Hybrid Post Quantum Cryptography Cryptosystems: ML-KEM (Kyber) Key Establishment with AES-256 AEAD

Jai Dayanand, Palak Gada, and Nnisarg Gada

Department of Computer Science
NMIMS University
Mumbai, India

Abstract. The rapid advancement of quantum computing threatens traditional public-key systems (RSA, ECC) by accelerating integer factorization and discrete logarithms. Post-Quantum Cryptography (PQC) offers schemes believed to resist quantum attacks, yet many introduce larger keys and higher latency. This paper proposes a hybrid model that uses a lattice-based Key Encapsulation Mechanism (KEM), e.g., CRYSTALS-Kyber / ML-KEM, to establish a shared secret and employs AES-256 for high-throughput bulk encryption. We outline an implementation agnostic workflow, discuss protocol integration (TLS-like handshakes), and provide indicative sizing/performance comparisons to motivate real-world feasibility. The approach balances near-term deployability with long-term security and provides a pragmatic on-ramp for PQC migration. We also address risk of 'store now, decrypt later', interoperability with legacy stacks, and operational considerations such as KDF transcript binding and AEAD non-ce management. Finally, we discuss deployment guidance aligned with NIST's transition roadmap and implications for resource-constrained environments.

Keywords: Post-Quantum Cryptography · Hybrid Cryptography · Lattice-Based KEM · ML-KEM / Kyber · AES-256 · TLS · Quantum Security

1 Introduction

Large-scale quantum computers would undermine classical public-key cryptography through Shor's algorithm, while Grover's algorithm provides quadratic speedups for key search against symmetric ciphers. As a result, RSA, classical Diffie–Hellman, and ECC are at risk; meanwhile, AES-256 remains practically robust by virtue of its key length. Lattice-based cryptography, particularly Learning With Errors (LWE) and Module-LWE constructions, underpins leading PQC KEMs such as CRYSTALS-Kyber (standardized as ML-KEM). However, PQC deployment can inflate key sizes and bandwidth and increase CPU cycles during handshakes. To address these transitional challenges, we advocate a hybrid architecture: use a lattice-based KEM solely for *ephemeral key establishment* and

rely on AES-256 (with an AEAD mode) for bulk encryption. This separates concerns, leveraging strengths of both paradigms while containing PQC overheads to the handshake phase.

Beyond the high-level motivation, two operational pressures drive this design. First, the *store-now-decrypt-later* (SNDL) threat model implies that adversaries can capture encrypted traffic today and retrospectively decrypt it once scalable quantum computers arrive. Deployments therefore need quantum-resistant *handshakes* immediately, even if data-plane ciphers remain unchanged. Second, migration must be incremental: organizations typically maintain heterogeneous stacks (legacy VPNs, IoT gateways, microservices) where a hard cutover to PQC is infeasible. Hybridization permits drop-in KEMs while preserving existing AEAD pipelines, key lifecycles, and compliance controls.

From a cryptographic engineering perspective, our focus is on (i) clean composition of a KEM-derived shared secret with a modern KDF (e.g., HKDF) to produce traffic keys bound to the handshake transcript; (ii) disciplined nonce management and rekeying for AES-256 AEAD to sustain high throughput under long-lived sessions; and (iii) authentication choices during transition (classical X.509 signatures vs. PQ signatures) without weakening security goals such as forward secrecy. We also discuss parameterization trade-offs (e.g., ML-KEM-768 vs. ML-KEM-1024) that balance security margins against bandwidth and CPU budgets in edge and data-center settings.

Finally, we outline protocol embedding patterns using a TLS-like handshake: the lattice KEM confines PQC cost to the control plane, while bulk encryption remains on AES-256, preserving performance-critical paths. We complement this with an indicative evaluation of artifact sizes and qualitative latency impacts, showing that the proposed hybrid retains near-baseline data-plane throughput while materially reducing SNDL risk.

2 Literature Review

The maturation of post-quantum cryptography (PQC) has progressed along two complementary axes: (i) protocol integration studies that embed PQ primitives into deployed key exchange protocols (e.g., TLS/VPN), and (ii) systems–security work addressing implementation hardness, hardware efficiency, and migration in constrained environments. Our work operationalizes these strands into a transport-agnostic hybrid blueprint—KEM $\rightarrow$ HKDF $\rightarrow$ AES-256 AEAD—that confines PQ cost to the control plane while preserving high-throughput symmetric record protection.

