Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd
Anatolian Science

ISSN,e-ISSN: 2548-1304
Volume: 2026
https://doi.org/10.53070/bbd.1898820
Research Paper

Hybrid ML-KEM in TLS 1.3: Performance Analysis on ARM64 Under Network Stress

Cemile İNCE*[1]

[1] Distance Education Research and Application Center, İnönü University, Malatya, Türkiye

(cemile.ince@inonu.edu.tr)

Received:Feb. 27, 2026 Accepted:Mar. 16, 2026 Published: Mar. 17, 2026

Abstract— In the post-quantum era, it is predicted that secure encryption algorithms like RSA and ECC will be broken within microseconds. In response, NIST has made the transition to post-quantum cryptography necessary by completing the ML-KEM standard (FIPS 203) in August 2024. However, integrating these new algorithms into the existing TLS 1.3 infrastructure raises some concerns, particularly in resource-constrained IoT devices where computing power and memory are limited. This paper evaluates the TLS 1.3 handshake performance of ML-KEM-512, ML-KEM-768, ML-KEM-1024, and hybrid X25519+ML-KEM-768 on a Raspberry Pi 4 (ARM Cortex-A72) in five different network scenarios (loopback, LAN (10 ms RTT), WAN (50 ms RTT), and packet loss rates of 1% and 5%). Experiments were performed using OpenSSL 3.x integrated with liboqs, with 100 iterations for each configuration. The results show that ML-KEM algorithms, which have high computational costs, introduce negligible computational overhead compared to the classic X25519 under low latency conditions, and base-state 1-RTT handshake times range from 11.3 to 13.3 ms. The ML-KEM 512 algorithm showed the best performance, particularly due to its small packet size. ML-KEM reached 180 ms with 5% loss, while X25519 reached 281 ms. It was also observed that session restart times consistently reduced handshake latency for all algorithms. In algorithm tests, ML-KEM 512 provided a 4.16-fold speedup at the base level. In WAN conditions, network RTT becomes the dominant bottleneck, and the choice of KEM algorithm becomes practically irrelevant. This demonstrates that ML-KEM algorithms are a usable standard even with limited hardware and that session restart is a significant example of optimization in IoT applications.

Keywords : *Post-Quantum Cryptography, ML-KEM, TLS 1.3, IoT Security, Raspberry Pi, Network Performance.*

1. Introduction

Existing algorithms such as RSA and Elliptic Curve Cryptography (ECC) provide adequate security levels. With the introduction of the Shor algorithm and the anticipated maturation of quantum computers, it is predicted that these classical encryption methods could be broken in seconds. Given the "store now, decrypt later" philosophy of quantum computers, securing network traffic with quantum-resistant protocols emerges as a critical need for long-term data security. (Montenegro et al., 2026). To overcome this challenge, the National Institute of Standards and Technology (NIST) launched the Post-Quantum Cryptography (PQC) Standardization Project. In August 2024, NIST formally standardized ML-KEM (Module Lattice Key Encapsulation Mechanism), a lattice-based algorithm derived from CRYSTALS-Kyber, with the publication of FIPS 203 (Nagy et al., 2025). ML-KEM is a lattice-based post-quantum cryptographic algorithm with three different security levels (ML-KEM-512, 768, and 1024). Currently, secure communication at the transport layer is primarily provided by the TLS protocol

over TCP and UDP. Integrating Post-Quantum Cryptography (PQC) into the existing Transport Layer Security (TLS) infrastructure presents significant experimental challenges (Aydeger et al., 2025). Post-Quantum Key Encapsulation Mechanisms (KEMs) involve significantly larger public keys and ciphertexts compared to classical cryptographic algorithms. For example, ML-KEM-768 uses 1184-byte public keys, significantly larger than the 32-byte keys used in the classical X25519 algorithm, leading to increased bandwidth overhead and computational latency. Internet of Things (IoT) applications are largely based on ARM-based architectures such as the Algorithms such as RSA and Elliptic Curve Cryptography (ECC), used in classical encryption, provide sufficient security levels (Kagai et al., 2025). With the introduction of the Shor algorithm and the expected developmental progress of quantum computers, it is predicted that these classical encryption methods can be broken in seconds. Switching network traffic to quantum-resistant protocols to protect against quantum computers operating on the "store now, decrypt later" principle has become an urgent requirement for long-term data security (Montenegro et al., 2026). To overcome this challenge, the National Institute of Standards and Technology (NIST) launched the Post-Quantum Cryptography (PQC) Standardization Project. In August 2024, NIST formally standardized ML-KEM (Module Lattice Key Encapsulation Mechanism), a lattice-based algorithm derived from CRYSTALS-Kyber, with the publication of FIPS 203. ML-KEM is a lattice-based post-quantum cryptographic algorithm with three different security levels (ML-KEM-512, 768, and 1024) (Jung et al., 2025). Currently, secure communication at the transport layer is primarily provided by the TLS protocol over TCP and UDP. Integrating Post-Quantum Cryptography (PQC) into the existing Transport Layer Security (TLS) infrastructure presents significant experimental challenges. Post-Quantum Key Encapsulation Mechanisms (KEMs) involve significantly larger public keys and ciphertexts compared to classical cryptographic algorithms. For example, ML-KEM-768 uses 1184-byte public keys, significantly larger than the 32-byte keys used in the classical X25519 algorithm, which increases bandwidth overhead and computational latency. Internet of Things (IoT) applications largely rely on ARM-based architectures such as Raspberry Pi, which operate under limited power, memory, and CPU constraints (Aissaoui et al., 2024). Much of the literature relies on simulated network environments rather than real-world wireless conditions where signal degradation and jitter are common. TLS 1.3, standardized by RFC 8446 (Chen et al., 2025) , is the fundamental protocol securing web communication today, and the vast majority of modern websites have adopted the HTTPS (Secure Hypertext Transfer Protocol) architecture. HTTPS is essentially an HTTP protocol built on top of the TLS (Transport Layer Security) layer. Its primary function is to derive session keys and authenticate endpoints. Its secondary function, the handshake phase, relies on efficient key exchange mechanisms like X25519, where the transaction time is typically around 2-3 milliseconds under ideal conditions. Algorithms like X25519, currently used for security purposes, are considered quite secure. However, the shift to post-quantum KEM (Key Encapsulation Mechanism) algorithms leads to a significant increase in computational overhead due to the large data sizes required by these algorithms. For example, while the key length for X25519 is only 32 bytes, the ML-KEM-768 algorithm requires 1184 bytes for public keys and 1088 bytes for ciphertext (Wei et al., n.d.). This size expansion results in a significant increase in both bandwidth consumption and computational latency. This paper presents an analysis of the hybrid TLS 1.3 performance for post-quantum algorithms on an ARM-based Raspberry Pi 4. Our work utilizes probabilistic delay modeling to discriminate between computational overhead and network transmission latency. The findings presented here highlight the critical importance of context-sensitive optimizations to ensure the feasibility of quantum-resistive protocols in various computing environments. This paper presents a systematic experimental study of post-quantum hybrid TLS performance on an ARM64 platform under real-world network conditions. Performance tests were conducted with a sample size of n=100 to ensure statistical significance. It remains unclear how practically the PQC algorithms standardized by NIST for the post-quantum era will work with TLS 1.3 under real network conditions. Overcoming this uncertainty has been the main motivation for this study.

