



Review

Post-Quantum PKI: A Survey of Applications and Benchmarking Practices

Maya Thabet ^{1,*} , Antonia Tsili ^{1,2,*} , Konstantinos Krilakis ^{1,2} and Dimitris Syvridis ^{1,2}

¹ Department of Informatics and Telecommunications, Panepistimioupolis, Ilisia, 16122 Athens, Greece; dsyvridi@di.uoa.gr (D.S.)

² Eulambia Advanced Technologies, Agiou Ioannou 24, Building Complex C, Ag. Paraskevi, 15342 Athens, Greece

* Correspondence: mthabet@di.uoa.gr (M.T.); tonyts@di.uoa.gr (A.T.)

Abstract

Post-quantum cryptography (PQC) is, and should be, currently dominating the field of cybersecurity, with many works designing and evaluating the transition of communications security to quantum-safe solutions. As the security level and implementations of post-quantum algorithms become more mature, the research on their application to realistic conditions changes accordingly, especially their application to widely adopted network architectures and corresponding protocols such as the Public Key Infrastructure (PKI). In this survey, we identified articles presenting ways of integrating PQC algorithms to PKI and classified related work according to the employed methods and benchmarking choices. The main results from many evaluations converge to similar conclusions on the performance of the most popular PC digital signature algorithms; however, modeling choices concerning architecture variants, hardware and measurement metrics vary. The diversity of the results and experimental setups makes comparison difficult and arrival at an objective conclusion regarding PKI requirements almost impossible. Ultimately, this review reveals a fragmented landscape of benchmarking practices for post-quantum PKI systems. The absence of standardized evaluation frameworks and common test environments limits the comparability and reproducibility of the findings. We aim to provide reference implementations, which are essential to guide the transition of PKI infrastructures toward robust, scalable, and quantum-resistant deployments.

Keywords: public key cryptography; post-quantum cryptography; pqc; pki; certificate; benchmarking framework; digital signatures



Academic Editors: Xianhui Lu and Jianfeng Wang

Received: 2 December 2025

Revised: 20 January 2026

Accepted: 9 February 2026

Published: 12 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

When two entities communicate, they must ensure they are interacting with the correct party and not an impostor. This is the purpose of authentication, in which digital signatures play a crucial role. Almost all modern networked communication relies on some form of authentication. Without it, entities cannot trust the origin of the messages they exchange. A common example is client–server communication, where a client (e.g., a browser) connects to an online service. The client must be able to verify that the server’s public key really belongs to the intended server and not to an attacker.

Public Key Infrastructure (PKI) addresses this public-key binding problem by issuing digital certificates. A certificate binds a public key to the identity of its owner through the digital signature of a certification authority. PKI is therefore a foundational layer of

trust in contemporary digital communication—by providing a mechanism to bind public keys to verified identities, it enables entities to authenticate each other and mitigates impersonation and man-in-the-middle attacks. Without PKI or an equivalent trust infrastructure, the confidentiality and integrity guarantees of higher-level protocols would be fundamentally weakened.

Today, the most widely deployed digital signature algorithms are Rivest–Shamir–Adleman (RSA) and the Elliptic Curve Digital Signature Algorithm (ECDSA). In 1994, Shor’s algorithm [1] showed that, on a sufficiently large quantum computer, the underlying hard problems of integer factorisation and discrete logarithms can be solved in polynomial time, breaking the security assumptions of RSA and ECDSA. As a consequence, once large-scale quantum computers become available, the integrity of the transferred data can no longer be reliably guaranteed and authentication between network entities will be compromised if classical signatures remain in use. This has motivated intense research into post-quantum cryptography (PQC), including quantum-safe authentication mechanisms. PQC-based digital signatures are, in principle, straightforward to adopt as software modules; however, in practice, they often come with non-trivial performance and memory costs.

This large-scale and costly integration must be carefully studied and benchmarked across realistic deployment settings. In spite of this, our literature review indicates that the study of PKI migration is underrated and not systematically explored. In a curated set of 600 PQC-related journal articles published in 2025, only $\approx 2.67\%$ focus explicitly on PKI and $\approx 14.33\%$ on digital signatures. A slightly broader OpenAlex [2] search for 2025 returns 655 PQC-related works overall, of which just 18 involve PKI and 30 focus on digital signatures. These results were obtained using tokenized queries to the respective catalogues for PQC-related papers, which were deduplicated and subsequently refined to PQ digital signature and PKI-focused articles. In this work, we narrow our attention to this gap. Our review methodology unfolded according to these PRISMA [3] steps:

1. **Selection criteria:** We initially defined the generic thematic scope of post-quantum digital signature algorithms and focused on articles published between 2020 and 2025, with more interest in recent works.
2. **Domain focus:** We refined our search to specifically target PKI-related articles, including concept works, but primarily focused on papers with experimental work, testbed descriptions and benchmarking records.
3. **Information sources:** We sourced articles published in prestigious conferences and journals, such as ACM CoNEXT, Internet of Things (Elsevier), and Cryptography (MDPI), and mainly excluded preprints. Some preprints were included to better reflect the latest trends.
4. **Data extraction:** We collected a range of performance metrics, as well as device specifications and network topology models. We also report code availability for reproducibility. We observe that the plethora of evaluation metrics and the impact of implementation choices hinder a comprehensive comparison across results, which motivates this work.

This paper’s aim is to provide a comprehensive survey of how PQC signature algorithms can be integrated in public key infrastructures instead of classical algorithms. The main contributions of the paper are listed as follows:

- An extensive accumulation and interpretation of recent papers that benchmark post-quantum signature algorithms in a PKI framework.
- The systematisation of collected works according to their topologies, for instance, the usage of a hierarchical chain-of-trust or simplistic client–server topologies.

- The classification of the assembled works by benchmarking metrics. While some of the papers have similar benchmarking techniques, an important subset of common practices produces results that are hard to interpret. For example, with each research group testing using different devices and the results being reliant on the underlying hardware, comparison of the calculations of the reviewed papers becomes difficult.
- The construction of a list of guidelines on how to make a benchmarking framework that adheres to standardized practices.

The roadmap of the current survey is as follows. In Section 2, we dive deeper into PKI fundamentals and explain how PKI enables secure communication and authentication using public key cryptography and digital certificates. In Section 3, we list different ways in which PQC can be integrated into PKIs, and provide recently proposed architecture paradigms. In Section 4, we evaluate and classify the reviewed works according to their benchmarking practices and metrics considerations. In Section 5, we summarise identified gaps in the existing literature and describe open problems, which can be easily mitigated through the widespread wide adoption of the practices that we introduce in Section 6, where we present guidelines on how to make a benchmarking framework. We conclude our paper by summarizing our observations and results.

2. Background and Foundations

2.1. PKI Fundamentals

A Public Key Infrastructure (PKI) is a system that enables secure communications and authentication over insecure networks. PKI relies on public key cryptography and the distribution of digital certificates. With public key cryptography, each network entity has a pair of cryptographic keys, a public key p_k that can be shared openly, and a private key s_k that must be kept in a safe place. Each public key uniquely matches the private key, so that the pair distinctively matches the entity's identity. Identification over the course of message exchange between peers is a crucial part of communication, and proper usage of (p_k, s_k) determines the communication's security and integrity.

Asymmetric cryptography provides the foundations for public key cryptography by associating pairs of keys (p_k, s_k) . In public key encryption, one encrypts with the public key and decrypts with the secret key: $C = \text{ENC}_{p_k}(M)$ and $M = \text{DEC}_{s_k}(C)$. In digital signatures, one signs with the secret key and verifies with the public key: $S = \text{SIGN}_{s_k}(M)$ and $\text{VERIFY}_{p_k}(M, S) = 1$. Security relies on the infeasibility of recovering M from C without s_k and the infeasibility of deriving s_k from p_k . These primitives underpin three main functionalities: public key encryption/KEMs, digital signatures and key exchange. While they all operate on a public key p_k corresponding to a secret key s_k , they serve different purposes by providing security to different aspects of communication. Public key encryption provides confidentiality, as only the holder of the secret key s_k can feasibly recover the message M from the ciphertext C . The purpose of key exchange is to establish a shared session key for confidentiality of subsequent communications over an insecure channel. While ephemeral Diffie–Hellman exchange (ECDHE) is usually employed, KEMs have emerged as the post-quantum cryptography counterparts that are expected to be universally adopted [4–6].

It becomes obvious that cryptographic algorithms from at least two of the aforementioned categories are needed. In realistic communication scenarios, the involved parties are authenticated, the integrity and non-repudiation of messages are preserved, and confidentiality is guaranteed, with ephemeral keys ensuring forward-secrecy; however, the currently employed protocols and schemes are vulnerable to quantum-assisted technologies. The scope of this work revolves around PKIs, where digital signatures are foundational components; thus, the use of digital signatures will be elaborated on in more

detail. The emergence of KEMs—and their interplay with digital signatures in future constructions—is briefly discussed.

2.1.1. Digital Signatures

Their general purpose is to ensure authentication, message integrity and non-repudiation, which are defined as shown in Definition 1. In a simplified example, Alice wants to send a message to Bob and Bob needs to make sure the received message was sent directly from Alice and it has not been amended during transfer.

1. Alice publishes her public key p_k^A and keeps her matching secret key s_k^A private.
2. Alice creates a plaintext M and computes a cryptographic hash $H = h(M)$.
3. Alice runs the signing algorithm with her secret key to obtain the digital signature $S = \text{SIGN}_{s_k^A}(H)$.
4. Alice sends the pair (M, S) to Bob over the insecure channel.
5. Bob computes $H' = h(M)$ and verifies the signature by checking $\text{VERIFY}_{p_k^A}(S) = H'$. If the check passes, the signature is valid.

Finally, Bob is able to verify the signature using p_k^A , which can only validate signatures produced with the matching s_k^A , with the latter assumed to be accessed only by Alice. Therefore, authentication is established and non-repudiation holds: Alice cannot plausibly deny having signed M . Due to the realisation of the comparison procedure $H' = h(M)$ with $\text{VERIFY}_{p_k^A}(S)$, Bob can also guarantee M 's integrity. Digital signatures are used in many aspects of everyday life concerning communication and security. Tan et al. [7], while studying the transition to post-quantum alternatives, identified 14 real-world applications across the financial, critical infrastructure, Internet, and enterprise sectors, with PKI variations spanning a significant subset of them.

Definition 1 (Bindel et al., 2017 [8]). *A digital signature scheme Σ is a tuple*

$$\Sigma = (\Sigma.\text{KEYGEN}, \Sigma.\text{SIGN}, \Sigma.\text{VERIFY})$$

of algorithms:

- $\Sigma.\text{KEYGEN}() \xleftarrow{\$} (s_k, p_k)$: the probabilistic key generation algorithm that returns a secret or signing key s_k and public or verification key $p_k \in \text{VK}_{\Sigma}$.
- $\Sigma.\text{SIGN}(s_k, m) \xleftarrow{\$} \sigma$: the probabilistic signature generation algorithm, which takes as input a signing key s_k and a message $m \in M_{\Sigma}$, and outputs a signature $\sigma \in S_{\Sigma}$. The space of random coins is R_{Σ} .
- $\Sigma.\text{VERIFY}(p_k, m, \sigma) \rightarrow 0$ or 1 : the verification algorithm, which takes as input a verification key p_k , a message m , and a signature σ , and returns a bit $b \in \{0, 1\}$. If $b = 1$, we say that the algorithm accepts; otherwise, we say that it rejects the signature σ for message m .