2.1 From PQ Key Exchange to Hybrid Handshakes

Foundational integration results showed that PQ key exchange can be slotted into TLS without re-architecting record protection. Bos et al. replaced (EC)DHE with a Ring-LWE construction and demonstrated feasibility in a standards-aligned handshake, catalyzing subsequent PQ-TLS experiments [14]. Building on

this, Döring and Geitz quantified real-world handshake overheads across network conditions, highlighting that WAN RTT dominates perceived latency while PQ compute contributes a modest incremental cost [12]. Mankowski et al. inspected Android's TLS landscape, mapping blockers to PQC uptake in mobile stacks (library availability, OS distribution cadence, and app-level control), which motivates a hybrid posture where PQ cost is amortized and deployability preserved [11]. Kupcova et al. contributed a reproducible PQ-TLS client–server framework, enabling end-to-end experimentation with ciphersuite negotiation, transcripts, and performance logging [13].

A key trajectory is the move from "pure PQ" to *hybrid* designs that compose PQ KEMs with classical mechanisms. Zafar and Iqbal integrate *code-based* PQC into SSL/TLS via an interoperable hybrid framework, validating multi-family composition (code-based with classical) and emphasizing interoperability during migration [1]. At higher assurance tiers, Aquina et al. show that hybrid PQC can coexist with quantum key distribution (QKD), diversifying control-plane key establishment across cryptographic and physical primitives for critical optical links [2]. In the data-confidentiality plane, Chittem and Pradhan's HQCE scheme combines classical and quantum-resistant ingredients while keeping bulk encryption symmetric, echoing our central thesis that AEAD should remain the fast path [3]. At the cellular layer, Turnip et al. survey hybrid PQC for 6G AKA, underscoring downgrade resistance, parameter agility, and backward compatibility—concerns directly mirrored in transport protocols [4]. Complementing this, Angom et al. survey lattice-based key establishment in depth, tracing the evolution of types and parameters that inform practical choices between ML-KEM-768 and ML-KEM-1024 under bandwidth/CPU budgets [8]. Sarıbaş and Tonyalı evaluate TLS 1.3 with Round-3 NIST KEMs/signatures on resource-constrained devices, reinforcing the value of isolating PQ cost to handshakes and keeping the data plane on accelerated AEAD [15].

2.2 Implementation Hardness: Side-Channels, Nonces, and Hardware Efficiency

Implementation security is pivotal. Jendral et al. demonstrate that a *single-trace* side-channel attack can break Kyber even with protections, highlighting the necessity of constant-time decapsulation, masked error paths, and robust randomness in any production handshake [5]. In response, Kuo et al. apply high-order Chebyshev filters to bolster Kyber's side-channel resistance in hardware, suggesting that board-level countermeasures can meaningfully shift the attack surface [6]. Nguyen et al. propose an area–time-efficient ML-KEM architecture, which can reduce handshake latency and energy per operation—critical for edge gateways and high-concurrency servers that must amortize PQ cost across many sessions [7].

On the symmetric plane, mismanaging AEAD nonces is a recurrent failure mode. Ali characterizes a generic nonce-misuse CPA attack on Ascon-128, illustrating how structural patterns under reuse degrade confidentiality; while our data plane uses AES-GCM, the lesson generalizes: strict per-direction counters,

rekey thresholds, and transcript-bound HKDF derivation are non-negotiable [9]. Yadav and Sharma introduce a randomized lightweight AEAD stream design for constrained settings, reflecting the diversity of symmetric options; nevertheless, AES-256-GCM remains the pragmatic default given widespread hardware acceleration and mature verification tooling [10].

2.3 Synthesis and Gap

The literature collectively supports a migration strategy that: (i) deploys *hybrid* handshakes to neutralize store-now-decrypt-later risk while safeguarding interoperability [1,4], (ii) isolates PQ overhead to the control plane and retains hardware-accelerated AEAD in the data plane to preserve throughput [12,15], and (iii) treats implementation hardness (constant-time KEM, nonce discipline, transcript binding) as first-class requirements [5,9]. Our contribution operationalizes these findings into an implementation-agnostic blueprint—KEM $\rightarrow$ HKDF $\rightarrow$ AES-256 AEAD—with parameterization guidance (ML-KEM-768 vs. 1024), downgrade-resistant negotiation, and deployment-level tuning (resumption, rekey policy, acceleration) aimed at immediate adoption in TLS/VPN and adjacent transports.

