

Role-Specific Signature Placement in TLS 1.3: Cost Propagation and PQC Migration

Moonsun Heo¹[0009–0005–6631–199X], Subeen Cho¹[0009–0004–4352–8021],
Jongbeom Ahn¹[0009–0005–7205–8409], Minseo Kim¹[0009–0008–1979–5778], and
Hwajeong Seo¹[0000–0003–0069–9061]

¹Department of Convergence Security,
Hansung University, Seoul (02876), South Korea,
{ky060261, chosubin1208, ajb11191128, msqoo0212, hwajeong84}@gmail.com

Abstract. Certificate issuance and per-connection signing impose different costs in TLS 1.3, and authentication cost can change depending on how signature algorithms are assigned to Root, Intermediate, and Leaf roles. PQC migration should therefore consider not only algorithm performance, but also role placement, certificate structure, network conditions, and server load. We evaluate all 13^3 role placements formed by 13 signature parameter sets while holding the algorithm composition fixed and changing only the role order. The evaluation covers cryptographic operations, certificate issuance, path validation, TLS authentication, mixed traditional–PQC chains, network profiles, session resumption, and concurrent load. Across 286 groups containing three distinct algorithms, the median maximum/minimum ratios are 2.085 for TLS latency, 4.310 for Certificate body size, and 4.551 for path validation. Thus, role placement alone can produce about a two-fold difference in TLS latency and more than four-fold differences in certificate size and path-validation cost. Direct Root–Intermediate swaps yield a median TLS ratio of 1.003, whereas swaps involving the Leaf yield 1.775–1.788, indicating a stronger influence of Leaf placement. In eight ML-DSA-65/SLH-DSA-SHAKE-192s placements, two candidates remain nondominated when only the Certificate message is considered, but only one remains when the Leaf signature in CertificateVerify is included. This shows that certificate size alone may not represent the transmission cost of the full TLS authentication path. PQC certificate-chain design should therefore consider role-specific computation and transmission costs under the intended network and load conditions.

Keywords: Post-quantum cryptography · TLS 1.3 · X.509 · Signature placement · Certificate chains · Performance evaluation

1 Introduction

Advances in quantum computing may threaten the long-term security of widely deployed RSA and elliptic-curve public-key cryptography [24]. In response, NIST has continued the standardization of post-quantum cryptography (PQC), including the standardization of ML-DSA and SLH-DSA and ongoing work on

additional digital-signature algorithms [12,13,1,14]. In Korea, KpqC has also evaluated domestic PQC candidates and supported related standardization efforts [9]. These developments increase the need to evaluate the performance and operational costs of introducing PQC into security protocols that currently rely on public-key cryptography.

TLS 1.3 is a widely deployed security protocol for Internet communication and provides confidentiality, integrity, and server authentication in applications such as web services. PQC migration therefore requires evaluating its cost in operational protocols such as TLS 1.3. In certificate-based server authentication, Root, Intermediate, and Leaf roles perform different signing and verification operations [2,17]. As a result, assigning the same signature algorithm to different roles can change certificate-issuance cost, path-validation cost, transmitted authentication data, and connection-time signing cost. These differences are particularly relevant to PQC signatures because key sizes, signature sizes, and signing and verification costs vary substantially across algorithms and parameter sets.

Prior work has examined PQ X.509 certificate sizes [8,7], network-dependent TLS costs [15,6], heterogeneous and mixed certificate chains [19,16], and role placement for ML-DSA and SLH-DSA [3]. However, existing evaluations generally focus on selected algorithm combinations or a limited set of role placements. This makes it difficult to isolate the cost caused by changing only the Root–Intermediate–Leaf assignment while keeping the algorithm types and counts unchanged. Analyses that jointly consider certificate issuance, path validation, CertificateVerify, and operational network and load conditions from a role-placement perspective are also limited.

To address this gap, we examine the full role-placement space formed by 13 signature parameter sets while keeping the algorithm types and counts fixed within each composition and changing only their role order. We relate placement-dependent differences to certificate fields and cryptographic operations, and further evaluate selected placements under mixed traditional–PQC migration, network profiles, session resumption, and concurrent load. This approach separates the effect of role reassignment from algorithm replacement and examines how role-specific signature placement affects the operational cost of TLS 1.3.

1.1 Main Contributions

- **Fixed-composition role-placement analysis.** We keep the algorithm types and counts fixed and change only their Root–Intermediate–Leaf assignment, separating role reassignment from algorithm replacement for controlled placement-performance analysis.
- **Role-specific cost propagation analysis.** We link placement-dependent costs to certificate issuance, path validation, Certificate, and CertificateVerify, identifying where role changes introduce computation or transmission costs and how message scope affects placement assessment.
- **Operational boundary of placement effects.** We examine mixed traditional–PQC migration, network profiles, response sizes, session resumption, and concurrent load to determine when role-placement differences

remain relevant to TLS latency and throughput. This provides evidence for selecting placements according to deployment conditions rather than from algorithm-level performance alone.

2 Background and Related Work

2.1 Certificate Issuance and TLS 1.3 Authentication

An X.509 certificate consists of a signed body (TBSCertificate) containing subject information and the public key, a signature-algorithm identifier, and an issuer signature. SubjectPublicKeyInfo (SPKI) in TBSCertificate identifies the subject public key and its algorithm, whereas the certificate signature is generated with the issuer’s private key. The subject and issuer signature algorithms can therefore differ [2]. Figure 1 shows this issuance structure together with the Root, Intermediate, and Leaf roles in TLS 1.3 server authentication. The Root acts as the trust anchor, the Intermediate issues subordinate certificates, and the Leaf is used as the server certificate. A pretrusted Root can be omitted from the certificate chain transmitted by the server [2,17].

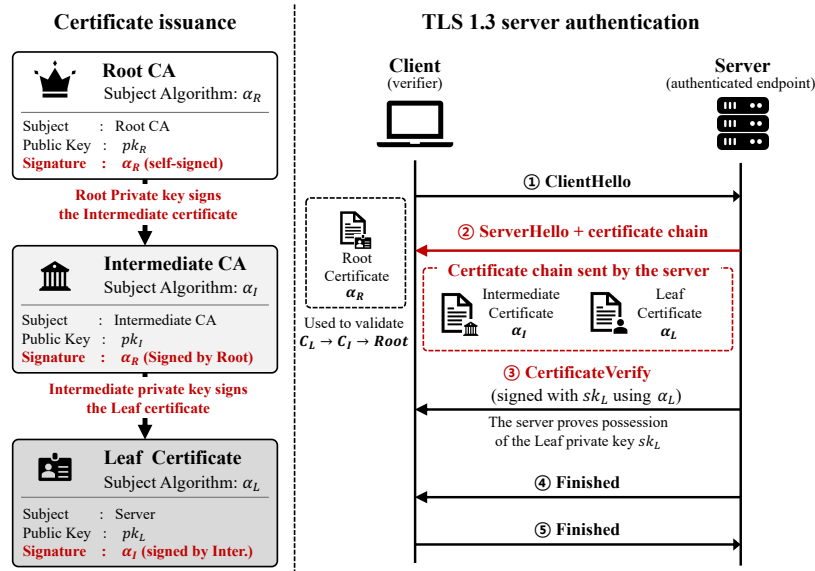


Fig. 1. Three-level certificate issuance and the TLS 1.3 server-authentication path used in the experiments.

In a certificate-based full TLS 1.3 handshake, the server sends Certificate and CertificateVerify, and the client validates the certification path and the Leaf signature [17]. Issuer signatures are generated during certificate issuance and reused with the certificates, whereas the Leaf generates a new CertificateVerify

signature for each full handshake. Upper-role signing cost therefore mainly affects issuance, while Leaf signing directly affects new-connection latency. Upper-role signature verification and certificate transmission remain part of the connection, so computation and transmission costs occur at different stages depending on the assigned role.