If a proof for Σ is provided in the random oracle model, we use H_{Σ} to denote the space of functions from which the random hash function is randomly sampled. We say that Σ is ϵ -correct if, for every message m in the message space, we have that

$$\Pr[\Sigma.\text{VERIFY}(p_k, m, \sigma) = 1 : (s_k, p_k) \xleftarrow{\$} \Sigma.\text{KEYGEN}(), \sigma \xleftarrow{\$} \Sigma.\text{SIGN}(s_k, m)] \geq 1 - \epsilon$$

where the probability is taken over the randomness of the probabilistic algorithms.

2.1.2. Certificates

The fundamental basis of PKI's operation is the digital certificate. This is an electronic credential that holds identifying information about a network entity and digitally binds the applicant's public key to their identity. The public key is bound to the entity through

the digital signature, which acts as a tamper-proof seal of approval that users can verify. The most widely used standard of the International Telecommunication Union (ITU) to define the format of public key certificates is X.509 [9].

X.509 specifies how a certificate should structure information such as the subject (the entity being authenticated), the issuer (the Certificate Authority or CA), the validity period, and the public key itself. In a typical public key infrastructure, these certificates are organized in a hierarchy: at the top sits a self-signed root CA certificate, which acts as a trust anchor and is distributed in operating-system or browser trust stores; below that are one or more intermediate CA (ICA) certificates, which are signed by the root (or by another intermediate) and are delegated the authority to issue certificates further down; and at the leaves of this hierarchy are end-entity (EE) certificates, which identify concrete subjects such as TLS servers, clients, or code-signing identities. The extensive standardisation and usage of the certificate and its structure, as well as the way that it is used in Transport Layer Security, have led to rigid implementations that cannot allow much alteration. This means that attempts to incorporate post-quantum features have limited scope; otherwise, they would require coordinated effort in updating the versatility of foundational operations, e.g., the way a certificate is read.

To obtain an X.509 certificate, an EE typically generates a Certificate Signing Request (CSR). The CSR contains the entity's public key and identifying information and is digitally signed with its private key to prove possession of that key. The CSR is then submitted to a trusted CA and that CA validates the information provided and signs the applicant's certificate. Root CA certificates are usually self-signed. X.509 also defines certificate revocation lists, which provide a mechanism for distributing information about certificates that have been revoked by a signing authority, as well as a certification path validation process. During path validation, a relying party verifies that an EE certificate was issued by a valid ICA (or sequence of intermediates), that each intermediate certificate was itself issued by a higher-level CA, and that this chain ultimately terminates at a trusted root CA in the verifier's trust store; this ordered sequence of certificates is known as a chain-of-trust. We hereafter imply that any certificates follow the X.509 standard.

2.1.3. Transport Layer Security

A widely adopted cryptographic protocol combining digital signatures with encryption to incorporate all security aspects described in the above is the Transport Layer Security (TLS), current version TLS 1.3. TLS combines asymmetric cryptography and symmetric cryptography; the asymmetric cryptography part is realized through the validation of the chain-of-trust that is embedded in the digital signature placed on PKI certificates; the shared session secrets are then established with an authenticated key agreement—typically ephemeral Diffie–Hellman (ECDHE) [10] in current deployments. The security of TLS is, thus, dependent on the public key algorithms used to produce and verify the digital signatures, which are presently vulnerable to quantum computer attacks. This further motivates the exploration of post-quantum alternatives.

The TLS handshake is the phase in which the client and server agree on cryptographic parameters, authenticate each other (at least the server), and derive shared keys for protecting application data. The handshake begins with a CLIENTHELLO, where the client sends its supported protocol version, cipher suites, extensions (e.g., signature algorithms, key exchange groups), and an ephemeral key share. The server replies with a SERVERHELLO selecting the version and cipher suite and returning its own key share. Both sides then compute a shared secret via the chosen key exchange and feed it into a KDF-based key schedule (e.g., HKDF) to derive handshake traffic keys. To authenticate the server, it sends its X.509 certificate chain and a CERTIFICATEVERIFY message containing a signature over

the handshake transcript. The client validates the certification path and verifies this signature. If client authentication is required, the server additionally requests a client certificate, and the client mirrors this procedure. Finally, both parties send a FINISHED message. The successful verification of FINISHED on both sides confirms transcript integrity and key agreement; all subsequent application data is encrypted and integrity-protected. This is a protocol comprising specific steps and features, and updating these to new versions can itself introduce issues [11]. Therefore, newly proposed post-quantum material cannot be straightforwardly incorporated due to its heavier size and, possibly, different usage (e.g., pre-shared keys). However, replacement with more advanced protocols would dictate extensive mobilisation and gradual modernisation of the global network.

2.2. Post-Quantum Digital Signature Schemes

Post-quantum digital signature techniques are intended to resist attacks from quantum computers, which are capable of breaking many classic public key algorithms. They are based on advanced mathematical foundations, different from the classical and vulnerable discrete logarithm and factorisation problems, leading to more complex implementations and larger keys and signatures that can be difficult to use in communication contexts, where performance is of importance. Post-quantum cryptography (PQC) algorithms can be categorised according to the problem they are based on, a problem that cannot be computationally solved by either conventional or quantum computers within a sensible timeframe. These categories are as follows.

2.2.1. Lattice-Based Cryptography

Such algorithms rely on the difficulty of solving problems on mathematical lattices, such as the Shortest Vector Problem (SVP) or Learning With Errors (LWE). LWE is defined as a list of $i \in [N]$ equations $\langle \mathbf{s}, \mathbf{a}_i \rangle \approx_{\chi} b_i \pmod{p}$ with error, where the error is generated by a probability distribution $\chi : \mathbb{Z}_p \rightarrow \mathbb{R}^+$ on $\mathbb{Z}_p$. The $\mathbb{Z}_p$ stands for the set of all integers modulo p . The problem is recovering $\mathbf{s}$ from such equations. In SVP, the target is to find a nonzero vector $\mathbf{v} \in \mathcal{L}$ of minimum Euclidean length (or approximate this length within some factor) given a lattice basis B for a lattice $\mathcal{L} \subset \mathbb{R}^n$. Efficiently solving LWE would let one solve standard lattice problems like (approximate) SVP/SIVP in the worst case. To obtain more compact keys and faster operations, the elements in module LWE (MLWE) are obtained from rings or modules rather than plain vectors. PQC algorithms derived from such problems have quickly become popular due to their lightweight nature, fast operations, and the hardness of the underlying problems. Recently, NIST-standardised versions of Dilithium [12] (ML-DSA [13]), Kyber [14] (ML-KEM [6]), and the soon-to-be-standardised Falcon [15] emphasise their applicability.

2.2.2. Code-Based Cryptography

The related mathematical problems concern decoding random linear error-correcting codes. A central problem is the syndrome decoding problem: given a parity-check matrix $H \in \mathbb{F}_q^{(n-k) \times n}$, a syndrome $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and a weight parameter w , find an error vector $\mathbf{e} \in \mathbb{F}_q^n$ with a Hamming weight of, at most, w , such that $H\mathbf{e}^T = \mathbf{s}$, where the notation $\mathbb{F}_q$ symbolises a finite field (Galois field) with q elements. In the underlying public key setting, the public key typically consists of a generator or parity check matrix of a code that is made to look ‘random’, while the secret key is a trapdoor. Code-based PQC schemes have a long history of cryptanalytic scrutiny and are regarded as conservative candidates; recent code-based KEMs, such as HQC [16,17] and CROSS [18], have been selected by NIST for standardisation, underlining their relevance in practice.

2.2.3. Hash-Based Cryptography

Hash-based schemes derive their security almost exclusively from the properties of cryptographic hash functions, typically collision resistance, (second) preimage resistance, and in some constructions, pseudorandomness. At their core, many hash-based signature schemes used as components build one-time or few-time signatures and authenticate large sets of public keys using Merkle trees, a hypertree whose branches correspond to a type of those signature components and the nodes are constructed with a type of signature that incorporates authentication path information. Stateless designs such as SPHINCS+ [19] (standardised by NIST as SLH-DSA [20]) provide post-quantum secure signatures without requiring long-term secret structures.

2.2.4. Multivariate Cryptography

Multivariate schemes base their security on the difficulty of solving systems of multivariate polynomial equations over finite fields. The underlying problem is often formulated as the MQ problem: Given m quadratic polynomials $f_i(x_1, \dots, x_n) \in \mathbb{F}_q[x_1, \dots, x_n]$, $i = 1, \dots, m$, and a target vector $\mathbf{y} \in \mathbb{F}_q^m$, find $\mathbf{x} \in \mathbb{F}_q^n$ such that $f_i(\mathbf{x}) = y_i$ for all i . In multivariate public-key schemes, the public key typically exposes a system of quadratic equations that appears random, while the secret key consists of affine transformations and a structured ‘central map’ that allow for efficient inversion. While several prominent multivariate proposals (e.g., Rainbow [21] and Mayo [22]) have faced significant cryptanalysis, ongoing research continues to refine designs and parameter choices [23].

2.2.5. Isogeny-Based Cryptography

Isogeny-based schemes rely on the difficulty of finding isogenies between elliptic curves, particularly within families of supersingular elliptic curves. An isogeny is a non-constant rational map $\varphi : E_1 \rightarrow E_2$ that is also a group homomorphism between the elliptic curve groups $E_1(\overline{\mathbb{F}}_q)$ and $E_2(\overline{\mathbb{F}}_q)$. The notation $\overline{\mathbb{F}}_q$ signifies the algebraic closure of the finite field $\mathbb{F}_q$. As previously, the notation $\mathbb{F}_q$ signifies the finite field with q elements. The underlying problem is typically phrased as: given two (supersingular) elliptic curves E_1 and E_2 over a finite field, known to be connected by a hidden isogeny of bounded degree, compute an explicit description of such an isogeny. Protocols then encode secret keys as compositions of small-degree isogenies in the isogeny graph, while public keys are the corresponding image curves. Despite offering very compact key sizes and an elegant group-action structure, several concrete isogeny-based proposals (e.g., SIKE) have been broken by powerful attacks, and no isogeny-based scheme is currently standardised.

3. Integration of Post-Quantum Signatures into PKI

There have been efforts to study plans and implementations for PKI migration to post-quantum solutions since 2017 [8,24], a year after NIST initiated the PQC standardisation process for digital signature and KEM proposed algorithms [25], in which 69 submissions were considered in total. Early works [26–28] investigate the performance of various promising PQ digital signature algorithms, including CRYSTALS–Dilithium [29], SPHINCS+ [30], qTESLA [31] (Section 2.2.1), Rainbow [21] (Section 2.2.4), MQDSS [32] (Section 2.2.4) and Picnic (symmetric-key cryptography), out of which only CRYSTALS–Dilithium and SPHINCS+ have been NIST-approved and formalised into the ML-DSA [13] and SLH-DSA [20] standards. Several of these works [27,28,33], and many in the following years [34–41], deal with ways of making TLS quantum-safe, while others explore alternative directions.

Several post-quantum PKI applications under study have emerged, with different aspects being explored in each work—creating an uneven benchmarking landscape.

