cryptography

IMPACT FACTOR 2.1

CITESCORE 5.0

Article

Post-Quantum Key Exchange in TLS 1.3: Further Analysis on Performance of New Cryptographic Standards

Konstantina Souvatzidaki and Konstantinos Limniotis

MDPI

https://doi.org/10.3390/cryptography9040073

cryptography

MDPI

Article

Post-Quantum Key Exchange in TLS 1.3: Further Analysis on Performance of New Cryptographic Standards

Konstantina Souvatzidaki [1] and Konstantinos Limniotis [1,2,*]

[1] Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, 15784 Athens, Greece; k.souvatzidaki@di.uoa.gr
[2] Hellenic Data Protection Authority, Kifissias 1-3, 11523 Athens, Greece
* Correspondence: klimn@di.uoa.gr

Abstract

The emergence of quantum computing presents a significant threat to classical cryptographic primitives, particularly those employed in securing internet communications via widely used protocols such as Transport Layer Security (TLS). As conventional key exchange mechanisms will become increasingly vulnerable in the post-quantum era, the integration of post-quantum cryptographic (PQC) algorithms into existing security protocols is of utmost importance. This study investigates the impact of incorporating PQC key encapsulation mechanisms—specifically, the recent standards CRYSTALS-Kyber and HQC, in conjunction with the candidate standard BIKE—into the TLS 1.3 handshake. A comprehensive experimental evaluation was conducted to measure handshake latency under emulated network conditions with varying packet loss probabilities. The findings offer useful insights into the performance trade-offs introduced by PQC integration and further highlight the necessity of a timely transition to post-quantum cryptographic standards.

Keywords: performance; post-quantum cryptography; Transport Layer Security

Academic Editor: Josef Pieprzyk

Received: 6 October 2025
Revised: 17 November 2025
Accepted: 19 November 2025
Published: 21 November 2025

Citation: Souvatzidaki, K.; Limniotis, K. Post-Quantum Key Exchange in TLS 1.3: Further Analysis on Performance of New Cryptographic Standards. *Cryptography* **2025**, *9*, 73. https://doi.org/10.3390/cryptography9040073

1. Introduction

Cryptography constitutes the main building block for ensuring confidentiality and authenticity of data, being present in a wide set of security protocols and applications such as in e-commerce, online banking, mobile phones, as well as in messaging apps, video conferencing, cloud storage, private browser analytics, etc. [1]. However, it is well-known that advances in quantum computing pose a serious risk to existing cryptographic algorithms and systems. Therefore, even today's secure communications may not be secure in the advent of powerful (i.e., cryptographically relevant) quantum computers and, thus, making long-term confidentiality especially vulnerable to the so-called "store now, decrypt later" attacks; the integrity and authenticity of data are also at stake (see, for example, [2]).

To mitigate these risks, it is essential to initiate a well-planned, timely, and coordinated migration toward post-quantum cryptography (PQC), i.e., cryptography that will remain secure even in the post-quantum era, that is, when cryptographically relevant computers are in place. To this end, the European Commission published on 11 April 2024 a "Recommendation on a Coordinated Implementation Roadmap for the transition to Post-Quantum Cryptography" [3], and in 2024 it has been announced that by 2035, the transition to post-quantum secure cryptosystems should be completed for as many systems as practically feasible. In the United States, the White House has already issued, since 4 May 2022, a policy directive through the "National Security Memorandum on Promoting United States

Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems" or NSM-10, which directed the U.S. government, among others, to prioritize timely and equitable transition to quantum-resistant cryptography across federal systems by 2035 [4].

Since quantum computers are expected to primarily threaten today's public-key cryptography—by efficiently solving the underlying hard problems such as integer factorization and discrete logarithms on which current systems rely. Post-quantum cryptography actually refers to asymmetric cryptographic algorithms whose security relies on hard problems that, up to our current knowledge, remain hard even for quantum computers. Therefore, the NIST (National Institute of Standards and Technology) launched a competition in 2017 to select new post-quantum secure algorithms as relevant cryptographic standards, applying an extensive evaluation process [5]. The NIST standardization process has, thus far, resulted in the selection of four post-quantum algorithms in 2022, following the conclusion of the third round of evaluations (namely, CRYSTALS-Kyber, CRYSTALS-Dilithium, Falcon, and SPHINCS+). In March 2025, after the completion of the fourth round, an additional algorithm, i.e., the HQC, was also standardized. More precisely, CRYSTALS-Kyber and HQC serve as cryptographic standards for secure key establishment, while CRYSTALS-Dilithium, Falcon, and SPHINCS+ constitute the post-quantum digital signature standards.