3 Methodology and Discussion

3.1 System Goals and Threat Model

Goals: (1) Quantum-resistant key establishment, (2) High data-plane throughput, (3) Minimal integration friction with existing stacks (e.g., TLS/VPN), (4) Phased migration where classical and PQC coexist.
Adversary: A Dolev–Yao network attacker capable of passive interception (store-now-decrypt-later) and active manipulation (message insertion, reordering, replay). We assume constant-time, side-channel-aware implementations and standard production mitigations (ASLR, hardened toolchains, verified randomness).

3.2 Cryptographic Primitives

KEM (IND-CCA2): A lattice-based KEM (e.g., ML-KEM/Kyber) providing *encapsulate/decapsulate* to agree on a high-entropy shared secret under chosen-ciphertext security. IND-CCA2 is critical to resist active adversaries during handshake.
KDF: HKDF (Extract-Expand) with domain-separated labels; inputs include the KEM shared secret, peer roles, and a running transcript hash to bind keys to the exact handshake context.
Symmetric Encryption: AES-256 in an AEAD mode (e.g., AES-256-GCM) with strict nonce discipline (per-connection monotonic counters or XoF-derived nonces) to guarantee confidentiality and integrity at high throughput.
Signatures (optional): PQ signatures (e.g., Dilithium) or classical signatures during transition for endpoint authentication and anti-impersonation.

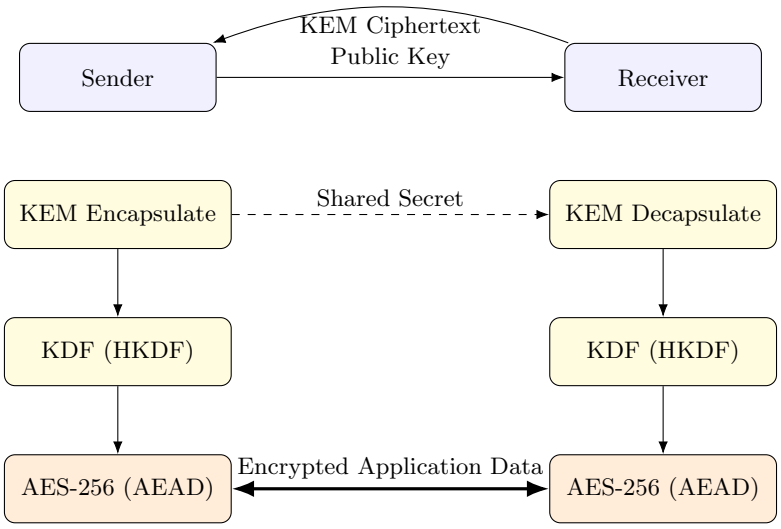

Fig. 1. Hybrid architecture: lattice KEM for key establishment; AES-256 AEAD for data.

3.3 Operational Flow

1. *KEM Setup:* The receiver publishes a KEM public key (fresh or rotated). Public-key distribution uses existing PKI or a pinned trust root.
2. *Encapsulation:* The sender encapsulates using the receiver's public key, yielding a KEM ciphertext and an agreed shared secret; the receiver decapsulates to recover the same secret.
3. *KDF:* Both parties run HKDF-Extract on the shared secret (salted with transcript hash), then HKDF-Expand to derive separate traffic keys, nonces, and exporter secrets.
4. *AEAD Channel:* Bulk data uses AES-256 AEAD with per-record nonces and sequence numbers; periodic rekeying can be scheduled by data volume or time.

