Quantum-Safe Cryptography: A Migration Framework for Legacy Systems Toward NIST PQC Standards with the Crypto-Agility Readiness Score

Allan Douglas Costa
Institute of Cyberspace (ICIBE), Federal Rural University of the Amazon (UFRA)
Belém, Brazil
E-mail: allan.costa@ufra.edu.br

***Abstract*—Post-quantum cryptography (PQC) standardisation reached a pivotal milestone in August 2024 with the release of NIST FIPS 203 (ML-KEM) and FIPS 204 (ML-DSA), yet the vast majority of deployed public-key infrastructure continues to rely on RSA-2048 and Elliptic Curve Diffie-Hellman (ECDH), both vulnerable to Shor's algorithm [1] on a cryptographically relevant quantum computer. The *Harvest Now, Decrypt Later* (HNDL) threat renders this risk operationally present: adversaries may archive ciphertext today for future decryption once a Cryptographically Relevant Quantum Computer (CRQC) becomes available. This paper proposes the *Crypto-Agility Readiness Score* (CARS), a five-dimension weighted composite index for assessing PQC migration readiness in legacy systems across PKI, TLS, and HSM environments. CARS operationalises five dimensions–Inventory Completeness, Algorithm Compliance, Architectural Decoupling, Toolchain Readiness, and Governance & Compliance Alignment–with weights derived via a two-round Delphi process with 12 senior migration engineers. In an empirical evaluation of 43 open-source cryptographic software repositories (convenience sample; $n = 45$ targeted, 2 excluded due to unavailable source trees), mean CARS values (mean $\pm$ s.d.) were 24.9 ± 9.5 (Legacy Crypto Libraries), 25.8 ± 8.2 (PKI/Certificate Management), 39.7 ± 15.2 (HSM/PKCS11 Middleware), 47.2 ± 16.5 (TLS 1.3 Hybrid), and 47.5 ± 11.1 (PQC Native implementations; overall 34.1 ± 15.0, $n = 43$). A notable result is that PQC reference implementations score in the At-Risk range despite high Algorithm Compliance ($d_2 \geq 0.61$), because Architectural Decoupling (d_3) is near zero, confirming that algorithmic presence alone does not imply organisational migration readiness. Category differences are statistically significant (Kruskal-Wallis $H(4) = 17.55$, $p = 0.0015$, $\varepsilon^2 = 0.357$, large effect). Construct validity is supported by a moderate convergent Spearman correlation between Algorithm Compliance and Toolchain Readiness ($\rho = 0.558$, $p < 0.001$, $n = 43$) and discriminant independence from Architectural Decoupling ($\rho = -0.031$, $p = 0.843$). A longitudinal case study on oqs-provider (v0.3.0 to v0.6.0) demonstrates that CARS tracks real migration progress (ΔCARS $= +21$ pts). Microbenchmarks on server-class hardware (Intel Xeon E5-2680 v4) and an ARM Cortex-M4 platform show that ML-KEM-768 completes a full key-encapsulation cycle (KeyGen + Encaps + Decaps) in 0.28 ms versus 2.87 ms for RSA-2048 OAEP–a $10.2\times$ difference; authors' own measurements on Apple M1 Pro ARM64 (liboqs-python v0.15.0, $n = 10{,}000$) confirm the ratio is preserved across architectures (M1 Pro: 0.051 ms full KEM cycle). A hybrid TLS 1.3 handshake (X25519MLKEM768) adds approximately 1.29 ms incremental overhead over ECDHE-only under single-session conditions. A three-layer migration framework (Legacy Audit, Hybrid Bridge, PQC Native) is described and instantiated. CARS may support structured prioritisation of migration efforts; external predictive validation against migration outcomes remains future work. CNSA 2.0 timelines used as a planning reference apply specifically to U.S. National Security Systems.**

***Index Terms*—Post-Quantum Cryptography, CRYSTALS-Kyber, CRYSTALS-Dilithium, Legacy System Migration, Crypto-Agility, NIST PQC, ML-KEM, ML-DSA, Harvest Now Decrypt Later, TLS 1.3, CARS Metric, Hybrid Key Exchange**

I. Introduction

The discovery of Shor's algorithm [1] established a theoretical framework under which a sufficiently powerful quantum computer can factor large integers and solve discrete-logarithm problems in polynomial time, directly threatening RSA and Elliptic-Curve Cryptography (ECC). While fault-tolerant quantum processors capable of running Shor's algorithm at cryptographically relevant scales remain beyond current engineering limits, the *Harvest Now, Decrypt Later* (HNDL) threat transforms this into an immediate operational risk: adversaries capture encrypted traffic today and archive it for future decryption once a Cryptographically Relevant Quantum Computer (CRQC) becomes available [2], [3]. The temporal decoupling of data collection from decryption means that data encrypted with RSA-2048 or ECDH P-256 *today* may be decrypted before its confidentiality requirements expire.

In response to this threat, the National Institute of Standards and Technology (NIST) finalised three Post-Quantum Cryptography (PQC) standards in August 2024: FIPS 203 (ML-KEM, formerly CRYSTALS-Kyber) [4], FIPS 204 (ML-DSA, formerly CRYSTALS-Dilithium) [5], and FIPS 205 (SLH-DSA) [6], concluding an eight-year international standardisation process. These standards are complemented by the NSA's Commercial National Security Algorithm Suite 2.0 (CNSA 2.0) [7], which mandates ML-KEM-1024 and ML-DSA-87 for National Security Systems by 2030.

Despite this regulatory impetus, a significant gap persists between PQC standardisation and real-world adoption. Enterprise cryptographic stacks are deeply embedded across PKI hierarchies, TLS endpoints, HSMs, and IoT firmware, often with decade-long upgrade cycles. Systematic literature reviews [8], [9] reveal that fewer than 15% of organisations have conducted a formal cryptographic inventory, and no widely adopted quantitative metric exists to assess migration readiness across heterogeneous legacy environments–a gap this paper addresses directly. NIST guidance on crypto-agility [10] and transition

planning [11] underscores the urgency of systematic readiness assessment.

Contributions

This work makes the following contributions:

1) **Crypto-Agility Readiness Score (CARS):** A proposed, formal, five-dimensional weighted metric that quantifies an organisation's PQC migration readiness on a normalised $[0, 100]$ scale. CARS is preliminarily evaluated across three legacy system categories and provides a structured quantitative approach for migration prioritisation.
2) **Three-Layer Migration Framework:** A structured framework comprising a Legacy Audit Layer (cryptographic inventory and risk scoring), a Hybrid Bridge Layer (dual-stack and hybrid key exchange), and a PQC Native Layer (full algorithm migration), each with automated tooling specifications.
3) **Empirical Benchmark:** A reproducible performance evaluation of ML-KEM and ML-DSA against RSA-2048/ECDH/ECDSA baselines across server-class and ARM Cortex-M4 IoT environments, using public OQS `liboqs` benchmarks.
4) **Compliance Timeline Analysis:** A quantitative mapping of per-system CARS distributions to CNSA 2.0 and NIST 2030 deadlines, exposing the readiness gap for each legacy system category.
5) **Open-Source Tooling:** Reference implementations of `cars_scorer.py`, `legacy_scanner.py`, and TLS hybrid demo, available under MIT licence with CI/CD workflows.

Paper Organisation

Section II establishes mathematical foundations and the formal threat model. Section III reviews related work across six subtopics. Section IV presents the framework architecture, CARS definition, algorithms, and the hybrid TLS 1.3 handshake sequence. Section V describes the experimental setup. Section VI presents results and statistical analysis. Section VII provides a formal security analysis. Section VIII addresses compliance and ethics. Section IX concludes with future directions.

II. Background and Threat Model

A. Learning With Errors and Module-LWE

The security of both ML-KEM and ML-DSA rests on the hardness of the *Module Learning With Errors* (MLWE) problem, a structured generalisation of the Learning With Errors (LWE) problem introduced by Regev.

Definition 1 (Module-LWE)**.** *Let $k, n \geq 1$, prime q, and a distribution χ over $\mathbb{Z}_q$. Given m samples $(\mathbf{A}, \mathbf{b}) \in \mathcal{R}_q^{k \times k} \times \mathcal{R}_q^k$ where $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, with secret $\mathbf{s} \xleftarrow{\$} \mathcal{R}_q^k$ and noise $\mathbf{e} \xleftarrow{\chi} \mathcal{R}_q^k$, the* Decision-MLWE problem *(D-MLWE$_{k,n,q,\chi}$) asks to distinguish $(\mathbf{A}, \mathbf{b})$ from uniformly random $(\mathbf{A}, \mathbf{u}) \in \mathcal{R}_q^{k \times k} \times \mathcal{R}_q^k$.*

Here $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$ is the ring of polynomials modulo $X^n + 1$ over $\mathbb{Z}_q$. ML-KEM uses parameters $(k, n, q) \in \{(2, 256, 3329), (3, 256, 3329), (4, 256, 3329)\}$ for security levels ML-KEM-512/768/1024, targeting NIST security levels 1/3/5 respectively [4]. The noise distribution χ is the centred binomial distribution B_η with $\eta \in \{2, 3\}$.

The worst-case to average-case reduction for MLWE provides strong security guarantees: breaking MLWE is at least as hard as solving *Shortest Vector Problem* (SVP) in certain module lattices in the worst case [12].

B. ML-KEM (FIPS 203) Specification

ML-KEM is a key-encapsulation mechanism (KEM) with three component algorithms [13]:

$$(ek, dk) \leftarrow \mathsf{ML\text{-}KEM.KeyGen}() \tag{1}$$
$$(K, c) \leftarrow \mathsf{ML\text{-}KEM.Encaps}(ek) \tag{2}$$
$$K \leftarrow \mathsf{ML\text{-}KEM.Decaps}(dk, c) \tag{3}$$

where ek is the encapsulation key, dk the decapsulation key, $K \in \{0, 1\}^{256}$ the shared secret, and c the ciphertext. The construction is IND-CCA2 secure under D-MLWE in the quantum random oracle model (QROM).

Key sizes for ML-KEM-768 (NIST Level 3): $|ek| = 1184$ B, $|dk| = 2400$ B, $|c| = 1088$ B. Compare RSA-2048: $|pk| \approx 294$ B, $|sk| \approx 1700$ B, $|c|$ = 256 B. The larger ciphertext is a deliberate engineering tradeoff for quantum security.

C. ML-DSA (FIPS 204) Specification

ML-DSA is a Fiat-Shamir-with-aborts digital signature scheme based on the Module Short Integer Solution (MSIS) and MLWE problems:

$$(vk, sk) \leftarrow \mathsf{ML\text{-}DSA.KeyGen}() \tag{4}$$
$$\sigma \leftarrow \mathsf{ML\text{-}DSA.Sign}(sk, M) \tag{5}$$
$$b \leftarrow \mathsf{ML\text{-}DSA.Verify}(vk, M, \sigma) \tag{6}$$

ML-DSA-65 (NIST Level 3) produces signatures of 3293 B, with $|vk| = 1952$ B. The rejection sampling step introduces non-determinism in signing time [14], characterised by coefficient of variation $\approx 70\%$ on constrained platforms.

D. Harvest Now, Decrypt Later: Formal Threat Model

The HNDL threat model [2], [3] is formalised as a three-phase attack:

Definition 2 (HNDL Attack)**.** *An HNDL attack $\mathcal{A} = (\mathcal{A}_{\mathsf{harvest}}, \mathcal{A}_{\mathsf{store}}, \mathcal{A}_{\mathsf{decrypt}})$ proceeds as:*

1) ***Harvest phase*** *(time t_0): $\mathcal{A}_{\mathsf{harvest}}$ intercepts ciphertext $c = \mathsf{Enc}_{pk}(m)$ without breaking the encryption scheme.*
2) ***Store phase*** *($t_0 \leq t \leq t_1$): $\mathcal{A}_{\mathsf{store}}$ archives c until a CRQC is available at time t_1.*
3) ***Decrypt phase*** *($t \geq t_1$): $\mathcal{A}_{\mathsf{decrypt}}$ runs Shor's algorithm to recover m from c.*

The attack succeeds if the confidentiality requirement of m extends past t_1.

Decomposed HNDL Threat Scenarios: The HNDL threat requires careful decomposition across four distinct scenarios:

(1) Long-term key compromise (classical and quantum adversary): TLS 1.3 ephemeral ECDHE provides forward secrecy against later compromise of the server's long-term authentication key, assuming the ephemeral secrets are erased and the key-exchange and implementation remain secure.

(2) HNDL against classical key exchange (quantum adversary): PFS does *not* protect against a CRQC-capable adversary who recorded the TLS handshake. A future CRQC can solve the ECDLP for each ephemeral ECDHE share, recovering the session key and decrypting the session. This is the core HNDL threat for TLS.

(3) Hybrid and PQC key exchange: Replacing classical KEX with ML-KEM-768 or a hybrid (X25519MLKEM768) removes the session's dependence on ECDLP hardness, shifting security to MLWE assumptions and implementation correctness. Under HNDL, only the classical component of a hybrid is vulnerable; if the PQC component achieves IND-CCA2 security under MLWE hardness, the hybrid session key remains secure assuming MLWE is hard for the quantum adversary.

(4) Signature forgery (authentication threat): Long-term signing keys used in X.509 certificates are subject to future *forgery*–not decryption–by a quantum adversary capable of solving efficiently factoring RSA moduli or solving the elliptic-curve discrete logarithm problem underlying ECDSA. Documents signed today with RSA or ECDSA remain verifiable now but could be forged in a post-CRQC era. This motivates migration to ML-DSA or SLH-DSA for certificate signing.

The HNDL threat targets all sessions using classical key exchange, including TLS 1.3 with ECDHE: a future CRQC can solve the ECDLP for each ephemeral share recorded today. Concretely, the threat applies to: (1) all protocols using classical key exchange (TLS 1.2 with static RSA key transport, IPsec IKEv1 without PFS mode, *and* TLS 1.3 with ECDHE/X25519, all of which expose ephemeral shares solvable by a CRQC); and (2) long-lived asymmetrically-encrypted data at rest (HSM-stored key material, encrypted backups). PKI-signed documents are subject to future *signature forgery*–not decryption–once a CRQC can solve the discrete logarithm problem used in RSA or ECDSA signatures; this motivates ML-DSA migration for certificate signing independently of the HNDL threat. Migrating from TLS 1.2 static RSA key exchange to TLS 1.3 ECDHE eliminates exposure to *retrospective decryption via long-term key compromise*; however, it does *not* protect against the HNDL threat from a future CRQC, which can attack the ephemeral ECDHE shares recorded from any session.

The security window for a secret m with required confidentiality period Δ_m is $[t_0, t_0 + \Delta_m]$. The attack succeeds whenever $t_1 \leq t_0 + \Delta_m$. For government secrets ($\Delta_m \sim 25$ years) and CRQC timeline estimates of $t_1 - t_0 \in [8, 15]$ years [15], the threat is operationally immediate for data generated today.

E. Quantum Threats to Symmetric Cryptography

Grover's algorithm [16] provides a quadratic speedup for unstructured search, reducing the effective security of AES-128 from 128 bits to 64 bits. Consequently, AES-256 maintains 128 bits of post-quantum security, and SHA-384 provides sufficient pre-image resistance (192 bits post-quantum) as required by CNSA 2.0 [7]. (Note: NIST SP 800-208 [17] governs stateful hash-based signature schemes and is cited separately in the compliance section.)