Session resumption uses a PSK derived from the initial connection and omits the server Certificate and CertificateVerify, substantially reducing the influence of role-specific certificate signing and verification costs [17]. Certificate compression instead reduces the transmitted Certificate representation while retaining decompression, path validation, and CertificateVerify [5]. Role-placement costs should therefore be evaluated separately for full and resumed connections while considering both computation and authentication-data transmission.

2.2 Signature Structures and Cost Characteristics

PQC signatures can introduce larger public keys or signatures and higher signing or verification costs than RSA or ECDSA, which may increase certificate size and issuance or validation costs during migration. Table 1 summarizes the PQC algorithms and parameter sets evaluated in this study, including NIST-standardized algorithms, additional candidates, and KpqC selections [12,13,14,1,9]. These schemes are based on different structures, including lattices, hashes, multivariate systems, MPC-in-the-Head, and isogenies. The Cat. column denotes the NIST security category. Categories 3 and 5 correspond to attack costs comparable to exhaustive key search against AES-192 and AES-256, respectively [11].

Table 1. Signature parameter sets, underlying structures, and security categories.

Class	Algorithm / parameter set	Structure	Cat.
NIST standard	ML-DSA-65 [12]	Module lattice	3
	SLH-DSA-SHAKE-192s [13]	Hash-based	3
NIST selected	Falcon-1024 [14,26]	NTRU lattice	5
NIST additional signature candidates	MAYO-3 [1,27]	Multivariate	3
	QR-UOV3-q127-L3 (v228, m78) [1,28]	Multivariate	3
	SNOVA-37-8-16-4 [1,29]	Multivariate	3
	UOV-III-pkc [1,30]	Multivariate	3
	FAEST-192s [1,31]	Symmetric-key	3
	MQOM2-Cat3-GF256-short-R3 [1,32]	Multivariate/MPC	3
	SQIsign-Level3 [1,33]	Isogeny	3
Measured variant	SDitH-Hypercube-Cat3-GF256 [1,18,34]	Code-based/MPC	3
KpqC selected	HAETAE-3 [9,35]	Module lattice	3
	AIMer-192s [9,36]	Symmetric-key	3

Algorithm structures and parameter sets lead to different public-key sizes, signature sizes, and KeyGen, Sign, and Verify costs. For example, ML-DSA and Falcon use different lattice-based signature constructions, whereas SLH-DSA uses a stateless hash-based construction [12,13,1]. Multivariate, MPC-in-the-Head, and isogeny-based candidates exhibit different size and computation trade-offs as well. A small public key therefore does not necessarily imply a short signature or low computation cost, and algorithm family or standardization status alone does not determine TLS performance. We therefore evaluate each parameter set using its actual key and signature sizes together with measured operation costs.

We classify certificate-chain compositions by their role assignments. A homogeneous placement uses the same parameter set for Root, Intermediate, and Leaf, whereas a heterogeneous placement uses at least two parameter sets. A mixed chain retains a traditional signature such as RSA or ECDSA in one or more roles while using PQC signatures in the remaining roles. These definitions are used throughout the role-placement and migration analyses.

2.3 Related Work and Scope of Comparison

Table 2 compares related prior work across five dimensions: chain design, primitive operations, network conditions, mixed chains, and concurrency. The Chain column covers certificate roles, hierarchy depth, or intermediate-certificate transmission, while Primitive denotes standalone cryptographic operations outside TLS. Network, Mixed, and Concurrency respectively denote network-condition evaluation, chains combining traditional and PQC signatures, and experiments that vary the number of simultaneous connections.

Kampanakis and Kallitsis [7] analyzed the transmission cost of PQ certificate chains and the omission of intermediate CA certificates. Sikeridis et al. [19] evaluated PQ signature operation costs, certificate-chain sizes, network conditions, and server throughput. Paul et al. [16] studied migration scenarios combining traditional and PQ signatures, while Delgado [3] analyzed ML-DSA and SLH-DSA costs across certificate roles and hierarchy depth. However, these studies do not systematically isolate the cost of changing only role positions while keeping the signature composition fixed. We separate algorithm-specific performance from role-reassignment effects and link the resulting differences to certificate fields and signing and verification operations.

Sosnowski et al. [25] analyzed PQ TLS latency and library computation under varying delay, loss, and bandwidth conditions. Montenegro et al. [10] connected standalone cryptographic operations with TLS and network-level costs. We combine these operational dimensions with role-specific placement comparisons and extend the evaluation to network conditions, response size, session resumption, and concurrent load. This allows us to examine how role-specific authentication costs are reflected in TLS latency and throughput.

Table 2. Evaluation scope of representative prior work across chain design, primitive operations, network conditions, mixed traditional/PQC chains, and concurrency. Chain covers certificate-role or hierarchy-level evaluation, while Model denotes model-based rather than directly measured network evaluation.

Study	Chain	Primitive	Network	Mixed	Concurrency
Kampanakis & Kallitsis [7]	✓	–	Model	–	–
Sikeridis et al. [19]	✓	✓	✓	–	✓
Paul et al. [16]	✓	✓	✓	✓	–
Delgado [3]	✓	–	–	–	–
Sosnowski et al. [25]	–	–	✓	–	–
Montenegro et al. [10]	–	✓	✓	–	–
This study (13 signatures)	✓	✓	✓	✓	✓

✓: measured; –: not separately reported.

3 Design and Analysis of Role-Placement Comparisons

3.1 Functional Qualification

Candidates $\mathcal{U}$ are algorithm-parameter-set pairs. Each is checked in the Root, Intermediate, and Leaf roles through source/provider recognition, Key-Gen/Sign/Verify with expected-failure inputs, X.509 issuance, path validation, and TLS authentication. With role-qualified sets Q_R, Q_I, Q_L , the common candidates are $Q_{\text{all}} = Q_R \cap Q_I \cap Q_L$ and placements are $\mathcal{P} = Q_{\text{all}}^3$. All 13 candidates qualified in every role, and certificate generation and TLS authentication were checked for all 2,197 placements.

3.2 Comparisons within a Fixed Composition

A placement $p = (a_R, a_I, a_L)$ assigns subject public-key algorithms to Root, Intermediate, and Leaf. Multiset S records algorithm types and counts, and $\Pi(S)$ contains its role permutations. Two ML-DSA instances and one SLH-DSA instance, for example, yield three placements. One-, two-, and three-algorithm compositions form 13, 156, and 286 groups, respectively. For a median time or body length $y(p)$, we measure role variation within a group as follows.

$$G_y(S) = \frac{\max_{p \in \Pi(S)} y(p)}{\min_{p \in \Pi(S)} y(p)}. \quad (1)$$

Equation (1) holds composition fixed. Each of the 286 three-algorithm ratios uses all six permutations. Direct Root–Intermediate, Root–Leaf, and Intermediate–Leaf swaps hold the remaining role fixed and yield 1,014 pairs per axis, counting both swap directions once.

3.3 Message Sizes and Operation Costs

The Intermediate certificate contains $\text{SPKI}(a_I)$ and an issuer signature $\text{sig}(a_R)$, while the Leaf certificate contains $\text{SPKI}(a_L)$ and $\text{sig}(a_I)$. The Leaf algorithm a_L is also used for the separate `CertificateVerify` signature. Figure 2 shows how the Root, Intermediate, and Leaf algorithms therefore propagate to the transmitted `Certificate` and `CertificateVerify` fields. Changing the role assignment can affect both certificate fields that are reused across connections and the Leaf signature generated for each full handshake.