3.4 System Architecture Diagram

Diagram rationale and relevance. Figure 1 separates the *control plane* (KEM + KDF) from the *data plane* (AEAD). This partition:

- confines the quantum-resistant cost to the handshake (single KEM round trip), keeping per-packet overhead unchanged,
- enforces transcript-bound key derivation via HKDF so that keys are inextricably tied to the session identity, roles, and negotiated parameters (preventing cross-protocol key reuse),

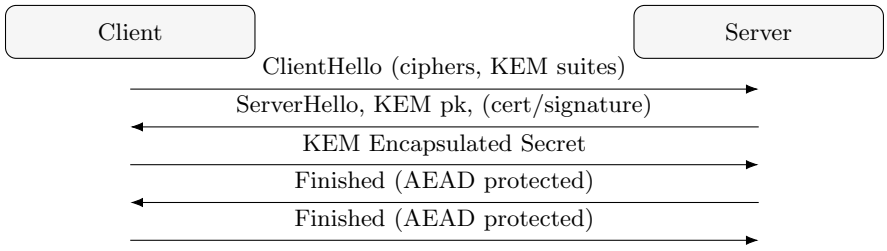

Fig. 2. Conceptual hybrid TLS-style handshake.

– permits independent lifecycle policies: aggressive rotation for KEM keys, volume/time-based rekeying for AEAD, and seamless swap of KEM parameter sets (e.g., ML-KEM-768 → ML-KEM-1024) without disturbing application encryption.

This mapping is directly relevant to TLS/VPN stacks where the bulk of CPU time is spent on data encryption; preserving AES-256 AEAD on the hot path maintains throughput while immediately mitigating store-now-decrypt-later risk via the KEM.

3.5 Hybrid TLS-Style Handshake Sketch

Handshake mapping and research considerations. Figure 2 follows the spirit of TLS 1.3 while substituting the (EC)DHE key share with a lattice KEM exchange:

– *Negotiation:* ClientHello advertises classical and PQC KEM suites; ServerHello selects a KEM suite and conveys (or certifies) its KEM public key. Downgrade resistance is ensured by hashing the negotiated suite into the transcript and including the transcript in Finished MACs.
– *Key schedule:* HKDF-Extract on the KEM shared secret (salted with transcript) prevents cross-session key reuse; HKDF-Expand derives independent client/server traffic keys and IVs. Exporter secrets enable application-layer keying without exposing the base secret.
– *Authentication:* During transition, servers may authenticate with classical X.509 signatures to retain ecosystem compatibility; deployments may upgrade to PQ signatures (e.g., Dilithium) when PKI/tooling support matures. Mutual authentication fits naturally by requesting client certificates after KEM completion.

– *Rekeying and resumption:* Rekey triggers (bytes/time) bound to AEAD sequence numbers mitigate nonce exhaustion. Session resumption can carry a PSK derived from exporter secrets, reducing future handshakes to PSK+KEM or PSK-only, depending on policy.

Research-wise, this hybridization preserves TLS 1.3's desirable properties (forward secrecy, key separation, transcript binding) while providing post-quantum security for the handshake. The data plane remains unchanged, simplifying performance validation and compliance audits.

3.6 Parameterization and Performance Trade-offs

Choosing ML-KEM-768 vs. ML-KEM-1024 reflects a bandwidth/CPU/security trade-off; the former reduces artifact sizes (public key ≈1184 B, ciphertext ≈1088 B) and CPU cost, whereas the latter raises the security margin with modest additional overhead. On commodity servers, KEM cost is amortized over long-lived sessions; on constrained IoT nodes, use of connection pooling, session resumption, and offloading (e.g., gateway-terminated KEM) can bound handshake frequency. Because AES-256 AEAD dominates the data plane, end-to-end throughput is effectively preserved.

3.7 Security Properties and Pitfalls

Transcript binding. All keying material derives from a transcript-hashed HKDF to prevent context confusion and downgrade.
Nonce discipline. AEAD nonces must never repeat under a fixed key; implementers should use monotonically increasing per-direction counters and enforce rekeying thresholds.
Randomness. KEM encapsulation requires high-quality randomness; deployments should monitor DRBG health and entropy sources.
Side channels. Constant-time decapsulation and masking of error paths are mandatory to avoid decryption oracles; production builds should be hardened and, ideally, fuzzed and formally verified at the interface level.
Compositional safety. No reuse of KEM shared secrets across protocols; HKDF labels must be domain-separated to avoid cross-protocol key compromise.