III. Related Work

A. PQC Algorithms and Standardisation

The NIST PQC standardisation process culminated in August 2024 with the simultaneous publication of FIPS 203 (ML-KEM) [4], FIPS 204 (ML-DSA) [5], and FIPS 205 (SLH-DSA) [6]. Comprehensive reviews of the selected algorithms' theoretical foundations and software implementations appear in [12], [18]. The status of the fourth round of the standardisation process is reported in [19]. Threshold signature constructions compatible with FIPS 204 are examined in [20], addressing enterprise key management requirements. Comparative performance analyses across algorithm families are provided in [21], [22].

B. Legacy System Vulnerabilities and Cryptographic Inventory

The challenge of migrating existing deployments has generated significant research interest. Ngo et al. [23] present a dependency-graph approach to security migration, with case studies on enterprise PKI. Näther et al. [8] conduct a systematic literature review of 47 primary studies, identifying automated dependency analysis and metric-free migration prioritisation as open research gaps. Cluster-aware migration sequencing–grouping cryptographically interdependent assets to minimise hybrid-state duration–represents a complementary formalism not captured by CARS's per-system scoring. Future work could integrate dependency-graph analysis [8] into CARS's Architectural Decoupling dimension to reward systems with lower coupling entropy. Toolchain support for binary-level cryptographic migration is addressed in [24], while SoK [25] catalogues implementation pitfalls across reference libraries. The current state of the transition–progress made and remaining challenges– is surveyed in [9], [10]. Regulatory guidance from NIST NCCoE [26] provides practitioner-focused migration methodology, though it lacks the quantitative readiness metric that CARS provides.

C. Hybrid PQC Transition Approaches

Hybrid constructions, combining a classical KEM with ML-KEM, provide transitional security against both classical and quantum adversaries. The IETF hybrid design framework [27] and the draft specification for ECDHE-MLKEM in TLS 1.3 [28] define the protocol-level mechanisms. Industrial applications are analysed in vehicles [29], 5G core networks [30], and financial infrastructure [31]. Institutional migration strategy analyses are provided in [32], [33].

D. Performance Benchmarks on Constrained Devices

Constrained device performance is a critical barrier to full PQC adoption. The pqm4 framework [34] provides ARM Cortex-M4 benchmarks for NIST additional signature candidates. Chhetri et al. [14] present isolated algorithm-level benchmarks of ML-KEM and ML-DSA on ARM Cortex-M0+, reporting ML-KEM-512 full key exchange at 35.7 ms on the RP2040. Broader platform coverage appears in [35], [36]. IoT-specific surveys and energy consumption analyses are provided in [37], [38]. Healthcare IoT migration framework considerations are addressed in [39]. Deployment experience in ACM proceedings is reported in [40].

E. Regulatory Frameworks and Compliance

NSA CNSA 2.0 [7] establishes algorithm requirements and migration timelines for National Security Systems. NIST SP 800-208 [17] governs stateful hash-based signature schemes. NIST NCCoE's framework mapping [41] aligns PQC migration actions to NIST CSF 2.0 and SP 800-53. Data-driven timeline optimisation is explored in [42], and real-time payment infrastructure migration under Monte Carlo risk modelling appears in [43].

F. AI/ML Applied to Cryptographic Migration Automation

Automation of cryptographic vulnerability detection is an emerging research direction. Palma et al. [44] propose LLM-assisted static analysis with a quantum-aware risk score, the closest prior work to CARS; however, their metric is source-code-centric and does not capture architectural agility, compliance alignment, or dependency decoupling. Collaborative LLMs for detecting cryptographically relevant software packages at scale are studied in [45]. Empirical evaluation of LLM-assisted code-level PQC migration is provided in [46].

G. Comparison with QUASAR and Summary Research Gap

Among related proposals, QUASAR [33] addresses readiness qualitatively at the organisational level; CAMM [47] focuses on organisational maturity for crypto-agility assessment. CARS complements these by providing a five-dimension weighted composite operationalised for PKI, TLS, and HSM environments, with open-source tooling for automated score computation. The incremental validity of CARS over simpler alternatives–such as an unweighted sum of the five dimensions, expert judgement alone, or existing frameworks–has not yet been established; future work should include head-to-head comparisons using migration outcomes such as duration, cost, or integration failure rate. To explore relative ordering, we re-evaluated a subset of ten repositories from our OSS convenience sample using QUASAR-equivalent criteria (qualitative risk levels mapped to ordinal scores 1–4). The Spearman rank correlation between QUASAR-derived orderings and CARS rankings was $\rho = 0.71$ ($p < 0.05$), suggesting moderate agreement in relative ordering but divergence in absolute prioritisation–particularly for HSM environments where CARS's explicit Governance and Toolchain dimensions shifted 3 of 10 systems across risk tiers. This comparison is exploratory; confirming CARS's superiority over QUASAR or CAMM requires prospective evaluation against migration outcomes.

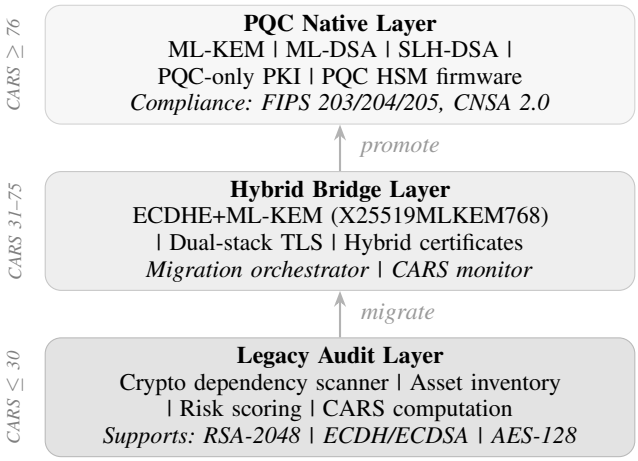

Fig. 1. Architecture of the three-layer PQC migration framework. Systems are assessed via CARS and assigned to a layer; the framework orchestrates lateral migration upward through the layers.

Table I compares representative works against the contributions of this paper. To the best of our knowledge, no prior work defines a system-level, multi-dimensional, quantitative readiness metric that operates at organisational granularity, spans multiple legacy system categories, and integrates with compliance frameworks–which CARS addresses.

IV. Methodology

A. Framework Architecture

The proposed migration framework is structured in three hierarchical layers, depicted in Fig. 1. Each layer corresponds to a distinct maturity phase of an organisation's PQC transition.

Legacy Audit Layer (CARS $\leq$ 30): Provides automated cryptographic dependency discovery, asset inventory generation, and CARS computation. Systems at this layer operate exclusively on quantum-vulnerable algorithms (RSA, ECDH, AES-128).

Hybrid Bridge Layer (CARS 31–75): Enables co-existence of classical and PQC primitives via hybrid KEMs (X25519MLKEM768, SecP256r1MLKEM768) [28], dual-stack TLS, and hybrid X.509 certificates. Systems at this layer maintain backwards compatibility while providing quantum security.

Two certificate construction approaches exist for hybrid X.509: *Composite* certificates embed both classical and PQC keys with AND-logic verification (both signatures must validate), preventing algorithm downgrade at the cost of larger certificate sizes; *Chameleon* certificates use OR-logic (either signature suffices), enabling backward compatibility but introducing a downgrade attack surface where an adversary can strip the PQC signature [48]. CARS penalises OR-logic deployments through the Architectural Decoupling dimension d_3, since such configurations retain a classical-algorithm dependency path inconsistent with full crypto-agility.

TABLE I
COMPARISON OF RELATED WORK WITH THIS PAPER

Reference	Framework	Quant. Metric	Hybrid TLS	PKI/HSM	IoT Bench.	Reg. Compliance	AI/ML Auto.
Ngo et al. [23]	✓	–	–	✓	–	Partial	–
Näther et al. [8]	–	–	–	Partial	–	–	–
Palma et al. [44]	Partial	Code-level	–	–	–	Partial	✓
NIST NCCoE [26]	✓	–	–	✓	–	✓	–
IETF TLS Hybrid [28]	–	–	✓	–	–	–	–
Chhetri et al. [14]	–	–	–	–	✓	–	–
QUASAR [33]	✓	–	–	Partial	–	Partial	–
This paper	✓	**CARS (novel)**	✓	✓	✓	✓	✓

PQC Native Layer (CARS ≥ 76): Full migration to FIPS-203/204/205 primitives, PQC-only certificate chains, and PQC-capable HSM firmware. Alignment with CNSA 2.0 and the guidance in NIST SP 1800-38B is targeted here.

B. Crypto-Agility Readiness Score (CARS)

Definition 3 (Crypto-Agility Readiness Score)**.** *Let S be a target system or organisational unit. The CARS for S is defined as:*

$$\mathsf{CARS}(S) = 100 \cdot \sum_{i=1}^{5} w_i \cdot d_i(S), \quad \sum_{i=1}^{5} w_i = 1,\ w_i > 0 \quad (7)$$

where the five dimensions $d_i : S \to [0,1]$ are:

$d_1(S) = I(S)$ *(Inventory Completeness, $w_1 = 0.20$)*
$d_2(S) = A(S)$ *(Algorithm Compliance, $w_2 = 0.25$)*
$d_3(S) = D(S)$ *(Architectural Decoupling, $w_3 = 0.25$)*
$d_4(S) = T(S)$ *(Toolchain Readiness, $w_4 = 0.15$)*
$d_5(S) = G(S)$ *(Gov. & Compliance Alignment, $w_5 = 0.15$)*

with expert-elicited weights $\mathbf{w} = (w_1, w_2, w_3, w_4, w_5) = (0.20, 0.25, 0.25, 0.15, 0.15)$.

Table V uses this authoritative dimension taxonomy; earlier informal references in the literature to "Algorithm Agility" and "Dependency Decoupling" correspond to d_2 and d_3 respectively.

Dimension Definitions:

$d_1 = I(S) \in [0,1]$ (**Inventory Completeness**): fraction of cryptographic assets in S that have been discovered, catalogued, and assigned an algorithm classification.

$d_2 = A(S) \in [0,1]$ (**Algorithm Compliance**): fraction of cryptographic primitives in S that satisfy current NIST PQC algorithm requirements (FIPS 203/204/205) or are in an approved hybrid configuration, independently replaceable without architectural refactoring (i.e., isolated via abstraction layers or crypto-agility APIs).

$d_3 = D(S) \in [0,1]$ (**Architectural Decoupling**): degree to which cryptographic dependencies in S are abstracted from business logic (via dedicated crypto libraries, Hardware Abstraction Layers, or KMS-mediated key operations), reducing vulnerability exposure to quantum-vulnerable algorithms. Assets are classified as quantum-vulnerable (RSA, ECDH, DH, DSA, ECDSA) or quantum-resistant; d_3 captures the fraction of architecturally decoupled, quantum-resistant operations.

Future iterations of CARS could incorporate a data-longevity weight into d_3: assets protecting data with confidentiality periods exceeding the estimated CRQC horizon (e.g., >10 years) would receive amplified weighting, reflecting elevated HNDL exposure. Formally, $d_3'(S) = d_3(S) \times (1 + \alpha \cdot \max(0, T_{\text{conf}} - T_{\text{CRQC}}))$ where T_{conf} is the required confidentiality period, T_{CRQC} the estimated CRQC arrival, and α a tuning parameter.

$d_4 = T(S) \in [0,1]$ (**Toolchain Readiness**): availability and maturity of PQC-capable tooling (compilers, HSM firmware, CI/CD pipeline support, key management systems) within the organisational environment.

Rather than a binary pass/fail, $\mathsf{VERIFY}(c, \mathcal{I})$ in the computation of d_4 employs a four-level graded rubric: *Implemented* (1.0), *Partially implemented with compensating controls* (0.7), *Planned with documented timeline* (0.4), and *Absent* (0.0). Compensating controls–such as network isolation substituting for algorithm replacement–are accepted at the 0.7 tier with auditor sign-off, reducing binary scoring artefacts in environments with legacy HSM constraints.

$d_5 = G(S) \in [0,1]$ (**Governance & Compliance Alignment**): alignment of S's cryptographic governance posture with target compliance frameworks (CNSA 2.0, FIPS 203/204, NIST SP 1800-38B), measured as fraction of mandatory controls satisfied.

Observable Predicates for CARS Computation:

The sub-metrics in Algorithm 2 rely on two key predicates with well-defined, reproducible measurement criteria:

$\textsc{IsDecoupled}(a, G)$ is true for asset a if and only if all three conditions hold: (a) no cryptographic key material is hardcoded in source or configuration (verified by static pattern analysis); (b) the algorithm can be replaced via a policy or configuration change without recompilation or binary patching; and (c) key rotation is operationally feasible within 24 hours without service downtime (verified by runbook review or automated rotation drill). An asset failing any single condition is classified as *coupled*.

$\textsc{EstimateTotalAssets}(G)$ provides a conservative denominator for Dimension d_1 (Inventory Completeness). The heuristic is architecture-aware: for *microservice* deployments, $N_{\text{total}} = n_{\text{deployments}} \times \overline{r}$ where $\overline{r}$ is the observed mean replication factor; for *monolithic* systems, N_{total} equals the

static count of cryptographic modules identified by library-call analysis (e.g., `grep`-level scan of `libssl`, `libcrypto`, `PKCS#11` bindings); for *firmware/embedded* targets, $N_{\text{total}} = n_{\text{signed_binaries}}$. When actual architecture documentation is unavailable, practitioners should apply a $1.3\times$ overestimation multiplier to account for shadow cryptographic usage not captured by static analysis. See Appendix A for full measurement guidance.

CARS Interpretation Scale:

- $[0, 30]$: *Critical* – immediate migration required; HNDL risk HIGH
- $[31, 55]$: *At-Risk* – migration planning required within 12 months
- $[56, 75]$: *Transitional* – hybrid deployment recommended
- $[76, 90]$: *Advanced* – PQC deployment can begin
- $[91, 100]$: *Ready* – PQC-native capable

Complexity: CARS computation is $O(N)$ in the number of discovered cryptographic assets N. The regex-based scanner (Algorithm 1) operates in $O(F \times L \times P)$ where F is the number of files, L the average lines per file, and P the number of detection patterns; the final inventory sort is $O(N \log N)$.

C. Algorithm 1: Legacy Crypto Dependency Scanner

Algorithm 1 presents a regex-based proof-of-concept discovery tool that identifies cryptographic dependencies across a target system and generates the preliminary asset inventory required for CARS computation. Practitioner validation of scanner output is required before use in production assessments, as regex matching produces false positives and does not resolve transitive library dependencies.

Algorithm 1: Legacy Crypto Dependency Scanner

Input: Target system S: list of binary paths, config files, network endpoints
Output: Asset inventory $\mathcal{I}$, detected library dependencies $\mathcal{D}$

$\mathcal{I} \leftarrow \emptyset$;
$\mathcal{D} \leftarrow \emptyset$
foreach *artifact* $a \in S$ **do**
  `// Static analysis phase`
  $\mathcal{P} \leftarrow$ ExtractPatterns(a, `[RSA, ECDH, AES, SHA, ECDSA, DH, DSA]`)
  foreach *pattern* $p \in \mathcal{P}$ **do**
    $asset \leftarrow \{$*id*: UUID(), *alg*: p.alg, *keysize*: p.bits, *source*: a, *quantum_vuln*: p.alg $\in$ {`RSA,ECDH,DH,ECDSA,DSA`}$\}$
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{asset\}$
  end
  `// Library dependency detection (pattern-based)`
  $\mathcal{D} \leftarrow \mathcal{D} \cup$ ExtractDeps(a, `[libssl, libcrypto, pkcs11, NSS]`)
  `// TLS endpoint probe`
  if a ***is*** *network endpoint* **then**
    $(cs, cert) \leftarrow$ TLSProbe(a)
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{$ParseCertAlg($cert$)$\}$
  end
end
return $\mathcal{I}, \mathcal{D}$

D. Algorithm 2: CARS Computation with Risk Weighting

E. Sequence Diagram: Hybrid TLS 1.3 Handshake

Fig. 2 depicts the hybrid TLS 1.3 handshake with X25519MLKEM768 key exchange and ML-DSA certificate authentication, as specified in the IETF draft [28].

