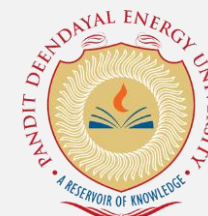


# 7<sup>th</sup> INTERNATIONAL CONFERENCE ON PUBLIC KEY INFRASTRUCTURE AND ITS APPLICATIONS (PKIA 2026)

SEPTEMBER 10-11<sup>th</sup>, 2026

**A Step Towards Migration from Classical to Quantum  
Secure TLS Handshake using PQC Primitives:  
Configurable to Multiple Security Levels**

Utkarsh Tiwari • Yagna Patel • Sanjeev Kumar Dwivedi  
School of Technology, Pandit Deendayal Energy University  
(PDEU), Gandhinagar, Gujarat, India



## Introduction & Motivation



Secure communication for defense, **BFSI** (Banking, Financial Services, and Insurance), healthcare, and critical infrastructure applications.

## Problem Statement & Contribution

- Classical TLS (RSA-2048 / ECDHE P-256) will not withstand quantum-capable adversaries
- PQC key exchange produces much larger keys & ciphertexts → bandwidth, latency and interoperability challenges for migration
- Most prior work targets performance benchmarking, hybrid transition or theoretical TLS integration — not primitive-level practical implementations
- Contribution: a lightweight, file-based client–server simulation exposing every Kyber primitive step of a TLS-like handshake
- NIST standardized Post-Quantum Cryptography (PQC); CRYSTALS-Kyber selected for key exchange (FIPS 203, ML-KEM)
- Design a configurable to Kyber512 / Kyber768 / Kyber1024 (NIST Level 1 / 3 / 5) with no code-structure changes

## Related Work

| Study                          | Focus                                                                                                         |
|--------------------------------|---------------------------------------------------------------------------------------------------------------|
| Henrich et al. [1]             | Benchmarked PQC KEMs in TLS 1.3 under varying network conditions; lattice-based schemes handshake efficiently |
| Zheng et al. [6]               | AVX-512 optimized ML-KEM (Kyber) implementation to cut TLS 1.3 handshake cost                                 |
| Sim et al. [3]                 | Zero-downtime bridge-server migration from classical → hybrid → PQC TLS                                       |
| Guo & Li [8], Gupta et al. [9] | Hardware accelerators for CRYSTALS-Kyber and CRYSTALS-Dilithium                                               |
| Liu et al. [10]                | Survey: lattice-based PQC (Kyber/Dilithium) is most practical for constrained IoT devices                     |