(a) **Certificate body**

Context length 1 B	List length 3 B	Certificate entry list Leaf $\rightarrow$ Intermediate
-----------------------	--------------------	---

Server authentication with empty request context and entry extensions.

(b) **Each certificate entry**

Certificate length 3 B	DER certificate SPKI + issuer signature + remaining fields	Extensions length 2 B
---------------------------	---	--------------------------

Entry overhead: 5 B. Two entries: $1 + 3 + 2 \times 5 = 14$ B.

(c) **CertificateVerify body**

Signature scheme 2 B	Signature length 2 B	Leaf signature payload B_{CVsig}
-------------------------	-------------------------	--

Separate from Certificate. Body length: $B_{\text{CVsig}} + 4$ B.

Fig. 2. Certificate and CertificateVerify fields used to derive authentication-body size and trace role-dependent signature and public-key contributions.

Let $B_{\text{served}} = |\text{DER}(C_I)| + |\text{DER}(C_L)|$. As shown in Fig. 2(a) and (b), with an empty request context and empty entry extensions, Certificate adds 1 B for the context length, 3 B for the certificate-list length, and $3+2$ B per entry for the certificate and extension lengths [17]. For the two transmitted certificates, the Certificate body length is defined as follows.

$$B_{\text{Cert}} = B_{\text{served}} + 1 + 3 + 2(3 + 2) = B_{\text{served}} + 14. \quad (2)$$

For a two-level hierarchy that transmits only the Leaf certificate, the corresponding overhead is 9 B. Figure 2(c) shows that the separate `CertificateVerify` message contains the Leaf signature payload B_{CVsig} plus 2 B each for SignatureScheme and signature length. We therefore define the authentication-body size as $B_{\text{auth,body}} = B_{\text{Cert}} + B_{\text{CVsig}} + 4$. This metric excludes the 4 B handshake header of each message, record-layer encryption overhead, and TCP/IP headers. Separating Certificate from CertificateVerify is important because changing the Leaf algorithm affects both its SPKI and the per-connection signature payload.

For primitive signing and verification times S, V in ms, the role-dependent costs are related as follows.

$$T_{\text{hs}}(p) \sim V(a_R) + V(a_I) + S(a_L) + V(a_L) + g(B_{\text{served}}, B_{\text{CVsig}}) + T_{\text{common}}. \quad (3)$$

Equation (3) identifies where role-dependent costs occur rather than directly summing measured components or fitting a regression model. Root and Intermediate algorithms contribute verification costs along the certification path, while the Leaf contributes both CertificateVerify signing and verification. The function g represents authentication-data processing and transmission, and T_{common} represents the fixed key exchange and other common TLS work. TLS measurements and the separate `openssl verify` process have different timing boundaries, so their difference is not interpreted as an individual internal-operation time.

Finally, we use a size model to explain the transmitted DER certificate size from the SPKI and issuer-signature lengths.

$$\hat{B}_{\text{served}} = |\text{sig}(a_R)| + |\text{SPKI}(a_I)| + |\text{sig}(a_I)| + |\text{SPKI}(a_L)| + C \quad (4)$$

In Eq. (4), all four coefficients are fixed to 1 and C represents the remaining size of the fixed certificate profile. We set C to the mean residual over the 2,197 observations and use prediction errors to assess how well role-dependent SPKI and issuer-signature sizes explain the measured DER size.

3.4 Variation, Alternatives, and Measurement Quality

Certificate body sizes range from 1,166 to 398,835 B, approximately 1.2–399 kB, while timing measurements span from milliseconds to seconds. To reduce these scale differences and compare relative role effects, we use the natural logarithm of representative values. Marginal means over the balanced 13^3 placement space are used to decompose the main effects of Root, Intermediate, and Leaf roles and their interactions.

$$\log y_{ril} = \beta_0 + R_r + I_i + L_l + (RI)_{ri} + (RL)_{rl} + (IL)_{il} + \epsilon_{ril}. \quad (5)$$

To make the role effects identifiable and remove arbitrary shifts caused by the reference level, the effects for each role are constrained to sum to zero. The explained share of each effect is computed as $SS_{\text{effect}}/SS_{\text{total}}$ over the between-placement log variation. Other combines the two-role interactions with higher-order interactions that are not separated explicitly and residual variation. We also repeat the same decomposition in 13 leave-one-algorithm-out subspaces to test whether a single algorithm with extreme size or computation cost dominates the overall role effects. Spearman correlation is used to examine whether the ordering of primitive Sign and Verify costs is reflected in system-level metrics such as TLS latency and path validation.

To compare role replacement at the same security level, we use the 12 algorithms assigned to NIST security category 3. Falcon provides parameter sets at categories 1 and 5 but not category 3, so it is excluded from this comparison [14]. Starting from each homogeneous placement, one of the Root, Intermediate, or

Leaf roles is replaced by another category-3 algorithm, producing $12 \times 11 = 132$ direct contrasts per role. A replacement is counted as an improvement only when both the TLS median and Certificate body size decrease. Requiring both metrics prevents a reduction in connection latency from being treated as an improvement when it is obtained by increasing authentication-data size.

For broader placement comparisons, nondominance is evaluated using TLS median, path-validation median, and Certificate body size. These metrics represent connection performance, client-side certification-path validation cost, and transmitted certificate size, respectively. A placement q dominates p only if q is no larger than p in all three metrics and is strictly smaller in at least one. Therefore, an increase in any of the three costs prevents a dominance relation. We do not introduce a separate equivalence tolerance and instead compare the measured representative values directly. This applies the same rule across all placements and avoids changing the nondominated set through an arbitrarily chosen tolerance. Because small measurement differences can still affect this strict comparison, the resulting candidates are interpreted together with repeatability and measurement-quality results.

Comparison groups are constructed incrementally from the standardization and selection status reported in Table 1. The 2^3 placements using only the NIST-standardized ML-DSA and SLH-DSA form the FIPS-only baseline. We then add the KpqC-selected HAETAE and AIMer and classify the 4^3 placements excluding the FIPS-only set as the FIPS/KpqC group. Placements containing the remaining category-3 candidates form the additional-candidate group. For the eight placements composed only of the NIST-standardized ML-DSA and SLH-DSA, we additionally compare the nondominated set when the size metric contains only the Certificate body and when it also includes the Leaf signature in CertificateVerify. This isolates how the scope of authentication data included in the metric affects placement selection.

Tail metrics such as p95 are sensitive to a small number of slow observations and to variation across repeated measurements. We therefore assess estimation uncertainty and repeatability in addition to representative values. Confidence intervals that preserve the measurement hierarchy and variation across repeated measurements are used as quality indicators, while the detailed repetition design, bootstrap procedure, and acceptance criteria are described with the experimental methodology.

4 Performance Evaluation and Placement Analysis

4.1 Experimental Setup and Measurement Scope

Table 3 summarizes the hardware, operating system, runtime, authentication path, and key-exchange settings shared across all experiments. TLS time covers the `SSL_connect` interval after initialization, trust-store loading, and TCP establishment, while path validation measures the entire separate `openssl verify` process. In the network experiments, TTFB starts immediately before client-

process creation and ends at first-byte reception, whereas TTLB includes response reception, cleanup, and process termination. Because these metrics have different measurement boundaries, only matching metrics are compared to reduce interpretation errors caused by incompatible timing scopes.

Table 3. Common experimental environment.