3.1. Certificate Adaptations

A full-stack, backwards-compatible and crypto-agile transition to PQC requires foundational changes in PKIs, starting with the structure of the X.509 certificate. A typical certificate is shown in Figure 1. Out of the described fields, the ‘signature algorithm identifier’, which specifies the digital signature algorithm the CA used to sign the certificate, along with the ‘signature value’ and the ‘extensions’, which allows X.509v3 certificates to be flexibly extended with extra information, are the most important for the PQC transition and, by extension, this survey. According to this structure, there are five (5) ways that certificates can be altered, giving way to the representative certificate types of Table 1 [42,43]. Versatile options that cover compatibility issues and introduce lower overheads seem to suffer from lower security guarantees. As a result, innovative solutions require many adaptations of the surrounding processes to accommodate them, such as protocol extensions and error corrections.

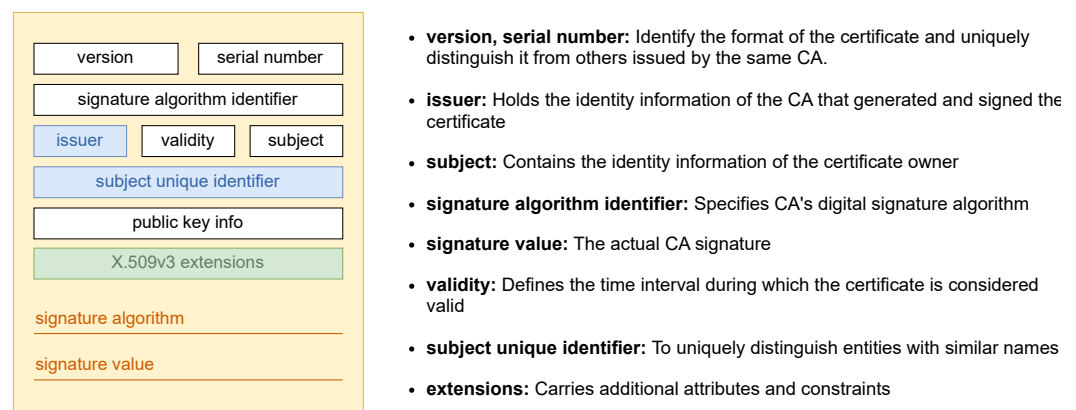


Figure 1. The structure of X.509 certificates with an explanation of each field. White boxes signify fields of version 1, blue boxes signify the additions of version 2, and green boxes signify additions of version 3.

Table 1. Certificate migration strategies towards PQC.

Certificate Type	Description	Advantages	Disadvantages
Pure PQC	A certificate where a post-quantum public key and signature are directly inserted into all relevant fields.	Directly replaces classical algorithms with PQC and enables a ‘pure’ post-quantum trust model without relying on classical signatures.	Increased key and signature sizes, potentially requiring segmented delivery [38]. Poor compatibility with legacy PKIs.
Parallel Chains	Two certificates, one with classical and one with PQC cryptographic material, are issued and validated independently in parallel.	Preserves backward compatibility, allows for the incremental deployment of PQC at endpoints that understand the additional chain.	Increases operational complexity, bandwidth and latency.

Table 1. *Cont.*

Certificate Type	Description	Advantages	Disadvantages
Hybrid [44]	The classical certificate structure is the primary carrier of identity and trust. Additional PQC material is embedded in a dedicated X.509 extensions. A certificate that embeds two keys: a classical key, which remains the primary bearer of identity and trust, and an additional post-quantum key incorporated as an extension (e.g., OIDs [45]).	Strong backward compatibility; pqc-aware clients can use the additional material to enable or test post-quantum validation. Supports gradual migration.	Security still depends on algorithms that may be broken by quantum adversaries. Adds some complexity in extension processing and policy design, alongside its increased size.
Composite	A certificate that embeds a single composite public key and composite signature. The composite cryptographic material emerges from the application of different component algorithms and the combination of their output. [46,47].	Strong robustness, flexible security policies and cryptographic agility during the transition period. A single certificate object compared to parallel chains.	Substantial size and performance overhead, due to aggregating multiple keys and signatures. Non-trivial interoperability challenges.
Chameleon [48]	A certificate with classical cryptographic material (base) that also embeds the Delta Certificate descriptor as a X.509 extension. The descriptor describes how to derive the PQC certificate (the delta) from the base.	Compact certificates and fully compatible with legacy infrastructure. Clear path for PQC-ready endpoints to obtain and validate a PQC-equivalent credential. Balances backward compatibility with PQC readiness.	Introduces additional complexity and new failure modes in reconstruction algorithms and delta-data handling. Authenticity initially remains anchored in the classical certificate.

3.2. Hybrid Authentication Approaches

We identify three distinct authentication approaches in modern cybersecurity applications: the classical, which relies on algorithms such as RSA and ECDSA, the post-quantum, based on the mathematical foundations described in Section 2.2, and a category that admits high-level combinations of both. One mode of such hybrid employment involves the use of a classical and a post-quantum algorithm for key exchange. TLS 1.3 already negotiates key exchange methods via the SUPPORTED_GROUPS extension and NAMEDGROUP values, so dedicating specific NAMEDGROUP values for new post-quantum and classical cryptography hybrids can easily be achieved through concatenating the shared secrets before feeding them to the Key Derivation Function (KDF). For authentication, separate classical and post-quantum signatures can be appended on a certificate, or can be concatenated together, as illustrated in Table 1. Some authors [40,49,50] have exploited the feasibility offered by the widely employed oqs-provider of OpenSSL, which was developed to readily integrate liboqs of the Open Quantum Safe project [24], and conducted some experiments to show how performance and available resources are affected. Other hybrid approaches comprise certificate chains linking classical and post-quantum digital signatures. We should note here that the divergent performance and security guarantees of different algorithms has led some authors to explore mixed-certificate chains [45], where different PKI levels make use of different post-quantum digital signature algorithms so as to match each algorithm's

advantages to each level’s requirements, as listed in Table 2. Some further hybrid solutions combine classical authentication with post-quantum KEMs, but these are out of this work’s scope.

Table 2. Reviewed papers that have either hybrid signature algorithms or key exchange, with reference to the best combination according to the claim of each paper.

Article	Hybrid	Best Combination
Sikeridis et al., 2020 [33]	key exchange	–
Raavi et al., 2021 [49]	digital signature	P256 + DilithiumII/Falcon512
Montenegro at al., 2025 [50]	digital signature	P521 + Falcon1024
Sim et al., 2025 [40]	digital signature	–

3.3. Emerging PKI Architecture Variants

Across the recent literature, the proposed or tested topologies—meaning those where the root CA, ICA and end entities actually live and the kind of devices they run on—fall into a small number of recurring patterns that are recombined in different ways. Deployment has been studied on a variety of hardware; every research article explores another type. A foundational classification can be derived from experiments run on moderate to high-performance computers or constrained devices. Tables A7 and A8 showcase articles falling in the respective categories.

3.3.1. Simulations on a Single Device

A first common pattern is that the root CA, any ICA and the EE are located on a single high-performance machine. In fact, some works [51,52] are specifically focused on benchmarking low-level digital signature cryptographic primitives, namely the key generation, signing and verification functions. Therefore, no interaction between processes or devices is needed. More advanced PKI operations include the TLS handshake, which can be measured on a simulated link between processes of the same device [36], as depicted in Figure 2. Here, the participating certificates are destined for direct usage and exchange; thus, there is no need to distinguish between PKI hierarchical levels. In this trivial case, the network entities use local, self-signed certificates and no CAs or separate devices are involved.

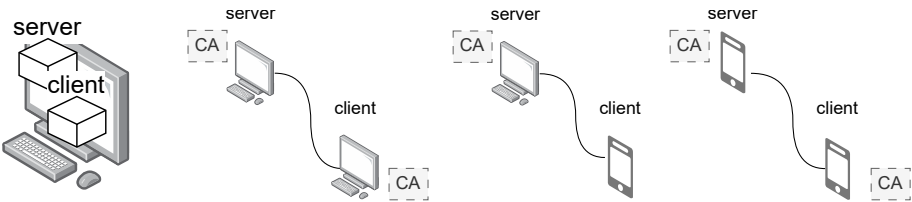


Figure 2. Client–server topologies with pre-provisioned or self-signed certificates. Common approaches include native or remote applications, with many works studying the behaviour of constrained devices.

3.3.2. Logical PKI Hierarchy Layers

A slightly more PKI-aware variant still uses one physical machine but treats CA and EE as separate logical roles [38,49]. Another major family of topologies keeps the PKI hierarchy logical, but separates client and server devices. Several works place a high-performance client in one location and remote servers in the cloud, with X.509 chains anchored in a CA/ICA structure [33,45], as illustrated in Figure 3. A closely related set of topologies pairs at least one constrained device with at least one high-performance machine. In [34], the constrained board is effectively an EE, the PC is a relatively powerful

EE or server, and the CA/ICA again exist only as the (native) issuers of the certificates used in the handshake. For instance, [27]’s bare-metal TLS study explores two scenarios: one with an x86 server and x86 client, and another with the same x86 server talking to an ARM Cortex-A53 client. Several works follow this trend into IoT space—Bürstinghaus-Steinbach et al. [28] dedicate the EE role to constrained devices, while the CA remains a background component that issued the certs. Mansoor and Döring [35,53] employ pre-provisioned certificates on IoT devices, or more capable servers. Zafar’s [39] hybrid McEliece framework, uses containerised client–server pairs across LAN, WAN emulation and IoT-mesh-like conditions, *but always on powerful Ubuntu hosts*.

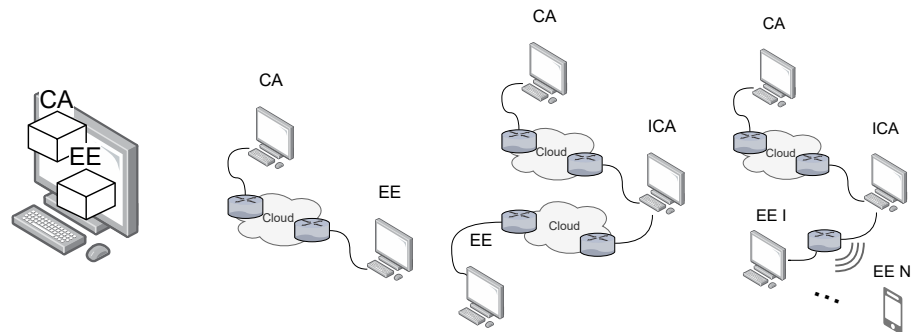


Figure 3. Topologies with a clear hierarchical distinction of PKI components. While the figure depicts a complex case where links connect remote network endpoints, they can also be inter-domain or simply direct.

3.3.3. Complete PKI Hierarchy Deployments

There is a small set of articles examining realistic scenarios of certificate management and distribution in a full-scale network (see Figure 3), showing the difficulty and the immaturity of quantum secure deployments that integrate a variety of hardware. Giron et al. [54] implement a complete hierarchical chain-of-trust, but they use only one device for the evaluation of their results. Sikeridis et al. [33] and Paul et al. [45] use the same topology, and also used different devices, but the diversity of the devices was only between the client and the server. Contrary to these examples, Tsili et al. [55] implemented a certificate distribution service with role-based access control through the use of ICAs, a root CA and EEs that function as the clients. They performed experiments that evaluate the effect of post-quantum algorithms within common network functionalities, including the dedication of a different device for each level in the topology and the deployment of multiple ICAs with the option to dynamically change the signature algorithm used to sign the certificates. Having a diversity of hardware and remote device interchange for each layer in the PKI hierarchy is deemed to be more realistic, as this adheres to the usual management concept that many EEs send requests to their local ICA, which then communicates with the root CA.