The concrete contributions of the study can be summarized as follows: The comprehensive testing of the ML-KEM-512, ML-KEM-768, ML-KEM-1024, and hybrid X25519+ML-KEM-768 algorithms on real hardware (Raspberry Pi 4 / ARM Cortex-A72) has been added to the literature. Measurements made in five different network scenarios, such as loopback, LAN, WAN, and different packet loss rates, reveal not only the computational performance of the algorithms but also their real-world performance under network conditions. In addition, the experimental proof that session restart is a meaningful optimization method for IoT applications and the fact that the selection of the KEM algorithm in WAN

conditions shows performance close to classical algorithms in terms of latency are among the important findings of the study.

2. Related Work

2.1 ML-KEM Algorithms

To transition to post-quantum cryptography, NIST initiated the Cryptography (PQC) Standardization process, culminating in the publication of FIPS 203 in August 2024. ML-KEM (Module Lattice-Based Key Encapsulation Mechanism), derived from CRYSTALS-Kyber, provides three sets of parameters corresponding to different NIST security levels. These are listed in Table 1.

Table 1.ML-KEM algorithms NIST Security Levels

Parameter Set	NIST Category	Security Equivalent	Public Key	Ciphertext	Characteristics
ML-KEM-512	I	AES-128	800 B	768 B	Baseline quantum resistance; requires 2^{64} quantum operations against Grover's algorithm.
ML-KEM-768	III	AES-192	1,184 B	1,088 B	Pragmatic Choice: Optimal balance between security and performance for most deployments.
ML-KEM-1024	V	AES-256	1,568 B	1,568 B	High-security applications (e.g., classified communications, financial infrastructure).

Security levels reflect the computational resources required to break the system using classical or quantum attacks. ML-KEM's lattice-based structure is based on the difficulty of the Error Module Learning (MLWE) problem, which is resistant to the Shor algorithm and other known quantum threats (Son & Cheon, 2019). The adoption of hybrid constructions is primarily driven by the relative novelty of lattice-based cryptography compared to the decades of cryptanalysis performed on elliptic curve schemes (Rubio García et al., 2025). Hybrid algorithms can be used by combining classical and post-quantum algorithms. Thus, the advantages of both algorithms are combined into a single algorithm. This deep defense strategy is one of the methods used and adopted by major applications including Google Chrome, Cloudflare and AWS (Asif, 2021). The hybrid structure combines both algorithms and derives the session key through a key derivation function (KDF), which guarantees that an adversary would have to break both fundamental elements to compromise the connection (Rubio García et al., 2025). As a result, the system maintains its integrity as long as at least one of the component algorithms remains secure.

2.2 TLS 1.3 Handshake

The TLS 1.3 handshake protocol, standardized in RFC 8446, consists of three basic phases: key exchange, server parameters, and authentication. Key Exchange Phase: TLS 1.3 uses a Key Encapsulation Mechanism (KEM) to generate the shared secret key. The protocol flow involves three cryptographic steps:

KeyGen(): The client generates a key pair (public key PK, secret key PK in the ClientHello message.

Encaps(pk): The server generates a ciphertext CT and shared secret key SS using the client's public key and sends the CT in ServerHello.

Decaps(sk, ct): The client uses its secret key and the received ciphertext to recover the shared secret SS. For the classic X25519, these operations are completed in approximately 0.05–0.1 milliseconds in total. Although the cryptographic operations of ML-KEM-768 are competitive, larger message sizes lead to network transmission delays that dominate the overall handshake latency, especially in high-latency or bandwidth-constrained networks (Kampanakis & Weibel, n.d.; Mamun et al., n.d.; Sikeridis et al., 2022). The KEM-based key exchange flow is illustrated in Figure 1.