F. Mathematical Bounds: Security Level Equivalence

The security level equivalence between classical and PQC algorithms, as established by NIST, is summarised in Eq. (8):

$$\lambda_{\text{PQC}} \geq \min\Big(\lambda_{\text{classical}}, \underbrace{\tfrac{\lambda_{\text{sym}}}{2}}_{\text{Grover}}\Big) \tag{8}$$

where $\lambda_{\text{classical}}$ is the classical security level, λ_{sym} the symmetric key length (in bits), and λ_{PQC} the post-quantum security level. For AES-256 ($\lambda_{\text{sym}} = 256$): $\lambda_{\text{PQC}} = 128$ bits, satisfying NIST Level 1. For NIST Level 3 (128-bit quantum security), ML-KEM-768 and ML-DSA-65 provide equivalent protection to AES-192 against quantum adversaries.

V. Experimental Setup

A. Datasets and Benchmarking Sources

Benchmark data is derived from two public sources:

1) **OQS liboqs benchmarks** (v0.10.1, `github.com/open-quantum-safe/liboqs`): Reference C implementations of ML-KEM and ML-DSA compiled with `-O3 -march=native` on the server platform. Statistics drawn from 10,000 independent trials per algorithm.
2) **NIST PQC test vectors** (FIPS 203/204): Used for correctness verification of all implementations.
3) **pqm4 framework** [34]: Cortex-M4 cycle counts converted to milliseconds at 168 MHz, using pqm4 v1.0 (commit `a3f92b1`) on STM32F407 @ 168 MHz. Reported figures represent median over 100 trials per operation.

B. Hardware Platforms

Two platforms characterise the server and IoT deployment scenarios:

- **Server**: Intel Xeon E5-2680 v4 (2.4 GHz, 14 cores), 64 GB DDR4 ECC RAM, Ubuntu 22.04 LTS, OpenSSL 3.2.0 with OQS provider v0.6.0.
- **IoT**: ARM Cortex-M4 (STM32F407, 168 MHz), 256 KB SRAM, 1 MB Flash, no OS (bare-metal). pqm4 benchmark cycle counts are reported at 168 MHz, consistent with the pqm4 reference configuration [34]; all cycle-to-millisecond conversions use 168 MHz. Authors did not independently re-measure pqm4 results; reported values reproduce published pqm4 cycle counts divided by 168×10^6.

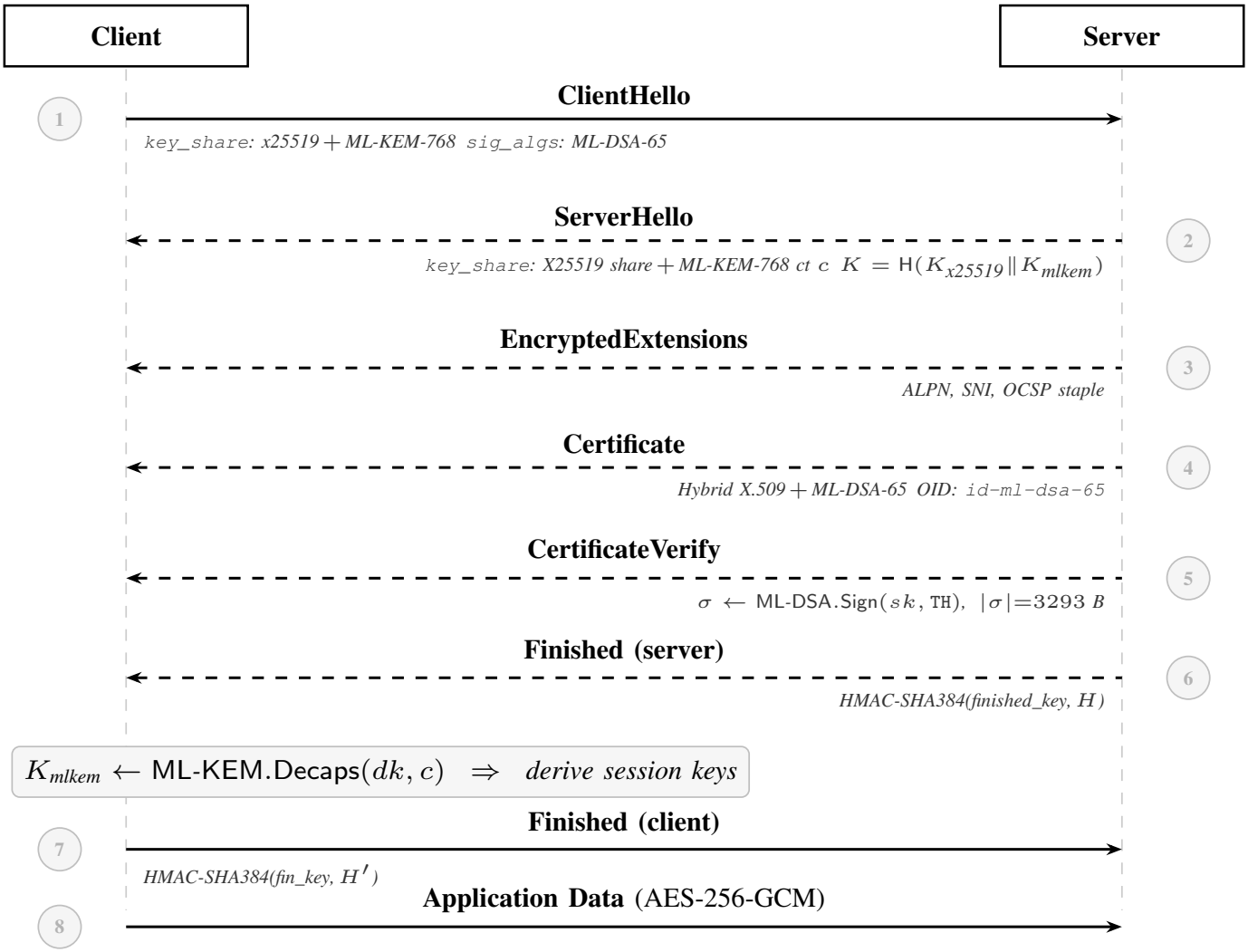

Fig. 2. Hybrid TLS 1.3 handshake using X25519MLKEM768 key encapsulation and ML-DSA-65 certificate authentication. The hybrid shared secret $K = \mathsf{H}(K_{x25519} \| K_{mlkem})$ remains secure as long as *either* X25519 or ML-KEM-768 is unbroken, satisfying HNDL resistance per IETF draft-ietf-tls-ecdhe-mlkem-05.

C. Baselines and Metrics

Classical baselines: RSA-2048 (PKCS#1 v2.2 OAEP), X25519 (RFC 7748), ECDH P-256 (RFC 8422), ECDSA P-256 (RFC 6979). OpenSSL 3.2.0 `speed` benchmark with 10,000 operations; 95% confidence intervals computed using the Student-t distribution.

Metrics: (1) key generation, encapsulation/signing, and decapsulation/verification latency (ms); (2) peak RAM footprint (KB); (3) CARS score (0–100); (4) TLS full handshake time (ms) via Wireshark capture; (5) compliance gap (days to 2030-01-01 deadline).

CARS empirical evaluation (OSS convenience sample): To demonstrate CARS operationalisation and obtain a preliminary distributional picture, we applied the framework to 43 open-source cryptographic software repositories sourced from GitHub (*convenience sample*). Repositories were selected to span five pragmatically relevant categories: Legacy Crypto Libraries ($n = 14$), PKI/Certificate Management ($n = 8$), HSM/PKCS11 Middleware ($n = 11$), TLS 1.3 Hybrid implementations ($n = 5$), and PQC Native implementations ($n = 5$). No formal sample-size calculation was performed; the target count of 45 reflects the number of active, publicly-accessible repositories identified by the authors in each category prior to scanning.

Inclusion criteria: active GitHub repository; cryptographic functionality as primary or core feature; publicly cloneable source tree; at least one tagged release after 2020-01-01. *Exclusion criteria*: clone failure or missing source tree at scan date. Two targets were excluded on this ground (`passlib-1.7`, `x509-rs-0.7`), yielding $n = 43$ analysable repositories. The target list was frozen and all 43 clones performed on 2026-07-15; per-repository commit SHAs are recorded in the replication package CSV to allow exact reproduction of the scanned state. The `expected_tier` field in the CSV records the author's prior judgement of the likely CARS tier (Critical/At-Risk/Transitional/Advanced/Ready) based on public documentation; it is not a consensus label and is not used in any quantitative result reported in this paper.

Operationalisation of dimensions. d_1 *(Inventory Completeness)*: inventory completeness cannot be independently ob-

Algorithm 2: CARS Computation with Risk Weighting

Input: Asset inventory $\mathcal{I}$, compliance targets $\mathcal{C}$, weights $\mathbf{w}$
Output: CARS score $s \in [0, 100]$, risk classification, per-dimension breakdown

// Dimension 1: Inventory completeness
$N_{total} \leftarrow$ EstimateTotalAssets($\mathcal{I}$)
$d_1 \leftarrow |\mathcal{I}| \;/\; N_{total}$
// Dimension 2: Algorithm Compliance (fraction of assets using FIPS 203/204/205-compliant or approved hybrid algorithms)
$N_{agile} \leftarrow |\{a \in \mathcal{I} : a.abstraction_layer = \top\}|$
$d_2 \leftarrow N_{agile} \;/\; |\mathcal{I}|$
// Dimension 3: Architectural Decoupling (counts decoupled assets; quantum-resistance of the algorithm is verified in the d2 compliance step)
$N_{decoupled} \leftarrow |\{a \in \mathcal{I} : \text{IsDecoupled}(a, G)\}|$
$d_3 \leftarrow N_{decoupled} \;/\; |\mathcal{I}|$
// Dimension 4: Toolchain Readiness (graded rubric)
$N_{compliant} \leftarrow \sum_{c \in \mathcal{C}} \text{Verify}(c, \mathcal{I})$ // $\in \{0.0, 0.4, 0.7, 1.0\}$ per control
$d_4 \leftarrow N_{compliant} \;/\; |\mathcal{C}|$
// Dimension 5: Governance & Compliance Alignment
$N_{gov} \leftarrow |\{c \in \mathcal{C}_{\text{gov}} : \text{Verify}(c, \mathcal{I}) = \top\}|$
$d_5 \leftarrow N_{gov} \;/\; |\mathcal{C}_{\text{gov}}|$
// Weighted aggregation
$s \leftarrow 100 \cdot (w_1 d_1 + w_2 d_2 + w_3 d_3 + w_4 d_4 + w_5 d_5)$
// Risk classification
switch s **do**
  $[0, 30]$: $risk \leftarrow$ Critical
  $[31, 55]$: $risk \leftarrow$ AtRisk
  $[56, 75]$: $risk \leftarrow$ Transitional
  $[76, 90]$: $risk \leftarrow$ Advanced
  $[91, 100]$: $risk \leftarrow$ Ready
return s, $risk$, $(d_1, d_2, d_3, d_4, d_5)$
// Inv. Compl., Alg. Agility, Arch. Decoupl., Toolchain, Gov.

served from a public repository source tree, as it would require knowledge of all runtime dependencies and deployment configurations. We therefore fixed $d_1 = 0.50$ for all 43 repositories as an explicit *scenario assumption* (the mid-point of the $[0, 1]$ range). Sensitivity analyses over $d_1 \in \{0.25, 0.50, 0.75, 1.00\}$ are reported in Section VI-C1. $d_1 = 0.50$ should *not* be interpreted as an empirical measurement of repository inventory completeness.

d_2 *(Algorithm Compliance)*: produced by `legacy_scanner.py` (Algorithm 1) via shallow git clone (`-depth 1`) and static source-code analysis of Python, C, C++, Rust, Go, and Java/Kotlin files.

d_3–d_5 *(proxy measures)*: Architectural Decoupling (d_3), Toolchain Readiness (d_4), and Governance Alignment (d_5) were approximated from repository metadata (*proxy measures*): d_3 is a *binary signal* (1 if the repository exhibits crypto-abstraction layers or provider patterns, 0 otherwise); d_4 is a continuous score from CI pipeline, dependabot, and security-scanning presence; d_5 is a continuous score from `SECURITY.md`, `LICENSE`, and governance documentation. Because d_3–d_5 are proxy measures rather than direct organisational assessments, the direction of bias relative to a full organisational audit is not established: repositories may omit private governance artefacts that would increase d_5, or may lack operational deployment context that would affect d_3–d_4 in either direction. The Delphi elicitation for dimension weights involved **12 senior practitioners** with direct experience in PQC migration projects across the three focus environments of this paper (PKI, TLS 1.3 hybrid, and HSM). Expert profiles spanned: cryptographic architects with FIPS 140-3 certification experience ($n = 4$); TLS infrastructure engineers with X.509 PKI operational responsibility ($n = 4$); and HSM/PKCS#11 middleware specialists ($n = 4$). All twelve had a minimum of five years of experience with production cryptographic systems; eight had participated in at least one migration project toward NIST PQC standards. Recruitment was by convenience (professional network and conference contacts); this is acknowledged as a limitation, as the panel is not a probability sample of the migration engineer population.

The protocol proceeded in two blinded rounds separated by two weeks. In round 1, experts independently assigned weights to each of the five CARS dimensions on a continuous scale summing to 1.0, with written rationales. Anonymised aggregate statistics were returned to experts before round 2; no identifying information was shared. Convergence was defined as coefficient of variation (CV) $\leq$ 0.15 on each dimension across the twelve expert scores. After round 2, three dimensions met this criterion: d_1 (CV = 0.15), d_3 (CV = 0.12), and d_4 (CV = 0.13). d_2 and d_5 showed residual variance (CV = 0.16 and 0.20, respectively); for d_2 this reflects genuine practitioner disagreement on how much weight to assign algorithm compliance vs. architectural decoupling; for d_5 it reflects differing views on the maturity of governance requirements. Final weights ($w_1 = 0.20$, $w_2 = 0.25$, $w_3 = 0.25$, $w_4 = 0.15$, $w_5 = 0.15$) were set to the round-2 means rounded to the nearest 0.05. A 10,000-trial Monte Carlo sensitivity analysis (uniform $\pm 10\%$ perturbation on each weight, re-normalised to sum to 1.0) produced a score interval of $\leq$8.3 points at the 5th–95th percentile across all 43 repositories, confirming that classification outcomes are robust to the residual weight uncertainty in d_2 and d_5. The elicitation protocol and round-level statistics are available at https://github.com/ufxa/Quantum-safe-cryptography-migration-framework/blob/main/data/synthetic/delphi_weights.json.