3.3.4. Specialised PKI Applications

Some of the more exotic topologies are driven by network architecture rather than by the PKI roles themselves. In [37], we find a triple-hybrid TLS experiment and client–server deployed as docker containers on separate QKD-enabled nodes; the CA and any ICA still behave as classical logical entities that issue X.509 certificates, not as extra physical devices. In [41] hybrid TLS in SDN networks is built on a multidomain QKD topology with and without trusted nodes; EEs are the TLS endpoints sitting in those SDN/QKD domains, while CA and ICA remain part of the control-plane infrastructure that authenticates them. Concept-heavy works on V2X certificates or the L-Band Digital Aeronautical Communi-

cation System (LDACS) propose domain-specific PKIs with root CAs, ICAs and EEs of specialised nature (Figure 4).

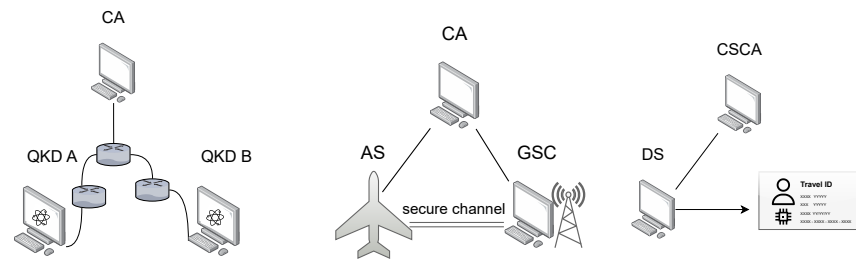


Figure 4. Topologies of specified applications (from left to right): (1) A QKD network with separate certificate provision. (2) A version of the L-Band Digital Aeronautical Communication System (LDACS), where the Aircraft Station (AS) and Ground Station Controller (GSC) are provided certificates from a CA. (3) PKI for electronic Machine Readable Travel Documents (MRTDs), where the Country Signing Certification Authorities (CSCA) provides certificates to the Document Signers (DS) that sign e-MRTD information.

4. Benchmarking Methodologies

After reviewing contemporary approaches to PKI architectures, it follows naturally to evaluate and classify relevant works according to their benchmarking practices and metrics considerations. When investigating the literature in this respect, we discover a fragmented and inconsistent ground with a wide range of measurable objects and scattered results. With each paper focusing placed on different angles of the PQC transition problem, we find little overlap among benchmarks, with most concerning the performance of basic, low-level digital signature operations (Definition 1). Across different benchmarkings, we find that most papers converge on SPHINCS+, Dilithium and Falcon, as depicted in Figure 5, as their primary candidates—SPHINCS+ and Dilithium because they are standardised, and Falcon because of its lightweight profile. Dilithium often emerges as the best solution in practice due to its balance of efficiency, memory footprint and standardisation status.

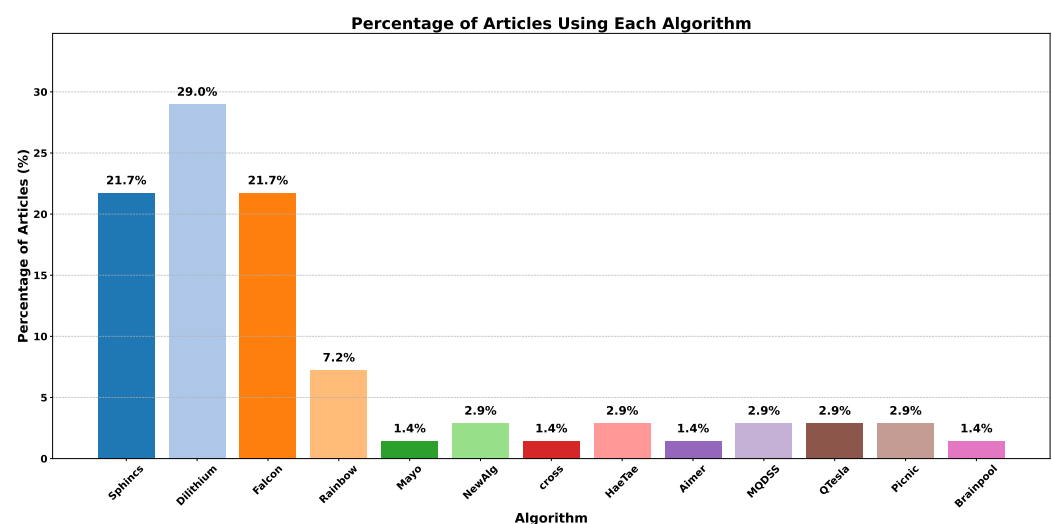


Figure 5. Percentage of articles that used each specified post-quantum algorithm family (e.g., Dilithium, Falcon, SPHINCS+, etc.) in their results and analysis. ‘NewAlg’ stands for the proposition of a new algorithm.

4.1. Microbenchmarks

We refer to post-quantum key generation, signing and verification measurements as microbenchmarks. These benchmarking metrics evaluate the performance of digital

signature algorithms in isolation (key generation, signing, verification), which is useful for understanding practical performance differences between algorithms and how these relate to their computational complexity. Hence, practical evaluations typically rely on dedicated benchmarking harnesses or tools, which are often included with the corresponding cryptographic libraries, to measure key generation, signing and verification performance. This arises as an easy way of understanding the behaviour of different digital signature algorithms and a common ground for researchers to make comparisons. Tables A1–A6 of Appendix A summarise our related findings with respect to the recently standardised algorithms. However, the speed of the operations makes measurements prone to noise and strongly affected by the underlying hardware and execution environment, which is never identical among works. While most measure wall-clock time, as seen in Figure 6, a more appropriate and meaningful metric is the estimation of clock cycles, as this reduces the impact of temporary effects of the operating system and clock frequency, enabling fairer comparisons across platforms, implementations and experimental setups. Furthermore, some CPU optimisations greatly benefit some algorithms in terms of performance, such as Advanced Vector Extensions 2 (AVX2), which is a CPU instruction set that improves performance for intensive computations by using SIMD (Single Instruction, Multiple Data) to process multiple data elements simultaneously. Nonetheless, many articles either avoid specifying whether such optimisations were activated or not, or omit baseline results without optimisations.

Percentage of Benchmarking Metric

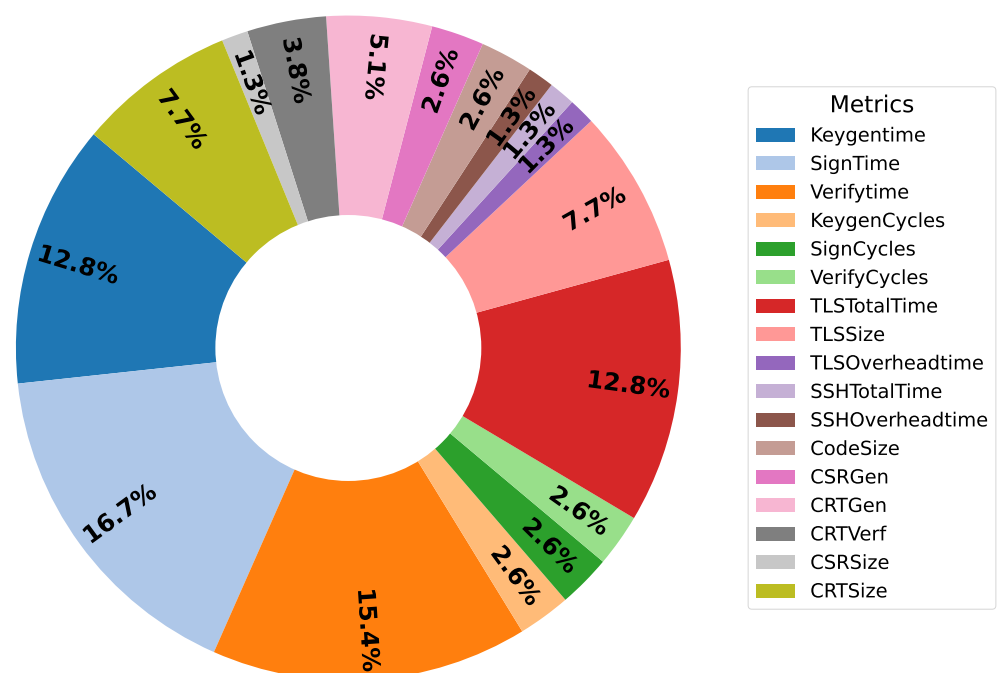


Figure 6. Percentage of benchmarking metrics related to post-quantum digital signatures according to 29 papers published between 2018 and 2025. ‘Keygen’, ‘Sign’ and ‘Verify’ components correspond to the core operations of digital signatures as seen in Definition 1. Labels that include the word ‘Time’ refer to wall-clock timing measurements. Labels that include the word ‘Cycles’ refer to CPU cycles. Labels that include the word ‘Size’ represent the calculation of memory requirements. ‘CRT’ stands for ‘Certificate’, ‘CSR’ stands for ‘Certificate Signing Request’; ‘Gen’ and ‘Verf’ mean generation and verification respectively. ‘TLSTotalTime’ is the calculated time for the TLS handshake completion and ‘TLSOverheadTime’ is the measured overhead introduced due to the adoption of post-quantum algorithms in place of the classical cryptography.

4.2. Certificate-Related Benchmarks

In addition to low-level cryptographic primitives, it is important to consider more high-level PKI operations [26,38,40,49,56], such as CSR creation, certificate signing, path validation, and revocation checks, as these constitute the essential functionalities of the infrastructure as seen by applications. Evaluating their performance after the addition of post-quantum algorithms in practice is challenging, since they are typically timed at the application level on a loaded operating system, where scheduling noise and background activity can affect fine-grained measurements. Nonetheless, such high-level timings are less prone to errors compared to microbenchmarks (Section 4.1) and thus remain meaningful for characterizing end-to-end behaviour. Finally, while latency is often the primary focus, the size of certificates and CSRs is also a critical factor, especially considering the memory overhead that post-quantum solutions, such as SPHINCS+, present; it affects storage, transmission, and caching, yet remains comparatively less explored in the literature.

4.3. Protocol Benchmarks

Regarding protocol benchmarks, we encounter a large number of works [27,34–41,49,56,57] measuring the total time to complete the TLS handshake, as expected. The next TLS-related metric to be most popular is the total memory that is occupied in order to bring the TLS handshake into conclusion. However, network conditions need to be properly modelled, with thorough benchmarking down to the recommended packet level, as such analysis can provide valuable insight about a post-quantum algorithm's empirical specifications—in [38], it is shown that the Falcon of the lowest Security Level is the only post-quantum algorithm that transfers its Server Hello message in a single packet. Also, realistic deployments on a variety of hardware are preferred. Some other certificate-related protocols, whose post-quantum migration is less explored, include the Simple Certificate Enrollment Protocol (SCEP) [58] and Automatic Certificate Management Environment (ACME) [54], where throughput and service latency are measured. While not pertaining to a specific protocol, the work of Tsili et al. [55], who developed certificate distribution services and a root CA from scratch, falls into the same category.

5. Summary and Open Problems

Shedding light on different aspects of PKI functionalities and operations during PQC transition is beneficial to the scientific community as it provides an insightful volume of data. Overall, we find that progress in transitioning is gradually introduced to large-scale deployments, with more significant and secure solutions focused on the algorithmic level, followed by protocol-level propositions and isolated application-specific presentations. Nevertheless, the lack of alignment impedes proper comparison and, eventually, the extraction of fruitful, broad and complete conclusions that would contribute to the successful conduct of common guidelines, universal standards and comprehensive recommendations.