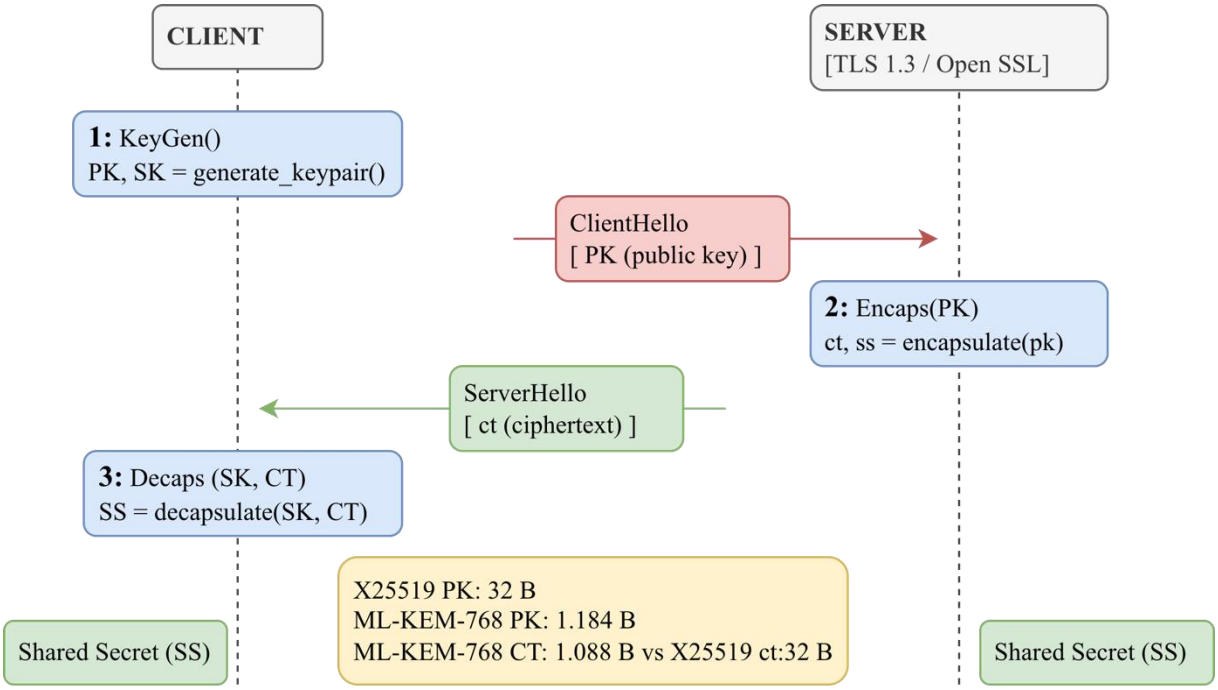

Figure 1. TLS 1.3 Handshake Protocol – KEM Protocol Flow

2.3 Literature Comparison Work

Sosnowski et al.(Sosnowski et al., 2023) demonstrated that ML-KEM in TLS 1.3 achieved handshake latencies that could compete with classical ECDH under network conditions with bandwidth constraints, latency, and packet loss. Their work concluded that Kyber outperformed classical algorithms at higher security levels, but their evaluation focused on x86-64 server hardware. Montenegro et al. [3] developed a generalized framework for post-quantum TLS evaluation by testing ML-KEM, hybrid KEMs, and multi-signature algorithms at NIST security levels I, III, and V. They identified hybrid KEMs as the configuration requiring the most bandwidth and noted that they have minimal computational overhead compared to pure post-quantum frameworks. The working frameworks were validated only in controlled network environments without performing actual WiFi or mobile network tests. PQC algorithms are algorithms that put extra overhead on the network due to their size. Various optimization methods have been proposed to reduce these overheads. Lim et al. (Lim et al., 2025) proposed PQTLS-AD, an optimization that pre-distributes ML-KEM public keys via DNS TXT records to reduce ClientHello size. This approach was reported to provide a 30% reduction in initial packet size, but it requires DNS infrastructure changes and is not compatible with link switching scenarios common in mobile networks. Kampanakis et al. (Kampanakis & Weibel, n.d.) proposed certificate compression and PKI constructs that do not require intermediate processing to reduce post-quantum signature overhead and achieved a 30-40% reduction in handshake size. These optimizations primarily focus on reducing bandwidth consumption and do not address the computational challenges faced by resource-constrained ARM devices.

The performance of PQC algorithms in large networks such as 5G/6G has also been a subject of interest. Hanna et al. (Hanna et al., 2024) observed that integrating post-quantum TLS into 5G network control planes increased the signal transmission overhead by 40-60% due to larger certificate chains. Deployment studies have shown the practical challenges of integrating post-quantum cryptography into the existing telecommunications infrastructure, and evaluations have revealed challenges even on x86-64 server platforms. The security levels of hybrid KEMs have been formally examined in many studies. The basic logic of these structures is actually quite simple: to ensure that the system can maintain its security even if one of the two components is broken by using classical and post-quantum algorithms together (Bindel et al., 2019). This approach has also found its place in practice; similar hybrid structures are preferred in Google Chrome's post-quantum TLS integration and Cloudflare's edge infrastructure (Schwabe et al., 2020). However, it is important to emphasize a crucial point here: formal security proofs

largely ignore the performance aspect. Especially on ARM processors with limited SIMD support, running two separate switch-switching algorithms sequentially leads to a significant computational overhead. This constitutes a major obstacle to the widespread adoption of the hybrid KEM approach across different hardware configurations.

Recent studies have begun to evaluate the QUIC protocol in conjunction with post-quantum KEMs. Montenegro et al.(Montenegro et al., 2026) achieved remarkable results in their comprehensive study comparing TLS and QUIC with post-quantum primitives: under ideal network conditions, QUIC reduces handshake time by 15-25% compared to TCP+TLS, while in environments where packet loss exceeds 2%, this gain can reach 30-50%. Kempf et al. (Kempf et al., 2024), on the other hand, examined post-quantum authentication mechanisms on QUIC; they pointed out that large post-quantum certificate chains pose serious problems in terms of PKI scalability. However, both studies are based on x86-64 architecture and simulated network environments; findings regarding real-world hardware and network conditions are still quite limited.