## Proposed Methodology — Kyber KEM Primitives

`indcpa_keypair(pk, sk)`

Key Generation — server creates a Kyber public / private key pair

`indcpa_enc(ct, message, pk, coins)`

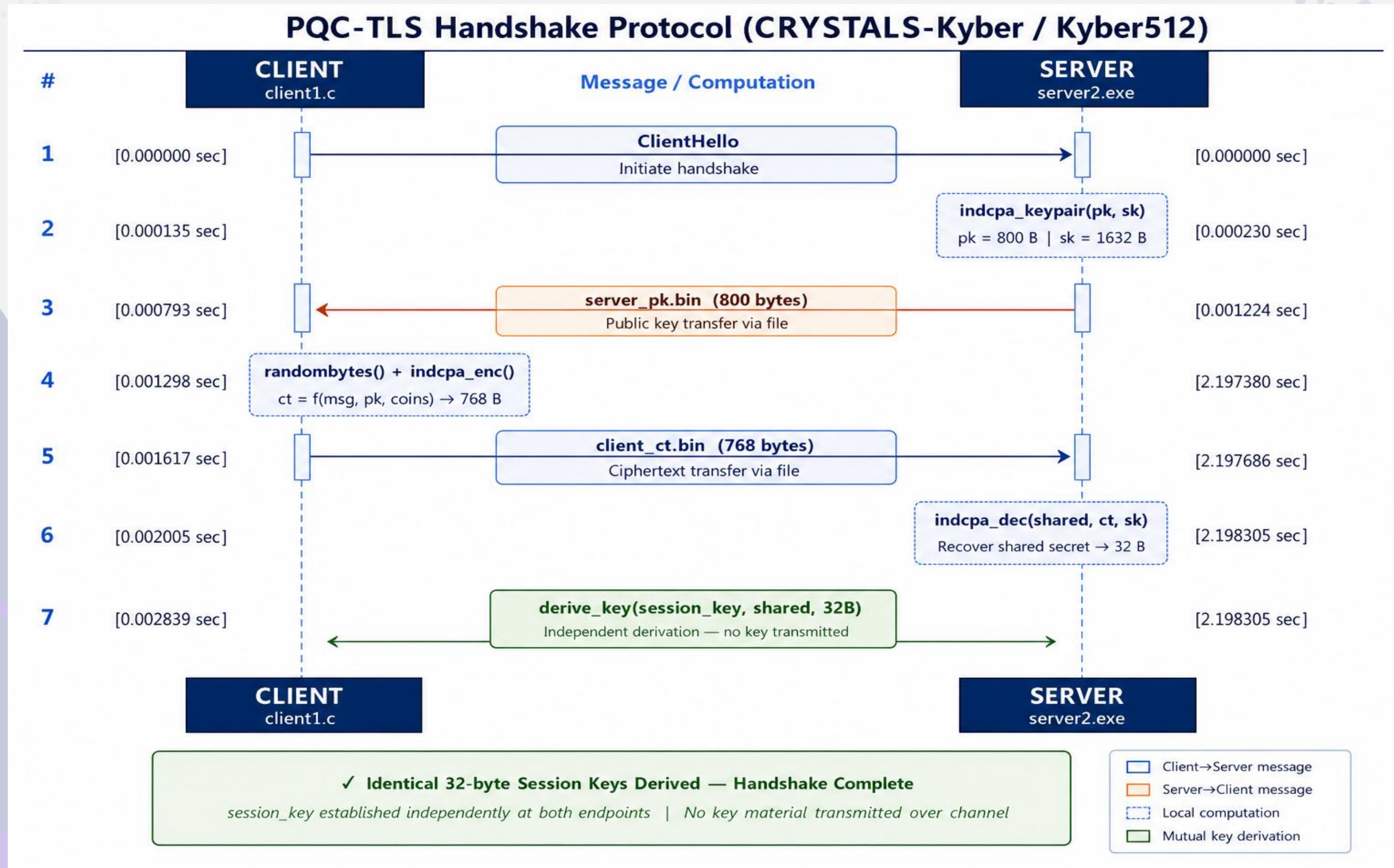
Encapsulation — client encrypts a random message with the server's public key, producing a ciphertext

`indcpa_dec(shared, ct, sk)`

Decapsulation — server recovers the shared secret from the ciphertext using its private key

*These lattice-based primitives (Module-LWE hardness) implement post-quantum key establishment; the same code path scales directly to Kyber768 / Kyber1024 for higher NIST security levels.*