3.8 Implementation Guidance and Interoperability

For brownfield environments, a pragmatic path is: (i) enable hybrid ciphersuites alongside classical ones, (ii) prioritize PQC for high-sensitivity links (SNDL exposure), (iii) progressively expand PQ signatures as PKI support arrives, and (iv) benchmark platform-specific KEM handshakes to select parameters. Telemetry should track handshake latency, failure modes (decapsulation failures, replay), and rekey events; SLOs can then be tuned via resumption and batching without touching the AEAD pipeline.

Table 1. Indicative Comparison (Handshake vs. Data Plane)

Scheme	Public Key (B)	Ciphertext (B)	Relative Speed
RSA-2048	∼256	∼256	Moderate (handshake)
ECC P-256	∼64	∼64	Fast (handshake)
ML-KEM/Kyber768	∼1184	∼1088	Moderate (handshake)
AES-256 (AEAD)	32 key	N/A	Very fast (data)
Hybrid (KEM + AES)	KEM-sized	KEM-sized	Fast data-plane

4 Results and Evaluation

4.1 Indicative Sizes and Relative Costs

Representative parameters (e.g., Kyber768 / ML-KEM-768) yield compact but larger artifacts than classical ECC. Table 1 contrasts typical public-key/ciphertext sizes and qualitative speed tiers (handshakes vs. data-plane). Exact figures depend on implementations (constant-time decapsulation, vectorization) and platforms (x86/ARM, availability of AES-NI/VAES).

4.2 Latency Model and Handshake Cost

For a single round-trip handshake, total setup latency can be decomposed as:

$$T_{\text{setup}} = \text{RTT} + t_{\text{KEM}}^{\text{enc}} + t_{\text{KEM}}^{\text{dec}} + t_{\text{KDF}} + t_{\text{auth}} + \varepsilon, \tag{1}$$

where $t_{\text{KEM}}^{\text{enc}}$ and $t_{\text{KEM}}^{\text{dec}}$ are encapsulation/decapsulation times, t_{KDF} is HKDF cost, t_{auth} is authentication (classical or PQ signatures), and ε captures implementation overheads (parsing, transcript hashing). In WAN settings, **RTT dominates**, so the incremental cost of KEM typically contributes a small fraction of T_{setup}; in LAN/edge scenarios, KEM and signature costs are more visible but remain one-time per session.

4.3 Handshake vs. Data-Plane Throughput

Handshake overhead. Hybrid adds a one-time KEM exchange; for multi-minute sessions or high request concurrency, the amortized cost per byte is negligible. Session resumption (PSK) further suppresses repeat KEM cost.
Throughput. The data plane is AES-256 AEAD; with hardware acceleration (AES-NI/ARMv8 Crypto), line-rate encryption remains achievable. Rekeying can be scheduled on byte/time thresholds to prevent nonce reuse without perturbing steady-state throughput.
CPU profile. CPU spikes concentrate at connection setup (KEM + signatures). Under sustained load, CPU usage is dominated by AEAD, which benefits from vector instructions and parallelizable pipelines.

4.4 Bandwidth and Memory Footprint

Bandwidth. KEM public keys/ciphertexts increase handshake bytes relative to ECC, but *do not* affect per-record payload size. In typical web/VPN sessions, the incremental handshake bandwidth is < 2 KB relative to ECC, which is amortized after a few application records.
Memory. ML-KEM implementations maintain modest key and workspace buffers; AES-GCM state is small (keys, counters, GHASH tables). Memory pressure is governed primarily by application I/O buffers and TLS/VPN record queues.

4.5 Parameter Sensitivity

Choosing ML-KEM-768 vs. ML-KEM-1024 trades marginal bandwidth/CPU for a higher security margin. For latency-sensitive mobile/edge links, 768 is typically preferred; for long-lived data-center tunnels or highly sensitive links, 1024 can be justified. Authentication choice (classical vs. PQ signatures) has a larger impact on handshake size than KEM selection and can be phased in as PKI/tooling mature.

4.6 Robustness and Failure Modes

Decapsulation failures. Rare by design under IND-CCA2; when they occur, the transcript-bound KDF prevents key mismatch escalation. Implementations should surface explicit alerts and avoid silent fallback.
Downgrade resistance. Negotiated KEM/cipher suites must be hashed into the transcript and authenticated in the Finished messages to prevent attacker-induced suite regression.
Nonce discipline. Enforce per-direction counters with rekey thresholds to preclude nonce reuse under AEAD; log and refuse records on counter exhaustion.