Table 2. Comprehensive Literature Summary on Post-Quantum Protocols

Study	Year	Protocol	KEM Type	Signature	Platform	Network	Focus
(Sosnowski et al.)	2023	TLS 1.3	Kyber	ECDSA	x86-64	Emulated	Performance
(Gonzalez and Wiggers)	2022	KEMTLS	ML-KEM-512	-	ARM (M4)	Localhost	Embedded
(Montenegro et al., 2026)	2026	TLS 1.3	MLKEM, Hybrid	ML-DSA	x86-64	Controlled	Framework
(Montenegro et al.)	2026	TLS+QUIC	ML-KEM	-	x86-64	Emulated	Comparison
(Hanna et al.)	2024	TLS 1.3	ML-KEM	ML-DSA	x86-64	5G	Control Plane
(Tasopoulos et al.)	2022	TLS 1.3	Kyber	-	ARM (RPi)	Localhost	IoT
(Kempf et al.)	2022	QUIC	-	Dilithium	x86-64	Cloud	Authentication
Proposed	**2026**	**TLS 1.3**	**X25519, ML-KEM (512/768/1024), Hybrid**	**RSA**	**ARM (RPi4)**	**Emulated**	**IoT / ARM**

Gonzalez et al. (Kim et al., 2025) compared KEMTLS with post-quantum TLS, an alternative protocol using KEM for authentication purposes in various embedded systems, including ARM Cortex-M4 microcontrollers. The results showed that ML-KEM 512 is feasible for IoT devices. Tasopoulos et al. (Tasopoulos et al., 2022) investigated ML-KEM in detail on Raspberry Pi and ESP32 platforms and reached an interesting finding: handshake latencies in ARM Cortex-A cores are 3-5 times higher compared to x86-64 due to limited SIMD (Single Instruction, Multiple Data) support and memory bandwidth constraints. However, the study only considered pure post-quantum KEMs; hybrid structures and real network environments were excluded. Dong et al. (Dong et al., n.d.) achieved a 40% speed increase in ML-KEM using a compilation-level vectorization method for embedded ARM devices. An important point that should not be overlooked in terms of current assessments is the NIST FIPS 203 standard, published in August 2024. Previous work [2][7] relied on Kyber implementations that were not yet standardized, differing from the final ML-KEM specification in terms of key generation procedures and ciphertext formats. Therefore, with the formal enactment of the standard, remeasuring post-quantum TLS performance through formal implementations has become a necessity, not a preference. Because compatibility with production environments can only be ensured in this way (National Institute of Standards and Technology (US), 2024). Table 2 summarizes the scope and contributions of relevant studies on post-quantum protocols.

Our study evaluates X25519, ML-KEM variants (512, 768, 1024), and the hybrid X25519+ML-KEM-768 on a Raspberry Pi 4 (ARM Cortex-A72) under emulated network conditions, identifying performance bottlenecks relevant to post-quantum TLS deployment on resource-constrained hardware.

3. Methods

In this study, a comprehensive performance evaluation environment was established on the Raspberry Pi 4 (RPi4) ARM64 architecture to measure the effect of ML-KEM algorithms, a type of post-quantum cryptography (PQC), on TLS 1.3 handshake times. The main reason for choosing the RPi4 as the experimental platform is to understand how PQC algorithms behave in practice in limited hardware environments such as IoT and embedded systems. The hardware specifications and algorithms used are given in Table 3.

Table 3. Experimental Setup and Test Parameters

Parameter	**Value**
Platform	Raspberry Pi 4 Model B (4 GB RAM)
Processor	Broadcom BCM2711, Cortex-A72 ARM64 @ 1.8 GHz
Operating System	Raspberry Pi OS 64-bit (Debian 12 Bookworm)
TLS Library	OpenSSL 3.x + liboqs integration
KEM Algorithms Tested	X25519, ML-KEM 512, ML-KEM 768, ML-KEM 1024, X25519+MLKEM768
Measurement Repetitions	100 runs (median value reported)
Network Scenarios	Loopback (0 ms), LAN (10 ms RTT), WAN (50 ms RTT)
Packet Loss Scenarios	1% and 5% (emulated via tc netem)
Measured Metrics	1-RTT handshake latency, Session Resumption latency

The basic encapsulation mechanisms (EEMs) used and classical methods are: the classical curve-based X25519, the three security levels of the ML-KEM standard ML-KEM 512, ML-KEM 768 and ML-EEM 1024, and the hybrid mode X25519+MLKEM768. This selection is based on a direct comparison of pure classical, pure post-quantum algorithms and hybrid approaches. TLS 1.3 handshake delay measurements were performed for two different connection tests: 1-RTT (full connection) requiring a full handshake and Session Restart based on session reuse. For each KEM algorithm, the delay gain provided by session reuse was calculated by measuring these two modes separately.Three different tests were conducted to test real-world network conditions. In the first test, only processor load was measured with network latency kept at zero. In the LAN scenario, a fixed 10 ms RTT latency was applied using the tc netem tool. In WAN tests, wide area network conditions were tested with a 50 ms RTT latency. In addition, 1% and 5% loss rates were tested to examine the effect of packet loss on handshake latency. To increase statistical reliability, each measurement was repeated at least 100 times. Median latency values were reported to prevent the results from being affected by extreme values caused by retransmission events. All tests were performed using the liboqs library integrated into OpenSSL 3.x.

4. Result and Discussion

This section presents TLS 1.3 handshake latency measurements obtained from four different network tests and interprets the differences between the algorithms. 1-RTT latency values and session reuse rates for all tests are summarized in Table 4.

Figure 2 shows the baseline latency values obtained in the loopback environment and the speed gains provided by session reuse. Examining the 1-RTT values, it is seen that all algorithms exhibit similar performance: X25519 at 11.7 ms, ML-KEM 512 at 11.3 ms, ML-KEM 768 at 11.5 ms, and ML-KEM 1024 at 11.8 ms. The hybrid mode X25519+MLKEM768 showed the highest latency at 13.3 ms due to the dual KEM operation. The highest speed gain in session reuse was achieved in ML-KEM 512 at 4.16 times, while this ratio decreased to 2.92 times in hybrid mode. Table 5 also provides the numerical equivalents of the values in Figure 2.