# Handshake Workflow — Seven Steps

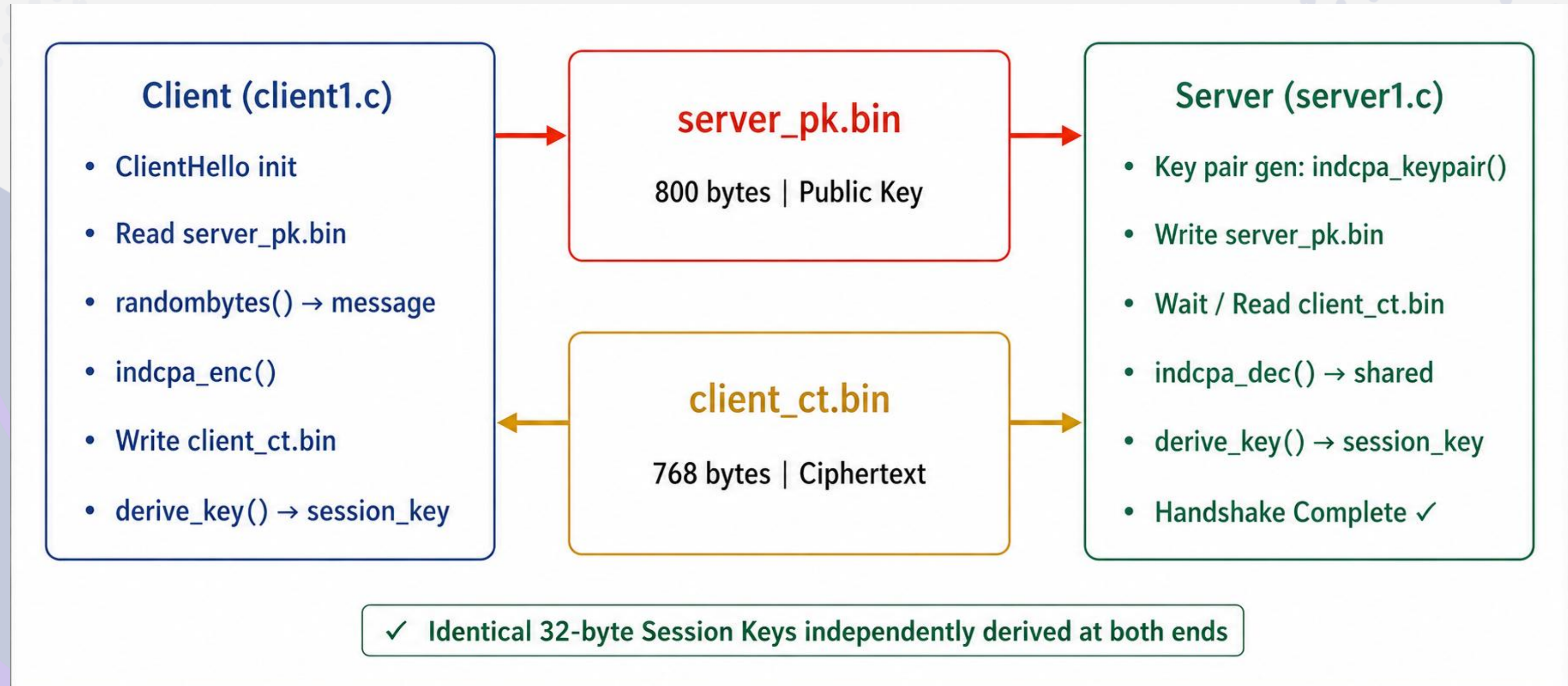


## Mapping with the TLS Handshake

| # | TLS Phase            | Server (server1.c)                         | Client (client1.c)                             |
|---|----------------------|--------------------------------------------|------------------------------------------------|
| 1 | ClientHello          | Log ClientHello receipt                    | Transmit ClientHello                           |
| 2 | ServerHello / KeyGen | indcpa_keypair() → pk (800 B), sk (1632 B) | Await server_pk.bin                            |
| 3 | ServerKeyExchange    | Write pk → server_pk.bin                   | Read server_pk.bin                             |
| 4 | ClientKeyExchange    | Poll for client_ct.bin                     | indcpa_enc() → ct (768 B); write client_ct.bin |
| 5 | Key Agreement        | indcpa_dec(ct, sk) → shared secret (32 B)  | Retain plaintext message (32 B)                |
| 6 | Key Derivation       | derive_key(shared) → session key (32 B)    | derive_key(message) → session key (32 B)       |
| 7 | Finished             | Print & verify session key; cleanup files  | Print & verify session key; terminate          |