Due to their importance, post-quantum algorithms are widely studied not only in terms of security but also in relation to their performance (see, for example, [6]). This paper aims to further contribute to this field by evaluating the performance of the widely used Transport Layer Security (TLS) protocol version 1.3 when its underlying public-key algorithms (which are not quantum-resistant) are being substituted by post-quantum secure algorithms—i.e., by purely post-quantum algorithms or its hybrid variants. In particular, we examine all the standardized post-quantum schemes for key exchange, including the recently adopted HQC standard, as well as the BIKE algorithm, which was still under consideration in the fourth round of the NIST standardization process during the course of this study. To this end, we performed an extensive experimental analysis across a large number of different algorithmic parameters and network conditions. Hence, the importance of this work lies in the presentation of TLS performance results that include the most recent cryptographic standard HQC within the TLS handshake, as well as BIKE, which, although not standardized, exhibits similar desirable properties to HQC.

1.1. Research Questions

To evaluate the performance of TLS 1.3 when post-quantum secure algorithms are being used, we identified the following research questions to be addressed:

- Q1 How efficient are the latest post-quantum key establishment algorithms, for various security levels, compared to classical key exchange ciphers, when deployed to TLS 1.3?
- Q2 Among the cryptographic standards for post-quantum secure key establishment algorithms, which is the most efficient for TLS 1.3?
- Q3 How efficient is the utilization of hybrid algorithms, i.e., combining post-quantum ciphers with classical key exchange components?
- Q4 How do the network conditions affect the overall performance of post-quantum secure key establishment in TLS 1.3?

It should be pointed out that since post-quantum cryptographic standards actually refer to asymmetric cryptographic operations within the TLS, the TLS Record Protocol—which is contingent entirely on symmetric primitives—remains unchanged (i.e., the AES-256 is known to be post-quantum secure), and thus, the performance evaluation in this work is restricted to the handshake phase. Likewise, the verification of the server's certificate

signature is excluded from our analysis, as all experiments employ the same post-quantum secure digital signature scheme.

1.2. Structure of the Paper

The paper is organized as follows: Section 2 presents the main background, i.e., the notion of post-quantum cryptography, the relevant challenges in the TLS 1.3 protocol, as well as the current status of the standardization procedure by the NIST. Section 3 provides a short description of relevant previous works. The main part of this work is presented in Section 4, which describes the adopted methodology, as well as all the results of our extensive set of experiments. An overall discussion based on the outcomes of the experiments is provided in Section 5. Finally, concluding remarks are given in Section 6.

2. Background

In this Section, we provide the necessary background information.

2.1. Post-Quantum Cryptography

Quantum computers are considered to be capable of performing computations at an exponential scale by harnessing the principles of quantum mechanics. The first quantum computers are currently under development. Although there are several different opinions on when a cryptographically relevant quantum computer will be a reality, there are several arguments stating that we will see a fully fault-tolerant quantum computer by 2035–2040 (see, for example, [7]).

The computational power of quantum devices is expected to pose significant challenges to the security of modern cryptosystems. Quantum computers leverage phenomena such as superposition and entanglement, enabling them to solve certain mathematical problems that are currently considered to be hard, such as integer factorization and discrete logarithms. Hence, widely used cryptosystems such as RSA, elliptic curve cryptography (ECC), and Diffie–Hellman will be extremely vulnerable, mainly due to the existence of the well-known quantum Shor's algorithm [8], which can efficiently break these schemes. In fact, the primary impact of quantum computing on cryptography is related to asymmetric (i.e., public key) cryptography since asymmetric cryptography relies on the difficulty of mathematical problems, which will not remain difficult in the post-quantum era. On the other hand, symmetric cryptography is less severely affected. More precisely, the so-called Grover's quantum algorithm [9] provides only a quadratic speedup in brute-force key searches, which means that doubling the size of keys in symmetric cryptographic primitives generally suffices to maintain security in the post-quantum era.

Post-quantum cryptography (PQC) refers to cryptographic algorithms designed to remain secure against adversaries equipped with quantum computers. Therefore, PQC focuses on developing new public-key cryptographic schemes whose security rests in mathematical problems that remain hard even under the assumption that large-scale quantum computers are in place. Such mathematical problems include, e.g., lattice-based, code-based, hash-based, and multivariate polynomial problems.

TLS in the Post-Quantum Era

Transport Layer Security (TLS) 1.3 is the latest standardized version of the TLS protocol, designed to provide confidentiality as well as message and entity authentication for communications over insecure networks. Published as RFC (RFC 8446) in 2018 by the Internet Engineering Task Force (IETF) [10], TLS 1.3 represents a major revision of its predecessors, both in terms of security and efficiency.