Table 4. TLS 1.3 1-RTT Latency Comparison (ms) (Best values highlighted in green)

KEM Algorithm	Baseline (ms)	LAN 10ms (ms)	WAN 50ms (ms)	Loss %1 (ms)	Loss %5 (ms)	Resumption Speedup
X25519	11.7	82	321	105	281	**3.79x**
ML-KEM 512	11.3	87	346	94	180	**4.16x**
ML-KEM 768	11.5	82	346	121	264	**3.84x**
ML-KEM 1024	11.8	88	346	150	213	**3.68x**
X25519+MLKEM768	13.3	84	346	134	217	**2.92x**

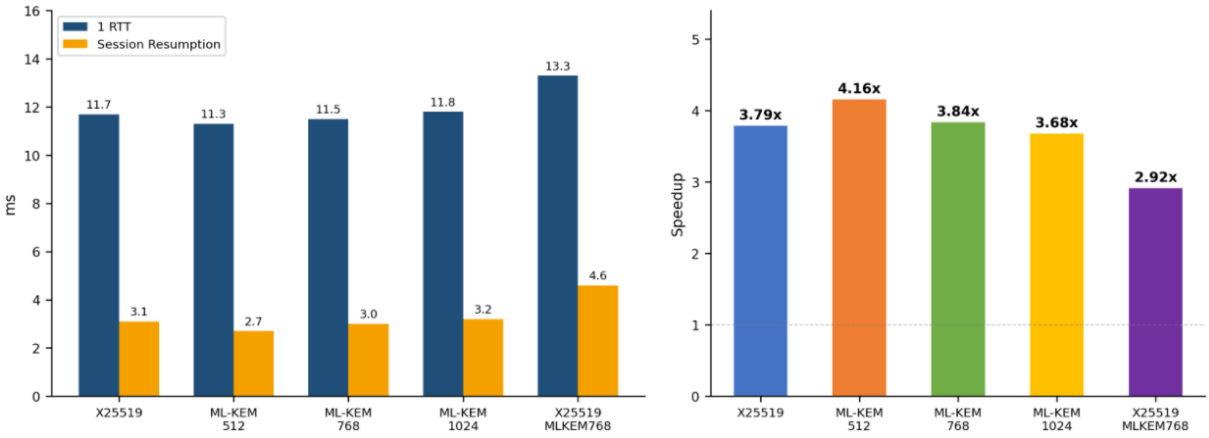

Figure 2. TLS 1.3 Loopback Latency and Session Reuse Speed Gains – RPi4 ARM64

Table 5. Post-Quantum KEM Latency and Resumption Performance

KEM	1-RTT (ms)	Reuse (ms)	Speedup
X25519	11.7	3.1	3.79x
ML-KEM 512	11.3	2.7	4.16x
ML-KEM 768	11.5	3.0	3.84x
ML-KEM 1024	11.8	3.2	3.68x
X25519 MLKEM768	13.3	4.6	2.92x

Figure 3 illustrates how the handshake delay changes with network RTT. As expected, as network latency increases, the absolute differences between the algorithms become increasingly insignificant. In the 50 ms WAN condition, 1-RTT delays reached the range of 320–346 ms; since the vast majority of these values are due to network latency, the effect of KEM algorithm selection was limited to only a few milliseconds. In the session reuse scenario, even in the WAN condition, the delays of all algorithms were very close to each other, and the difference between them narrowed to the limit of statistical significance.

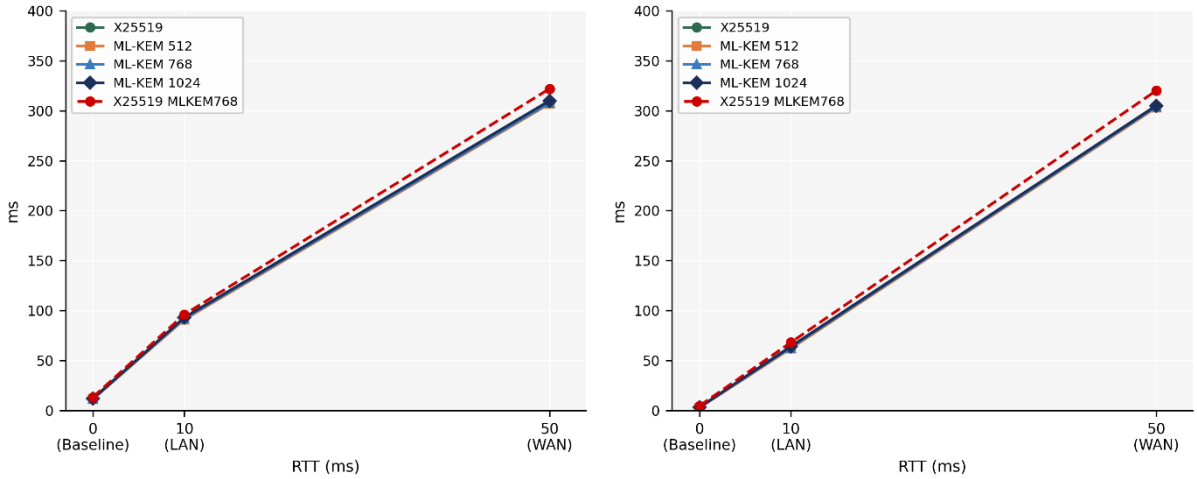

Figure 3. TLS 1.3 Latency – Network RTT Relationship – RPi4 ARM64

Figure 4 illustrates the significant negative impact of packet loss on handshake performance. At a 1% loss rate, delays increased significantly; at a 5% loss rate, X25519 reached 281 ms, while ML-KEM 512 recorded the lowest value among all algorithms at 180 ms. This result shows that ML-KEM 512 is less affected by packet loss thanks to its smaller handshake message sizes. ML-KEM 768 achieved 264 ms, ML-KEM 1024 213 ms, and in hybrid mode, 217 ms.