## System Architecture — Client–Server Modules



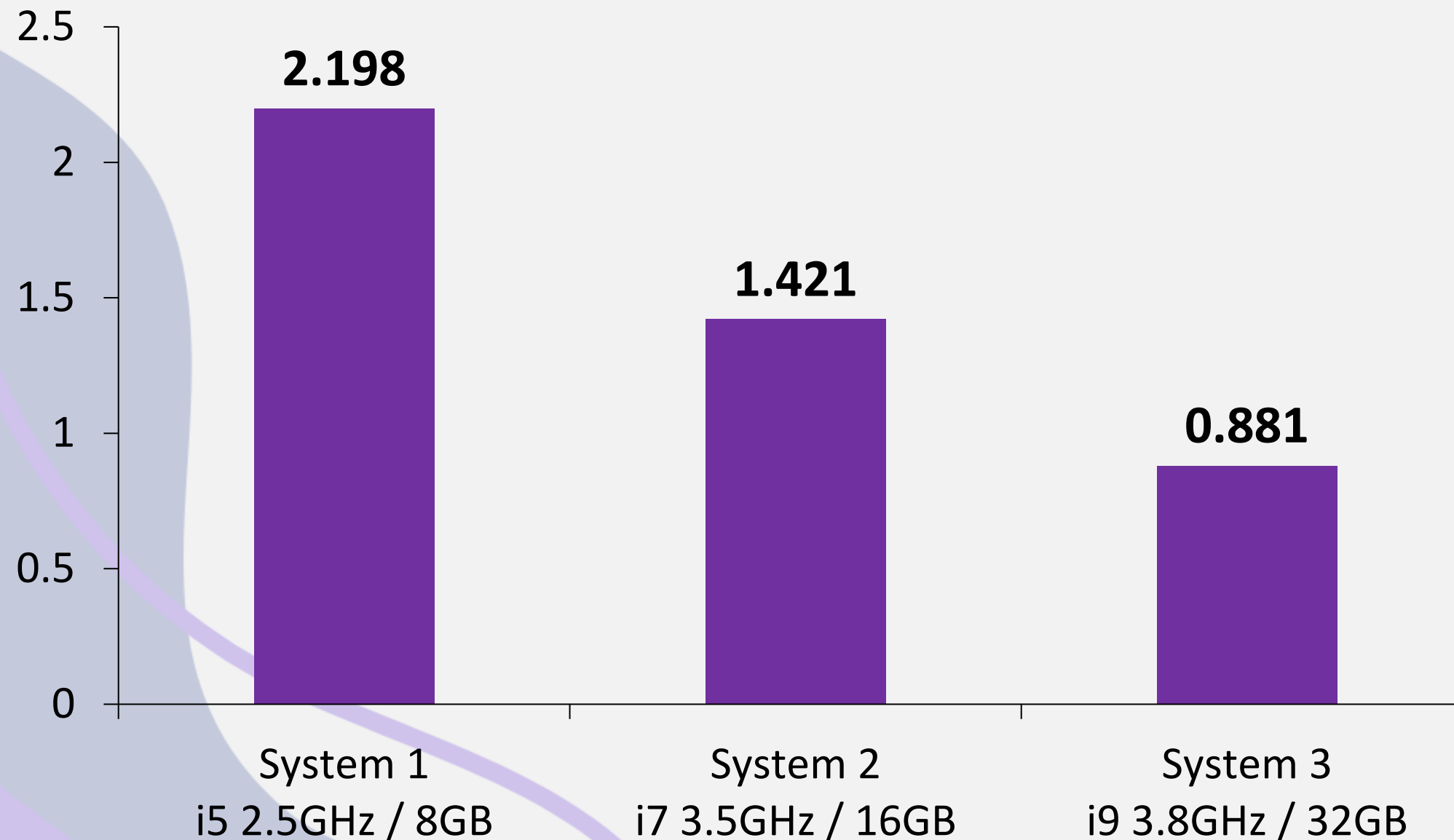
# Cryptographic Parameter Comparison: Kyber-512 vs Classical Algorithm

| Parameter                  | RSA-2048<br>(Classical) | ECDH<br>(P-256) | ECDH<br>(P-384) | Kyber512<br>(This Work) | Kyber768<br>(PQC) | Kyber1024<br>(PQC) |
|----------------------------|-------------------------|-----------------|-----------------|-------------------------|-------------------|--------------------|
| Security Level (bits)      | 112                     | 128             | 192             | 128 (PQC)               | 184 (PQC)         | 256 (PQC)          |
| Public Key Size (bytes)    | 256                     | 64              | 96              | 800 *                   | 1184              | 1568               |
| Ciphertext Size (bytes)    | 256                     | 64              | 96              | 768 *                   | 1088              | 1568               |
| Shared Secret Size (bytes) | 256                     | 32              | 48              | 32 *                    | 32                | 32                 |
| Private Key Size (bytes)   | 1192                    | 32              | 48              | 1632                    | 2400              | 3168               |
| Quantum Resistant          | ✗                       | ✗               | ✗               | ✓                       | ✓                 | ✓                  |
| Based on Hard Problem      | Factoring               | ECDLP           | ECDLP           | Module-LWE              | Module-LWE        | Module-LWE         |
| NIST Standardized          | ✓                       | ✓               | ✓               | ✓ (FIPS 203)            | ✓                 | ✓                  |
| Key Gen Speed              | Slow                    | Fast            | Medium          | Very Fast               | Fast              | Fast               |
| Enc/Dec Speed              | Slow                    | Fast            | Medium          | Very Fast               | Fast              | Fast               |