TLS 1.3 consists of two main phases: the handshake protocol and the record protocol. The handshake protocol is responsible for the negotiation of cryptographic parameters be-

tween the client and the server, while simultaneously establishing mutual trust. In TLS 1.3, the handshake has been significantly streamlined compared to earlier versions, thereby reducing latency and eliminating obsolete or insecure mechanisms such as static RSA key exchange which does not provide forward secrecy. Authentication of the server is mandatory and is performed through the exchange and validation of digital certificates, whereas client authentication remains optional. Ephemeral Diffie–Hellman key exchange (using elliptic curve or finite-field variants) is employed to establish session keys with forward secrecy. Therefore, public-key cryptography is present both in the authentication process and key exchange, whereas symmetric cryptography is being used to secure the communication itself.

After the completion of the handshake, the so-called record protocol is initiated for the exchange of application data (i.e., the content of the communication). In this phase, data streams are protected using symmetric cryptographic algorithms agreed upon during the handshake. Unlike earlier versions, TLS 1.3 requires the use of authenticated encryption with associated data (AEAD) schemes, which combine encryption and integrity protection in a single operation, thereby removing the need for separate message authentication codes.

Based on the above, it is evident that TLS 1.3 would be highly vulnerable in the post-quantum era unless its asymmetric cryptographic algorithms are upgraded to post-quantum secure counterparts (the length of the key for the symmetric encryption should not be less than 256 bits). To this end, there are two main approaches (see, for example, [11]):

- Purely post-quantum implementations, in which classical asymmetric cryptographic algorithms in TLS are being replaced by post-quantum ones.
- Hybrid post-quantum implementations, in which a combination of classical and a post-quantum implementations is being used.

2.2. Post-Quantum Cryptographic Standards

Due to importance of having post-quantum secure cryptographic standards, the National Institute of Standards and Technology (NIST) launched an evaluation procedure in 2017 aiming to receive proposals for post-quantum secure cryptographic algorithms, assess them in terms of their security and performance characteristics, and eventually decide which will be selected for standardization. In order to ensure that post-quantum cryptographic algorithms align with the security guarantees of well-established symmetric primitives, NIST classifies the post-quantum algorithms into different algorithm security levels.

Specifically, NIST defines five security categories, ranked in order of increasing strength [12]:

- Level 1: Equivalent to the security of a 128-bit block cipher against brute-force attacks (e.g., AES-128).
- Level 2: Comparable to the difficulty of finding collisions in a 256-bit hash function (e.g., SHA-256/SHA3-256).
- Level 3: Equivalent to the security of a 192-bit block cipher against key search attacks (e.g., AES-192).
- Level 4: Comparable to the difficulty of finding collisions in a 384-bit hash function (e.g., SHA-384/SHA3-384).
- Level 5: Equivalent to the security of a 256-bit block cipher against key search attacks (e.g., AES-256).

In 2022, after three evaluation rounds, NIST announced its selection of algorithms to be advanced as post-quantum cryptographic standards: (a) The CRYSTALS-Kyber for key encapsulation mechanisms, and the CRYSTALS-Dilithium for digital signatures, (b) The signature schemes FALCON and SPHINCS+ for digital signatures. CRYSTALS-

Kyber, CRYSTALS-Dilithium, and FALCON are all representatives of lattice-based cryptography, whereas SPHINCS+ is founded on hash-based techniques. In August 2024, NIST published the standard FIPS203 for ML-KEM, the standardized version of CRYSTALS-Kyber [13].

In July 2022, NIST initiated the fourth round of the competition to identify alternative algorithms to CRYSTALS-Kyber that rely on the hardness on a different mathematical problem (i.e., not a lattice-based algorithm). Four candidates for key encapsulation mechanisms were advanced for further evaluation: (a) BIKE, Classic McEliece, and HQC, which are all based on code-based cryptography, and (b) SIKE, which is derived from isogeny-based cryptography. During this stage, the designers of SIKE disclosed vulnerabilities and, thus, recommended discontinuing its use. At the same time, in parallel with key encapsulation candidates, a second round of submissions for digital signature schemes was ongoing [14].

Very recently, while this work was in progress, NIST released its latest status report on the fourth round of the post-quantum cryptography standardization process [15], which provides a comprehensive summary of all candidate algorithms and the outcomes of the round. The results can be summarized as follows:

- BIKE was recognized for its balanced performance and its foundation in code-based cryptography. NIST determined that BIKE would serve as a suitable complement to the lattice-based ML-KEM, derived from CRYSTALS-Kyber, offering smaller key sizes and ciphertexts compared to HQC.
- Classic McEliece, while acknowledged for its strong security properties, suffers from very large public key sizes, limiting its potential use for most applications. It may still be prefered in cases where ciphertext size is critical and public keys are transmitted infrequently. Consequently, Classic McEliece was not selected for standardization at this stage.
- HQC, which shares the same underlying mathematical principles as BIKE, has also been identified as a strong counterpart to ML-KEM (i.e., the CRYSTALS-Kyber). Despite having larger public key sizes, HQC benefits from a more stable and thoroughly analyzed security profile, leading to its selection for standardization.