4.7 Operational Evaluation: Tuning Guidance

Table 2 summarizes practical knobs that materially influence observed performance/security without redesigning the protocol.

4.8 Discussion

Handshake overhead. The incremental KEM cost is amortized quickly in typical sessions; PSK resumption suppresses repeat costs and stabilizes tail latency.
Throughput. Because the data plane remains AES-256 AEAD with hardware acceleration, the hybrid retains near-baseline throughput even at high concurrency; rekeying policies preserve nonce safety without disturbing steady state.
Migration path. Hybrid ciphersuites can co-exist with classical suites to enable staged rollouts and fallback. Authentication can remain classical initially and migrate to PQ signatures as PKI/tooling mature, without altering the data plane.

Table 2. Operational Tuning Recommendations (Hybrid KEM + AES-256 AEAD)

Dimension	Recommendation / Effect
KEM parameter set	ML-KEM-768 for latency-sensitive/mobile; ML-KEM-1024 for maximum margin on long-lived tunnels.
Session resumption	Prefer PSK-based resumption; reduces future handshakes to PSK(+KEM) and lowers tail latency.
AEAD record sizing	Use MTU-aware records to maximize throughput and minimize fragmentation; batch GHASH where possible.
Rekey policy	Rotate on bytes/time (e.g., 2^{32} records or 24h), synchronized with sequence counters; avoids nonce exhaustion.
CPU offload	Enable AES acceleration (AES-NI/VAES/ARMv8 Crypto) and, if available, KEM offload/optimized libraries.
Telemetry	Track handshake latency percentiles, decapsulation error rate, rekey events, and AEAD decryption failures; feed SLOs.

5 Conclusion

This work articulated and motivated a hybrid PQC architecture that confines post-quantum cost to the control plane—via a lattice-based KEM for ephemeral key establishment—while preserving AES-256 AEAD on the data plane for line-rate throughput. By separating handshake and bulk-encryption concerns, the design offers a practical migration path that mitigates store-now-decrypt-later risk without disrupting mature symmetric pipelines or operational tooling. We detailed a transcript-bound key schedule (HKDF with domain separation), downgrade-resistant negotiation, and strict nonce discipline, and we mapped the construction onto a TLS-style handshake to illustrate deployability in widely used transport stacks.

Our indicative evaluation highlights that, although KEM artifacts are larger than ECC and add marginal CPU and bandwidth at setup, the one-time handshake cost is quickly amortized in realistic session lifetimes; steady-state performance remains dominated by hardware-accelerated AEAD. Parameterization guidance (e.g., ML-KEM-768 vs. ML-KEM-1024) enables operators to tune security margins against latency and bandwidth budgets across data-center, WAN, and edge/IoT settings. Operationally, coexistence with classical suites supports phased rollout and safe fallback, while PSK-based resumption curbs repeat handshake overheads.

Limitations. We did not present implementation-specific microbenchmarks, formal proofs of composition beyond standard KEM+KDF+AEAD assumptions, or an analysis of side-channel leakage on diverse hardware targets. In addition, ecosystem readiness for PQ signatures and PKI integration remains uneven and may influence end-to-end handshake size and latency more than the KEM itself.

Future work. We will (i) produce end-to-end benchmarks on x86/ARM with and without acceleration, (ii) evaluate handshake tail latency under loss and reordering, (iii) integrate PQ signatures and study PKI operational impacts, (iv) formalize transcript-binding and exporter interfaces for application keys, (v) investigate constant-time decapsulation hardening and continuous RNG health monitoring, and (vi) extend the model to multi-party settings (group keying, gateways) and constrained nodes with gateway-terminated KEM. Collectively, these steps aim to turn the hybrid design into a rigorously validated blueprint for PQC adoption at scale.