\* = values observed in this implementation | PQC = Post-Quantum Cryptography | LWE = Learning With Errors

## Results — Handshake Timing Analysis

Handshake completion time by hardware tier



- Kyber decryption (Step 5) dominates handshake time on every tier
- Higher clock speed & memory bandwidth cut total handshake time by  $\approx 60\%$ , i5  $\rightarrow$  i9
- No synchronization failures observed between client and server processes
- Delay before Step 5 is caused by process start-up ordering, not Kyber's computational cost
- Timings are application-level (not a standardized benchmarking harness) — illustrate relative, not absolute, performance.

## Comparison of Existing Research Works with Proposed Implementation

| No. Features                            | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | Proposed Work |
|-----------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|---------------|
| 1 TLS 1.3 PQC Integration               | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓    | ✓             |
| 2 Practical Implementation              | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓    | ✓             |
| 3 Kyber-based Implementation            | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓    | ✓             |
| 4 Direct Primitive-Level Implementation | ×   | ×   | P   | ×   | ×   | ×   | P   | ×   | ×   | ×    | ✓             |
| 5 Client–Server Handshake Simulation    | ×   | ×   | ×   | ×   | ×   | ×   | ×   | ×   | ×   | ×    | ✓             |
| 6 File-Based Communication              | ×   | ×   | ×   | ×   | ×   | ×   | ×   | ×   | ×   | ×    | ✓             |
| 7 Session Key Derivation Demonstration  | ×   | ×   | ✓   | ✓   | ×   | ×   | ✓   | ×   | ×   | ×    | ✓             |
| 8 Lightweight Educational Design        | ×   | ×   | ×   | ×   | ×   | ×   | ×   | ×   | ×   | ×    | ✓             |
| 9 End-to-End Handshake Workflow         | ×   | ×   | ×   | ×   | ×   | ×   | P   | ×   | ×   | ×    | ✓             |

**Legend:** ✓ = Fully Supported

× = Not Included

P = Partially Addressed

## Discussion, Limitations & Future Work

- Kyber artifacts are larger than classical ECDH (800/768 B vs  $\approx 33$  B) but the handshake remains practically fast and deterministic
- Current limitations: XOR-based key derivation is a placeholder for HKDF; no authentication, certificates, or TLS 1.3 socket layer yet
- Findings align with existing literature: structured lattice schemes (Kyber) give the best efficiency among PQC KEMs
- Future work: full TLS 1.3 integration, certificate-based authentication, hybrid classical+PQC handshakes, socket-based transport
- Further evaluation planned under realistic network conditions and on constrained IoT devices, benchmarked against OpenSSL / liboqs

## References:

- [1] J. Henrich, A. Heinemann, A. Wiesmaier, and N. Schmitt, “Performance impact of pqc kems on tls 1.3 under varying network characteristics,” in ISC 2023, 2023.
- [2] P. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” in Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124–134.
- [3] M. Sim et al., “Zero-downtime post-quantum tls 1.3 migration: A bridgeserver based approach,” Cryptology ePrint Archive, 2025.
- [4] K. Souvatzidaki and K. Limniotis, “Post-quantum key exchange in tls 1.3,” Cryptography, 2025.
- [5] A. Olushola and S. P. Meenakshi, “Design and implementation of an authenticated post-quantum session protocol,” Frontiers in Physics, 2026.
- [6] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, “Crystals - kyber: A cca-secure module-lattice-based kem,” in 2018 IEEE European Symposium on Security and Privacy (EuroSP), 2018, pp. 353–367.
- [7] J. Zheng et al., “Faster post-quantum tls 1.3 based on ml-kem: Implementation and assessment,” IEEE Access, 2024.
- [8] M. Sim et al., “Integrating and benchmarking kpqc in tls/x.509,” Electronics, 2025.
- [9] S. Paul, F. Schick, and J. Seedorf, “Tpm-based post-quantum cryptography: A case study,” in ARES 2021, 2021.
- [10] J. A. Montenegro, R. Rios, and J. Lopez-Cerezo, “A performance evaluation framework for post-quantum tls,” Future Generation Computer Systems, vol. 175, p. 108062, 2026.

# THANK YOU