Therefore, on the basis of the above, NIST announced that HQC will be the new cryptographic standard which will serve as a backup for ML-KEM. Therefore, with the announcement of HQC's standardization, the fourth round of NIST's standardization process is now concluded.

3. Relevant Previous Works

Due to its high importance, there has been considerable research interest in recent years with respect to performance issues of post-quantum TLS implementations. In this section, we briefly present the main outcomes from relevant works.

In [16], wide-scale experiments were co-launched by Cloudflare and Google, which aimed to evaluate the efficiency of post-quantum algorithms when used by real clients over real networks. For the purposes of the experiment, Cloudflare's TLS stack was modified to integrate NIST candidate algorithms. The results showed that combining post-quantum ciphers with classical ECDH did not affect the performance of TLS handshakes significantly. This experiment highlighted the significance of post-quantum cryptography, and many works focusing on the evaluation of quantum-safe algorithms have been published thereafter.

In [17], Paquin et al. developed a benchmarking framework for their experiment with quantum-safe key exchange and signature algorithms on TLS 1.3 using Open Quantum Safe implementations. They integrated post-quantum cryptography into both emulated and internet-based network environments. Their results showed that data transmission

times were affected more significantly than the duration of TLS handshakes in unreliable quantum-enabled networks.

In [18], Tzinos et al. used the Open Quantum Safe framework to examine the performance of all third-round key encapsulation mechanisms (KEMs) finalists, by comparing them with elliptic curve Diffie–Hellman (ECDH) ciphers. The results highlighted the effect of the key and ciphertext sizes on the overall algorithm performance. This work also concluded that CRYSTALS-Kyber is a promising option in the post-quantum era given its good performance properties, in line with the subsequent outcomes of the NIST evaluation process.

In [19], Steinbach et al. integrated selected post-quantum algorithms, more specifically CRYSTALS-Kyber for key establishment and SPHINCS+ as a signature scheme, into the mbedTLS library for embedded systems. The results showed that CRYSTALS-Kyber performed better than the classical ECDH on all platforms, while SPHINCS+ underperformed compared to an ESDSA scheme. The work also highlighted that computing SPHINCS+ signatures posed a significant overhead during TLS handshakes. A relatively low overhead was observed in terms of code and RAM size.

In [20], Kampanakis et al. adopted an OQS-enabled setup to focus on the impact of the time-to-last-byte (TTLB) in TLS 1.3 connections. They compared the performance of the standardized versions of KRYSTAL-Kyber and CRYSTALS-Dilithium against classical ECDH. Their findings showed that the impact of using the post-quantum ciphers is less significant on the TTLB, especially in high-density data transmissions, compared to the impact on the handshake process.

In [21], the authors illustrate how TLS 1.3 can be transformed into quantum-safe by modifying the TLS 1.3 architecture in constrained environments to accommodate the PQC algorithms, also evaluating the execution time, memory, and bandwidth requirements of the proposed post-quantum variant of TLS 1.3. However, given the time frame of its completion (2022), this study could not have incorporated the most recent NIST PQC standards.

Very recently, while the present work was being prepared, Montenegro et al. in [22] introduced a framework for evaluating post-quantum algorithms within the TLS protocol, with the goal of supporting the transition to quantum-safe TLS. Their approach leveraged the latest OpenSSL library from the Open Quantum Safe project to integrate several digital signature schemes from the NIST competition, namely, the standardized version ML-DSA of CRYSTALS-Diithium, FALCON, SPHINCS+, MAYO, and CROSS, while employing the standardized version ML-KEM of KRYSTALS-Kyber for key encapsulation. Their results showed that post-quantum algorithms introduced non-trivial overheads in the TLS handshake and highlighted that hybrid schemes may be a good compromise during the transition period.

It should also be pointed out that an extensive recent survey on post-quantum TLS is given in [11], in which results from a wide set of experiments is also provided. According to the analysis and the results therein, there are existing reasonably secure and efficient purely post-quantum solutions for authentication and key exchange as well as efficient hybrid post-quantum solutions. However, as similarly concluded in [11], the performances of post-quantum secure implementations seem to be restricted in low-bandwidth environments with significant packet loss. Finally, an important observation in [11] is that existing implementations are identified as being of high-quality, with explicit reference to the Open Quantum Safe (OQS) project as *the most comprehensive, offering both post-quantum (hybrid) authentication and key exchange, as well as easy-to-use benchmarking facilities* [11]; the experimental environment for this study is similarly designed, as will be discussed next.

Finally, during the review process of this paper, a new paper [23] presented performance results on post-quantum secure TLS connection for various network parameters,

illustrating that embedding post-quantum secure algorithms does not negatively affect the overall performance as substantially as poor network conditions; the effect of such poor network conditions is more evident when PQC algorithms at the highest security levels are being used. The work in [23] mainly emphasized on various PQC digital signature schemes.