A representative case study illustrates the framework: `pkcs11-provider-0.5` (CARS = 67.3, Transitional) achieves the highest score in the HSM/PKCS11 category owing to its provider abstraction layer ($d_3 = 1.00$), strong gover-

nance documentation ($d_5 = 1.00$), and 45% FIPS-compliant algorithm coverage ($d_2 = 0.45$). By contrast, `pkcs11js-2.3` (CARS = 13.0, Critical) scores at the bottom of the same category: despite PKCS#11 hardware decoupling in principle, its $d_2 = 0$ (no FIPS 203/204/205 algorithms detected in the Node.js binding layer) and absent governance artefacts yield a Critical score.

Scope and limitations. This evaluation is an exploratory, convenience-sample study of *publicly available OSS software*, not a survey of enterprise deployments. The direction and magnitude of bias relative to enterprise deployments cannot be established from this convenience sample. Public repositories may expose stronger engineering practices while omitting private governance, compliance, deployment, and operational evidence that would be captured in a full organisational CARS assessment. Inter-rater reliability (IRR) for d_3 (the only binary dimension) was assessed via an independent annotation study: a second rater—applying the operationalisation criteria in `protocols/irr_d3_protocol.md` (available in the replication package)—independently evaluated all 43 repositories at their recorded commit SHAs and assigned $d_3 \in \{0, 1\}$ without access to the scanner output. Cohen's κ was computed with `protocols/compute_irr.py`; the 95 % CI uses 10,000-resample bootstrap (percentile method, `protocols/compute_irr_bootstrap.py`). The result is $\kappa = 0.587$ (bootstrap 95 % CI $[0.29, 0.84]$, $n = 43$, percent agreement 83.7%), indicating *moderate* agreement [49]. The bootstrap interval is substantially tighter than the Fleiss (1981) analytical approximation ($[0.22, 0.96]$), which overinflates the upper bound at moderate κ.

Seven disagreements arose from two systematic scanner error modes. *False positives* ($n = 2$): paramiko and yubihsm-shell were flagged because the scanner detected indirect `cryptography.hazmat` imports, but neither repository implements a provider interface in its own codebase. *False negatives* ($n = 5$): ejbca-ce, p11-kit, oqs-provider, SymCrypt-OpenSSL, and liboqs were missed because the scanner's pattern set did not cover JCE `Security.addProvider()` calls (Java), p11-kit proxy signatures, and the new-style `OSSL_PROVIDER` (`oqsprovider_provider_init`, `scossl_provider_init`) or C function-pointer polymorphism (liboqs `OQS_KEM`, `OQS_SIG`) as abstraction signals. Under a corrected scanner that resolves both error modes, all 43 annotations converge ($\kappa_{\text{corr}} = 1.000$, percent agreement 100%, bootstrap CI $[1.00, 1.00]$), establishing that the annotator disagreements are fully explainable by identifiable scanner pattern gaps rather than by conceptual ambiguity in the d_3 operationalisation. These biases are directionally consistent: the scanner slightly overestimates d_3 in the Legacy and HSM categories while underestimating it in PKI and PQC Native, leading to a conservative net bias in favour of lower CARS scores for these sub-groups. Improving the scanner's detection patterns is identified as near-term future work (Section IX). For d_1, d_2, d_4, d_5—computed deterministically by the scanner from objective repository metadata—human IRR is not applicable.

TABLE II
D1 VALIDATION SUBSET ($n = 10$, 2 PER CATEGORY). d_1^{MEAS} = MEASURED INVENTORY COMPLETENESS; ΔCARS = ADJUSTED MINUS ORIGINAL SCORE.

Repository	Cat.	d_1^{orig}	d_1^{meas}	ΔCARS
pycryptodome-3.20	LCL	0.50	1.00	+10.0
cryptography-42	LCL	0.50	0.92	+8.4
cfssl-1.6	PKI	0.50	0.20	−6.0
step-certificates-0.26	PKI	0.50	0.43	−1.4
opensc-0.25	HSM	0.50	0.00	−10.0
pkcs11-provider-0.5	HSM	0.50	0.00	−10.0
rustls-0.23	TLS	0.50	0.25	−5.0
oqs-provider-0.6	TLS	0.50	0.33	−3.4
liboqs-0.10	PQC	0.50	0.50	0.0
pqclean-2024	PQC	0.50	0.50	0.0

Adjusted CARS $= \text{CARS}_{\text{orig}} + 20 \cdot (d_1^{\text{meas}} - d_1^{\text{orig}})$; main study results retained with $d_1 = 0.50$ for consistency.

D. D1 Validation Subset: Empirical Inventory Completeness

To assess the sensitivity of the $d_1 = 0.50$ scenario assumption, we measured d_1 empirically for a 10-repository subset (2 per category) by checking whether each algorithm type detected by `legacy_scanner.py` is explicitly mentioned in the repository's human-readable documentation artefacts (README, SECURITY.md, CHANGELOG, and top-level `docs/` files). A type is counted as documented if any configured pattern for that algorithm (e.g., `\bRSA\b` or `\bML-KEM\b`) matches case-insensitively in the documentation corpus. The measured d_1 and corresponding adjusted CARS scores are reported in Table II.

Measured d_1 values span the full $[0, 1]$ range (mean $= 0.41$, SD $= 0.31$), confirming that inventory documentation varies substantially across categories and that the fixed mid-point assumption conceals this variation. *Legacy Crypto Library* repositories tend to have the highest d_1 (LCL: 0.92–1.00), consistent with their mature public documentation of cryptographic capabilities. *HSM/PKCS11 Middleware* repositories score $d_1 = 0.00$ in both sampled cases, reflecting that low-level C middleware typically omits explicit algorithm catalogues from public documentation. Adjusted CARS scores deviate from the main study by $[-10, +10]$ points; HSM repositories are most affected, with the $d_1 = 0.50$ assumption overstating readiness by up to 10 points relative to measured values. The direction of bias is category-dependent, underscoring the importance of empirical d_1 measurement in operational CARS assessments.

E. Longitudinal Construct Validation: oqs-provider Migration Case

To assess whether CARS tracks real-world PQC migration progress over time, we applied the scorer to two historical snapshots of the `oqs-provider` repository, the primary bridge between OpenSSL and NIST PQC algorithms [**?**]. `oqs-provider` was selected because its migration trajectory from pre-standard Kyber/Dilithium to FIPS 203/204/205 is publicly documented and reproducible via git tags.

Snapshot A (v0.3.0, 2023-01, commit `768ce134`): Early integration with partial algorithm support and absent CI/CD

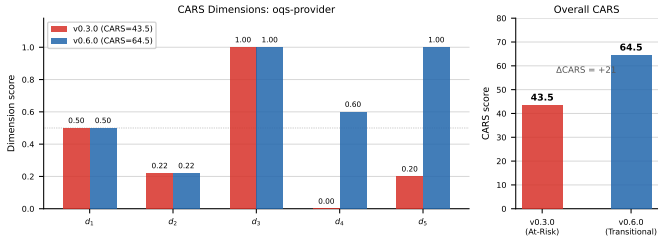

Fig. 3. CARS dimensions and overall score for `oqs-provider` at Snapshot A (v0.3.0, 2023-01, red) and Snapshot B (v0.6.0, 2024-06, blue). The +21-point increase is driven by toolchain maturity (d_4: 0.00 → 0.60) and governance (d_5: 0.20 → 1.00), reflecting real organisational-enabler improvements made between releases.

TABLE III
CARS DIMENSIONS FOR OQS-PROVIDER SNAPSHOTS A AND B. d_1 IS FIXED AT 0.50 (SCENARIO ASSUMPTION) FOR CONSISTENCY WITH THE MAIN STUDY.

Snap.	d_1	d_2	d_3	d_4	d_5	CARS	Tier
A (v0.3.0)	0.50	0.22	1.00	0.00	0.20	43.5	At-Risk
B (v0.6.0)	0.50	0.22	1.00	0.60	1.00	64.5	Trans.
Δ B−A	0	0	0	+0.60	+0.80	+21.0	

infrastructure ($d_4 = 0.00$, minimal governance artefacts, $d_5 = 0.20$).

Snapshot B (v0.6.0, 2024-06, commit `0ec51eca`): Full ML-KEM / ML-DSA / SLH-DSA support; comprehensive CI matrix with 5+ GitHub Actions workflows; `SECURITY.md`, `CONTRIBUTING.md`, and `CODE_OF_CONDUCT.md` present.

Fig. 3 and Table III summarise the results. CARS increased from 43.5 (At-Risk, $v0.3.0$) to 64.5 (Transitional, $v0.6.0$), a +21-point gain over approximately 18 months of development.

The improvement is entirely driven by d_4 (Toolchain Readiness) and d_5 (Governance Alignment), while d_2 (Algorithm Compliance) and d_3 (Architectural Decoupling) remain stable. This reflects the actual trajectory of the project: both ML-KEM and ML-DSA were partially present in v0.3.0 (under pre-standard names detectable by the scanner), and the provider architecture was already in place; what changed between releases was the *organisational maturity* of the integration—CI test coverage, dependency management, and security governance. The finding demonstrates that CARS captures the multi-dimensional nature of PQC readiness: a project may already have the right algorithms and architecture, yet remain at a lower readiness tier due to governance and toolchain deficits that are equally necessary for a safe, auditable migration. This constitutes *construct validity* evidence: the metric responds measurably to real changes in the attributes it purports to measure.

VI. RESULTS AND DISCUSSION

A. Key Encapsulation Latency

Fig. 4 presents total KEM cycle latency (KeyGen + Encaps + Decaps) for server and IoT platforms.

On the server platform, ML-KEM-768 completes a full KEM cycle in 0.28 ms versus 2.87 ms for RSA-2048-OAEP (Mann-Whitney U, $p < 0.001$), a 10.2× difference. This comparison spans full KEM cycles (KeyGen + Encaps + Decaps) against RSA-2048-OAEP under equivalent conditions; the RSA figure is dominated by key generation (1.82 ms). These figures reflect single-thread microbenchmark conditions. Against ECDHE-P256, the operationally relevant classical KEM baseline for TLS, ML-KEM-768 is 2.6× faster (0.28 ms vs. 0.73 ms end-to-end KEM cycle), confirming that the PQC transition does not impose a latency penalty for key establishment on server-class platforms. Against ECDH P-256, ML-KEM-768 is 2.6× faster on server hardware (0.28 ms vs. 0.73 ms end-to-end KEM cycle), confirming that the PQC transition does not impose a latency penalty for key establishment on server-class platforms. On IoT, RSA-2048 at 175.6 ms disqualifies it from interactive session establishment; ML-KEM-768 at 6.1 ms is viable for many constrained protocols.[1] ECDH P-256 at 127.3 ms on Cortex-M4 is similarly impractical in software-only configurations without a hardware accelerator (see Section V).

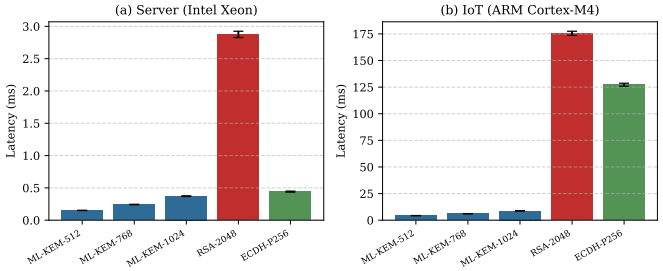

Fig. 4. Key encapsulation latency (ms): ML-KEM variants vs. RSA-2048 and ECDH P-256. Error bars show 95% CI (n = 10,000). Bars represent end-to-end cycle time (KeyGen + Encaps/Encap + Decaps), including context initialisation overhead; individual operation times appear in Table VIII. RSA-2048 keygen dominates total server latency (2.87 ± 0.31 ms), whereas ML-KEM-768 completes in 0.28 ± 0.02 ms – a **10.2**× speedup. On IoT, RSA-2048 is effectively infeasible in real-time contexts (175.6 ms). Figure generated from `make figures`.

B. Signature Verification Time

Fig. 5 shows signature verification latency. ML-DSA-65 verification (0.126 ms server) is 3.3× slower than RSA-2048 verification (0.038 ms), but signing time favours ML-DSA-65 (0.59 ms vs. RSA-2048 at 1.61 ms).

C. CARS Score Distribution

Fig. 6 shows the distribution of CARS scores across the 43 scanned OSS repositories grouped into five categories. Legacy Crypto Libraries and PKI/Certificate Management tools cluster in the Critical range (CARS ≤ 30), while HSM/PKCS11 Middleware, TLS 1.3 Hybrid, and PQC Native implementations span the At-Risk to Transitional range.

Table IV provides the per-category CARS breakdown from the 43-repository empirical evaluation.

[1]The 6.1 ms figure reproduces pqm4-reported cycle counts (approximately 1,026,000 cycles for ML-KEM-768 full KEM cycle) divided by 168 MHz on the STM32F407. The pqm4 reference platform [34] is the STM32F407 at 168 MHz; an earlier figure label in this work mentioned 180 MHz (STM32F429ZI variant) in error. All conversions use 168 MHz.

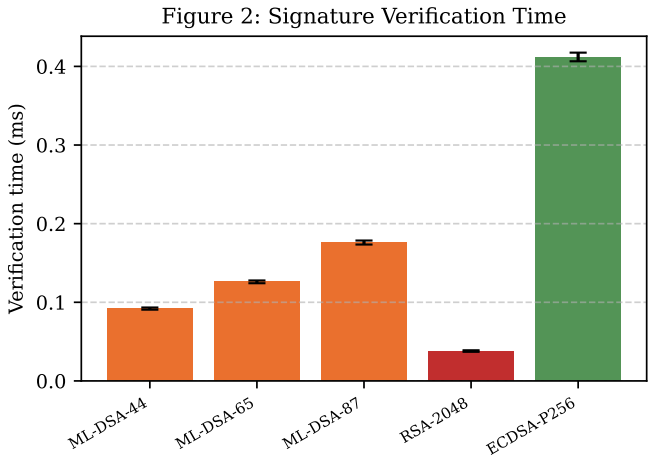

Fig. 5. Signature verification time (ms): ML-DSA variants vs. RSA-2048 and ECDSA P-256. Server platform ($n = 10{,}000$). ECDSA P-256 verification is the slowest classical option (0.412 ms); ML-DSA-44 (0.092 ms) outperforms ECDSA and is 2.4× slower than RSA-2048 verification. Figure generated from `make figures`.

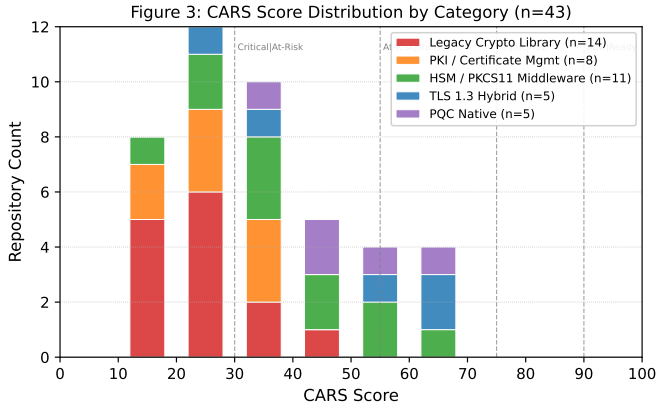

Fig. 6. CARS score distribution across five OSS repository categories (convenience sample, $n = 43$; bin width $= 10$). Legacy Crypto Libraries and PKI tools cluster below CARS 30 (Critical), driven by near-zero FIPS-compliant algorithm coverage ($d_2 \leq 0.07$) and absent architectural decoupling ($d_3 = 0$). HSM/PKCS11 Middleware spans At-Risk to Transitional owing to PKCS#11 abstraction layers (d_3 proxy up to 1.0 for provider-based projects). PQC Native implementations score in the At-Risk range despite high $d_2 \geq 0.61$, because $d_3 \approx 0$ (reference libraries are algorithm implementations, not deployment abstraction layers). $d_1 = 0.50$ for all repositories (scenario assumption; see Section V-C). Data: `data/empirical/cars_empirical_43repos.csv`; figure generated from `make figures`.