Item	Setting
Host	Scaleway EM-B220E-NVMe; AMD EPYC 7232P; 8 cores/16 threads; 64 GiB DDR4 ECC
OS	Ubuntu 24.04 LTS; Linux 6.8.0-88
CPU	Server/client: logical CPUs 2/3 (separate physical cores); performance governor
Runtime	OpenSSL 3.3.1 [20]; liboqs [21]; oqs-provider 0.12.0-dev [22]; KpqClean_ver2 [23]; research signature provider
Auth. path	Pretrusted Root; Leaf and Intermediate transmitted
Key exchange	X25519MLKEM768

Table 4 summarizes the complete experimental flow from functional qualification to role placement, migration, network sensitivity, and load evaluation. Functional qualification (Q) checks whether all 13 candidates operate correctly in the Root, Intermediate, and Leaf roles, while the Primitive and Issuer experiments measure basic cryptographic operations and certificate-issuance costs. F Baseline and Role/Depth comparisons are then used to establish representative role effects before the full-space exploration (E1 Full) evaluates all 2,197 Root–Intermediate–Leaf placements. Placements with large variation across repeated batches or a p95 confidence interval whose relative width exceeds the configured quality limit are selected for additional measurements through E1 Review, E1 Recheck, E2 Confirm, and E2 Extended to assess repeatability and cost-order consistency. This separates full-space exploratory results from results supported by additional repetitions.

The operational evaluation extends the analysis to migration, network and session conditions, and concurrent load. M1 Migration and M2 Confirm examine mixed chains containing traditional RSA or ECDSA together with PQC signatures to analyze the effect of where PQC is introduced. E3 Network, E3 Refine, E3 Precision, and M3 Network evaluate changes in TTLB and tail latency across network and session conditions, while E4 Load measures throughput and completion-conditional p95 as concurrency increases. These experiments determine whether role-placement cost differences remain visible in latency and throughput under operational conditions. Unless otherwise noted, the repeat count in Table 4 denotes the number of measurement batches multiplied by the number of runs per batch.

Table 4. Campaign scope, repetitions, and counts. Repeats are batches $\times$ runs per batch unless noted. E1 reports TLS and path-validation counts separately. E3 Precision includes the corresponding Refine observations.

Campaign	Scope / purpose	Repeats	Count
Qualification and Components			
Q	13 candidates, role qualification	Per role	13
Primitive	13 candidates, KeyGen/Sign/Verify	5 batches	1,140
Issuer	338 issuer–subject links	3×10	10,140
Role Placement and Depth			
F Baseline	8 ML-DSA/SLH-DSA placements	3×50	1,200
Role/Depth	15 role/depth comparisons	3×50	2,250
E1 Full	2,197 placements, full-space exploration	3×10	65,910
E1 Review	19 flagged placements, targeted repeat	3×40	2,280
E1 Recheck	6 remaining placements, second repeat	3×40	720
E2 Confirm	34 placements, independent confirmation	3×40	4,080
E2 Extended	1 placement, extended repeat	6×40	240
Migration			
M1 Migration	319 migration placements, controls included	2×5	3,190
M2 Confirm	30 migration placements, confirmation	3×40	3,600
Network and Session Sensitivity			
E3 Network	306 conditions, network/session sweep	3×10	9,180
E3 Refine	80 conditions, tail remeasurement	3×10	2,400
E3 Precision	42 conditions, tail precision	10×10	4,200
M3 Network	279 conditions, mixed-chain network sweep	3×10	8,370
Concurrency and Load			
E4 Load	48 conditions, concurrency sweep	3 trials	144

Placement: one Root–Intermediate–Leaf assignment.

Condition: one placement under an operational setting.

4.2 Primitive and Certificate-Issuance Costs (Primitive, Issuer)

Table 5 compares primitive-operation costs and encoded sizes. UOV signs fastest at 0.082 ms but has a 189,253 B SPKI. SLH-DSA has a 66 B SPKI and a 16,224 B signature, but signing takes 1,790.573 ms. SQIsign is compact with a 112 B SPKI and a 224 B signature, yet signing takes 907.272 ms. Small authentication data and fast signing therefore do not necessarily occur in the same algorithm.

The certificate-issuance experiment evaluates all 13×13 issuer–subject algorithm combinations for both Root–Intermediate and Intermediate–Leaf links. Command time measures certificate generation and issuer signing through `openssl x509 -req` [20], while total time additionally includes subject-key and CSR generation, DER conversion, and related processing.

Matching the same 169 issuer–subject combinations across the two links gives Spearman correlations of 0.9994 for command time, 0.9996 for total time, and

Table 5. Median primitive-operation times and encoded sizes over five batches.

Signature	Operation time (ms)			Encoded size (B)	
	KeyGen	Sign	Verify	SPKI	Signature
AIMer-192s	0.131	41.566	41.532	63	9,120
FAEST-192s	0.036	2,654.375	2,446.268	63	11,260
Falcon-1024	20.590	0.527	0.106	1,811	1,270
HAETAE-3	0.540	1.631	0.134	1,491	2,349
MAYO-3	0.177	0.445	0.191	3,005	681
ML-DSA-65	0.069	0.151	0.064	1,974	3,309
MQOM	2.172	126.977	61.281	136	7,900
QR-UOV	48.901	4.242	3.284	71,913	292
SDitH-Hypercube	5.131	42.106	38.967	197	19,544
SLH-DSA-SHAKE-192s	192.031	1,790.573	1.460	66	16,224
SNOVA	0.399	1.136	0.275	4,131	376
SQIsign-Level3	426.619	907.272	30.236	112	224
UOV-III-pkc	4.487	0.082	0.317	189,253	200

FAEST and SLH-DSA use 20 samples, all others use 100. SPKI is full DER TLV, signature is payload length, and Falcon reports the median signature size.

1.000 for DER size. Log command-time variation is explained by the issuer, subject, and their interaction at 57.9%, 34.9%, and 7.2%, respectively, while issuer and subject shares for total time are 48.5% and 43.5%. Even with AIMer fixed as the issuer, changing the subject from ML-DSA to FAEST increases total time from 155.731 to 7,722.580 ms. Subject-key and CSR processing can therefore materially affect issuance cost in addition to the issuer algorithm. Subsequent TLS connection experiments use certificates generated in advance.

4.3 Signature Roles and Hierarchy Depth (F Baseline, Role/Depth)

This experiment compares how Root, Intermediate, and Leaf role changes and certificate-hierarchy depth affect TLS performance for ML-DSA and SLH-DSA. The baseline repetition experiment (F Baseline) measures eight role combinations of the two algorithms, while the Role/Depth experiment compares Leaf changes, upper-role changes, and the presence of an Intermediate certificate. With the upper roles fixed to ML-DSA, replacing only the ML-DSA Leaf with SLH-DSA increases TLS latency from 2.642 to 1,793.835 ms, approximately 679-fold, while an independent repetition shows an approximately 707-fold difference. In contrast, the Certificate body decreases from 11,133 to 9,225 B by 17.1%, and path-validation cost changes by only 0.3%. This shows that the approximately 1.79 s online Leaf-signing cost can dominate connection latency even when the smaller Leaf SPKI reduces transmission size.

Table 6. Role/depth repetitions with TLS p50 and Certificate body size.

Test	Depth	Signature algorithm by role			TLS p50 (ms)	Body (B)
		Root	Intermediate	Leaf		
Baseline (P)	3	ML-DSA	ML-DSA	ML-DSA	2.640	11,133
		SLH-DSA	SLH-DSA	SLH-DSA	1,798.354	33,171
Leaf (L)	3	ML-DSA	ML-DSA	ML-DSA	2.642	11,133
		ML-DSA	ML-DSA	SLH-DSA	1,793.835	9,225
Upper (U)	3	ML-DSA	ML-DSA	ML-DSA	2.637	11,133
		ML-DSA	SLH-DSA	ML-DSA	4.235	22,156
		SLH-DSA	ML-DSA	ML-DSA	4.223	24,056
Depth (H)	2	ML-DSA	None	ML-DSA	2.575	5,585
		ML-DSA	None	SLH-DSA	1,795.149	3,677
		SLH-DSA	None	ML-DSA	4.170	18,508
		SLH-DSA	None	SLH-DSA	1,795.964	16,600
Depth (H)	3	ML-DSA	ML-DSA	ML-DSA	2.644	11,133
		ML-DSA	ML-DSA	SLH-DSA	1,795.919	9,225
		SLH-DSA	ML-DSA	ML-DSA	4.211	24,056
		SLH-DSA	ML-DSA	SLH-DSA	1,793.627	22,148

ML-DSA: ML-DSA-65, SLH-DSA: SLH-DSA-SHAKE-192s.