Contribution of This Work

In this work, we use the well-known open-source implementation of Open Quantum Safe in a similar setup to previous works, aiming to evaluate the performance of post-quantum key encapsulation algorithms within emulated networks, using the newest OpenSSL Open Quantum Safe provider [24]. Our objective is to conduct a large set of experiments for various algorithm and network parameters, using the latest post-quantum KEMs from the NIST competition, i.e., HQC and BIKE. Their performance is evaluated based on total TLS handshake duration and is compared against CRYSTALS-Kyber, the already standardized post-quantum algorithm which is known to achieve high performance. Both post-quantum and hybrid versions of the key encapsulation algorithms are being considered in our analysis for various algorithms security levels and for several different packet loss ratios. This extensive set of experiments is feasible due to the large number of algorithms supported by the Open Quantum Safe project, which also provides different hybrid algorithms, i.e., algorithms that combine classic and quantum-safe methods in order to support the parallel use of classic and quantum-secure cryptography (see [25]). Moreover, regarding the digital signatures, the standardized algorithm CRYSTALS-Dilithium is used for all cases, ensuring full post-quantum security for the whole protocol.

Compared to previous works in the field, the main contribution of this work lies in providing, for the first time, a comprehensive and systematic evaluation of all standardized post-quantum key exchange schemes in the TLS handshake context (including the recent HQC), considering a wide range of algorithmic and network parameters, and these experiments were carried out using the updated version of the Open Quantum Safe project [24].

4. Measuring Performance of Post-Quantum Secure Key Exchange in TLS

This section constitutes the main part of this work, presenting the main results derived from an extensive set of experiments evaluating the performance of TLS when post-quantum secure key exchange algorithms are being used.

4.1. Experimental Environment

In order to evaluate the efficiency of post-quantum algorithms in the TLS protocol, we focused on measuring the duration of a TLS handshake explicitly, using post-quantum secure algorithms for both key exchange and digital signatures. To this end, we set up an appropriate environment for the experiments. The setup consisted of a virtual machine (VM) running on VMware Workstation, while the hypervisor was a Windows 10 personal computer equipped with a 3.9 GHz processor and 32 GB of RAM. The VM allocated 2 vCPUs and 4 GB of RAM and used Ubuntu 24.04.1 as an operating system.

After configuring the virtual device, we proceed as follows:

- To emulate a client–server network within a single virtual machine, Linux network namespaces were utilized. More precisely, two network namespaces were initialized, one for the client and one for the server party. For this purpose, we were based on a shell script implemented by C. Paquin et al. in [17].
- With the VM in place and the client–server network namespaces set up, the next step was to install all the required utilities to run the experiment. These include the latest version of OpenSSLv3 and the Open Quantum Safe components to enable PQC.

In a similar approach to the one adopted in [17,18,20], the latest Liboqs library was used to integrate post-quantum cryptography into OpenSSL. However, compared to other works which relied on an old (now-deprecated) OQS OpenSSL fork with Liboqs integrated to enable PQC on TLS, here the most recent (during the period that this research was conducted) OpenSSL OQS Provider (version 0.8) was used [25] to enable post-quantum cryptography support in OpenSSL 3.4. The chosen version of the OQS Provider fully supports the necessary post-quantum algorithms and their hybrid variants: CRYSTALS-Kyber, HQC, BIKE, and CRYSTALS-Dilithium.

- Self-signed certificates for both the server and the client were generated using CRYSTALS-Dilithium.
- Finally, on the server network namespace, the Nginx web server was configured with SSL, using the locally installed OQS-enabled OpenSSL.

4.2. Experimental Parameters

The goal of the experiment was to evaluate the latest post-quantum algorithms for key exchange and compare them with classical ECDH KEMs as well as the already standardized CRYSTALS-Kyber post-quantum KEM. As stated above (see Section 2), during the fourth round of the NIST post-quantum cryptography competition, four candidates were under consideration for public-key encryption and key exchange: BIKE, SIKE, Classic McEliece, and HQC. HQC has been recently selected for standardization, while SIKE was disqualified in 2024 after being shown to be insecure [15] and was, therefore, excluded from the experiments. Classic McEliece was also omitted due to its well-documented slow performance (see, for example, the relevant discussion in [18]). BIKE, however, remained a viable candidate at the time this work was conducted and was, thus, included in the experiments.

The experiments were designed based on the following:

- All combinations of the available variants of CRYSTALS-Kyber, HQC, and BIKE shall be tested for key exchange.
- Both post-quantum and hybrid key exchange implementations shall be included in the evaluation.
- For TLS authentication, self-signed certificates were generated for both client and server. These certificates were digitally signed using the Level 3 pure-quantum variant of the standardized CRYSTALS-Dilithium algorithm.
- Two parameters were defined for each experiment, the key encapsulation mechanism (KEM) and the packet loss probability, which were applied to both ends of the emulated client–server network. Packet loss values ranged from 0 to 12%.