Across the surveyed works, several systematic gaps emerge. First, most benchmarks report wall-clock execution time, but raw time measurements are highly platform-dependent, which limits their usefulness for cross-platform comparison and for drawing general conclusions about algorithmic efficiency. Such variation can be observed in our Tables A1–A6, even when considering one of the high-performance or constrained groups. Despite the high dispersion among the results, there is a clear preference for Dilithium's balanced efficiency and memory requirements for both groups, as shown in Figure 7. Nevertheless, Dilithium is lattice-based, a type of algorithm that is not yet as meticulously studied and challenged as other types, e.g., the memory-hungry SPHINCS+. Though scarce, some results were accompanied by publicly available implementations that support the adoption

and evaluation of post-quantum cryptographic schemes. The reference implementations are listed in Table 3.

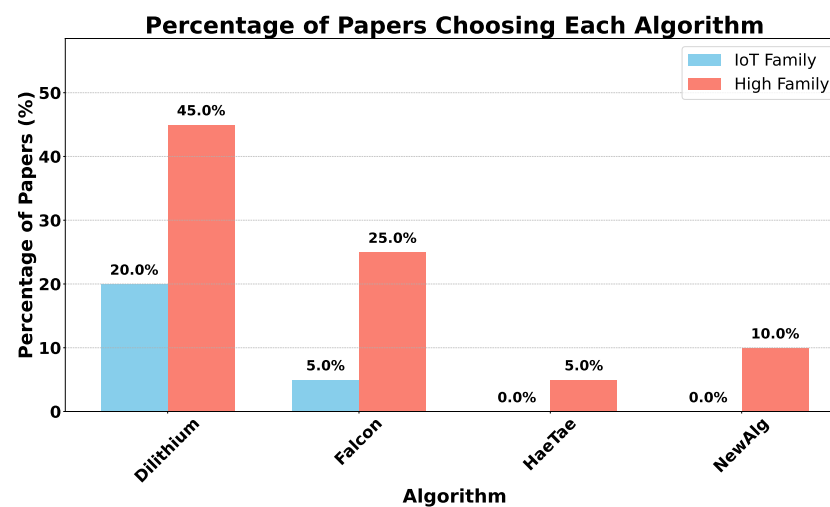


Figure 7. Percentage of articles recommending Dilithium, Falcon, the Korean HaeTae or a newly-proposed ('NewAlg') digital signature algorithm for high-performance or IoT scenarios.

Table 3. Reference implementations found in the reviewed papers.

Author	Year	Implementation	Description
DeAbiega et al. [27]	2020	ponchohunt1980/pq_tls [https://gitlab.com/ponchohunt1980/pq_tls] (accessed on 20/1/2026)]	Migration towards post-quantum TLS performance evaluation
Doring et al. [35]	2022	praxisprojekt-notebook [https://gitlab.com/3v3ryb0dy/praxisprojekt-notebook] (accessed on 20/1/2026)]	Migration towards post-quantum TLS performance evaluation
Sim et al. [40]	2025	minjoo97 [https://github.com/minjoo97] (accessed on 20/1/2026)]	Integrating kpqc in TLS, benchmarking of X.509 certificates, PQC migration
Giron et al. [54]	2024	AAGiron [https://github.com/AAGiron/acme-newchallenge] (accessed on 20/1/2026)]	implementation of the ACME protocol with Post-Quantum Cryptography (PQC)
Rios et al. [36]	2025	AAGiron [https://github.com/montenegro-montes/TLS-PQ] (accessed on 20/1/2026)]	A complete environment to test and benchmark the TLS protocol using classical, hybrid, and post-quantum Key Encapsulation Mechanisms (KEMs) and signature cryptographic primitives inside Docker containers.
Spalas et al. [59]	2025	LDACS [https://github.com/konnnGit/LDACS-Security-Evaluation] (accessed on 20/1/2026)]	Transforming the L-Band Digital Aeronautical Communication System (LDACS) aviation communications into a quantum-safe setup

In general, we find important information about the conduct of experiments is omitted, for instance the activation of AVX2 or the employment of DER-encoded certificates, which affect both the popular metrics of latency and memory usage. Second, metrics that are particularly important in IoT settings—such as energy consumption, power draw, and supply voltage behaviour—are clearly relevant yet remain relatively under-reported. Third, while TLS is the dominant focus, other protocols that are increasingly important in practice, such

as QUIC, and ACME, receive comparatively little attention and warrant further exploration. Lower-level protocol benchmarking, such as TCP, could also prove quite insightful and provide results that help understand, predict and thoroughly explain the benchmarks obtained for higher-level schemes. Moreover, cost-benefit ratios (for example, latency per bit of security, or bandwidth overhead per security level) are almost never quantified explicitly, even though such normalised metrics would make trade-offs much clearer. A summary of the consistency problems emerging from the reviewed literature is outlined in Table 4.

Table 4. A summary of the open problems that emerged from the reviewed literature, along with some works that include the addressed problem.

Description	References
Platform-dependent benchmarking using raw execution time.	[27,33–35,40]
Limited use of hardware-normalised metrics (e.g., CPU cycles)	Observed in all except: [40,51]
Overreliance on latency-only performance metrics.	[33,35,36]
Lack of reporting energy and power consumption metrics for IOT devices.	[28,34,53]
Narrow focus on TLS-centric evaluations.	[33,34,36]
Use of algorithm families instead of concrete variants	[53,58,60]
Limited benchmarking of certificate-level PKI operations.	[40,45]
Unclear or missing reporting of optimisation settings.	[27,28,33,35,36,50,53]

Finally, many inconsistencies arise regarding the presentation of results. Examples include some works referring to families of algorithms, instead of specific variants, where cryptographic material shows vast differences in terms of size. Other works use classical along with post-quantum methods and explore only a subset of these hybrid pairs, which is, for the most part, distinct compared to subsets selected from other works. In addition, reported metrics are often heterogeneous in both units and granularity (e.g., mixing cycles, milliseconds and qualitative labels), and are sometimes given without full disclosure of parameters such as security level, implementation details, hardware configuration or compiler settings. Together, these issues significantly hinder direct comparison between studies, conceal meaningful performance and security trade-offs, and complicate any attempt to derive clear, implementation-agnostic guidance for practitioners.

6. Toward a Standardised Benchmarking Framework

A move towards standardised benchmarking for post-quantum and hybrid public key schemes requires a more disciplined and transparent methodology than what is currently common practice. First, algorithm identifiers should always refer to specific variants rather than families (e.g., “Dilithium3” rather than simply “Dilithium”), since sizes and performance differ substantially across variants, and the exact implementation and library versions used must be clearly specified.

Benchmark suites should clearly distinguish between microbenchmarks, certificate- and PKI-level operations (where DER-encoding should be clearly reported), and end-to-end protocol benchmarks. Microbenchmarks of key generation, signing and verification should always be reported and kept in a separate table from higher-level protocol results. Reported metrics should explicitly distinguish between measurements taken on constrained, edge and high-performance devices, and each paper should designate at least one concrete baseline platform that is inexpensive and widely available, so that the results can be reproduced and extended (for example, a particular off-the-shelf laptop or development board). For constrained and IoT environments, energy- and power-related metrics (e.g., energy per operation, average and peak power draw) should be treated as results of primary priority.

All measurements should be reported in a small, agreed set of metrics and units: at a minimum, estimated clock cycles, together with data sizes in bytes. Wherever possible,

performance should be expressed in terms of operations per second (throughput) rather than only time per operation, as this yields clearer and more directly comparable metrics across implementations and platforms. For digital signature benchmarks that depend on input size, file sizes should be chosen to reflect realistic use cases (for example, certificate-sized messages), while research based on testbeds including certificate circulation should benchmark CSR creation, certificate creation and verification, with clear correlation indications to the primitives involved in each operation, such as CSR creation versus key generation. At the protocol level, benchmarks should include network-level metrics such as TCP segment sizes and the total number of handshake packets. When network distance is varied, a baseline distance expressed in hops should be given to make topologies less ambiguous. Along the proposed lines, a checklist for reporting benchmarks inspired by MLPerf [61] guidelines is given below:

- I Benchmark Goal and Scope
 - **Objective:** Hypothesis definition.
 - **Readiness:** Development stage.
 - **Target level:** Microbenchmarks, application, protocol or system-wide performance.
- II System specifications
 - **Device fleet:** Number of devices.
 - **Network topology:** Physical and simulated links.
 - **Hardware:** Computational power, memory availability, processor type, memory type.
 - **System specifications:** Operating system.
- III Implementation details
 - **Algorithmic variant:** Name of the algorithmic family and the variant defined by the cryptographic material specifications, such as key size.
 - **Software:** Programming language, framework functionality and version.
 - **Dependencies:** Cryptographic libraries, cryptoengine.
 - **Certificate format:** Field organisation, version, encoding, size of the certificates.
- IV Metrics per scenario
 - In Tables 5 and 6, we provide a template of recommended metrics, units and parameters that should be recorded during performing benchmarking experiments on a cybersecurity system or framework. It is organised according to the corresponding research question and focus, so that benchmarks of more complex scenarios (towards the bottom of the table) build on top of lower-level measurements.

Table 5. Benchmarking template organised by target level. Wall-clock times should be reported as distributions with variance and explicit external dependencies. ‘DS’ stands for ‘digital signature’ and ‘Ops’ stands for ‘operations’.

Target Level	Example Scenario	Operation	Primary Metrics		
			Latency	Throughput	Optimizations
Algorithm	New DS	Key generation	Clock cycles	Ops/s	AVX2, other or none
		Sign	Clock cycles	Ops/s	AVX2, other or none
		Verify	Clock cycles	Ops/s	AVX2, other or none
Application	PKI workflow	CSR creation	Wall-clock	–	–
		Certificate creation	Wall-clock	–	–
		Certificate verification	Wall-clock	–	–
		Service response	Wall-clock	Served req./s	–

Table 5. *Cont.*

Target Level	Example Scenario	Operation	Primary Metrics		
			Latency	Throughput	Optimizations
Protocol	TLS	Handshake	Wall-clock	Total packets	–
			Wall-clock	Total packets	–
System	Deployment	End-to-end exchange	Wall-clock	No. of clients	–
		Hardware variety	Per device	Request latency	–
		Energy (constrained)	–	–	–

Table 6. Auxiliary benchmarking metrics organised by target level. These metrics should be reported alongside primary performance measurements.

Target Level	Scope	Auxiliary Metric	Unit/Representation
Algorithm	Key material	Key size	Bytes/bits
	Signature operations	Signature size	Bytes
Application	PKI workflow	CSR size	Bytes
	Certificates	Certificate size	Bytes
	Service response	Failed requests	Count/ratio
Protocol	TLS handshake	Total handshake size	Bytes
	Transport layer	TCP segment size	Bytes
System	Network topology	No. of hops	Count
	Deployment constraints	Energy consumption	Joules/power profile

7. Conclusions

In conclusion, we thoroughly surveyed recent papers focusing on PKI applications and their migration to PQC digital signatures. We examined the provided benchmarks and identified an uneven landscape of results that is hard to fruitfully interpret. We listed emerging topologies with regards to PKI components and network infrastructure, ultimately hovering over a clearly defined set of physical layouts—either all processes are co-located on a powerful machine, e.g., a single CA plus one or more EEs; high-performance clients exchange messages with cloud-hosted servers whose certs come from a CA/ICA chain; mixed high-performance and constrained EEs communicate over LAN or Internet while CAs stay offline; containerised EEs are placed on QKD or SDN nodes; and embedded-only EEs work with pre-issued certificates. Overall, despite varied application domains (web, IoT, V2X, aeronautical, SDN/QKD), the underlying deployment patterns are surprisingly uniform while complex configurations are mostly left unexplored (e.g., fully distributed PKIs, dynamic CA placement, on-device certificate issuance, multi-hop CA chains under realistic routing). We also reviewed various benchmarking techniques and summarised the results, noticing an overreliance on wall-clock time without normalisation, leading to platform-specific results, the undermining of energy/power/voltage metrics that are crucial for IoT/edge devices, and the inconsistent reporting of certificate-level metrics (sizes, encoding, generation/verification time) and network metrics (segment sizes, number of handshake packets, hops). Due to these observations, we proposed a set of guidelines for producing meaningful and consistent benchmarks across devices, although integration of a variety of high-level protocols requires new considerations. In order to facilitate adoption, we provided a practical template that can be used by researchers to report benchmarks in a uniform and comprehensive way. Eventually, if the community converges on a small, well-specified set of multi-complex topologies and benchmark conventions, then the results across papers will become directly comparable. This, in turn, will make it easier to select

concrete algorithms and configurations for real-world deployments during and after the PQC transition.