Each row reports 150 TLS runs over three batches. Repeated compositions are independent comparisons. None denotes no intermediate CA.

Replacing only the Intermediate or Root with SLH-DSA increases TLS latency by approximately 1.62-fold relative to the all-ML-DSA placement. Path-validation cost increases by 23.6% and 22.3%, respectively, while Certificate body size increases by 99.0% and 116.1%. Because the ML-DSA Leaf still performs the online signature, the SLH-DSA cost in the upper roles is mainly reflected in transmission and verification of the precomputed issuer signatures. This indicates that Leaf changes and upper-role changes affect TLS performance through different cost paths.

Hierarchy depth mainly affects certificate transmission and validation costs. Adding an Intermediate to a two-level all-ML-DSA chain increases the Certificate body from 5,585 to 11,133 B, while TLS latency increases by only 2.65%. With an ML-DSA Root and SLH-DSA Leaf, the body increases from 3,677 to 9,225 B by 150.9%, while the TLS difference is only 0.04%. Thus, adding or removing an Intermediate primarily changes transmission size and validation steps, whereas changing the Leaf algorithm directly changes the online signing cost incurred in each full handshake.

4.4 Full-Space Role Effects (E1 Full)

The full-space placement experiment (E1 Full) evaluates all 2,197 placements, including the 19 placements retained for review. Figure 3(a) shows the performance distribution obtained by changing only the role order while keeping the algorithm composition fixed. The maximum/minimum ratio is the largest measured value divided by the smallest value within a fixed composition and represents the performance difference caused by role placement alone. Across 286 groups containing three distinct algorithms, the median ratios are 2.085 for TLS latency, 4.310 for Certificate body size, and 4.551 for path validation. The TLS ratio is at least 2 in 188 groups (65.7%) and at least 10 in 93 groups (32.5%). Across 156 groups containing two distinct algorithms, the median TLS ratio is 1.631.

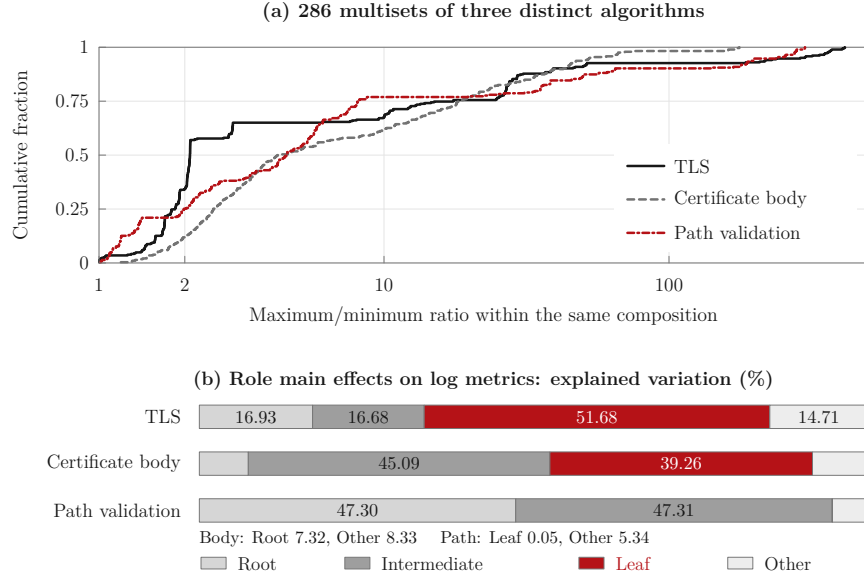


Fig. 3. Role effects in the exploratory exhaustive space. (a) Distribution of fixed-composition maximum/minimum ratios. (b) Role-wise explained variation in log metrics.

The largest TLS difference occurs in a composition containing two ML-DSA instances and one SLH-DSA. With the Root fixed to ML-DSA, swapping the Intermediate and Leaf increases TLS latency from 4.214 to 1,793.330 ms, a 425.52-fold increase, while reducing the Certificate body from 22,156 to 9,225 B by 58.4%. In contrast, the authentication-body sizes including CertificateVerify are nearly identical at 25,469 and 25,453 B. This occurs because signature bytes removed from the Certificate are shifted to CertificateVerify, which is generated for each full handshake. Certificate size alone therefore does not fully represent authentication-transmission cost.

Across 1,014 direct role-swap pairs, the median TLS maximum/minimum ratios are 1.003 for Root–Intermediate, 1.775 for Root–Leaf, and 1.788 for Intermediate–Leaf. Unlike upper-role swaps, swaps involving the Leaf change the online signing algorithm. Figure 3(b) shows that the Leaf explains the largest share of TLS variation at 51.68%, while Root and Intermediate account for 47.30% and 47.31% of path-validation variation. For Certificate body size, Intermediate and Leaf account for 45.09% and 39.26%, respectively. The same role-dependent pattern appears in the primitive-operation results. The Spearman correlation between Leaf signing cost and TLS latency is $\rho = 0.671$, while the correlation between the summed upper-role verification cost and path validation is $\rho = 0.978$. These role effects also remain when each algorithm is excluded in turn.

When the algorithm in only one role is changed from a homogeneous category-3 baseline, both TLS latency and Certificate body size decrease in 27.3% of Root changes, 28.0% of Intermediate changes, and 11.4% of Leaf changes. Under the three-metric nondominance rule, the remaining candidate shares are 25.0% for FIPS-only placements, 17.9% for the KpqC-inclusive group, and 3.7% for the additional-candidate group. Among the eight ML-DSA/SLH-DSA placements, two remain nondominated when only the Certificate body is used as the size metric, whereas including the Leaf signature in CertificateVerify leaves only the all-ML-DSA placement. Thus, placement assessment depends not only on the role and algorithm group, but also on the scope of authentication data included in the metric.

4.5 Traditional-to-PQC Migration (M1, M2)

The migration experiment (M1 Migration) evaluates 319 placements combining ECDSA P-256/P-384, RSA-2048/3072, ML-DSA, SLH-DSA, and MAYO to examine how TLS performance changes depending on which roles retain traditional signatures and where PQC is introduced. The set includes 64 all-traditional placements, 144 placements retaining traditional signatures in two roles, 108 retaining them in one role, and three homogeneous PQC baselines. The median TLS latency of the all-traditional placements is 4.518 ms. Even when Root and Intermediate remain traditional, TLS latency ranges from 2.652 to 1,798.009 ms depending on the Leaf algorithm. Across placements grouped by PQC Leaf, the median TLS latencies are 3.405 ms for ML-DSA, 3.533 ms for MAYO, and 1,796.423 ms for SLH-DSA. These results show that migration position alone is insufficient to explain connection cost, and that the computational characteristics of the PQC algorithm placed at the Leaf directly affect TLS latency.

When only one role retains a traditional signature, the Certificate body size also depends on which role remains traditional. The median body sizes are 8,004.5 B for a traditional Root, 6,239.5 B for a traditional Intermediate, and 17,753 B for a traditional Leaf. Thus, retaining the same number of traditional signatures can produce different transmission costs depending on their role positions. The independent confirmation experiment (M2 Confirm) selects

30 representative placements across latency, size, issuance, validation, and role diversity. Figure 4 compares their TLS latency and Certificate body size.