For each experiment, 500 TLS handshakes were initiated. Several statistical metrics were computed to provide a comprehensive analysis of the total duration across all 500 handshakes, including the mean, standard deviation, minimum, maximum, and multiple percentiles (25th, median, 75th, and 95th).

All the post-quantum ciphers that were used for our experiments are shown in Table 1. It should be stressed that in this table, as well as in the presentation of all subsequent experiment results, we used the same names of the algorithms as those used in the openssl OQS library, that is, (a) kyber512, kyber768, and kyber1024 for the L1, L3, and L5 versions of the CRYSTALS-Kyber, respectively; (b) hqc128, hqc192, and hqc256 for the L1, L3, and L5 versions of the HQC, respectively; (c) bikel1, bikel3, and bikel5 for the L1, L3, and L5 versions of the BIKE, respectively. Similarly, in the same library, the L3 version of the CRYSTALS-Dilithium is denoted by dilithium3.

Table 1. A list of the post-quantum ciphers used in our experiments.

Algorithm	NIST Security Level	Hybrid Component	Post-Quantum Component
kyber512	L1 (~AES-128)	-	Kyber512
p256_kyber512	L1 (~AES-128)	ECDH	Kyber512
kyber768	L3 (~AES-192)	-	Kyber768
p384_kyber768	L3 (~AES-192)	ECDH	Kyber768
p256_kyber768	L3 (~AES-192)	ECDH	Kyber768
kyber1024	L5 (~AES-256)	-	Kyber1024
p521_kyber1024	L5 (~AES-256)	ECDH	Kyber1024
hqc128	L1 (~AES-128)	-	HQC128
p256_hqc128	L1 (~AES-128)	ECDH	HQC128
hqc192	L3 (~AES-192)	-	HQC192
p384_hqc192	L3 (~AES-192)	ECDH	HQC192
hqc256	L5 (~AES-256)	-	HQC256
p521_hqc256	L5 (~AES-256)	ECDH	HQC256
bikel1	L1 (~AES-128)	-	BIKE L1
p256_bikel1	L1 (~AES-128)	ECDH	BIKE L1
bikel3	L3 (~AES-192)	-	BIKE L3
p384_bikel3	L3 (~AES-192)	ECDH	BIKE L3
bikel5	L5 (~AES-256)	-	BIKE L5
p521_bikel5	L5 (~AES-256)	ECDH	BIKE L5
dilithium3	L3 (~AES-192)	-	Dilithium L3

Once the environment was fully set up with all the necessary components, a Python 3.12 script, was created to automate the execution of all experiments in a single run. This script allows the execution of the experiments for the combination of KEM algorithms and network packet loss variations under varying network reliability scenarios. More specifically, for every unique pair of a post-quantum KEM cipher and a packet loss probability value, the script conducts an experiment consisting of 500 individual TLS connection attempts; for each such connection, the client initiates a TLS handshake by generating a request for the post-quantum SSL-enabled Nginx server, running on the server network namespace and using certificates signed with CRYSTALS-Dilithium. This choice was made in order to work on a fully PQC environment.

Before initiating the TLS connections, the script dynamically applies the specified packet loss probability to the emulated network environment. This is accomplished by configuring the queuing discipline (qdisc) settings of both the client and server network namespaces using Linux utilities.

Once the network conditions are set, the experiment proceeds with the execution of 500 TLS handshakes. These are carried out using the OpenSSL `s_client` utility, which is invoked within the client network namespace to initiate connections to the server.

The script measures the duration of each TLS handshake by recording timestamps immediately before and after each handshake is completed, using the built-in time() function in Python's standard libraries. Each `s_client` connection is terminated upon handshake completion, and no additional payload is transmitted in order to isolate and evaluate the impact of the post-quantum KEM and certificate on the handshake protocol alone. As a result, the amount of transmitted data is limited to a few kilobytes, approximately 3–5 KB for a full handshake on the TLS 1.3 protocol, corresponding solely to the handshake messages. Hence, for our measurements, we do not consider certificate verification or TCP setup. The complete implementation of the TLS timer script is available in the Github repository [26].

4.3. Simulation Results

This section outlines the outcomes of the conducted experiments using a twofold approach: numerical results are first presented in tabular form, followed by graphical representations to support comparative analysis and facilitate the derivation of conclusions.

First, Tables 2 and 3 display the median and 95th percentile values, respectively, of the total handshake time using classical ECDH curves for key exchange, for various packet loss ratios.

Table 2. The 50th percentile of 500 handshakes for classical key exchange ciphers (in s).