Author Contributions: Conceptualisation, M.T. and A.T.; Methodology, M.T. and A.T.; Formal Analysis, M.T. and A.T.; Investigation, M.T., K.K. and A.T.; Data Curation, M.T. and A.T.; Writing – Original Draft Preparation, M.T.; Writing – Review and Editing, A.T.; Visualisation, M.T.; Supervision, A.T. and D.S.; Project Administration, D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the European Union through the projects ALLEGRO (Grant Agreement No. 101092766) and HellasQCI (Grant Agreement No. 101091504).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: Authors Antonia Tsili, Konstantinos Krilakis, and Dimitris Syvridis were employed by the company Eulambia Advanced Technologies. The remaining author declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Tables with Results of Dilithium, SPHINCS+ and Falcon with Infrastructure Specifications

Table A1. Metrics of papers with high-performance devices that use Dilithium with security levels I, V. The time to generate keys, sign, verify and complete a TLS handshake is measured in milliseconds. Certificate sizes are measured in bytes. The device specifications are listed in Table A7. An asterisk (*) signifies that the corresponding certificate is DER-encoded.

Security	Authors	Year	KeyGen	Sign	Verify	TLS (ms)	CRTSize
II	Sikeridis et al. [33]	2020	–	–	–	–	400
	Pradel et al. [26]	2020	0.00300	–	0.01300	8000	–
	Pradel et al. [26]	2020	0.00340	–	0.01270	7000	–
	DeAbiega et al. [27]	2020	0.128	0.557	0.144	–	–
	Raavi et al. [49]	2021	3.07	–	–	5340	–
	Tasopoulos et al. [34]	2022	–	–	–	91.10	–
	Paul et al. [45]	2022	0.107	0.414	0.121	–	–
	Doring et al. [35]	2022	–	0.887	0.197	0.902	–
	Opilka et al. [52]	2024	0.500	25.920	378.400	–	–
	Sim et al. [40]	2025	–	–	–	–	* 3977
	Seo et al. [56]	2025	–	–	–	–	4097
	Al-Dabbagh et al. [58]	2025	–	1.950	–	5621	–
	Abbasi et al. [60]	2025	0.220	0.310	0.190	–	–
	Raavi et al. [38]	2025	0.00010	0.00019	0.00005	3.710	5340
	Khan et al. [62]	2025	–	–	–	2.690	–
III	Raavi et al. [49]	2021	3.160	–	–	7391	–
	Sikeridis et al. [33]	2022	–	–	–	80	–
	Tasopoulos et al. [34]	2022	–	–	–	153.492	–
	Opilka et al. [52]	2024	0.510	25.380	363.000	–	–
	Fitzgobbon et al. [57]	2024	–	–	–	–	–
	Al-Dabbagh et al. [58]	2025	–	2.080	–	7716	–
	Abbasi et al. [60]	2025	0.360	0.420	0.250	–	–
	Montenegro et al. [50]	2025	–	0.164	0.065	–	7504
V	Raavi et al. [49]	2021	3.160	–	–	7391	–
	Sikeridis et al. [33]	2022	–	–	–	80	–
	Tasopoulos et al. [34]	2022	–	–	–	153.492	–

Table A2. Metrics of papers with constrained devices that use Dilithium with security levels I, V. The time to generate keys, sign, verify and complete a TLS handshake is measured in milliseconds. Certificate sizes are measured in bytes. The device specifications are listed in Table A8.

Security	Authors	Year	KeyGen	Sign	Verify	TLS (ms)	CRTSize
II	DeAbiega et al. [27]	2020	0.202	0.882	0.227	–	–
	Paul et al. [45]	2022	2.040	11.900	2.210	–	–
	Tasopoulos et al. [34]	2022	12.10	25.40	69.60	96.30	–
	Mansoor et al. [53]	2024	0.064	0.171	0.061	8.650	–
	Abbasi et al. [60]	2025	1.050	1.250	0.900	–	–
	Khan et al. [62]	2025	–	–	–	177.550	–

Table A2. Cont.

Security	Authors	Year	KeyGen	Sign	Verify	TLS (ms)	CRTSize
III	Tasopoulos et al. [34]	2022	19.438	39.309	16.244	157.126	–
	Mansoor et al. [53]	2024	0.105	0.277	0.100	14.140	–
	Fitzgobbon et al. [57]	2024	–	–	–	–	–
	Abbasi et al. [60]	2025	1.350	1.550	1.030	–	–
V	Tasopoulos et al. [34]	2022	19.438	39.309	16.244	157.126	–

Table A3. Metrics of papers with high-performance devices that use Falcon with security levels I, V. The time to generate keys, sign, verify and complete a TLS handshake is measured in milliseconds. Certificate sizes are measured in bytes. The device specifications are listed in Table A7. An asterisk (*) signifies that the corresponding certificate is DER-encoded.

Security	Authors	Year	KeyGen	Sign	Verify	TLS	CRTSize
I	Pradel et al. [26]	2020	0.0578	–	–	0.0919	5200
	Pradel et al. [26]	2020	0.0229	–	–	0.0501	3700
	Raavi et al. [49]	2021	12.030	–	–	–	2366
	Paul et al. [45]	2022	20.100	5.900	0.042	–	–
	Tasopoulos et al. [34]	2022	–	–	–	285.951	–
	Doring et al. [35]	2022	–	8.333	0.096	0.761	–
	Opilka et al. [52]	2024	8.910	25.290	358.800	–	–
	Giron et al. [54]	2024	–	–	–	–	–
	Khan et al. [63]	2024	–	–	–	2.360	–
	Al-Dabbagh et al. [58]	2025	–	2.050	–	–	2581
	Montenegro et al. [50]	2025	–	0.422	0.060	–	2450
	Abbasi et al. [60]	2025	0.550	0.650	0.090	–	–
	Raavi et al. [38]	2025	0.0083	0.00042	0.00005	3.830	2358
	Sim et al. [40]	2025	–	–	–	–	* 1788
	Rios et al. [36]	2025	–	–	–	1.600	–
V	Raavi et al. [49]	2021	26.390	–	–	–	4414
	Tasopoulos et al. [34]	2022	–	–	–	589.058	–
	Doring et al. [35]	2022	–	18.182	0.194	1.124	–
	Opilka et al. [52]	2024	20.310	26.020	358.100	–	–
	Al-Dabbagh et al. [58]	2025	–	2.310	–	–	4666
	Montenegro et al. [50]	2025	–	0.837	0.122	–	4506
	Abbasi et al. [60]	2025	0.850	1.050	0.150	–	–
	Raavi et al. [38]	2025	0.02275	0.00072	0.00010	4.540	4426
	Rios et al. [36]	2025	–	–	–	2.000	–
	Sim et al. [40]	2025	–	–	–	–	* 3304

Table A4. Metrics of papers with constrained devices that use Falcon with security levels I, V. The time to generate keys, sign, verify and complete a TLS handshake is measured in milliseconds. Certificate sizes are measured in bytes. The device specifications are listed in Table A8.

Security	Authors	Year	KeyGen	Sign	Verify	TLS	CRTSize
I	Tasopoulos et al. [34]	2022	1266.667	243.881	3.275	288.305	–
	Paul et al. [45]	2022	158.000	35.700	0.435	–	–
	Mansoor et al. [53]	2024	–	–	–	25.950	–
	Khan et al. [63]	2024	–	–	–	189.890	–
	Abbasi et al. [60]	2025	2.200	2.450	0.450	–	–
V	Tasopoulos et al. [34]	2022	4802.667	527.789	6.852	594.495	–
	Mansoor et al. [53]	2024	45.530	1.003	0.178	53.500	–
	Abbasi et al. [60]	2025	3.400	3.850	0.700	–	–

Table A5. Metrics of papers with high-performance devices that use SPHINCS+-fast with security level II, III, V. The time to generate keys, sign, verify and complete a TLS handshake is measured in milliseconds. Certificate sizes are measured in bytes. The device specifications are listed in Table A7.

Security	Authors	Year	KeyGen	Sign	Verify	TLS	CRTSize
I	Paul et al. [45]	2022	1.75	43.3	2.46	–	–
	Opilka et al. [52]	2024	0.7	52.23	270.9	–	–
	Al-Dabbagh et al. [58]	2025	–	5.91	–	–	24168
	Montenegro et al. [50]	2025	–	11.86	1.1	–	23535
	Raavi et al. [38]	2025	–	–	–	53.73	23442
III	Opilka et al. [52]	2024	0.87	43.38	194.2	–	–
	Montenegro et al. [50]	2025	–	17.36	1.32	–	48710
V	Opilka et al. [52]	2024	1.62	55.94	192.1	–	–
	Montenegro et al. [50]	2025	–	36	1.41	–	67954
	Rios et al. [36]	2025	–	–	–	47	120000

Table A6. Metrics of papers with constrained devices that use SPHINCS+ fast with security level II, III, V. The time to generate keys, sign, verify and complete a TLS handshake is measured in milliseconds. Certificate sizes are measured in bytes. The device specifications are listed in Table A8.

Security	Authors	Year	KeyGen	Sign	Verify	TLS	CRTSize
I	Bürstinghaus et al. [28]	2020	710	22000	950	970	–
	Tasopoulos et al. [34]	2022	137.75	3361	190.167	–	–
	Paul et al. [45]	2022	7.33	183	10.2	–	–
	Mansoor et al. [53]	2024	0.813	19.197	1.535	–	–

Appendix B. Tables with Technical Specifications for Each Experimental Article

Table A7. All the libraries and hardware specifications that were used by the reviewed papers employing high-performance devices.

Article	Year	Library	High-Performance Environment
Sikeridis et al. [33]	2020	OpenSSL	Ubuntu 18.04, Intel i7-8665u, 4 cores, 1.9 GHz, 8 GB RAM
Pradel et al. [26]	2020	liboqs	Ubuntu, 8 GB RAM, Intel i5-3470 @ 3.20 GHz
Raavi et al. [49]	2021	liboqs	AMD Ryzen 7 1700x, 8-core, 3.4 GHz, 32 GB RAM
Kuang et al. [51]	2022	liboqs	Intel i7-10700, 16 cores, 2.90 GHz
Tasopoulos et al. [34]	2022	pqm4	Ubuntu 20.04, Intel i7-1165G7, 8 cores, 2.8 GHz
Paul et al. [45]	2022	WolfSSL	Intel i5-6300U quad-core @ 2.4 GHz, 8 GB RAM

Table A7. Cont.

