecosystem scores use explicit criteria so that each value maps to cited documentation rather than author judgment alone. For operational readiness, level 3 requires documented configuration guidance, migration or compatibility procedures, and operator-facing deployment controls. Windows, Linux, and macOS meet this threshold: Microsoft documents post-quantum cryptography application programming interfaces and platform enablement for Schannel [12], the OQS project documents provider installation and TLS configuration steps for OpenSSL 3 [13], and Apple documents compatibility handling for larger ClientHello messages and temporary fallback controls during migration [11]. Android receives operational level 2 because Google documents post-quantum work for boot, attestation, and keystore [15] but does not publish equivalent migration guidance for Conscrypt-based application TLS in the sources cited here; operators must therefore treat browser and application paths separately.

For ecosystem readiness, level 3 requires public vendor guidance or deployment reporting that helps practitioners track PQ-TLS transition progress. Windows, Linux, and macOS each score 3 because Microsoft, the OQS project, Apple, and major browser vendors publish ongoing PQ-TLS roadmaps or deployment announcements in the cited material [2,12–14]. Android receives ecosystem level 2 because the cited Google and Mankowski sources document partial ecosystem movement (browser hybrid TLS and platform signature work) without comparable public reporting for Conscrypt TLS adoption [2,10,15].

The cross-layer result is the paper's primary finding. Dimensions A and B are uniformly at level 4, so cryptographic standardization and protocol specification are not the limiting factors in this case study. Dimensions C and D show the widest spread (0–4) and the lowest bottleneck scores. Platform readiness is the dominant constraint for Android (level 0 on Conscrypt) and the lowest-scoring layer for Linux (level 2). Application readiness separates macOS 26 (level 4) from Android (level 2), where browser traffic can use hybrid TLS but many platform-API applications cannot according to the cited documentation.

5. Discussion

The case study demonstrates that the framework produces a reproducible, evidence-linked result rather than a narrative label. Analysts can update Table 2 when vendors publish new guidance by reapplying the same rubric to the same six dimensions. The key insight for Internet infrastructure is that PQ-TLS standardization has largely matured at the cryptographic and protocol layers, but platform and application readiness remain the primary bottlenecks for software that inherits OS TLS APIs.

Reported handshake overhead remains small relative to wide-area network delay [6], so transition planning should weight enablement policy, API path selection, and observability alongside algorithm choice. The same six dimensions apply beyond desktop OS families to cloud load balancers, Internet of Things firmware, and fifth-generation (5G) control planes [8,9].

 doi:10.20944/preprints202606.0067.v1

6. Limitations and Future Work

This letter presents a framework and a documentation-based case study. It does not measure live PQ-TLS negotiation, interoperability between arbitrary stack pairs, or central processing unit and packet-size effects. Maturity scores apply only to the cited sources at the time of review.

Future work should add longitudinal handshake telemetry, including measurements over QUIC (Quick UDP Internet Connections) and HTTP/3 (Hypertext Transfer Protocol version 3), interoperability tests against public reference servers [16], and repeated application of the rubric to cloud, mobile, and regulated environments.

7. Conclusions

PQ-TLS migration should be evaluated as a layered adoption problem, not as a single algorithm upgrade. This letter contributes a readiness and adoption framework, a reproducible level 0–4 rubric, and an OS case study grounded in cited vendor documentation. The case study shows that cryptographic and protocol readiness are already high in the material reviewed, while platform and application readiness lag and form the primary adoption bottlenecks, especially where applications depend on OS TLS APIs rather than bundled browser stacks. Empirical work can extend the same matrix with live negotiation and performance measurements when those signals become available.

References

1. National Institute of Standards and Technology. FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. https://csrc.nist.gov/pubs/fips/203/final, 2024.
2. Google Chrome Team. A new path for Kyber on the web. https://security.googleblog.com/2024/09/a-new-path-for-kyber-on-web.html, 2024.
3. Garcia, C.R.; Aguilera, A.C.; Olmos, J.J.V.; Monroy, I.T.; Rommel, S. Quantum-Resistant TLS 1.3: A Hybrid Solution Combining Classical, Quantum and Post-Quantum Cryptography. In Proceedings of the 2023 IEEE 28th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). IEEE, 2023.
4. Rescorla, E. "The transport layer security (tls) protocol version 1.3," 2018.
5. Kagai, F.; Branch, P.; But, J.; Allen, R. Harvest-Now, Decrypt-Later: A Temporal Cybersecurity Risk in the Quantum Transition. *Telecom* **2025**, *6*, 100.
6. Döring, R.; Geitz, M. Post-Quantum Cryptography in Use: Empirical Analysis of the TLS Handshake Performance. In Proceedings of the NOMS 2022 - IEEE/IFIP Network Operations and Management Symposium. IEEE, 2022.
7. Kupcová, E.; Simko, J.; Pleva, M.; Drutarovsky, M. Experimental Framework for Secure Post-Quantum TLS Client-Server Communication. In Proceedings of the 2024 International Symposium ELMAR. IEEE, 2024.
8. Hanna, Y.; Pineda, D.; Veksler, M.; Paudel, M.; Akkaya, K.; Anastasova, M.; Azarderakhsh, R. Integrating Post-Quantum TLS into the Control Plane of 5G Networks. In Proceedings of the 2024 IEEE International Performance, Computing, and Communications Conference (IPCCC). IEEE, 2024.
9. Marchsreiter, D.; Sepúlveda, J. Hybrid Post-Quantum Enhanced TLS 1.3 on Embedded Devices. In Proceedings of the 2022 25th Euromicro Conference on Digital System Design (DSD). IEEE, 2022.
10. Mankowski, D.; Wiggers, T.; Moonsamy, V. TLS → Post-Quantum TLS: Inspecting the TLS Landscape for PQC Adoption on Android. In Proceedings of the 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). IEEE, 2023.
11. Apple. Prepare your network for quantum-secure encryption in TLS. https://support.apple.com/en-us/122756, 2025.
12. Microsoft. Post-Quantum Cryptography APIs Now Generally Available on Microsoft Platforms. https://techcommunity.microsoft.com/blog/microsoft-security-blog/post-quantum-cryptography-apis-now-generally-available-on-microsoft-platforms/4469093, 2025.
13. Open Quantum Safe Project. liboqs and oqs-provider for OpenSSL. https://openquantumsafe.org, 2024.
14. Apple. Quantum-secure cryptography in Apple operating systems. https://support.apple.com/guide/security/quantum-secure-cryptography-apple-devices-secc7c82e533/web, 2026.

 doi:10.20944/preprints202606.0067.v1

15. Google. Security for the quantum era: Implementing post-quantum cryptography in Android. https://blog.google/security/security-for-the-quantum-era-implementing-post-quantum-cryptography-in-android/, 2025.
16. Open Quantum Safe Project. Post-Quantum TLS Interoperability Test Server. https://test.openquantumsafe.org/, 2024.