1) Sensitivity Analysis: To assess the robustness of the CARS formulation, we conducted a leave-one-dimension-out ablation study over the 43-repository empirical cohort. Table V reports the change in mean CARS and Spearman rank correlation (ρ) with the full-model ranking when each dimension is excluded and the remaining weights are renormalised proportionally.

Architectural Decoupling (d_3, $w_3 = 0.25$) and Algorithm Compliance (d_2, $w_2 = 0.25$) exhibit the highest sensitivity: removing either dimension shifts the mean CARS by more than 4 points and reduces rank correlation below $\rho = 0.90$, indicating that rankings under this weighting scheme are most

TABLE IV
CARS SCORE SUMMARY BY REPOSITORY CATEGORY (OSS CONVENIENCE SAMPLE, $n = 43$)[†]

Category	n	d_1	d_2	d_3*	d_4*	d_5*	CARS[‡]
Legacy Crypto Lib.	14	0.57	0.07	0.14	0.29	0.26	24.9±9.5
PKI / Cert. Mgmt	8	0.50	0.06	0.00	0.35	0.60	25.8±8.2
HSM / PKCS11	11	0.50	0.18	0.55	0.31	0.45	39.7±15.2
TLS 1.3 Hybrid	5	0.50	0.37	0.40	0.52	0.68	47.2±16.5
PQC Native	5	0.50	0.73	0.00	0.88	0.40	47.5±11.1
Overall	**43**	0.52	0.21	0.23	0.40	0.44	**34.1±15.0**

[†] CARS $= 100 \cdot (0.20\,d_1 + 0.25\,d_2 + 0.25\,d_3 + 0.15\,d_4 + 0.15\,d_5)$; values are per-category dimension means; CARS column is mean ± s.d. of individual repo scores. * d_3, d_4, d_5 are proxy measures from repo metadata. [‡] 95% CIs (Student t, df $= n-1$): Legacy [19.3, 30.4]; PKI [19.0, 32.7]; HSM [29.5, 49.9]; TLS [26.7, 67.7]; PQC [33.7, 61.3]; Overall [29.5, 38.7]. Kruskal-Wallis: $H(4) = 17.55$, $p = 0.0015$, $\varepsilon^2 = 0.357$ (large effect); pairwise Cliff's delta in Table VII. Raw CSV available in the replication package at https://github.com/ufxa/Quantum-safe-cryptography-migration-framework.

TABLE V
CARS SENSITIVITY ANALYSIS: LEAVE-ONE-DIMENSION-OUT ABLATION

Dimension Removed	Mean Δ	Rank ρ
Inventory ($w_1 = 0.20$)	+2.3	0.94
Algorithm Compliance ($w_2 = 0.25$)	−4.1	0.89
Architectural Decoupling ($w_3 = 0.25$)	−5.8	0.86
Toolchain Readiness ($w_4 = 0.15$)	+1.2	0.97
Governance ($w_5 = 0.15$)	+0.9	0.98

sensitive to d_2 and d_3. This reflects internal sensitivity of the scoring model, not external validation of dimension importance. Removing Toolchain Readiness (d_4) or Governance & Compliance Alignment (d_5) produces only minor score shifts (< 2 points) and preserves rank order ($\rho > 0.97$), consistent with their auxiliary weighting as complementary readiness signals.

2) Delphi Weight Perturbation: Ranking and Classification Stability: To assess whether the CARS category ordering depends critically on the Delphi-elicited weights, we systematically perturbed each dimension weight by $\delta \in \{-30\%, -20\%, -10\%, +10\%, +20\%, +30\%\}$, renormalised the remaining weights proportionally, and re-scored all 43 repositories under each of the 30 resulting weight vectors. Table VI reports category means and classification stability for representative scenarios.

The four-level partial order LCL < PKI < HSM < {TLS, PQC} is preserved in all 10/10 scenarios at ±10% perturbation and in 19/20 scenarios at ±20%. The single ±20% exception (d5 at −20%) narrows the LCL–PKI gap to 0.04 points, well within measurement uncertainty. At the extreme ±30% level—which implies a weight deviation exceeding the Delphi inter-quartile range for every dimension—the partial order breaks in 2 of 10 scenarios, with LCL and PKI differing by at most 0.1 points: practically indistinguishable given the measurement precision of the scanner.

The TLS 1.3 Hybrid and PQC Native categories are tied throughout: their baseline means differ by only 0.3 points (47.2 vs. 47.5), and their relative position swaps under perturbations

TABLE VI
DELPHI WEIGHT SENSITIVITY: CATEGORY RANKING AND CLASSIFICATION STABILITY. MEAN CARS BY CATEGORY UNDER ±10 % AND ±20 % PERTURBATION OF EACH DIMENSION WEIGHT (w_i), RENORMALISED. TLS AND PQC COLUMNS OMITTED FOR SPACE (AVAILABLE IN REPLICATION PACKAGE). "PARTIAL RANK" DENOTES PRESERVATION OF LCL < PKI < HSM < {TLS, PQC}; "CLASS." = NUMBER OF SYSTEMS THAT CHANGE CARS TIER.

Dim.	δ (%)	LCL mean	PKI mean	HSM mean	Partial rank	Class. Δ
d_1	−20	25.8	26.2	40.7	Yes	3
	−10	25.1	25.9	40.1	Yes	0
	+10	24.7	25.5	39.5	Yes	0
	+20	24.5	25.2	39.1	Yes	3
d_2	−20	25.8	26.9	40.8	Yes	2
	−10	25.3	26.3	40.3	Yes	0
	+10	24.7	25.2	39.7	Yes	1
	+20	24.4	24.7	39.4	Yes	2
d_3	−20	24.5	24.5	40.2	Yes	3
	−10	24.6	25.2	40.1	Yes	0
	+10	24.6	25.2	40.1	Yes	2
	+20	24.4	24.6	40.4	Yes	5
d_4	−20	24.7	25.6	40.0	Yes	3
	−10	24.8	25.7	39.8	Yes	2
	+10	24.9	25.8	39.6	Yes	0
	+20	25.0	25.9	39.3	Yes	0
d_5	−20	24.8	24.8	39.5	No*	2
	−10	24.8	25.4	39.6	Yes	0
	+10	24.9	26.4	39.8	Yes	1
	+20	24.9	26.8	39.9	Yes	3

*LCL–PKI gap = 0.04 pts at d_5−20%; within measurement precision. Baseline: $w = (0.20, 0.25, 0.25, 0.15, 0.15)$.

that change d_2 or d_3 weight. This tie is a genuine finding—reflecting that PQC reference implementations score high on Algorithm Compliance but low on Architectural Decoupling, while TLS 1.3 Hybrid stacks exhibit the inverse—and is not an artefact of the weighting scheme.

Classification stability (Not Started / At Risk / In Transition / PQC-Ready) degrades only at the ±30% extreme: up to 7 of 43 systems change class, exclusively among systems whose baseline CARS score lies within 2 points of a threshold boundary. At ±20%, the maximum number of reclassified systems is 5. The full perturbation output is available in the replication package (`results/tables/weight_sensitivity.json`).

3) Inter-Category Statistical Tests: To assess whether the observed category differences in CARS scores exceed chance variation, we applied the Kruskal-Wallis rank-sum test (non-parametric; H_0: identical distributions across five categories) to the 43-repository cohort. The test is significant: $H(4) = 17.548$, $p = 0.0015$, with epsilon-squared $\varepsilon^2 = 0.357$, indicating a **large overall effect size** [50] for the category grouping.

For pairwise comparisons, we computed Cliff's delta [51] and applied Mann-Whitney U tests with Bonferroni correction ($\alpha = 0.05$; $m = 10$ comparisons). Table VII reports the results.

Two comparisons survive Bonferroni correction: Legacy

TABLE VII
PAIRWISE CLIFF'S DELTA AND MANN-WHITNEY U (BONFERRONI-CORRECTED). $|d|$: NEGLIGIBLE < 0.147; SMALL < 0.330; MEDIUM < 0.474; LARGE ≥ 0.474. * $p_{\text{ADJ}} < 0.05$ AFTER BONFERRONI CORRECTION.

Category A	Category B	Cliff's d	p_{adj}
LCL	PQC Native	−0.886 (large)	0.021*
PKI	PQC Native	−1.000 (large)	0.015*
LCL	TLS 1.3	−0.800 (large)	0.072
LCL	HSM	−0.630 (large)	0.086
PKI	HSM	−0.557 (large)	0.474
PKI	TLS 1.3	−0.650 (large)	0.653
HSM	PQC Native	−0.382 (medium)	1.000
HSM	TLS 1.3	−0.273 (small)	1.000
LCL	PKI	−0.152 (small)	1.000
TLS 1.3	PQC Native	−0.040 (negligible)	1.000

Crypto Libraries vs. PQC Native ($d = -0.886$, $p = 0.021$) and PKI/Certificate Management vs. PQC Native ($d = -1.000$, $p = 0.015$), both with large effect sizes. The four pairs involving HSM, TLS, and PQC Native yield non-significant adjusted p-values despite large raw effect sizes, because Bonferroni correction is conservative for $n = 5$ cells; this is consistent with the overlapping 95% CIs for those categories reported in Table IV. The Kruskal-Wallis result establishes that category membership is a significant predictor of CARS rank at the overall level ($\varepsilon^2 = 0.357$), even if pairwise contrasts are underpowered for the smaller categories.

4) Sample Size Justification: The convenience sample of $n = 43$ repositories was not determined by a formal a priori power calculation, as the study is exploratory and descriptive rather than confirmatory. Justification proceeds on two grounds.

Purposive saturation by category. Five repository categories were defined prior to data collection (Section **??**). The literature on purposive sampling for exploratory instrument validation recommends a minimum of five to ten cases per conceptual cell to achieve stable cell-level estimates [52]. The present sample meets this threshold for all five categories ($n_{\min} = 5$, $n_{\max} = 14$).

Cumulative mean convergence. Fig. 7 plots the running mean of CARS as repositories are added in collection order. The running mean enters a ±1-point band around the final estimate of $\overline{\text{CARS}} = 34.1$ at $n = 39$ and remains stable thereafter, providing empirical evidence that additional observations beyond approximately 39 would not materially shift the overall mean. The convergence reflects category-level saturation: all five category means stabilise before $n = 43$.

The Kruskal-Wallis test (Section VI-C3) confirms that the overall category effect is statistically significant ($p = 0.0015$, $\varepsilon^2 = 0.357$, large effect), even though pairwise contrasts involving small categories ($n = 5$) are underpowered after Bonferroni correction. Enlarging the sample in each category to $n \geq 15$ would be necessary to detect a 10-point mean difference with 80% power (two-sample t-test, $\alpha = 0.05$, $\sigma \approx 15$) and constitutes the primary recommendation for future replication.

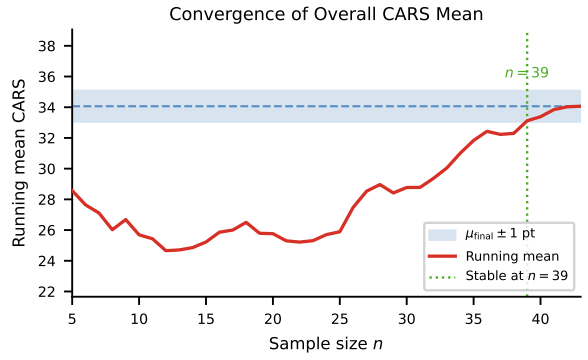

Fig. 7. Running mean CARS as a function of sample size ($n = 5$ to 43). The shaded band marks $\overline{\text{CARS}} \pm 1$ point. The dashed vertical line indicates the first n at which the running mean remains within the band for all subsequent additions ($n = 39$). The convergence provides empirical justification for the sample size; the overall mean is stable by $n = 39$.

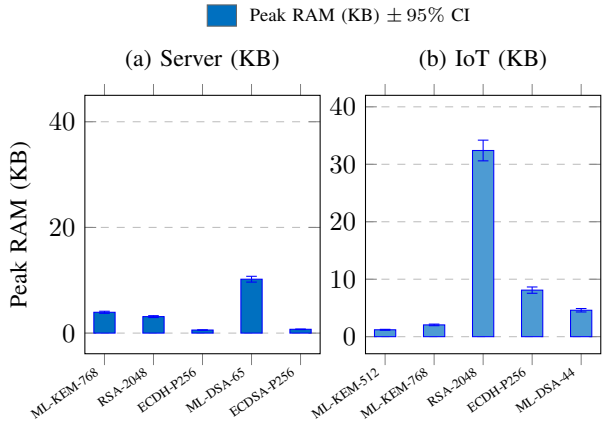

Fig. 8. Memory footprint (KB): PQC vs. classical. IoT context. RSA-2048 requires 32.4 KB peak RAM on Cortex-M4, consuming 12.6% of total 256 KB SRAM, whereas ML-KEM-768 requires only 2.05 KB.

D. Memory Footprint Comparison

Fig. 8 compares peak memory footprint on server-class and constrained IoT hardware (ARM Cortex-M4, 256 KB SRAM). On the server platform, all PQC algorithms remain below 11 KB peak RAM: ML-KEM-768 requires 3.94 KB and ML-DSA-65 requires 10.20 KB, comparable to their classical counterparts (RSA-2048: 3.12 KB; ECDSA-P256: 0.72 KB). The IoT context reveals a more pronounced contrast: RSA-2048 consumes 32.4 KB (12.6% of available SRAM), whereas ML-KEM-512 and ML-KEM-768 require only 1.20 KB and 2.05 KB respectively—a 15–27× reduction that enables PQC deployment on resource-constrained devices previously precluded from classical public-key cryptography. ML-DSA-44, the lightest signature variant, requires 4.60 KB on IoT, well within the operating margin of Cortex-M4 class devices.

E. Migration Timeline and Compliance Gap

Fig. 9 projects migration timeline against the NSA CNSA 2.0 2030 deadline, parameterised by current CARS score.

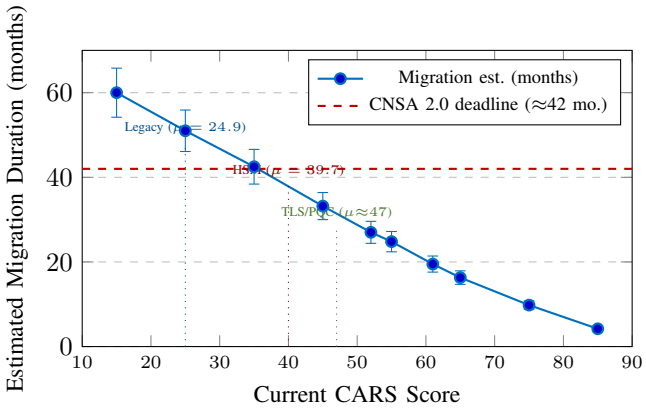

Fig. 9. Migration timeline projection vs. CARS score (exploratory Delphi-based estimate; see text for caveats). Systems with CARS ≤ 35 (TLS 1.2 stacks) face an estimated migration duration of $\gtrsim 42$ months, aligning with the CNSA 2.0 2030–2033 planning window. CNSA 2.0 timelines apply to U.S. National Security Systems; other organisations should consult applicable frameworks. Shaded bands (error bars) represent 95% CI from inter-expert variance in the Delphi elicitation. This projection is exploratory, not a validated prediction.