Packet Loss	prime256v1	secp384r1	secp521r1
0	0.018	0.018	0.034
0.1	0.018	0.018	0.034
0.5	0.018	0.018	0.035
1	0.018	0.018	0.034
1.5	0.018	0.018	0.035
2	0.018	0.018	0.034
2.5	0.018	0.019	0.034
3	0.018	0.018	0.034
4	0.018	0.018	0.034
6	0.018	0.019	0.035
8	0.019	0.020	0.036
10	0.034	0.034	0.036
12	0.220	0.220	0.269

Table 3. The 95th percentile of 500 handshakes for classical key exchange ciphers (in s).

Packet Loss	prime256v1	secp384r1	secp521r1
0	0.020	0.034	0.036
0.1	0.019	0.034	0.036
0.5	0.020	0.035	0.038
1	0.219	0.036	0.270
1.5	0.220	0.271	0.271
2	1.030	1.079	0.271
2.5	1.030	0.274	1.034
3	1.030	1.028	1.029
4	1.035	1.080	1.030
6	1.081	1.083	1.082
8	1.283	1.280	1.284
10	1.285	1.436	1.537
12	2.567	1.490	2.093

Next, Tables 4 and 5 present the corresponding results for the purely post-quantum (PQ) implementations of Kyber, HQC, and BIKE, again for various packet loss ratios.

Finally, Tables 6 and 7 illustrate the results for the hybrid implementations of the three algorithms, which combine classical components with the post-quantum algorithms, and the same network parameters as previously have also been considered.

Table 4. The 50th percentile of 500 handshakes for post-quantum key exchange ciphers (in s).

Packet	kyber			hqc			bike		
Loss	512	768	1024	128	192	256	l1	l3	l5
0	0.017	0.017	0.017	0.034	0.067	0.118	0.017	0.017	0.034
0.1	0.017	0.017	0.017	0.034	0.067	0.118	0.017	0.017	0.034
0.5	0.017	0.017	0.017	0.034	0.067	0.118	0.017	0.017	0.034
1	0.017	0.017	0.017	0.034	0.068	0.118	0.017	0.017	0.034
1.5	0.017	0.017	0.017	0.034	0.068	0.118	0.017	0.017	0.034
2	0.017	0.017	0.017	0.034	0.068	0.118	0.017	0.017	0.034
2.5	0.018	0.017	0.017	0.034	0.068	0.118	0.017	0.017	0.034
3	0.017	0.017	0.017	0.034	0.067	0.118	0.017	0.018	0.034
4	0.018	0.017	0.017	0.034	0.068	0.118	0.018	0.017	0.034
6	0.018	0.018	0.018	0.034	0.068	0.118	0.018	0.018	0.034
8	0.018	0.018	0.018	0.035	0.068	0.119	0.018	0.019	0.034
10	0.019	0.018	0.019	0.035	0.068	0.120	0.018	0.034	0.035
12	0.019	0.034	0.034	0.036	0.269	0.318	0.019	0.221	0.268

Table 5. The 95th percentile of 500 handshakes for post-quantum key exchange ciphers (in s).

Packet	kyber			hqc			bike		
Loss	512	768	1024	128	192	256	l1	l3	l5
0	0.019	0.019	0.019	0.037	0.069	0.119	0.019	0.019	0.035
0.1	0.019	0.019	0.019	0.037	0.069	0.120	0.019	0.019	0.035
0.5	0.019	0.019	0.019	0.037	0.070	0.120	0.019	0.020	0.036
1	0.044	0.033	0.218	0.038	0.071	0.122	0.032	0.033	0.269
1.5	0.219	0.268	0.221	0.270	0.072	0.320	0.270	0.267	0.269
2	1.027	0.790	0.326	0.427	0.271	0.320	0.269	1.027	0.320
2.5	0.499	0.269	1.078	0.836	0.274	0.554	0.319	0.269	1.028
3	1.078	0.643	1.033	0.318	1.079	1.129	1.076	1.079	1.078
4	1.078	1.078	1.080	1.081	1.081	1.128	1.079	1.029	1.079
6	1.081	1.081	1.291	1.281	1.131	1.135	1.081	1.081	1.128
8	1.483	1.436	1.433	1.584	1.389	1.588	1.280	1.281	1.283
10	1.689	1.446	1.701	1.334	1.593	1.534	1.483	1.687	1.486
12	2.090	2.342	2.209	1.852	2.351	1.650	1.583	2.295	2.246

Table 6. The 50th percentile of 500 handshakes for hybrid key exchange ciphers (in s).