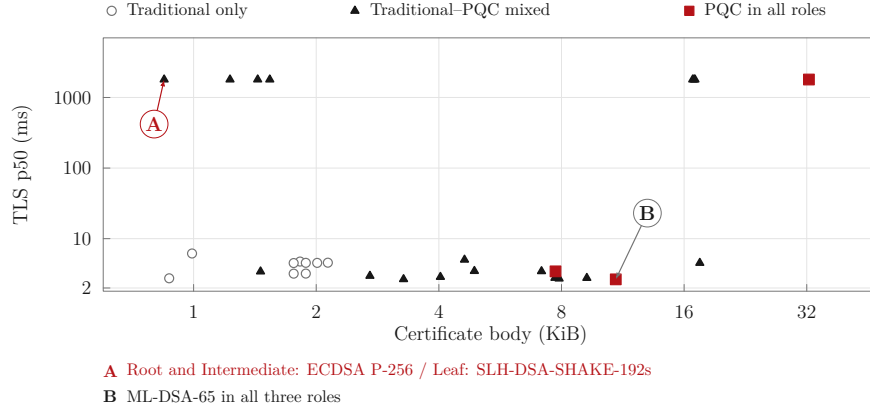


Fig. 4. TLS latency and Certificate body size for 30 representative migration placements. Red squares denote baselines using the same PQC signature in all three roles. Both axes are logarithmic.

Starting from the all-ECDSA P-256 baseline, replacing only the Leaf with ML-DSA increases TLS latency slightly from 2.743 to 2.968 ms, while the Certificate body grows from 892 to 2,774 B. In contrast, a mixed chain with an SLH-DSA Leaf has a smaller 867 B Certificate body but increases TLS latency to 1,795.855 ms (point A in Fig. 4). These results show that PQC migration should consider not only how many roles are converted, but also which algorithm is assigned to each role. In addition, any mixed chain that retains RSA or ECDSA also retains the quantum exposure of the traditional signature and therefore should not be treated as security-equivalent to a fully PQC certificate chain [4,16].

4.6 Network and Session Effects (E3, M3)

The network and session experiment (E3 Network) evaluates 17 PQC placements selected from the quality-passing results of the independent confirmation experiment. The placements were chosen to represent different characteristics in TLS p95 latency, Certificate size, certificate-issuance time, and repeated-measurement variation. Each placement is evaluated under loopback, moderate, and constrained network profiles with response bodies of 0 B, 50 KiB, and 200 KiB, comparing full and resumed connections. Because delay, loss, and bandwidth vary together within each profile, the results represent the effect of the complete network condition rather than an individual network factor.

Across the 17 placements, the median TTLB p50 values for 0 B responses are 97.48, 273.45, and 435.54 ms for full connections and 12.03, 156.24, and 310.68 ms

for resumed connections in loopback, moderate, and constrained profiles, respectively. At 200 KiB, these values increase to 97.84, 374.56, and 1,101.77 ms for full connections and 12.38, 295.48, and 932.39 ms for resumption. Larger responses and stronger network constraints therefore increase completion time beyond the authentication computation itself.

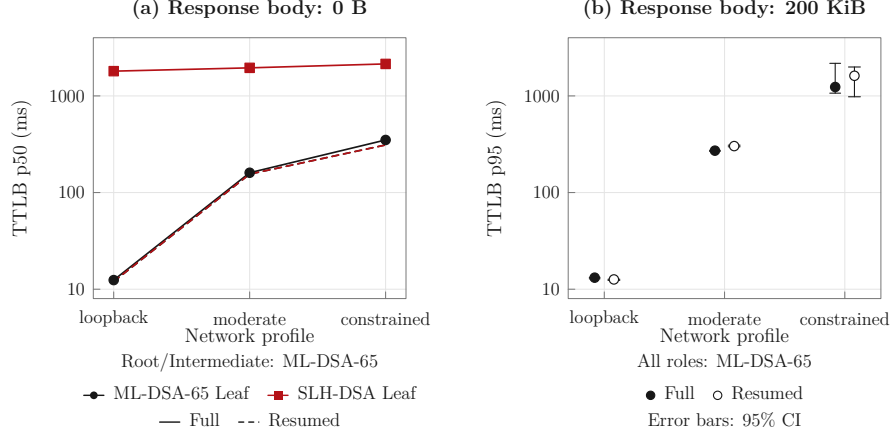


Fig. 5. TTLB under network and session conditions. (a) p50 with 0 B responses, with Root and Intermediate fixed to ML-DSA-65 and the Leaf varied between ML-DSA-65 and SLH-DISA-SHAKE-192s. Solid and dashed lines denote full and resumed connections. (b) p95 and 95% confidence intervals for the all-ML-DSA-65 baseline with 200 KiB responses. Filled and open markers denote full and resumed connections.

Figure 5 illustrates the interaction between role-dependent authentication cost and network conditions. With the Root and Intermediate fixed to ML-DSA and SLH-DISA placed at the Leaf, the 0 B full-connection TTLB ranges from 1,804.033 to 2,143.070 ms, whereas resumption ranges from 12.025 to 310.654 ms. The absolute difference remains approximately 1.8 s, but the relative contribution of authentication decreases as network delay grows. For larger responses, some conditions also show that resumption is not faster than a full connection, indicating that omitting authentication work does not necessarily reduce total completion time when network and response-transfer costs dominate.

The mixed-transition network experiment (M3 Network) applies the same network and session conditions to 14 mixed placements selected from the passing independent migration-confirmation results using the same selection criteria. Because E3 and M3 contain different placement samples, differences between their group medians also include algorithm-composition effects. Resumption effects are therefore interpreted by comparing full and resumed connections within the same placement. Actual CompressedCertificate messages were observed only for a subset of mixed placements, and the observed cases did not show a consistently large latency reduction [5].

4.7 Concurrency and Load Effects (E4)

The concurrent-load experiment (E4 Load) evaluates 12 placements at concurrency levels of 1, 8, 32, and 128 with three trials per condition. Each trial measures throughput and `SSL_connect` latency for requests completed within a 30 s window on a single server process, while new client tasks replace completed requests to maintain the configured concurrency. Figure 6 reports the trial medians. For the all-ML-DSA placement, throughput is 82.90, 88.13, 87.50, and 85.23 requests/s across the four concurrency levels. Throughput therefore changes little from concurrency 1 to 128, while completion-conditional p95 latency increases from 2.64 to 1,414.98 ms, approximately 536-fold. When only the Root is replaced with Falcon, HAETAE, or MAYO, or only the Intermediate is replaced with Falcon or HAETAE, throughput similarly remains within about 1.03-fold of its low-concurrency level while p95 latency increases by several hundred times. Thus, even with upper-role algorithm changes, higher concurrency primarily appears as throughput saturation and rapidly increasing tail latency among completed requests.