Article	Year	Library	High-Performance Environment
Doring et al. [35]	2022	liboqs	Intel Pentium Silver J5005, Quad-core, 1.5–2.8 GHz, 16 GB DDR4
Garcia et al. [37]	2023	liboqs	Remote Docker containers on QKD nodes
Opilka et al. [52]	2024	liboqs	Ubuntu 22.04.1, 12 GB RAM, 8 cores
Fitzgobbon et al. [57]	2024	liboqs	Windows 11, Ryzen 9 5950x (16 cores, 4.9 GHz), 64 GB RAM
Al-Dabbagh et al. [58]	2025	liboqs	Intel i7-6820HQ, 8 GB RAM
Abbasi et al. [60]	2025	liboqs	Intel i5-10210U (1.6/4.2 GHz), 4 cores, 8 logical cores, 8 GB RAM, Windows 10 with WSL2
Raavi et al. [38]	2025	liboqs	4 cores, 8 GB RAM
Sim et al. [40]	2025	OpenSSL	MacBook Pro (M1 Pro, 16 GB RAM, macOS 14.4)
Seo et al. [56]	2025	liboqs	Ubuntu
Rios et al. [36]	2025	liboqs	Docker instances
Giron et al. [54]	2024	—	8 GB memory, 2 CPUs
Khan et al. [62]	2025	liboqs	AMD Ryzen 7 5700U @ 1.8 GHz, 16 GB RAM

Table A8. All the libraries and hardware specifications that were used by the reviewed papers employing constrained devices.

Article	Year	Library	IoT/Embedded Environment
Bürstinghaus-Steinbach et al. [28]	2020	OpenSSL	ESP32-PICO-KIT V4 (Xtensa LX6, 240 MHz, 520 kB RAM, 16 MB Flash)
Tasopoulos et al. [34]	2022	pqm4	NUCLEO F439ZIlb (ARM Cortex-M4, 180 MHz, 192 KB SRAM, 2 MB Flash)
Paul et al. [45]	2022	WolfSSL	Raspberry Pi 3 Model B (ARM Cortex-A53 @ 1.2 GHz, 1 GB RAM)
Mansoor et al. [53]	2024	liboqs	IM30 A53 Quad-core ARM Embedded Board (2.0 GHz, 4 GB RAM, 32 GB storage)
Fitzgobbon et al. [57]	2024	liboqs	Raspberry Pi 4 (ARMv8 Cortex-A72, 1.5 GHz, 4 GB RAM)
Abbasi et al. [60]	2025	liboqs	ARM Cortex-A53 (1.4 GHz), 4 cores, 1 GB RAM, Debian 11
Sim et al. [40]	2025	OpenSSL	Raspberry Pi 5 (Cortex-A76, 8 GB RAM, Ubuntu 24.04)
Khan et al. [63]	2025	liboqs	Raspberry Pi (Quad-core Cortex-A72 @ 1.8GHz, 8 GB RAM)

References

- Shor, P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* **1997**, *26*, 1484–1509. <https://doi.org/10.1137/S0097539795293172>.
- Priem, J.; Piwowar, H.; Orr, R. OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *arXiv* **2022**. <http://arxiv.org/abs/2205.01833>.
- Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, Egyptian Agency for Research, B.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71.
- Moody, D.; Perlner, R.; Regenscheid, A.; Robinson, A.; Cooper, D. *Transition to Post-Quantum Cryptography Standards*; NIST Internal Report 8547 ipd IR 8547; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2024. <https://doi.org/10.6028/NIST.IR.8547.ipd>.
- Alagic, G.; Barker, E.; Chen, L.; Moody, D.; Robinson, A.; Silberg, H.; Waller, N. *Recommendations for Key-Encapsulation Mechanisms*; NIST Special Publication 800-227 SP 800-227; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2025. <https://doi.org/10.6028/NIST.SP.800-227>.
- National Institute of Standards and Technology (US). *Module-Lattice-Based Key-Encapsulation Mechanism Standard*; Technical Report NIST FIPS 203; National Institute of Standards and Technology: Washington, DC, USA, 2024. <https://doi.org/10.6028/NIST.FIPS.203>.
- Tan, T.G.; Szalachowski, P.; Zhou, J. Challenges of post-quantum digital signing in real-world applications: A survey. *Int. J. Inf. Secur.* **2022**, *21*, 937–952. <https://doi.org/10.1007/s10207-022-00587-6>.

8. Bindel, N.; Herath, U.; McKague, M.; Stebila, D. Transitioning to a Quantum-Resistant Public Key Infrastructure. In *Post-Quantum Cryptography*; Lecture Notes in Computer Science; Lange, T., Takagi, T., Eds.; Springer International Publishing: Cham, Switzerland, 2017; Volume 10346, pp. 384–405. https://doi.org/10.1007/978-3-319-59879-6_22.
9. International Telecommunication Union. X.509: *Information Technology—Open Systems Interconnection—The Directory: Public-Key and Attribute Certificate Frameworks*; Technical Report Recommendation X.509; ITU-T: Geneva, Switzerland, 2019.
10. Diffie, W.; Hellman, M.E., New Directions in Cryptography. In *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, 1st ed.; Association for Computing Machinery: New York, NY, USA, 2022; pp. 365–390.
11. Souppaya, M.; Polk, T.; Barker, W.; Kent, J. *Addressing Visibility Challenges with TLS 1.3 Within the Enterprise*; NIST Special Publication 1800-37; National Institute of Standards and Technology (NIST), National Cybersecurity Center of Excellence (NCCoE): Gaithersburg, MD, USA, 2025. <https://doi.org/10.6028/NIST.SP.1800-37>.
12. Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. In *IACR Transactions on Cryptographic Hardware and Embedded Systems*; Ruhr-Universität Bochum Universitätsstr: Bochum, Germany, 2018; pp. 238–268. <https://doi.org/10.13154/tches.v2018.i1.238-268>.
13. National Institute of Standards and Technology (US). *Module-Lattice-Based Digital Signature Standard*; Technical Report NIST FIPS 204; National Institute of Standards and Technology: Washington, DC, USA, 2024. <https://doi.org/10.6028/NIST.FIPS.204>.
14. Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehle, D. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In *Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, London, UK, 24–26 April 2018; pp. 353–367. <https://doi.org/10.1109/EuroSP.2018.00032>.
15. Fouque, P.A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Prest, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z.; et al. *Falcon: Fast-Fourier lattice-Based Compact Signatures over NTRU*; Post-Quantum Cryptography Standard; IBM: Armonk, NY, USA, 2018; Volume 36, pp. 1–75.
16. Gaborit, P.; Zémor, G.; Bos, J.; Dion, A.; Lacan, J.; Robert, J.M.; Véron, P.; Barreto, P.L.; Ghosh, S.; Gueron, S.; et al. *Hamming Quasi-Cyclic (HQC)—Specifications (Version 2025-08-22)*; Technical Report, Team HQC/PQC-HQC; 2025. Available online: https://pqc-hqc.org/doc/hqc_specifications_2025_08_22.pdf (accessed on 8 February 2026).
17. National Institute of Standards and Technology. *NIST Selects HQC as Fifth Algorithm for Post-Quantum Encryption*; Technical Report; NIST: Gaithersburg, MD, USA, 2025.
18. A. N. iCROSS Security Details. *iCROSS Security Details: Deliverable/Report Type, Version 2.1*. iCROSS Consortium, 2025. Available online: https://www.cross-crypto.com/CROSS_SecurityDetails_v2.0.pdf (accessed on 8 February 2026).
19. Bernstein, D.J.; Hülsing, A.; Kölbl, S.; Niederhagen, R.; Rijneveld, J.; Schwabe, P. The SPHINCS+ Signature Framework. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS '19*, New York, NY, USA, 11–15 November 2019; pp. 2129–2146. <https://doi.org/10.1145/3319535.3363229>.
20. National Institute of Standards and Technology (US). *Stateless Hash-Based Digital Signature Standard*; Technical Report NIST FIPS 205; National Institute of Standards and Technology: Washington, DC, USA, 2024. <https://doi.org/10.6028/NIST.FIPS.205>.
21. Ding, J.; Schmidt, D. Rainbow, a New Multivariable Polynomial Signature Scheme. In *Proceedings of the Applied Cryptography and Network Security (ACNS 2005)*, New York, NY, USA, 7–10 June 2005. https://doi.org/10.1007/11496137_12.
22. Beullens, W.; Campos, F.; Celi, S.; Hess, B.; Kannwischer, M.J.; PQMayo. *MAYO: A Multivariate Quadratic Signature Scheme*. Online Resource, 2023. Available online: <https://pqmayo.org/> (accessed on 8 February 2026).
23. Furue, H.; Ikematsu, Y. A New Security Analysis Against MAYO and QR-UOV Using Rectangular MinRank Attack. In *Proceedings of the Advances in Information and Computer Security*; Shikata, J., Kuzuno, H., Eds.; Springer Nature: Cham, Switzerland, 2023; pp. 101–116.
24. Stebila, D.; Mosca, M. Post-quantum key exchange for the Internet and the Open Quantum Safe project. In *Proceedings of the Selected Areas in Cryptography (SAC) 2016*; Lecture Notes in Computer Science; Avanzi, R., Heys, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10532, pp. 1–24. https://doi.org/10.1007/978-3-319-69453-5_2.
25. Chen, L.; Jordan, S.; Liu, Y.K.; Moody, D.; Peralta, R.; Perlner, R.; Smith-Tone, D. *Report on Post-Quantum Cryptography*; NIST Interagency/Internal Report (NISTIR) 8105; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2016. <https://doi.org/10.6028/NIST.IR.8105>.
26. Pradel, G.; Mitchell, C.J. Post-quantum Certificates for Electronic Travel Documents. In *Computer Security*; Lecture Notes in Computer Science; Boureanu, I.; Drăgan, C.C.; Manulis, M.; Giannetsos, T.; Dadoyan, C., Gouvas, P., Hallman, R.A., Li, S., Chang, V., Pallas, F., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 12580, pp. 56–73. https://doi.org/10.1007/978-3-030-66504-3_4.
27. De Abiega-L'Eglise, A.F.; Delgado-Vargas, K.A.; Valencia-Rodriguez, F.Q.; Gonzalez-Quiroga, V.G.; Gallegos-Garcia, G.; Nakano-Miyatake, M. Performance of New Hope and CRYSTALS-Dilithium Postquantum Schemes in the Transport Layer Security Protocol. *IEEE Access* **2020**, *8*, 213968–213980. <https://doi.org/10.1109/ACCESS.2020.3040324>.
28. Bürstinghaus-Steinbach, K.; Krauß, C.; Niederhagen, R.; Schneider, M. Post-Quantum TLS on Embedded Systems: Integrating and Evaluating Kyber and SPHINCS+ with mbed TLS. In *Proceedings of the 15th ACM Asia Conference on Computer and*