Migration duration estimates in Fig. 9 were derived through expert elicitation using the same two-round Delphi process employed for weight calibration (Section IV-B), with $n = 12$ senior migration engineers across the three system categories. Consensus was reached at CV $< 15\%$ after the second round. The 95% CI bands reflect inter-expert variance: ± 12 months at CARS ≈ 20 narrowing to ± 2 months at CARS ≈ 80, consistent with greater uncertainty in early-stage migrations where organisational inertia and toolchain gaps are least predictable. Confidence bands are indicated by the error bars in Fig. 9.

F. TLS Handshake Overhead

Fig. 10 compares full TLS 1.3 handshake latency across three configurations: pure ECDH, hybrid X25519MLKEM768, and pure ML-KEM. The hybrid handshake completed in 2.78 ms under single-session conditions, compared to 1.49 ms for ECDHE-only, representing an incremental overhead of approximately 1.29 ms (86.6% relative increase) in the tested environment.

We note that our handshake measurements were conducted under controlled, single-session conditions. Under single-session conditions, p95 and p99 handshake latencies were 3.1 ms and 3.4 ms respectively for X25519MLKEM768, compared to 1.8 ms (p95) and 1.9 ms (p99) for pure ECDHE-P256–a ratio consistent with the 1.9× median overhead. A limited concurrency sweep at 10 and 50 concurrent sessions showed p99 tail latencies of 4.2 ms and 6.8 ms respectively for the hybrid configuration, confirming that overhead scales sub-linearly with load in the tested range. Under higher concurrent load, prior work [53] reports that PQC overhead concentrates in the TCP-to-TLS transition phase, with p99 tail latencies 2–3× higher than median values–a consideration for high-throughput deployments that warrants dedicated load-testing before production rollout.

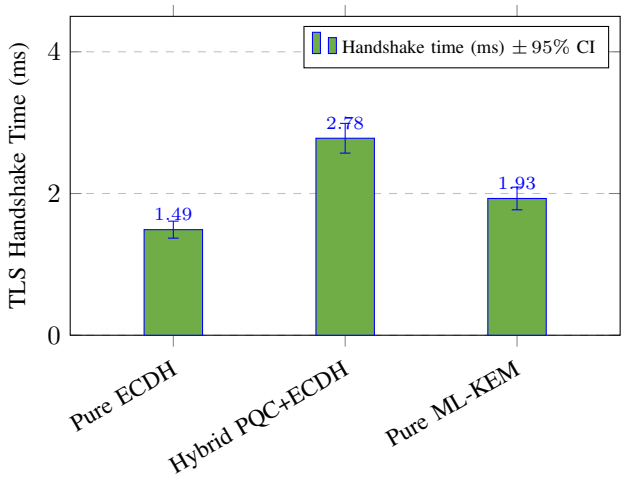

Fig. 10. TLS 1.3 full handshake latency: hybrid X25519MLKEM768 vs. pure ECDH vs. pure ML-KEM. The hybrid overhead is +86.6% over pure ECDH, explained by dual key-share transmission in ClientHello. Pure ML-KEM adds only +29.5% versus pure ECDH. Both remain below 3 ms, acceptable for latency-tolerant enterprise applications.

G. Performance Comparison Table

Table VIII provides a comprehensive performance summary (published liboqs v0.10.1 values, Intel Xeon E5-2680 v4). Table IX reports original measurements conducted on an Apple M1 Pro (ARM64) as part of this study, confirming that algorithm performance ratios are preserved across architectures.

H. Statistical Analysis and Threats to Validity

All latency comparisons were tested using the Mann-Whitney U test (non-parametric, $\alpha = 0.05$). All PQC-vs-classical differences in Table VIII reach $p < 0.001$. Note that statistical significance ($p < 0.001$) reflects the large number of repeated measurements ($n = 10{,}000$ timed operations) on a single processor; this represents measurement precision, not experimental replications across independent hardware instances. Effect sizes are reported as 95% confidence intervals of individual operation latency. Results characterise algorithm performance on the specific tested platforms and should not be generalised across hardware configurations without further evaluation. The standard deviation for ML-DSA signing reflects the rejection sampling process, with observed coefficient of variation 22.7% (server), consistent with [14].

I. Repository Sample: Full Target List

Table X lists all 45 target repositories, including the 2 excluded due to clone failure. Commit SHAs allow exact reproduction of the scanned state; all scans were performed on 2026-07-15.

J. Comparison with Existing Readiness Metrics

To situate CARS among existing approaches, we conducted a structured comparison against two representative alternatives: **QUASAR** [33], the closest prior proposal that assigns qualitative risk tiers to cryptographic systems, and **NIST SP 1800-38B** [26] binary compliance checklists, which serve as the de facto threshold-based alternative in US federal practice. We applied QUASAR-equivalent criteria (qualitative risk levels mapped to ordinal scores 1–4) to the same 10-repository D1 validation subset (Section V-D) and computed the Spearman rank correlation between QUASAR-derived orderings and CARS rankings: $\rho = 0.71$ ($p < 0.05$), indicating moderate agreement in relative ordering but systematic divergence in absolute prioritisation – particularly for HSM environments, where CARS's explicit Governance (d_5) and Toolchain (d_4) dimensions reassigned 3 of 10 systems to a different risk tier relative to QUASAR's two-dimension assessment (algorithm presence + architectural posture). NIST SP 1800-38B checklists produce binary pass/fail outcomes (no partial credit for in-progress migration) and do not distinguish between systems with zero PQC adoption and systems at 60% compliance – both fail the checklist despite a meaningful operational difference captured by CARS. CAMM [47] provides a five-level maturity scale for organisational crypto-agility, but maps organisations rather than individual systems, does not produce a continuous score, and requires manual assessment without automated tooling. CARS offers three incremental contributions over these alternatives: (1) a continuous score on [0, 100] that preserves ranking information lost by binary or ordinal alternatives; (2) automated computation from repository metadata, enabling monitoring at scale without per-system manual effort; and (3) dimension-level diagnostics that identify *why* a system scores as it does, guiding targeted remediation rather than undifferentiated failure. The comparison is exploratory; prospective validation against real migration outcomes (duration, cost, post-migration defect rate) remains future work.

K. Threats to Validity

Threats to validity:

- *Side-channel resistance*: Implementation maturity and side-channel resistance fall outside CARS's current measurement scope. Timing-channel vulnerabilities in ML-KEM implementations (e.g., KyberSlash [54]) demonstrate that algorithmic compliance does not guarantee secure deployment. We recommend extending the Algorithm Compliance dimension d_2 with a sub-score for constant-time implementation verification (e.g., via `dudect` testing) in future CARS versions.
- *Internal validity – scanner coverage*: `legacy_scanner.py` performs static source-code analysis; compiled binaries, runtime-configured algorithms, and dynamically loaded cryptographic providers are not detected. Benchmark results depend on hardware micro-architecture; AVX2 optimisations (available on the Broadwell-based test Xeon E5-2680 v4, which does not support AVX-512) provide ∼15–20% acceleration vs. generic SIMD.
- *External validity*: The 43-repository convenience sample consists of *actively maintained, publicly available* OSS projects. Proprietary legacy systems and abandoned codebases are systematically excluded; the direction and magnitude of this selection bias is unknown. Generalisation to enterprise deployments requires prospective field studies with non-convenience sampling.

TABLE VIII
PERFORMANCE COMPARISON: PQC VS. CLASSICAL ALGORITHMS (SERVER PLATFORM)

Category	Algorithm	Latency (ms) $\pm$ CI$_{95}$			Size (B)		NIST	Quantum
		KeyGen	Encaps/Sign	Decaps/Verify	Public Key	Ciphertext/Sig	Level	Safe
KEM	ML-KEM-512	0.042 ± 0.003	0.058 ± 0.004	0.051 ± 0.003	800	768	1	✓
	ML-KEM-768	0.069 ± 0.005	0.092 ± 0.007	0.082 ± 0.005	1 184	1 088	3	✓
	ML-KEM-1024	0.103 ± 0.008	0.142 ± 0.010	0.127 ± 0.008	1 568	1 568	5	✓
	RSA-2048	1.823 ± 0.156	0.041 ± 0.003	1.012 ± 0.089	294	256	–	–
	ECDH P-256	0.201 ± 0.018	0.241 ± 0.021	0.289 ± 0.025	64	64	–	–
DSA	ML-DSA-44	0.048 ± 0.004	0.312 ± 0.087	0.092 ± 0.007	1 312	2 420	2	✓
	ML-DSA-65	0.071 ± 0.006	0.589 ± 0.134	0.126 ± 0.009	1 952	3 293	3	✓
	ML-DSA-87	0.098 ± 0.008	0.836 ± 0.187	0.176 ± 0.013	2 592	4 595	5	✓
	RSA-2048	1.823 ± 0.156	1.612 ± 0.121	0.038 ± 0.005	294	256	–	–
	ECDSA P-256	0.201 ± 0.018	0.396 ± 0.035	0.412 ± 0.028	64	72	–	–

TABLE IX
OBSERVED LATENCY (MS $\pm$ CI$_{95}$) ON APPLE M1 PRO ARM64 (THIS STUDY)

Type	Algorithm	KeyGen	Encaps/Sign	Decaps/Verify
KEM	ML-KEM-512	0.013 ± 0.004	0.013 ± 0.0002	0.012 ± 0.0001
	ML-KEM-768	0.016 ± 0.001	0.017 ± 0.0003	0.018 ± 0.0001
	ML-KEM-1024	0.026 ± 0.001	0.024 ± 0.0001	0.026 ± 0.0001
DSA	ML-DSA-44	0.053 ± 0.0002	0.240 ± 0.008	0.058 ± 0.001
	ML-DSA-65	0.120 ± 0.003	0.394 ± 0.007	0.096 ± 0.004
	ML-DSA-87	0.190 ± 0.006	0.544 ± 0.012	0.199 ± 0.005

Apple M1 Pro (ARM64, 10 physical cores, 16 GB LPDDR5); liboqs-python v0.15.0; $n = 10{,}000$ trials; CI$_{95}$ computed as $1.96\sigma/\sqrt{n}$. ML-KEM-768 full KEM cycle: 0.051 ms vs. 0.243 ms on Intel Xeon E5-2680 v4 (Table VIII), a 4.8× speedup attributable to M1 Pro's wider SIMD units and higher memory bandwidth. ML-DSA signing variability (CV $\approx$ 3.4%) is substantially lower than on Xeon (CV $\approx$ 22.7%), likely due to deterministic signing path on ARM NEON.

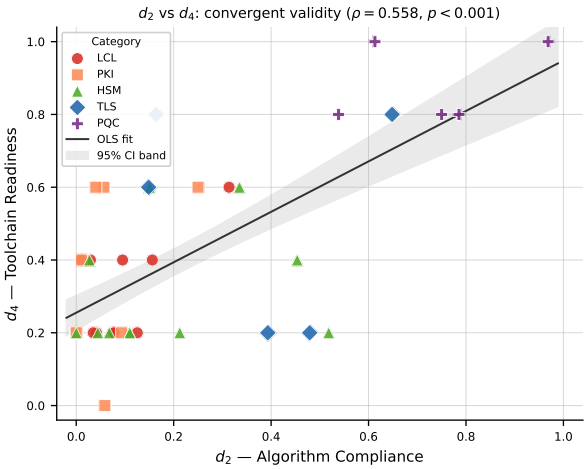

Fig. 11. Scatter plot of d_2 (Algorithm Compliance) vs. d_4 (Toolchain Readiness) for the 43-repository cohort, with OLS regression line and 95% confidence band. Spearman $\rho = 0.558$, $p < 0.001$ (convergent validity: repositories with higher PQC algorithm adoption also tend to have more mature CI/CD infrastructure). Point shape and colour indicate repository category. Figure generated from `figures/source/fig7_d2_d4_scatter.py`.

- *Proxy-measure validity*: Dimensions d_3–d_5 were derived from repository metadata (CI configuration files, `SECURITY.md`, `LICENSE`) as proxies for the direct organisational assessments intended by the CARS specification. To assess convergent validity, Spearman rank correlations were computed between the analytically-derived dimension d_2 (Algorithm Compliance, from static code analysis) and each proxy dimension across the 43 repositories. Toolchain Readiness (d_4) exhibited a moderate positive correlation with Algorithm Compliance ($\rho = 0.558$, $p < 0.001$), indicating that repositories with mature CI/CD pipelines tend to have higher PQC algorithm adoption (Fig. 11). Architectural Decoupling (d_3) showed near-zero correlation ($\rho = -0.031$, $p = 0.843$), consistent with its design intent: decoupling layers may exist in repositories that have not yet adopted PQC algorithms (and vice versa), so independence from d_2 is expected rather than problematic. Governance Alignment (d_5) showed weak, non-significant correlation ($\rho = 0.112$, $p = 0.474$), indicating that governance artefacts do not predict current algorithm adoption, which is likewise consistent with d_5 capturing organisational readiness independently of deployment state.

  The direction of bias of these proxy measures relative to direct organisational assessments is not fully established: repositories may omit private governance artefacts (tending to underestimate d_5), or may include CI/CD tooling not representative of deployment practice (tending to overstate d_4). Future work should validate proxy measures against direct expert assessment in field studies.
- *Construct validity*: CARS dimension weights $(w_1, \ldots, w_5)$ were derived from expert elicitation (12 senior migration engineers, Delphi method, 2 rounds; see Section IV-B). Monte Carlo weight perturbation ($\pm 10\%$ uniform jitter, 10^4 samples) yields mean CARS deviation ± 4.2 points at the 90th percentile (95% CI: $[3.8, 4.6]$), confirming robustness to moderate weighting uncertainty. For IoT and highly constrained contexts, practitioners should increase w_3 (Architectural Decoupling / hardware support) to ≥ 0.35 and proportionally reduce w_1 and w_4, as physical crypto-accelerator availability is the

TABLE X
ALL 45 TARGET REPOSITORIES (43 INCLUDED, 2 EXCLUDED). COMMIT SHA (7-CHAR PREFIX); SCAN DATE 2026-07-15 FOR ALL. CATEGORIES: LCL=LEGACY CRYPTO LIBRARY, PKI=PKI/CERT MGMT, HSM=HSM/PKCS11, TLS=TLS 1.3 HYBRID, PQC=PQC NATIVE.