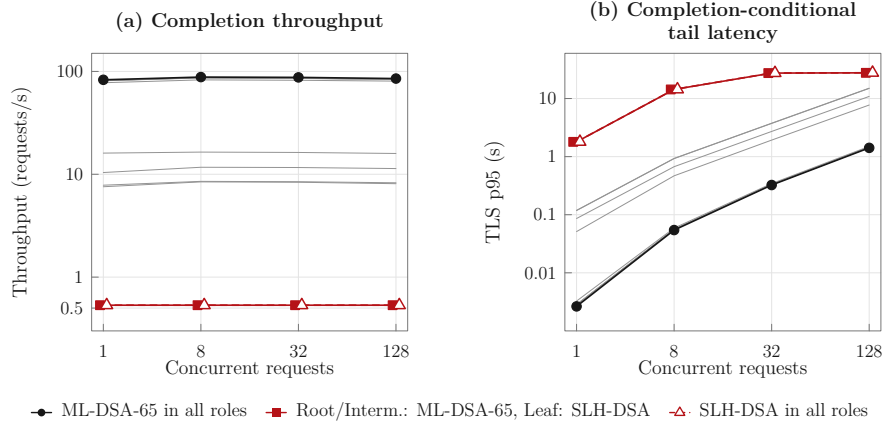


Fig. 6. Throughput and completion-conditional p95 across concurrency levels for 12 placements. Each point is the median of three trials.

For the two placements with SLH-DSA at the Leaf, regardless of whether the upper roles use ML-DSA or SLH-DSA, all 24 trials complete only 16 requests per 30 s, corresponding to 0.533 requests/s. Their p95 latency is approximately 1.80 s at concurrency 1 and increases to about 27.7 s at concurrency 128. Because these p95 values summarize only requests completed within the measurement window, they should not be interpreted as the latency distribution of all submitted requests. These results show that role-specific signing costs can appear as per-connection latency at low concurrency and develop into throughput saturation and substantially larger completion-conditional tail latency under high concurrency.

4.8 Repeatability and Measurement Limits

Table 7 summarizes p50 consistency and p95 precision across the repeated measurements. The independent Role/Depth, E2 Confirm, and M2 Confirm experiments preserve the main role-placement effects. The lower-role change in Role/Depth shows a 423.62-fold TLS difference, close to the 425.52-fold difference in E1 Full, while E2 Confirm and M2 Confirm also show high consistency with their corresponding earlier experiments. These results indicate that relative cost relationships across role placements remain stable across repeated measurements.

Table 7. Repeatability of matching placements and p95 precision under a 50% relative-width limit. Matching-placement rows compare independent repetitions of the same placement, while the lower block reports conditions exceeding the p95 precision limit.

Check	Scope	Result
Matching-placement TLS p50		
E2 / E1	34 placements	Median 0.9997, range 0.9099–1.0582, $\rho = 0.995$
M2 / M1	30 placements	Median 0.9998, range 0.9867–1.0083, $\rho = 0.992$
Relative p95 Interval Width		
E2 Extended	All-SQIsign-Level3	55.43% $\rightarrow$ 34.89%
E3 Refine	80 conditions	42 exceed the limit
E3 Precision	42 conditions	39 exceed, 37 constrained-profile conditions
M3 Network	279 conditions	75 exceed, 58 constrained-profile conditions

ρ : Spearman correlation.

In contrast, E1 Review/Recheck, E3 Refine/Precision, and M3 Network retain wide p95 confidence intervals in some conditions, particularly under constrained networks. This shows that large role-placement cost differences remain visible across repetitions, whereas tail-latency stability is more sensitive to network conditions.

4.9 Operational Placement Criteria

Table 8 summarizes placement criteria derived from the preceding experiments. Security categories and roles that may retain traditional signatures should first be fixed. When certificate-list size is the main constraint, B_{Cert} should be considered, whereas $B_{\text{auth,body}}$ is more appropriate when CertificateVerify is also relevant. Latency-sensitive deployments should prioritize Leaf signing cost, while certificate transmission and path-validation constraints require attention to Intermediate and Root signature sizes and verification costs. As network constraints, response size, or concurrency increase, transmission cost, throughput, and tail latency become more important. Appropriate role placement should therefore be selected according to the target deployment conditions.

Table 8. Key operational checks for role-specific signature placement.

Condition	Key check
Full	Leaf signing, upper-role verification, TLS p50/p95
Auth.	SPKIs, issuer signatures, Leaf CertificateVerify
Depth	Transmission savings versus Leaf signing cost
Migration	Traditional roles retained, PQC-affected operations
Resume	Full/resumed TTLB, omitted authentication work
Net./resp.	Response size, network profile, TTLB
Load	Throughput, completion p95, unfinished requests

Full: full connection, Auth.: authentication,
 Net./resp.: network/response, Load: concurrency/load.

5 Conclusion

This study isolates role-placement costs in TLS 1.3 certificate chains by fixing the signature algorithms and their counts while changing only the Root, Intermediate, and Leaf assignments. Across all 2,197 placements, the same algorithm composition can produce substantially different TLS latency, Certificate size, and path-validation cost depending on role order, with Leaf changes having the strongest effect on connection latency. In one representative case, moving SLH-DSA to the Leaf reduces the Certificate body by 58.4% but increases TLS latency by 425.52-fold. These results show that PQC certificate-chain performance cannot be evaluated from standalone algorithm performance or certificate size alone, and that role placement should be treated as a primary design factor.

Placement effects also vary with operational conditions. Session resumption reduces the influence of online Leaf signing, while stronger network constraints and larger responses increase the contribution of transmission cost. Under high concurrency, throughput saturation and increased tail latency become more important. Latency-sensitive deployments should therefore prioritize Leaf signing cost, whereas environments constrained by certificate transmission or path validation should place greater emphasis on upper-role signature sizes and verification costs. PQC role placement can thus be viewed as a design problem that balances latency, validation, transmission, and throughput costs under the required security and migration conditions.

The evaluation is limited to a fixed OpenSSL-based implementation, hardware platform, and certificate profile, and some constrained-network conditions show wide p95 confidence intervals. The concurrent-load results are also limited to a single server core. Future work should evaluate the reproducibility of these role-placement effects across different TLS and PQC implementations, hardware platforms, multi-core servers, and certificate structures. It should also extend the analysis toward deployment-aware placement policies that incorporate resumption rate, network conditions, response size, and concurrency.

References

1. Alagic, G., Bros, M., Ciadoux, P., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Silberg, H., Smith-Tone, D., Waller, N.: Status report on the second round of the additional digital signature schemes for the NIST post-quantum cryptography standardization process. NIST Interagency/Internal Report 8610, National Institute of Standards and Technology (May 2026). <https://doi.org/10.6028/NIST.IR.8610>
2. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.T.: Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. RFC 5280, Internet Engineering Task Force (May 2008). <https://doi.org/10.17487/RFC5280>
3. Delgado Jiménez, J.L.: Signature placement in post-quantum TLS certificate hierarchies: An experimental study of ML-DSA and SLH-DSA in TLS 1.3 authentication. arXiv:2604.06100, version 3 (2026). <https://doi.org/10.48550/arXiv.2604.06100>, <https://arxiv.org/abs/2604.06100v3>
4. Driscoll, F., Parsons, M., Hale, B.: Terminology for post-quantum traditional hybrid schemes. RFC 9794, Internet Engineering Task Force (Jun 2025). <https://doi.org/10.17487/RFC9794>
5. Ghedini, A., Vasiliev, V.: TLS certificate compression. RFC 8879, Internet Engineering Task Force (Dec 2020). <https://doi.org/10.17487/RFC8879>
6. Kampanakis, P., Childs-Klein, W.: The impact of data-heavy, post-quantum TLS 1.3 on the Time-To-Last-Byte of real-world connections. In: Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb). Internet Society (2024). <https://doi.org/10.14722/madweb.2024.23010>
7. Kampanakis, P., Kallitsis, M.G.: Faster post-quantum TLS handshakes without intermediate CA certificates. In: Cyber Security, Cryptology, and Machine Learning. Lecture Notes in Computer Science, vol. 13301, pp. 337–355. Springer (2022). https://doi.org/10.1007/978-3-031-07689-3_25
8. Kampanakis, P., Panburana, P., Daw, E., Van Geest, D.: The viability of post-quantum X.509 certificates. Cryptology ePrint Archive, Paper 2018/063 (2018), <https://eprint.iacr.org/2018/063>
9. KpqC Research Group: Selected algorithms from the KpqC competition round 2. KpqC Competition official announcement, 16 January 2025 (2025), https://kpsc.or.kr/competition_02.html, AIMER and HAETAE selected as the final digital-signature algorithms; accessed 2026-09-24
10. Montenegro, J.A., Rios, R., Lopez-Cerezo, J.: A performance evaluation framework for post-quantum TLS. Future Generation Computer Systems **175**, 108062 (2026). <https://doi.org/10.1016/j.future.2025.108062>
11. National Institute of Standards and Technology: Post-quantum cryptography: Security (evaluation criteria). NIST Computer Security Resource Center, [https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-\(evaluation-criteria\)](https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria)), accessed 2026-09-25
12. National Institute of Standards and Technology: Module-lattice-based digital signature standard. Federal Information Processing Standards Publication 204, National Institute of Standards and Technology (Aug 2024). <https://doi.org/10.6028/NIST.FIPS.204>
13. National Institute of Standards and Technology: Stateless hash-based digital signature standard. Federal Information Processing Standards Publication 205, National