- Communications Security, ASIA CCS '20, Taipei, Taiwan, 5–9 October 2020; ACM: New York, NY, USA, 2020; pp. 841–852. <https://doi.org/10.1145/3320269.3384725>.
29. Ducas, L.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. *CRYSTALS–Dilithium: Digital Signatures from Module Lattices*; Cryptology ePrint Archive 2017/633; IACR: Santa Barbara, CA, USA, 2017.
 30. Bernstein, D.J.; Hopwood, D.; Hülsing, A.; Lange, T.; Niederhagen, R.; Papachristodoulou, L.; Schneider, M.; Schwabe, P.; Wilcox-O’Hearn, Z. SPHINCS: Practical Stateless Hash-Based Signatures. In *Proceedings of the Advances in Cryptology–EUROCRYPT 2015*; Lecture Notes in Computer Science; Oswald, E., Fischlin, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9056, pp. 368–397. https://doi.org/10.1007/978-3-662-46800-5_15.
 31. Alkim, E.; Barreto, P.S.L.M.; Bindel, N.; Krämer, J.; Longa, P.; Ricardini, J.E. The Lattice-Based Digital Signature Scheme qTESLA. In *Proceedings of the Applied Cryptography and Network Security (ACNS 2020)*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12146, pp. 441–460. https://doi.org/10.1007/978-3-030-57808-4_22.
 32. Chen, M.S.; Hülsing, A.; Rijneveld, J.; Samardjiska, S.; Schwabe, P. From 5-pass MQ-based Identification to MQ-based Signatures. In *Proceedings of the Advances in Cryptology–ASIACRYPT 2016*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2016; Volume 10032, pp. 135–165.
 33. Sikeridis, D.; Kampanakis, P.; Devetsikiotis, M. Assessing the overhead of post-quantum cryptography in TLS 1.3 and SSH. In *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT '20*, Barcelona, Spain, 1–4 December 2020; pp. 149–156. <https://doi.org/10.1145/3386367.3431305>.
 34. Tasopoulos, G.; Li, J.; Fournaris, A.P.; Zhao, R.K.; Sakzad, A.; Steinfeld, R. Performance Evaluation of Post-Quantum TLS 1.3 on Resource-Constrained Embedded Systems. In *Information Security Practice and Experience*; Lecture Notes in Computer Science; Su, C., Gritzalis, D., Piuri, V., Eds.; Springer International Publishing: Cham, Switzerland, 2022; Volume 13620, pp. 432–451. https://doi.org/10.1007/978-3-031-21280-2_24.
 35. Döring, R.; Geitz, M. Post-Quantum Cryptography in Use: Empirical Analysis of the TLS Handshake Performance. In *Proceedings of the NOMS 2022–2022 IEEE/IFIP Network Operations and Management Symposium*, Budapest, Hungary, 25–29 April 2022; pp. 1–5. <https://doi.org/10.1109/NOMS54207.2022.9789913>.
 36. Rios, R.; Montenegro, J.A.; Muñoz, A.; Ferraris, D. Towards the Quantum-Safe Web: Benchmarking Post-Quantum TLS. *IEEE Netw.* **2025**, *39*, 247–253. <https://doi.org/10.1109/MNET.2025.3531116>.
 37. Garcia, C.R.; Aguilera, A.C.; Olmos, J.J.V.; Monroy, I.T.; Rommel, S. Quantum-Resistant TLS 1.3: A Hybrid Solution Combining Classical, Quantum and Post-Quantum Cryptography. In *Proceedings of the 2023 IEEE 28th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Edinburgh, UK, 6–8 November 2023; pp. 246–251. <https://doi.org/10.1109/CAMAD59638.2023.10478407>.
 38. Raavi, M.; Khan, Q.; Wuthier, S.; Chandramouli, P.; Balytskyi, Y.; Chang, S.Y. Security and Performance Analyses of Post-Quantum Digital Signature Algorithms and Their TLS and PKI Integrations. *Cryptography* **2025**, *9*, 38. <https://doi.org/10.3390/cryptography9020038>.
 39. Zafar, A.; Iqbal, S.S. Integrating code-based post-quantum cryptography into SSL/TLS protocols through an interoperable hybrid framework. *Discov. Comput.* **2025**, *28*, 202. <https://doi.org/10.1007/s10791-025-09735-7>.
 40. Sim, M.; Song, G.; Eum, S.; Lee, M.; Yoon, S.; Baksi, A.; Seo, H. Integrating and Benchmarking KpqC in TLS/X.509. *Electronics* **2025**, *14*, 3717. <https://doi.org/10.3390/electronics14183717>.
 41. Buruaga, J.S.; Méndez, R.B.; Brito, J.P.; Martin, V. Hybrid Quantum-Safe integration of TLS in SDN networks. *Comput. Netw.* **2025**, *267*, 111355. <https://doi.org/10.1016/j.comnet.2025.111355>.
 42. Wang, C.; Xue, W.; Wang, J. Integration of Quantum-Safe Algorithms into X.509v3 Certificates. In *Proceedings of the 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*, Changchun, China, 26–28 May 2023; pp. 384–388. <https://doi.org/10.1109/ICETCI57876.2023.10176713>.
 43. Ricchizzi, N.; Schwinne, C.; Pelzl, J. Applied Post Quantum Cryptography: A Practical Approach for Generating Certificates in Industrial Environments. *arXiv* **2025**. <https://doi.org/10.48550/arXiv.2505.04333>.
 44. *ISO/IEC 9594-8:2020*; Information Technology—Open Systems Interconnection—Part 8: The Directory: Public-Key and Attribute Certificate Framework. International Organization for Standardization: Geneva, Switzerland, 2020.
 45. 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, ASIA CCS '22*, Nagasaki, Japan, 30 May–3 June 2022; pp. 727–740. <https://doi.org/10.1145/3488932.3497755>.
 46. Ounsworth, M.; Pala, M. *Composite Signatures for Use in Internet PKI*. Internet-Draft draft-ounsworth-pq-composite-sigs-07, Internet Engineering Task Force (IETF), 2022. Available online: <https://datatracker.ietf.org/doc/html/draft-ounsworth-pq-composite-sigs-07> (accessed on 8 February 2026).
 47. Ounsworth, M.; Pala, M.; Klaussner, J. *Composite Public and Private Keys for Use In Internet PKI*. Internet-Draft draft-ounsworth-pq-composite-keys-02, Internet Engineering Task Force, 2022. Work in Progress. Available online: <https://datatracker.ietf.org/doc/html/draft-ounsworth-pq-composite-keys-02> (accessed on 8 February 2026).

48. Bonnell, C.; Gray, J.; Hook, D.; Okubo, T.; Ounsworth, M. A Mechanism for Encoding Differences in Paired Certificates. Internet-Draft draft-bonell-lamps-chameleon-certs-05, Internet Engineering Task Force, 2024. Work in Progress. Available online: <https://datatracker.ietf.org/doc/draft-bonell-lamps-chameleon-certs/05/> (accessed on 8 February 2026).
49. Raavi, M.; Chandramouli, P.; Wuthier, S.; Zhou, X.; Chang, S.Y. Performance Characterization of Post-Quantum Digital Certificates. In Proceedings of the 2021 International Conference on Computer Communications and Networks (ICCCN), Athens, Greece, 19–22 July 2021; pp. 1–9. <https://doi.org/10.1109/ICCCN52240.2021.9522179>.
50. Montenegro, J.A.; Rios, R.; Lopez-Cerezo, J. A performance evaluation framework for post-quantum TLS. *Future Gener. Comput. Syst.* **2026**, *175*, 108062. <https://doi.org/https://doi.org/10.1016/j.future.2025.108062>.
51. Kuang, R.; Perepechaenko, M.; Toth, R.; Barbeau, M. Benchmark Performance of a New Quantum-Safe Multivariate Polynomial Digital Signature Algorithm. In Proceedings of the 2022 IEEE International Conference on Quantum Computing and Engineering (QCE), Broomfield, CO, USA, 18–23 September 2022; pp. 454–464. <https://doi.org/10.1109/QCE53715.2022.00067>.
52. Opilka, F.; Niemiec, M.; Gagliardi, M.; Kourtis, M.A. Performance Analysis of Post-Quantum Cryptography Algorithms for Digital Signature. *Appl. Sci.* **2024**, *14*, 4994. <https://doi.org/10.3390/app14124994>.
53. Mansoor, K.; Afzal, M.; Iqbal, W.; Abbas, Y.; Mussiraliyeva, S.; Chehri, A. PQCAIE: Post quantum cryptographic authentication scheme for IoT-based e-health systems. *Internet Things* **2024**, *27*, 101228. <https://doi.org/https://doi.org/10.1016/j.iot.2024.101228>.
54. Giron, A.A.; Schardong, F.; Perin, L.P.; Custódio, R.; Valle, V.; Mateu, V. Automated Issuance of Post-Quantum Certificates: A New Challenge. In *Applied Cryptography and Network Security*; Lecture Notes in Computer Science; Pöpper, C., Batina, L., Eds.; Springer Nature: Cham, Switzerland, 2024; Volume 14584, pp. 3–23. https://doi.org/10.1007/978-3-031-54773-7_1.
55. Tsili, A.; Kordolaimis, K.; Krilakis, K.; Syvridis, D. A Scalable Framework for Post-Quantum Authentication in Public Key Infrastructures. In Proceedings of the 2025 International Conference on Quantum Communications, Networking, and Computing (QCNC), Tokyo, Japan, 1–3 September 2025; pp. 279–286. <https://doi.org/10.1109/QCNC64685.2025.00052>.
56. Seo, Y.; Kim, Y.S. Post-Quantum Cryptography Migration on V2X Certificate using KpqC Algorithms*. In Proceedings of the 2025 Sixteenth International Conference on Ubiquitous and Future Networks (ICUFN), Lisbon, Portugal, 8–11 July 2025; pp. 12–14. <https://doi.org/10.1109/ICUFN65838.2025.11170005>.
57. Fitzgibbon, G.; Ottaviani, C. Constrained Device Performance Benchmarking with the Implementation of Post-Quantum Cryptography. *Cryptography* **2024**, *8*, 21. <https://doi.org/10.3390/cryptography8020021>.
58. Al-Dabbagh, R.; Alkhatib, M.; Albalawi, T. Efficient Post-Quantum Cryptography Algorithms for Auto-Enrollment in Public Key Infrastructure. *Electronics* **2025**, *14*, 1980. <https://doi.org/10.3390/electronics14101980>.
59. Spalas, K.; Kolokotronis, N. Post-Quantum Security Evaluation of Aeronautical Communications. In Proceedings of the 2025 IEEE International Conference on Cyber Security and Resilience (CSR), Chania, Crete, Greece, 14–16 August 2025; pp. 295–301. <https://doi.org/10.1109/CSR64739.2025.11130013>.
60. Abbasi, M.; Cardoso, F.; Váz, P.; Silva, J.; Martins, P. A Practical Performance Benchmark of Post-Quantum Cryptography Across Heterogeneous Computing Environments. *Cryptography* **2025**, *9*, 32. <https://doi.org/10.3390/cryptography9020032>.
61. Mattson, P.; Tang, H.; Wei, G.Y.; et al.. MLPerf: An industry standard benchmark suite for machine learning performance. *IEEE Micro* **2020**, *40*, 8–16. <https://doi.org/10.1109/MM.2020.2974843>.
62. Khan, Q.; Purification, S.; Chang, S.Y. Post-Quantum Key Exchange and Subscriber Identity Encryption in 5G Using ML-KEM (Kyber). *Information* **2025**, *16*, 617. <https://doi.org/10.3390/info16070617>.
63. Khan, Q.; Purification, S.; Cheruiyot, R.; Kim, J.; Kim, J.; Chang, S.Y. Toward Post-Quantum Digital Certificate for eSIM. In Proceedings of the 2024 Silicon Valley Cybersecurity Conference (SVCC), Seoul, Republic of Korea, 18–20 June 2024; pp. 1–3. <https://doi.org/10.1109/SVCC61185.2024.10637354>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.