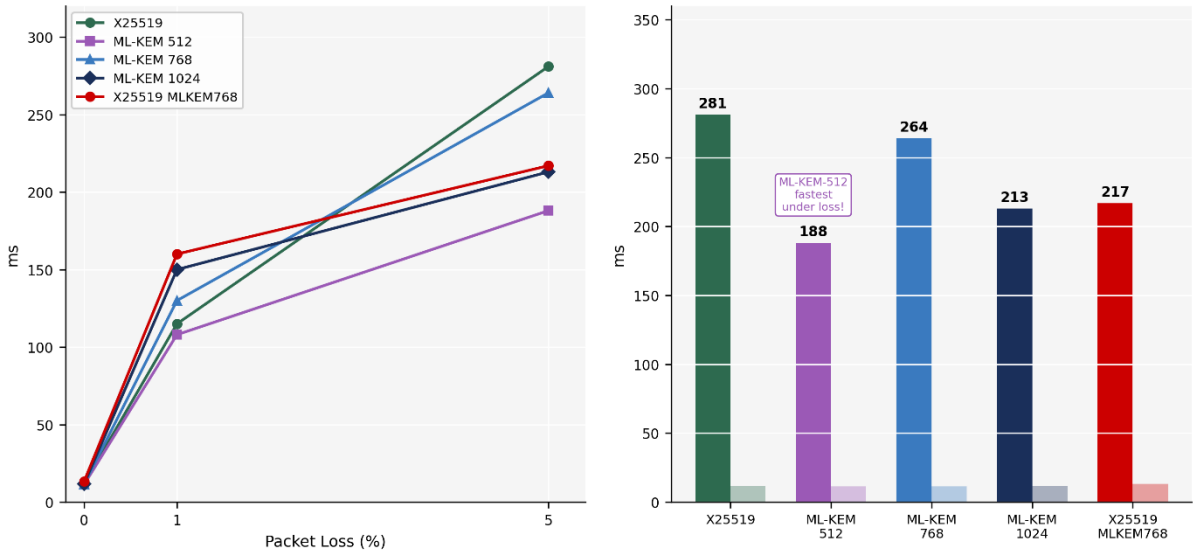

Figure 4. TLS 1.3 Latency under pocket loss

Figure 5 is a heat map visualization showing five different tests and the performance of all algorithms simultaneously. While clearly demonstrating how 1-RTT latencies increase in the WAN and loss scenarios, it also confirms that session reuse is an effective optimization tool outside of the WAN. In the WAN scenario, the difference between session reuse and full handshake almost disappears, revealing that network latency completely outweighs the computational cost. It almost surpasses the computational costs of PQC algorithms.

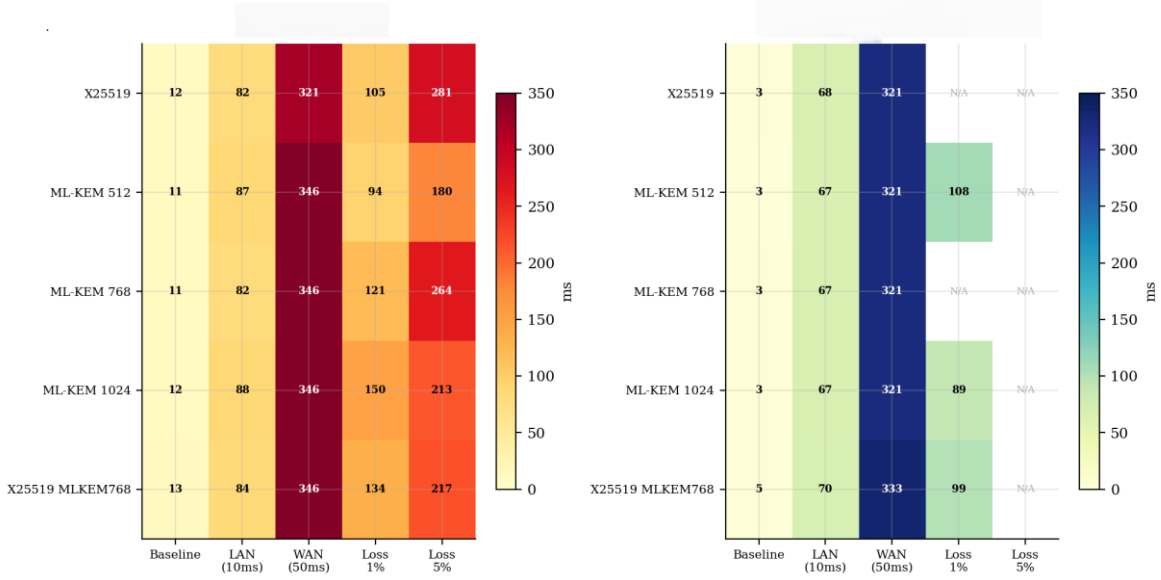

Figure 5. TLS 1.3 KEM Performance Across All Network Scenarios – RPi4 ARM64

Figure 5 presents a comparative analysis of the 1-RTT and Session Restart algorithms. In the baseline condition with no network latency, the gain from session reuse is highest because computational cost plays a more decisive role. While the gain in LAN tests is still significant, it is limited to 1.2-1.3 times. When switching to WAN conditions, this gain largely disappears, and the ratio drops to 1.0-1.1 times across all algorithms.