- Institute of Standards and Technology (Aug 2024). <https://doi.org/10.6028/NIST.FIPS.205>
14. National Institute of Standards and Technology: Post-Quantum Cryptography: PQC Standardization Process. NIST Computer Security Resource Center (2026), <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization>, Falcon/FN-DSA remains under standardization development; accessed 2026-09-24
 15. Paquin, C., Stebila, D., Tamvada, G.: Benchmarking post-quantum cryptography in TLS. In: Ding, J., Tillich, J.P. (eds.) *Post-Quantum Cryptography. Lecture Notes in Computer Science*, vol. 12100, pp. 72–91. Springer (2020). https://doi.org/10.1007/978-3-030-44223-1_5
 16. Paul, S., Kuzovkova, Y., Lahr, N., Niederhagen, R.: Mixed certificate chains for the transition to post-quantum authentication in TLS 1.3. In: *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. pp. 727–740. Association for Computing Machinery (2022). <https://doi.org/10.1145/3488932.3497755>
 17. Rescorla, E.: The transport layer security (TLS) protocol version 1.3. RFC 9846, Internet Engineering Task Force (Jul 2026). <https://doi.org/10.17487/RFC9846>
 18. SDitH Project: SDitH-Hypercube: Reference implementation, category 3 over GF256. API version 1.01, archived source, <https://github.com/sdith/sdith/tree/fff41ae5b12ba3c6b90d8f97b2ede57aa277a92f>, measured implementation; accessed 2026-09-24
 19. Sikeridis, D., Kampanakis, P., Devetsikiotis, M.: Post-quantum authentication in TLS 1.3: A performance study. In: *Network and Distributed System Security Symposium*. Internet Society (2020). <https://doi.org/10.14722/ndss.2020.24203>
 20. OpenSSL Project: OpenSSL 3.3 Documentation. <https://docs.openssl.org/3.3/>, accessed 2026-09-25.
 21. Open Quantum Safe: liboqs: C library for quantum-safe cryptographic algorithms. <https://openquantumsafe.org/liboqs/>, accessed 2026-09-25.
 22. Open Quantum Safe: oqs-provider: Open Quantum Safe provider for OpenSSL. Software repository and documentation. <https://github.com/open-quantum-safe/oqs-provider>, accessed 2026-09-25.
 23. KpqClean Developers: KpqClean_ver2: Clean reference implementations of KpqC algorithms. Software repository. https://github.com/kpqc-cryptocraft/KpqClean_ver2, accessed 2026-09-25.
 24. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing* **26**(5), 1484–1509 (1997). <https://doi.org/10.1137/S0097539795293172>
 25. Sosnowski, M., Wiedner, F., Hauser, E., Steger, L., Schoinianakis, D., Gallenmüller, S., Carle, G.: The performance of post-quantum TLS 1.3. In: *Companion of the 19th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. pp. 19–27. Association for Computing Machinery (2023). <https://doi.org/10.1145/3624354.3630585>
 26. Fouque, P.-A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Specification version 1.2, NIST Post-Quantum Cryptography Standardization submission (2020), <https://falcon-sign.info/falcon.pdf>
 27. Beullens, W.: MAYO: Practical post-quantum signatures from oil-and-vinegar maps. In: *Selected Areas in Cryptography – SAC 2021. Lecture Notes in Com-*

- puter Science, vol. 13203, pp. 355–376. Springer (2022). https://doi.org/10.1007/978-3-030-99277-4_17
28. Furue, H., Ikematsu, Y., Kiyomura, Y., Takagi, T.: A new variant of unbalanced oil and vinegar using quotient ring: QR-UOV. In: *Advances in Cryptology – ASIACRYPT 2021, Part IV*. Lecture Notes in Computer Science, vol. 13093, pp. 187–217. Springer (2021). https://doi.org/10.1007/978-3-030-92068-5_7
29. Wang, L.-C., Tseng, P.-E., Kuan, Y.-L., Chou, C.-Y.: A simple noncommutative UOV scheme. *Taiwanese Journal of Mathematics* **30**(3) (2026). <https://doi.org/10.11650/TJM/251110>
30. Beullens, W., Chen, M.-S., Hung, S.-H., Kannwischer, M.J., Peng, B.-Y., Shih, C.-J., Yang, B.-Y.: Oil and vinegar: Modern parameters and implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems* **2023**(3), 321–365 (2023). <https://doi.org/10.46586/tches.v2023.i3.321-365>
31. Baum, C., Braun, L., Delpech de Saint Guilhem, C., Klooß, M., Orsini, E., Roy, L., Scholl, P.: Publicly verifiable zero-knowledge and post-quantum signatures from VOLE-in-the-Head. In: *Advances in Cryptology – CRYPTO 2023*. Lecture Notes in Computer Science, vol. 14085, pp. 581–615. Springer (2023). https://doi.org/10.1007/978-3-031-38554-4_19
32. Benadjila, R., Feneuil, T., Rivain, M.: MQ on my Mind: Post-quantum signatures from the non-structured multivariate quadratic problem. In: *2024 IEEE European Symposium on Security and Privacy (EuroS&P)*. pp. 468–485. IEEE (2024). <https://doi.org/10.1109/EuroSP60621.2024.00032>
33. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact post-quantum signatures from quaternions and isogenies. In: *Advances in Cryptology – ASIACRYPT 2020, Part I*. Lecture Notes in Computer Science, vol. 12491, pp. 64–93. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_3
34. Aguilar Melchor, C., Gama, N., Howe, J., Hülsing, A., Joseph, D., Yue, D.: The return of the SDitH. In: *Advances in Cryptology – EUROCRYPT 2023, Part V*. Lecture Notes in Computer Science, vol. 14008, pp. 564–596. Springer (2023). https://doi.org/10.1007/978-3-031-30589-4_20
35. Cheon, J.H., Choe, H., Devevey, J., Güneysu, T., Hong, D., Krausz, M., Land, G., Möller, M., Stehlé, D., Yi, M.: HAETAE: Shorter lattice-based Fiat–Shamir signatures. *IACR Transactions on Cryptographic Hardware and Embedded Systems* **2024**(3), 25–75 (2024). <https://doi.org/10.46586/tches.v2024.i3.25-75>
36. Kim, S., Ha, J., Son, M., Lee, B., Moon, D., Lee, J., Lee, S., Kwon, J., Cho, J., Yoon, H., Lee, J.: AIM: Symmetric primitive for shorter signatures with stronger security. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. pp. 401–415. ACM (2023). <https://doi.org/10.1145/3576915.3616579>