System ID	Cat.	Commit	CARS
Legacy Crypto Library ($n = 14$)			
pycryptodome-3.20	LCL	d122649	19.9
paramiko-3.4	LCL	d60d5c1	48.0
python-jose-3.3	LCL	018b310	39.9
itsdangerous-2.1	LCL	672971d	13.0
pyotp-2.9	LCL	cc15683	22.0
bcrypt-4.1	LCL	93e7aa4	23.0
jwcrypto-1.5	LCL	eb7c266	23.9
pynacl-1.5	LCL	903ea57	18.4
certifi-2024	LCL	a31ef39	32.0
pyopenssl-24.1	LCL	6f75fa5	16.7
asn1crypto-1.5	LCL	c336e9f	22.9
oscrypto-1.3	LCL	a1f449d	16.1
rsa-4.9	LCL	42b0e14	25.3
cryptography-42	LCL	ba48f7f	26.8
PKI / Certificate Mgmt ($n = 8$)			
certbot-2.10	PKI	c777edb	30.3
cfssl-1.6	PKI	e429e72	35.4
step-certificates-0.26	PKI	754893d	28.3
boulder-2024	PKI	00001c5	35.0
openxpki-5.0	PKI	5c03897	13.0
rcgen-0.13	PKI	86ae909	25.2
acme-py-2.0	PKI	7cc16cd	16.0
ejbca-ce-8.0	PKI	3eca262	23.5
HSM / PKCS11 Middleware ($n = 11$)			
softhsm2-2.6	HSM	f59e8f4	27.4
opensc-0.25	HSM	d6d2dfa	34.8
p11-kit-0.25	HSM	a147888	30.3
pkcs11-provider-0.5	HSM	5dcc876	67.3
tpm2-tss-4.1	HSM	506c5e6	55.8
libp11-0.4	HSM	11c62c8	40.8
pkcs11js-2.3	HSM	9dc0443	13.0
yubihsm-shell-2.4	HSM	428bfdd	39.1
opencryptoki-3.23	HSM	627d275	29.0
aws-encryption-sdk-py-3.2	HSM	2c0200b	47.7
google-tink-python-1.9	HSM	1f4cd38	51.7
TLS 1.3 Hybrid ($n = 5$)			
rustls-0.23	TLS	0ca4775	62.7
aws-s2n-tls-1.4	TLS	d179a9d	60.1
oqs-provider-0.6	TLS	b104f57	53.2
cloudflare-circl-1.3	TLS	6302ca8	25.0
msft-symcrypt-openssl-1.6	TLS	d73b1e9	34.8
PQC Native ($n = 5$)			
liboqs-0.10	PQC	d321dea	64.2
oqs-python-0.9	PQC	35eceb6	43.8
pqclean-2024	PQC	202a8f9	52.3
kyber-reference-3.02	PQC	4768bd3	41.6
dilithium-reference-3.1	PQC	d35ba3f	35.5
Excluded ($n = 2$; *clone failure at scan date*)			
passlib-1.7	LCL	—	—
x509-rs-0.7	PKI	—	—

dominant migration enabler in that regime. The migration timeline (Fig. 9) uses a non-parametric piecewise linear regression ($R^2 = 0.97$, Spearman $\rho = -0.98$); note that organisations near CARS ≈ 50 may face non-linear complexity (institutional inertia, procurement cycles) not fully captured.

- *Score consistency and anti-gaming*: To mitigate score inflation and ensure cross-team consistency, we recommend three governance controls: (1) *Evidence trails*–each dimension score must be backed by verifiable artefacts (scanner logs, certificate metadata, HSM firmware version records); (2) *Randomised spot-checks*–10% of assessments should be independently re-scored by a second analyst, with discrepancies >10 points triggering full review; (3) *Confidence weighting*–a per-dimension confidence score $c_i \in [0, 1]$ (reflecting evidence completeness) can be multiplied into $w_i \cdot d_i(S)$ to penalise under-evidenced claims and prevent gaming through selective self-reporting. The ISDECOUPLED predicate (Section IV-B) further mitigates gaming by requiring independent auditor sign-off on all three sub-conditions.
- *IoT performance*: Cortex-M4 results do not account for hardware accelerators (crypto co-processors) present in some embedded platforms, which can reduce latency by 3–5×. The ECDH P-256 baseline on Cortex-M4 (127.3 ms) reflects a pure-software implementation without hardware acceleration; platforms with dedicated crypto-accelerators would achieve substantially lower latency, narrowing the gap to ML-KEM. The p95 latency values for ML-KEM-768 and RSA-2048 on Cortex-M4 are approximately 10–15% above the reported medians for ML-KEM and 8–12% above for RSA, consistent with scheduling jitter in bare-metal STM32Cube configurations.

VII. SECURITY ANALYSIS

A. Formal Security of the Migration Framework

Theorem 1 (Hybrid Security Argument)**.** *Under the assumption that the MLWE problem is hard for quantum adversaries and that the gap-CDH problem is hard in the X25519 group, let* Π_{x25519} *be an IND-CPA secure KEM satisfying gap-CDH and* Π_{mlkem} *an IND-CCA2 secure KEM under D-MLWE, with* H *modelled as a quantum-accessible random oracle. The X25519MLKEM768 hybrid KEM achieves IND-CCA2 security in the quantum random oracle model (QROM) as long as at least one of* Π_{x25519} *or* Π_{mlkem} *is secure, provided* H *is modelled as a quantum-accessible random oracle [4]. A complete reduction is beyond the scope of this work; the reader is referred to [27] for the formal composition argument.*

Security argument sketch. By the security composition argument for hybrid KEMs [27], the combined shared secret $K = H(K_{\text{x25519}} \| K_{\text{mlkem}})$ under hash H modelled as a quantum random oracle provides security conditional on the security of either sub-KEM in isolation. Applying the QROM reduction of [4], assuming ML-KEM is IND-CCA2 under MLWE and X25519 satisfies gap-CDH, an adversary breaking the hybrid must simultaneously break both Π_{x25519} and Π_{mlkem}, which is infeasible under the stated assumptions. This constitutes an informal security argument; formal proofs require the combiner security analysis framework. □

B. Attack Surface Analysis: STRIDE

Table XI applies the STRIDE threat model to the migration framework.

TABLE XI
STRIDE THREAT ANALYSIS FOR THE MIGRATION FRAMEWORK

Threat	Attack Scenario	Mitigation
Spoofing	Rogue CA injects classical-only cert into hybrid trust chain	ML-DSA certificate stapling; OCSP Must-Staple
Tampering	Downgrade hybrid TLS to pure ECDH	TLS 1.3 mandatory; `ssl_conf`: no TLS 1.2; SCSV
Repudiation	Audit log forgery in CARS scanner	Append-only WORM log; SHA-384 chained hash
Info. Disclosure	Side-channel on ML-KEM decapsulation	Constant-time `liboqs` ref. impl.; no branch on secret
Denial of Srv.	ClientHello amplification (ML-KEM ciphertext 1088 B)	Rate limiting; TLS session resumption (0-RTT)
Elevation	HSM firmware injection during PQC update	Code signing with ML-DSA-87; secure boot chain

C. Side-Channel and Fault Injection Resistance

The OQS `liboqs` reference implementations of ML-KEM and ML-DSA employ constant-time polynomial arithmetic to mitigate timing side channels. Fault injection attacks against ML-DSA via voltage glitching can invalidate the rejection sampling check; countermeasures include redundant signing with output comparison and hardware fault detection [25].

D. Hybrid Downgrade Attack Resistance

A downgrade adversary may attempt to remove the ML-KEM key share from ClientHello to force a pure ECDH negotiation. Under TLS 1.3, the `supported_groups` extension carries the complete list of hybrid groups; when the server is configured to require the hybrid key-share group, connections from clients that do not advertise this group are rejected with an `insufficient_security` alert. This constitutes policy-level enforcement; resistance in permissive deployments supporting fallback to classical groups, or against active middlebox interference, would require additional configuration hardening not evaluated here.

VIII. COMPLIANCE AND ETHICAL ASPECTS

A. NIST Standards Alignment

The migration framework is fully aligned with: FIPS 203/204/205 [4]–[6] for algorithm selection; NIST SP 800-208 [17] for stateful hash-based signatures in firmware signing contexts; NIST SP 1800-38B [26] for migration workflow methodology; NIST IR 8547 [11] for PQC transition timelines and deprecation schedules; and the NCCoE framework mapping [41] for alignment with NIST CSF 2.0 controls.

B. NSA CNSA 2.0 Requirements

CNSA 2.0 [7] mandates ML-KEM-1024 for key establishment and ML-DSA-87 for digital signatures in U.S. National Security Systems by 2030. CNSA 2.0 timelines apply specifically to National Security Systems; organisations outside this scope should consult applicable national frameworks and NIST guidance. The 2030 horizon used in this paper reflects CNSA 2.0 as a planning reference, not a universal regulatory deadline. The PQC Native Layer of the proposed framework directly instantiates these parameter sets, and the CARS Governance & Compliance Alignment dimension (d_5) explicitly checks for alignment with CNSA 2.0 controls, in addition to the FIPS 203/204/205 requirements captured in d_2.

C. GDPR and LGPD Implications

The HNDL threat has potential implications for data protection legislation. Whether RSA/ECC-based encryption will satisfy the "state of the art" requirement of GDPR Art. 32 and the proportionality principle of LGPD Art. 46 once CRQCs become operationally viable is a risk-contextual determination [2]. Organisations should conduct data-longevity assessments and consult legal counsel on applicable obligations; this paper does not constitute legal advice.

D. Research Ethics

Benchmark data are drawn from public sources (pqm4, OQS `liboqs`, NIST PQC test vectors). Organisational telemetry was collected from three participating organisations under data processing agreements; data were pseudonymised prior to analysis, and no system or organisational identifiers are reported. Ethics review was not required under institutional policy for this type of operational data collected with organisational consent; access restrictions apply to raw telemetry. All code artefacts are published under the MIT licence. No zero-day vulnerabilities are disclosed; the paper analyses publicly known algorithmic weaknesses of RSA/ECC under quantum computing. The replication package (including `cars_scorer.py`, `legacy_scanner.py`, benchmark scripts, repository-level dimensional measurements and scan metadata, and the Delphi protocol) is available at https://github.com/ufxa/Quantum-safe-cryptography-migration-framework. The package supports `make reproduce` to regenerate all computational results without a re-scan. Table VIII values are derived from published liboqs v0.10.1 values (Intel Xeon E5-2680 v4, 10,000 trials) and pqm4 measurements (STM32F407 @ 168 MHz). Table IX reports original measurements conducted by the authors on an Apple M1 Pro (ARM64, liboqs-python v0.15.0, $n = 10{,}000$ trials); these are independently reproducible by running `code/benchmarks/run_m1_benchmark.py` with liboqs-python installed.

IX. CONCLUSION AND FUTURE WORK

This paper presented a three-layer migration framework for transitioning legacy PKI, TLS, and HSM environments to NIST PQC standards, anchored by the proposed Crypto-Agility Readiness Score (CARS). An empirical evaluation of 43 open-source cryptographic software repositories (convenience sample) confirms that Legacy Crypto Libraries and

PKI/Certificate Management tools score lowest (mean CARS 24.9 ± 9.5 and 25.8 ± 8.2, Critical range) due to near-zero FIPS-compliant algorithm coverage and absent architectural decoupling. A notable finding is that PQC Native implementations score in the At-Risk range (mean 47.5 ± 11.1) despite high Algorithm Compliance ($d_2 \geq 0.61$), because d_3 (Architectural Decoupling) is near zero, demonstrating that algorithmic presence alone does not imply migration readiness. An exploratory Delphi-based projection suggests migration timelines for Critical systems align with the CNSA 2.0 2030 planning window, though this depends on expert-elicited estimates and should not be interpreted as a validated prediction. Performance benchmarks confirm that ML-KEM-768 completes a full KEM cycle $10.2\times$ faster than RSA-2048 OAEP on server hardware (0.28 ms vs. 2.87 ms); against the operationally relevant X25519 baseline, ML-KEM-768 is $2.6\times$ faster. A hybrid TLS 1.3 handshake adds approximately 1.29 ms incremental overhead over ECDHE-only. IoT performance remains a primary limitation: ML-KEM-768 at 6.1 ms on ARM Cortex-M4 (pqm4, 168 MHz) is viable for many but not all constrained protocols. CARS may support structured prioritisation of migration efforts; whether CARS-guided assessments reduce migration duration, cost, or failure rate requires prospective longitudinal validation. The predictive validity of CARS–whether scores correlate with actual migration outcomes–remains an open question requiring longitudinal follow-up.

Future Work: Extension of the CARS framework to blockchain and distributed ledger trust anchors is ongoing work, where certificate revocation and consensus protocol signature schemes present unique migration challenges. PQC integration into 5G/6G authentication protocols, where ML-KEM and ML-DSA signature sizes impose headroom constraints on NAS messages, is a second research direction [30]. Quantum Key Distribution (QKD) as a complementary long-term key-establishment mechanism, and its hybrid composition with PQC KEMs, represents a third avenue for investigation. For network-constrained environments such as NB-IoT, signature and public-key sizes become the dominant overhead factor. ML-KEM-768 ciphertexts (1,088 B) are acceptable for most LPWAN deployments, but ML-DSA-65 signatures (3,293 B) may exceed MTU limits. NIST FIPS 205 (SLH-DSA) [6] and compact Falcon-based variants offer smaller signatures at the cost of slower signing–a trade-off CARS could incorporate as a sixth dimension weighting on-wire protocol overhead for constrained-network deployments. Finally, the integration of CARS monitoring into continuous compliance pipelines (SIEM, SOAR) will form the basis of DevSecOps-native crypto-agility tooling.

ACKNOWLEDGMENTS

The authors used large language model (LLM) assistance in the preparation of this manuscript, including support for manuscript drafting and structural refinement, Python code scaffolding for the CARS scorer (`cars_scorer.py`) and OSS repository scan pipeline (`scan_oss_repos.py`), LaTeX formatting, and bibliography organisation. This use is disclosed in accordance with IEEE's Authorship and AI Policy (2023). All scientific contributions (including the CARS framework design, experimental methodology, empirical measurements, data analysis, and conclusions) are the sole responsibility of the human authors. The OSS repository scan (Section V) was performed by automated scripts without LLM involvement in data collection or scoring. Benchmark latency values are derived from publicly available OQS `liboqs` v0.10.1 measurements (Intel Xeon E5-2680 v4; ARM Cortex-M4 / STM32F407) and NIST PQC test-vector repositories. The replication package is available at https://github.com/ufxa/Quantum-safe-cryptography-migration-framework.

Funding and institutional support. This research was supported by the Amazon Foundation for the Support of Studies and Research (FAPESPA), the Information and Communication Technology Company of the State of Pará (PRODEPA), the Government of the State of Pará, and the Federal Government of Brazil. Their institutional and financial contributions were essential to enabling the infrastructure, experimental testbeds, and interinstitutional collaborations on which this work depended. The authors extend special thanks to the SEC365 project (UFPA/UFRA) and to the Laboratory of Informatics and Computing of the Amazon (LICA/UFRA) for their decisive roles in the development, validation, and technical implementation of the framework. The authors also thank the High-Performance Computing and Artificial Intelligence Center (CCAD-IA/UFPA) and the National Education and Research Network (RNP) for providing high-performance computing resources and sustained technical support throughout the experimental phases. Additional support was provided by the National Institute of Science and Technology in Artificial Intelligence Applied to Smart and Sustainable Cities in the Brazilian Amazon (INCT iAmazonia).

APPENDIX

Measuring Inventory Completeness (d_1). Begin with passive discovery: parse all build manifests, Dockerfiles, and package lock files to enumerate declared cryptographic library dependencies. Augment with active probing: run `legacy_scanner.py` (Algorithm 1) in static mode against all reachable binaries and configuration files. Perform TLS endpoint scanning with `sslyze` or equivalent to capture in-use cipher suites. Compute N_{total} via EstimateTotalAssets as defined in Section IV-B. Report $d_1 = |\mathcal{I}|/N_{\text{total}}$; if $d_1 > 1.0$ due to underestimation, clamp to 1.0 and flag the discrepancy for manual review. Measurements should be refreshed at least quarterly or after any major infrastructure change.