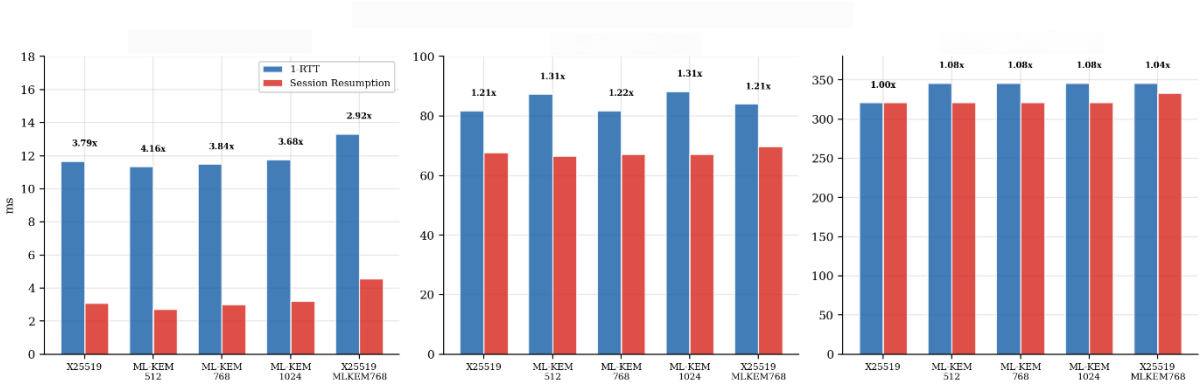

Figure 6. TLS 1.3-1-RTT vs. session resumption

Figure 6 compares the latencies of 1-RTT and Session Resumption for five different key exchange algorithms (X25519, ML-KEM 512/768/1024, and hybrid X25519 MLKEM768) in the TLS 1.3 handshake protocol across three network scenarios. In a loopback environment, Session Resumption offers a significant speed advantage of 3.68x–4.16x compared to 1-RTT, but this gain decreases to 1.21x–1.31x under LAN (10ms RTT) conditions and almost disappears (1.00x–1.08x) under WAN (50ms RTT) conditions. This shows that, regardless of algorithm choice, the advantage provided by Session Resumption becomes increasingly insignificant as network latency increases, and real performance gains are only achievable in low-latency environments.

Overall, it has been observed that even on limited hardware like the RPi4, ML-KEM algorithms do not significantly increase TLS 1.3 handshake times. The main contributor to latency appears to be network RTT and packet loss, independent of the algorithm used. Therefore, it can be said that switching to PQC in today's systems with sufficient processing power would have a negligible impact on latency.

5. Conclusion

This study investigated the performance impact of both post-quantum cryptography algorithms, namely ML-KEM and the classical X25519 algorithm, on the TLS 1.3 protocol using a Raspberry Pi 4 ARM64 platform. The results showed that ML-KEM-based PQC algorithms exhibited performance close to the classical X25519 algorithm in real-world applications under real network conditions. ML-KEM 512 demonstrated the best performance among all algorithms. With a latency of 180 ms at a 5% loss rate, significantly lower than X25519's 281 ms latency, this algorithm proved to be more robust to network conditions thanks to its small message sizes. This result shows that PQC can offer an additional advantage, especially in wireless networks and lossy channel environments. Session reuse was validated as a robust optimization mechanism for both classical and post-quantum algorithms. In the baseline condition, the 4.16x gain of ML-KEM 512 demonstrated how critical this technique is in IoT and distributed system scenarios with high repetitive connections. It is important that all algorithms exhibit similar latencies in the WAN scenario, with the difference between them falling below 25 milliseconds. This is because, in high-latency networks, the choice of KEM algorithm becomes almost irrelevant, and the bottleneck shifts to the network's RTT. This clearly shows that PQC transition does not create latency concerns for wide-area networks. The fact that the hybrid mode X25519+MLKEM768 produces the highest latency in the baseline condition and lags behind pure ML-KEM variants in loss scenarios reveals that the hybrid approach is not always the best solution. However, it can be considered a reasonable compromise for transition scenarios that need to provide protection against both classical and quantum attacks. In conclusion, this study has experimentally demonstrated that ML-KEM family algorithms are practically applicable even on limited hardware such as RPi4 and introduce an acceptable level of overhead to the TLS 1.3 handshake process. This provides a strong technical rationale for transitioning to PQC, particularly in IoT, smart infrastructure, and embedded systems applications. Future work should aim to support these findings with comprehensive comparisons across different ARM architectures, RISC-V platforms, and real-world network connections.

References

Aissaoui, R., Deneuville, J.-C., Guerber, C., & Pirovano, A. (2024). A Performant Quantum-Resistant KEM for Constrained Hardware: Optimized HQC: *Proceedings of the 21st International Conference on Security and Cryptography*, 668–673. https://doi.org/10.5220/0012757800003767

Asif, R. (2021). Post-Quantum Cryptosystems for Internet-of-Things: A Survey on Lattice-Based Algorithms. *IoT*, *2*(1), 71–91. https://doi.org/10.3390/iot2010005

Aydeger, A., Hoque, S., & Zeydan, E. (2025). Challenges of DNS in the Post-Quantum Era: Improving Security with Post-Quantum TLS. *Infocommunications Journal*, *17*(3), 11–21. https://doi.org/10.36244/ICJ.2025.3.2

Bindel, N., Brendel, J., Fischlin, M., Goncalves, B., & Stebila, D. (2019). Hybrid Key Encapsulation Mechanisms and Authenticated Key Exchange. In J. Ding & R. Steinwandt (Eds.), *Post-Quantum Cryptography* (Vol. 11505, pp. 206–226). Springer International Publishing. https://doi.org/10.1007/978-3-030-25510-7_12

Chen, J., Peng, W., Wang, Y., & Bian, Y. (2025). On the Security and Efficiency of TLS 1.3 Handshake with Hybrid Key Exchange from CPA-Secure KEMs. *Entropy*, *27*(12), 1242. https://doi.org/10.3390/e27121242

Dong, J., Hou, Y., Wang, S., Sha, L., Xiao, F., Dong, Z., & Lin, J. (n.d.). *HIGH: Harnessing GPU Parallelism for Optimized HQC Performance*.