References

1. Zafar, A., Iqbal, S.S.: Integrating code-based post-quantum cryptography into SSL/TLS protocols through an interoperable hybrid framework. *Discover Computing* **28**, 202 (2025). https://doi.org/10.1007/s10791-025-09735-7
2. Aquina, N., Rommel, S., Monroy, I.T.: Quantum secure communication using hybrid post-quantum cryptography and quantum key distribution. In: *2024 24th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4 (2024). https://doi.org/10.1109/ICTON62926.2024.10648124
3. Chittem, M.B., Pradhan, A.K.: Hybrid Quantum–Classical Encryption (HQCE) Algorithm: A Post-Quantum Secure Solution for Data Encryption. In: *2024 OITS International Conference on Information Technology (OCIT)*, pp. 321–326 (2024). https://doi.org/10.1109/OCIT65031.2024.00063
4. Turnip, T.N., Andersen, B., Vargas-Rosales, C.: Towards 6G Authentication and Key Agreement Protocol: A Survey on Hybrid Post Quantum Cryptography. *IEEE Communications Surveys & Tutorials* (2025). https://doi.org/10.1109/COMST.2025.3567439
5. Jendral, S., Ngo, K., Wang, R., Dubrova, E.: Breaking SCA-Protected CRYSTALS-Kyber with a Single Trace. In: *2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 70–73 (2024). https://doi.org/10.1109/HOST55342.2024.10545390
6. Kuo, C.-W., Hong, W.-C., Hsu, Y.-K., Hong, Y.-Y.: Enhancing Side-Channel Attack Resistance of Postquantum Cryptographic Algorithm CRYSTALS-Kyber Using High-Order Chebyshev Filters. In: *2025 9th International Conference on Cryptography, Security and Privacy (CSP)*, pp. 28–32 (2025). https://doi.org/10.1109/CSP66295.2025.00013
7. Nguyen, T.-H., Nguyen, N.T., Tran, D.M., Nguyen, P.H., Nguyen, T.C.: An Area–Time Efficient Hardware Architecture for ML-KEM Post-Quantum Cryptography Standard. *IEEE Access* **13**, 103834–103847 (2025). https://doi.org/10.1109/ACCESS.2025.3579379
8. Angom, A., Kar, N., Debbarma, T., Biswas, P.: A Survey on Lattice-Based Key Establishment Schemes: Types, Evolution and Advances. In: *2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, pp. 1–6 (2025). https://doi.org/10.1109/IATMSI64286.2025.10984484
9. Ali, M.T.: Generic CPA Decryption Attack on Ascon-128 in Nonce-Misuse Setting by Exploiting XOR Patterns. In: *2024 14th International Conference on Electrical Engineering (ICEENG)*, pp. 172–174 (2024). https://doi.org/10.1109/ICEENG58856.2024.10566378

10. Yadav, J., Sharma, D.K.: Implementation of an Efficient Randomized Lightweight Stream Cipher RRSC 128-AEAD. In: *2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT)*, pp. 23–28 (2024). https://doi.org/10.1109/IC2SDT62152.2024.10696085
11. Mankowski, D., Wiggers, T., Moonsamy, V.: TLS → Post-Quantum TLS: Inspecting the TLS Landscape for PQC Adoption on Android. In: *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 526–538 (2023). https://doi.org/10.1109/EuroSPW59978.2023.00065
12. Döring, R., Geitz, M.: Post-Quantum Cryptography in Use: Empirical Analysis of the TLS Handshake Performance. In: *NOMS 2022 – IEEE/IFIP Network Operations and Management Symposium*, pp. 1–5 (2022). https://doi.org/10.1109/NOMS54207.2022.9789913
13. Kupcova, E., Simko, J., Pleva, M., Drutarovsky, M.: Experimental Framework for Secure Post-Quantum TLS Client–Server Communication. In: *2024 International Symposium ELMAR*, pp. 213–216 (2024). https://doi.org/10.1109/ELMAR62909.2024.10694092
14. Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem. In: *2015 IEEE Symposium on Security and Privacy (SP)*, pp. 553–570 (2015). https://doi.org/10.1109/SP.2015.40
15. Sarıbaş, S., Tonyalı, S.: Performance Evaluation of TLS 1.3 Handshake on Resource-Constrained Devices Using NIST's Third Round Post-Quantum Key Encapsulation Mechanisms and Digital Signatures. In: *2022 7th International Conference on Computer Science and Engineering (UBMK)*, pp. 294–299 (2022). https://doi.org/10.1109/UBMK55850.2022.9919545