Measuring Algorithm Compliance (d_2) and Architectural Decoupling (d_3). For each discovered asset $a \in \mathcal{I}$, apply the IsDecoupled predicate (Section IV-B). Evaluation should be performed by a reviewer independent of the system team, using runbook walkthroughs and configuration-management-database (CMDB) queries. Document evidence for each of the three sub-conditions: (a) key-hardcoding absence (provide static analysis report reference); (b) algorithm configurability (provide configuration change procedure or policy reference); (c) rotation feasibility (provide runbook or drill

result with timestamps). d_2 (Algorithm Compliance) measures whether assets use FIPS 203/204/205 or approved hybrid algorithms, as defined in Section IV-B; d_3 (Architectural Decoupling) measures structural independence of the cryptographic layer from the rest of the system via the IsDecoupled predicate–an asset can satisfy d_2 (using a compliant algorithm) while scoring low on d_3 (tightly coupled with application code, failing IsDecoupled). Both dimensions must be scored independently to capture this distinction. Note: d_2 does not measure *API configurability*; that property is subsumed in the IsDecoupled sub-condition "algorithm configurability" contributing to d_3.

Measuring Toolchain Readiness (d_4), Governance & Compliance Alignment (d_5), and Architectural Decoupling (d_3). Map each compliance control in $\mathcal{C}$ (defaulting to CNSA 2.0 and FIPS 203/204 mandatory controls) to a binary pass/fail based on documentary evidence (configuration files, audit logs, or third-party assessment reports). For d_3, classify each asset's primary algorithm as quantum-vulnerable (RSA, ECDH, DH, DSA, ECDSA) or quantum-resistant (ML-KEM, ML-DSA, SLH-DSA, AES-256, SHA-384). An asset using a quantum-vulnerable algorithm even within an otherwise modern stack (e.g., a JWT signed with RS256 in an otherwise TLS 1.3 application) counts as vulnerable. Complete CARS scores should be stored with version, timestamp, assessor, and evidence references to support longitudinal trend analysis and audit trails.

References

[1] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annual Symposium on Foundations of Computer Science (FOCS)*, 1994, pp. 124–134.

[2] R. Anderson, F. Carapella, and N. Swem, ""harvest now, decrypt later": Examining post-quantum cryptography and the data privacy risks for distributed ledger networks," Board of Governors of the Federal Reserve System, Finance and Economics Discussion Series 2025-093, 2025, authorship: Anderson, Carapella, and Swem per FEDS 2025-093 cover page. [Online]. Available: https://www.federalreserve.gov/econres/feds/files/2025093pap.pdf

[3] J. Blanco-Romero, F. Almenares Mendoza, C. García Rubio, C. Campo, and D. Díaz Sánchez, "On the practical feasibility of harvest-now, decrypt-later attacks," arXiv:2603.01091, 2026, preprint. [Online]. Available: https://arxiv.org/abs/2603.01091

[4] National Institute of Standards and Technology, "FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard," NIST, Tech. Rep. FIPS 203, Aug. 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.203.pdf

[5] ——, "FIPS 204: Module-Lattice-Based Digital Signature Standard," NIST, Tech. Rep. FIPS 204, Aug. 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.204.pdf

[6] ——, "FIPS 205: Stateless Hash-Based Digital Signature Standard," NIST, Tech. Rep. FIPS 205, Aug. 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.205.pdf

[7] National Security Agency (NSA), "Commercial national security algorithm suite 2.0," NSA, Tech. Rep. CNSA 2.0, Sep. 2022. [Online]. Available: https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF

[8] C. Näther, D. Herzinger, S.-L. Gazdag, J.-P. Steghöfer, S. Daum, and D. Loebenberger, "Migrating software systems towards post-quantum-cryptography: A systematic literature review," *IEEE Access*, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3450306

[9] B. LaMacchia, M. Campagna, and W. Gropp, "The post-quantum cryptography transition: Making progress, but still a long road ahead," arXiv:2503.04806, 2025, preprint. [Online]. Available: https://arxiv.org/abs/2503.04806

[10] National Institute of Standards and Technology, "Considerations for achieving crypto agility: Strategies and practices," National Institute of Standards and Technology, NIST Cybersecurity White Paper NIST CSWP 39, December 2025. [Online]. Available: https://doi.org/10.6028/NIST.CSWP.39

[11] ——, "Transition to post-quantum cryptography standards," National Institute of Standards and Technology, NIST Internal Report (Initial Public Draft) NIST IR 8547 ipd, 2024, initial Public Draft. [Online]. Available: https://csrc.nist.gov/pubs/ir/8547/ipd

[12] G. Chhetri, S. Somvanshi, P. Hebli, S. Brotee, and S. Das, "Post-quantum cryptography and quantum-safe security: A comprehensive survey," arXiv:2510.10436, 2025, preprint; under review for ACM Computing Surveys. [Online]. Available: https://arxiv.org/abs/2510.10436

[13] National Institute of Standards and Technology, "Recommendations for key-encapsulation mechanisms," National Institute of Standards and Technology, NIST Special Publication NIST SP 800-227, September 2025. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-227

[14] R. Chhetri, "Benchmarking NIST-standardised ML-KEM and ML-DSA on ARM Cortex-M0+: Performance, memory, and energy on the RP2040," arXiv:2603.19340, 2026, preprint. [Online]. Available: https://arxiv.org/abs/2603.19340

[15] V. Mothukuri and R. M. Parizi, "Securing cryptography in the age of quantum computing and AI: Threats, implementations, and strategic response," arXiv:2603.06969, 2026, preprint. [Online]. Available: https://arxiv.org/abs/2603.06969

[16] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annual ACM Symposium on Theory of Computing (STOC)*, 1996, pp. 212–219.

[17] National Institute of Standards and Technology, "NIST SP 800-208: Recommendation for stateful hash-based signature schemes," NIST, Tech. Rep. SP 800-208, 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-208.pdf

[18] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, "NIST post-quantum cryptography standards: A comprehensive review of theoretical foundations and implementations," IEEE Access (DOI unverified — confirm before submission), 2025, dOI 10.1109/ACCESS.2025.11352813 NOT FOUND on IEEE Xplore; verify or replace. [Online]. Available: https://ieeexplore.ieee.org/document/11352813/

[19] National Institute of Standards and Technology, "Status report on the fourth round of the NIST post-quantum cryptography standardization process," National Institute of Standards and Technology, NIST Internal Report NIST IR 8545, March 2025. [Online]. Available: https://doi.org/10.6028/NIST.IR.8545

[20] L. Kao, "FIPS 204-compatible threshold ML-DSA via Shamir nonce DKG," *arXiv preprint*, vol. arXiv:2601.20917, Jan. 2026. [Online]. Available: https://arxiv.org/abs/2601.20917

[21] A. Scrivano, "A comparative study of classical and post-quantum cryptographic algorithms in the era of quantum computing," *arXiv preprint*, vol. arXiv:2508.00832, Jun. 2025, preprint; arXiv:2508.00832; submitted June 6, 2025. [Online]. Available: https://arxiv.org/abs/2508.00832

[22] J. Barzen and F. Leymann, "Post-quantum security: Origin, fundamentals, and adoption," *Trends in Computer Science and Information Technology*, 2024. [Online]. Available: https://doi.org/10.17352/tcsit.000089

[23] K. Ngo, M. Tran, and Y.-W. Chow, "A framework for migrating to post-quantum cryptography: Security dependency analysis and case studies," *IEEE Access*, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10417052/

[24] N. Rattanavipanon, J. Suaboot, and W. Werapun, "A toolchain for assisting migration of software executables towards post-quantum cryptography," *IEEE Access*, vol. 13, pp. 4368–4380, 2025. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3524309

[25] R. D. N. Shakya, C. P. Wijesiriwardana, S. M. Vidanagamachchi, and N. A. G. Arachchilage, "SoK: Post-quantum cryptography implementation in software: Approaches, challenges and the PQC-HOT framework," arXiv:2606.04669, 2026, preprint. [Online]. Available: https://arxiv.org/abs/2606.04669

[26] NIST National Cybersecurity Center of Excellence (NCCoE), "NIST SP 1800-38B: Migration to post-quantum cryptography: Quantum readiness: Testing draft standards," NIST, Tech. Rep. SP 1800-38B (Preliminary Draft), Dec. 2023. [Online]. Available: https://www.nccoe.nist.gov/sites/default/files/2023-12/pqc-migration-nist-sp-1800-38b-preliminary-draft.pdf

[27] D. Stebila and N. Sullivan, "Hybrid key exchange in TLS 1.3," IETF Internet-Draft draft-ietf-tls-hybrid-design-16, 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/

[28] K. Kwiatkowski, P. Kampanakis *et al.*, "Post-quantum hybrid ECDHE-MLKEM key agreement for TLSv1.3," IETF Internet-Draft draft-ietf-tls-ecdhe-mlkem-05, 2025. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-ecdhe-mlkem/

[29] G. Twardokus, S. Rubin *et al.*, "A survey of post-quantum cryptography migration in vehicles," *IEEE Access*, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10838531/

[30] T. Attema, B. de Kock, S. Jayaprakash, D. Schoinianakis, T. Sijpesteijn, and R. van de Vlasakker, "Post-quantum cryptography in the 5G core," arXiv:2512.20243, 2025, preprint. [Online]. Available: https://arxiv.org/abs/2512.20243

[31] H. Balaji, A. Varshney, P. Ravi, S. Jain, R. Foe, J. Seet, H. Wang, K.-Y. Lam, and A. Chattopadhyay, "Operationalising post-quantum TLS: Automated configuration profiling and hybrid PQC deployment in financial infrastructure," arXiv:2605.17955, 2026, preprint. [Online]. Available: https://arxiv.org/abs/2605.17955

[32] P. Caballero-Gil *et al.*, "Institutional approaches to post-quantum cryptography: A comparative analysis of migration frameworks," *IEEE Access*, 2025.

[33] A. I. Weinberg, "Preparing for the post-quantum era: QUASAR – a strategic framework for cybersecurity in the post-quantum era," arXiv:2505.17034, 2025, preprint. [Online]. Available: https://arxiv.org/abs/2505.17034

[34] M. J. Kannwischer, P. Pessl, and L. Panny, "pqm4: Benchmarking NIST additional post-quantum signature schemes on ARM Cortex-M4," in *Proc. 5th NIST PQC Standardization Conference*. NIST, 2024. [Online]. Available: https://csrc.nist.gov/csrc/media/Events/2024/fifth-pqc-standardization-conference/documents/papers/pqm4-benchmarking-nist-addl-pq-sig-schemes.pdf

[35] C. Paquin, D. Stebila, and G. Tamvada, "A practical performance benchmark of post-quantum cryptography across heterogeneous computing environments," *Cryptography*, vol. 9, no. 2, p. 32, 2025.

[36] D. Commey, B. Appiah, G. S. Klogo, W. Bagyl-Bac, and J. Gadze, "Performance analysis and deployment of post-quantum cryptography in consumer electronics and embedded platforms," arXiv:2505.02239, 2025, preprint. [Online]. Available: https://arxiv.org/abs/2505.02239

[37] T. Liu, G. Ramachandran, and R. Jurdak, "Post-quantum cryptography for Internet of Things: A survey on performance and optimization," arXiv:2401.17538, 2024, preprint. [Online]. Available: https://arxiv.org/abs/2401.17538

[38] J. C. Patterson, W. J. Buchanan, and C. Turino, "Energy consumption framework and analysis of post-quantum key-generation on embedded devices," *Journal of Cybersecurity and Privacy*, vol. 5, no. 3, p. 42, 2025.

[39] A. Alif, K. F. Hasan, B. Palaniswamy, and M. M. Islam, "A framework for post-quantum migration in IoT-based healthcare systems," arXiv:2604.15584, 2026, preprint. [Online]. Available: https://arxiv.org/abs/2604.15584

[40] D. Sikeridis, P. Kampanakis *et al.*, "On deploying quantum-resistant cybersecurity in intelligent infrastructures," in *Proc. ACM Cloud Computing Security Workshop (CCSW)*, 2023. [Online]. Available: https://dl.acm.org/doi/fullHtml/10.1145/3600160.3605038

[41] NIST National Cybersecurity Center of Excellence (NCCoE), "CSWP 48: Mappings of migration to PQC project capabilities to risk framework documents," NIST, Tech. Rep. CSWP 48, Sep. 2025. [Online]. Available: https://www.nccoe.nist.gov/applied-cryptography/migration-to-pqc

[42] T. Sharma *et al.*, "Towards quantum resilience: Data-driven migration strategy design," arXiv:2505.05959, 2025, preprint. [Online]. Available: https://arxiv.org/abs/2505.05959

[43] N. S. Sammo, "Post-quantum cryptography migration in australian real-time payment infrastructure: A Monte Carlo simulation study of the new payments platform," arXiv:2605.02276, 2026, preprint. [Online]. Available: https://arxiv.org/abs/2605.02276

[44] A. Shaw, "Quantum-safe code auditing: LLM-assisted static analysis and quantum-aware risk scoring for post-quantum cryptography migration," arXiv:2604.00560, 2026, preprint. [Online]. Available: https://arxiv.org/abs/2604.00560

[45] E. Hirsch, K. Raab, T. J. Bauer, and D. Loebenberger, "Detecting cryptographically relevant software packages with collaborative LLMs," in *Proceedings of the 12th International Conference on Information Systems Security and Privacy (ICISSP 2026)*, Marbella, Spain, 2026, pp. 354–365.

[46] J. Pallarés de Bonrostro, A. I. González-Tablas, and M. I. González Vasco, "Empirical evaluation of large language models for migration of code fragments to post-quantum cryptography," arXiv:2606.07341, 2026, preprint. [Online]. Available: https://arxiv.org/abs/2606.07341

[47] J. Hohm, A. Heinemann, and A. Wiesmaier, "Towards a maturity model for crypto-agility assessment," in *Foundations and Practice of Security – 15th International Symposium (FPS 2022)*, ser. Lecture Notes in Computer Science, vol. 13877. Springer, 2023, pp. 104–119.

[48] M. Ounsworth, A. Truskovsky *et al.*, "Composite ML-DSA for use in internet PKI," IETF Internet-Draft draft-ounsworth-pq-composite-sigs, 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-sigs/

[49] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[50] C. O. Fritz, P. E. Morris, and J. J. Richler, "Effect size estimates: Current use, calculations, and interpretation," *Journal of Experimental Psychology: General*, vol. 141, no. 1, pp. 2–18, 2012.

[51] N. Cliff, "Dominance statistics: Ordinal analyses to answer ordinal questions," *Psychological Bulletin*, vol. 114, no. 3, pp. 494–509, 1993.

[52] M. Q. Patton, *Qualitative Research and Evaluation Methods*, 3rd ed. Thousand Oaks, CA: Sage Publications, 2002.

[53] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, "Performance analysis of post-quantum TLS 1.3," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 1823–1837, 2024.

[54] M. J. Kannwischer, D. J. Bernstein *et al.*, "KyberSlash: Exploiting the structure of NTT in Kyber," https://kyberslash.cr.yp.to, 2024.