Gonzalez, R., & Wiggers, T. (2022). KEMTLS vs. Post-quantum TLS: Performance on Embedded Systems. In L. Batina, S. Picek, & M. Mondal (Eds.), *Security, Privacy, and Applied Cryptography Engineering* (Vol. 13783, pp. 99–117). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-22829-2_6

Hanna, Y., Pineda, D., Veksler, M., Paudel, M., Akkaya, K., Anastasova, M., & Azarderakhsh, R. (2024). Integrating Post-Quantum TLS into the Control Plane of 5G Networks. *2024 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, 1–8. https://doi.org/10.1109/IPCCC59868.2024.10850437

Jung, H., Dang Truong, Q., & Lee, H. (2025). Highly-Efficient Hardware Architecture for ML-KEM PQC Standard. *IEEE Open Journal of Circuits and Systems*, *6*, 356–369. https://doi.org/10.1109/OJCAS.2025.3591136

Kagai, F., Branch, P., But, J., & Allen, R. (2025). Harvest-Now, Decrypt-Later: A Temporal Cybersecurity Risk in the Quantum Transition. *Telecom*, *6*(4), 100. https://doi.org/10.3390/telecom6040100

Kampanakis, P., & Weibel, A. (n.d.). *(Yet another) Analysis of MLWE's performance impact on specific TLS Use-cases*.

Kempf, M., Gauder, N., Jaeger, B., Zirngibl, J., & Carle, G. (2024). A Quantum of QUIC: Dissecting Cryptography with Post-Quantum Insights. *2024 IFIP Networking Conference (IFIP Networking)*, 195–203. https://doi.org/10.23919/IFIPNetworking62109.2024.10619916

Kim, M.-S., Jeon, S.-B., & Kim, S. (2025). SRAM based Gaussian noise generation for post quantum cryptography. *Scientific Reports*, *15*(1), 43573. https://doi.org/10.1038/s41598-025-27929-3

Lim, S., Lee, H., Jeong, G., & Kwon, T. T. (2025). PQTLS-AD: Post-Quantum TLS Accelerated with DNS. *2025 34th International Conference on Computer Communications and Networks (ICCCN)*, 1–6. https://doi.org/10.1109/ICCCN65249.2025.11133832

Mamun, A. A., Abrar, A., Rahman, M., Salek, M. S., & Chowdhury, M. (n.d.). *Post-Quantum Cryptography for Intelligent Transportation Systems: An Implementation-Focused Review*.

Montenegro, J. A., Rios, R., & Bonilla, J. (2026). Comparative analysis of post-quantum handshake performance in QUIC and TLS protocols. *Computer Networks*, *275*, 111957. https://doi.org/10.1016/j.comnet.2025.111957

Montenegro, J. A., Rios, R., & López-Cerezo, J. (n.d.). *A Performance Evaluation Framework for Post-Quantum TLS*.

Nagy, N., Alnemer, S., Alshuhail, L. M., Alobiad, H., Almulla, T., Alrumaihi, F. A., Ghadra, N., & Nagy, M. (2025). Module-Lattice-Based Key-Encapsulation Mechanism Performance Measurements. *Sci*, *7*(3), 91. https://doi.org/10.3390/sci7030091

National Institute of Standards and Technology (US). (2024). *Module-lattice-based key-encapsulation mechanism standard* (NIST FIPS 203; p. NIST FIPS 203). National Institute of Standards and Technology (U.S.). https://doi.org/10.6028/NIST.FIPS.203

Rubio García, C., Cano Aguilera, A., Stan, C., Vegas Olmos, J. J., Rommel, S., & Tafur Monroy, I. (2025). Enhanced Network Security Protocols for the Quantum Era: Combining Classical and Post-Quantum Cryptography, and Quantum Key Distribution. *IEEE Journal on Selected Areas in Communications*, *43*(8), 2765–2781. https://doi.org/10.1109/JSAC.2025.3568011

Schwabe, P., Stebila, D., & Wiggers, T. (2020). Post-Quantum TLS Without Handshake Signatures. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 1461–1480. https://doi.org/10.1145/3372297.3423350

Sikeridis, D., Huntley, S., Ott, D., & Devetsikiotis, M. (2022). Intermediate certificate suppression in post-quantum TLS: An approximate membership querying approach. *Proceedings of the 18th*

International Conference on Emerging Networking EXperiments and Technologies, 35–42. https://doi.org/10.1145/3555050.3569127

Son, Y., & Cheon, J. H. (2019). Revisiting the Hybrid Attack on Sparse Secret LWE and Application to HE Parameters. *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 11–20. https://doi.org/10.1145/3338469.3358941

Sosnowski, M., Wiedner, F., Hauser, E., Steger, L., Schoinianakis, D., Gallenmüller, S., & Carle, G. (2023). The Performance of Post-Quantum TLS 1.3. *Companion of the 19th International Conference on Emerging Networking EXperiments and Technologies*, 19–27. https://doi.org/10.1145/3624354.3630585

Tasopoulos, G., Li, J., Fournaris, A. P., Zhao, R. K., Sakzad, A., & Steinfeld, R. (2022). Performance Evaluation of Post-Quantum TLS 1.3 on Resource-Constrained Embedded Systems. In C. Su, D. Gritzalis, & V. Piuri (Eds.), *Information Security Practice and Experience* (Vol. 13620, pp. 432–451). Springer International Publishing. https://doi.org/10.1007/978-3-031-21280-2_24

Wei, H., Li, W., Shen, S., Yang, H., & Zhao, Y. (n.d.). *Optimized Implementation of ML-KEM on ARMv9-A with SVE2 and SME*.