Packet Loss	p256_ kyber512	p384_ kyber768	p256_ kyber768	p521_ kyber1024	p25_ hqc128	p384_ hqc192	p521_ hqc256	p256_ bikel1	p384_ bikel3	p521_ bikel5
0	0.017	0.033	0.017	0.034	0.034	0.067	0.118	0.017	0.034	0.034
0.1	0.017	0.033	0.017	0.034	0.034	0.067	0.118	0.017	0.034	0.034
0.5	0.017	0.033	0.017	0.034	0.034	0.067	0.118	0.017	0.034	0.034
1	0.017	0.033	0.017	0.034	0.034	0.067	0.118	0.017	0.034	0.034
1.5	0.017	0.033	0.017	0.034	0.034	0.068	0.118	0.017	0.034	0.035
2	0.017	0.033	0.017	0.034	0.034	0.067	0.118	0.017	0.034	0.035
2.5	0.017	0.033	0.017	0.034	0.034	0.068	0.118	0.017	0.034	0.035
3	0.017	0.033	0.017	0.034	0.034	0.068	0.118	0.017	0.034	0.035
4	0.017	0.033	0.018	0.034	0.034	0.068	0.118	0.017	0.034	0.035
6	0.018	0.034	0.018	0.035	0.034	0.068	0.118	0.018	0.034	0.035
8	0.019	0.034	0.019	0.035	0.035	0.068	0.119	0.019	0.035	0.035
10	0.033	0.034	0.033	0.035	0.036	0.069	0.120	0.033	0.035	0.036
12	0.035	0.035	0.034	0.036	0.268	0.269	0.318	0.035	0.268	0.269

Table 7. The 95th percentile of 500 handshakes for hybrid key exchange ciphers (in s).

Packet Loss	p256_ kyber512	p384_ kyber768	p256_ kyber768	p521_ kyber1024	p25_ hqc128	p384_ hqc192	p521_ hqc256	p256_ bikel1	p384_ bikel3	p521_ bikel5
0	0.019	0.035	0.033	0.036	0.036	0.070	0.120	0.019	0.036	0.036
0.1	0.018	0.034	0.019	0.036	0.036	0.070	0.120	0.019	0.036	0.036
0.5	0.019	0.035	0.020	0.036	0.036	0.070	0.120	0.020	0.036	0.037
1	0.044	0.268	0.033	0.037	0.037	0.083	0.121	0.020	0.037	0.036
1.5	0.271	0.269	0.034	0.269	0.270	0.271	0.319	0.269	0.270	0.270
2	0.269	0.270	1.026	0.344	0.271	0.270	0.363	0.693	1.077	0.270
2.5	0.270	1.078	1.077	0.269	0.270	0.679	0.322	0.271	0.402	1.078
3	0.500	1.077	1.033	1.077	0.327	1.086	1.129	1.078	1.030	1.079
4	1.076	1.079	1.077	1.080	1.080	1.130	1.129	1.080	1.080	1.081
6	1.081	1.082	1.283	1.131	1.130	1.131	1.181	1.231	1.081	1.130
8	1.282	1.641	1.287	1.333	1.286	1.485	1.332	1.233	1.286	1.284
10	1.284	1.889	1.483	1.846	2.091	1.535	1.810	1.922	1.891	1.546
12	1.897	2.293	2.051	2.094	2.901	2.055	2.353	2.328	2.770	2.547

To present a more comprehensive view of the outcomes, we next present, for specific packet loss ratios, more detailed information, such as the minimum and maximum values—among the 500 handshakes—that have been measured for the handshake time, as well as the 25th percentile and the 75th percentile in conjunction with the median and the 95th percentile. Hence, such results are presented in Table 8 (for packet loss ratio = 0%), in Table 9 (for packet loss ratio = 1%), in Table 10 (for packet loss ratio = 3%), in Table 11 (for packet loss ratio = 6%), as well as in Table 12 (for packet loss ratio = 10%).

Table 8. Detailed information of measuring handshake times (in s) for packet loss = 0%.

cipher	Min	25th Percentile	Median	75th Percentile	Max	95th Percentile
prime256v1	0.010	0.017	0.018	0.019	0.021	0.020
secp384r1	0.016	0.018	0.018	0.025	0.11	0.034
secp521r1	0.016	0.018	0.034	0.035	0.037	0.036
kyber512	0.008	0.009	0.017	0.018	0.02	0.019
kyber768	0.008	0.01	0.017	0.019	0.019	0.019
kyber1024	0.008	0.009	0.017	0.019	0.02	0.019
hqc128	0.010	0.017	0.034	0.037	0.038	0.037
hqc192	0.016	0.018	0.067	0.069	0.070	0.069
hqc256	0.017	0.019	0.118	0.119	0.120	0.119
bikel1	0.009	0.01	0.017	0.019	0.019	0.019
bikel3	0.009	0.011	0.017	0.019	0.020	0.019
bikel5	0.011	0.018	0.034	0.035	0.036	0.035
p256_kyber512	0.01	0.017	0.017	0.019	0.021	0.019
p384_kyber768	0.011	0.017	0.033	0.035	0.036	0.035
p256_kyber768	0.010	0.017	0.017	0.033	0.035	0.033
p521_kyber1024	0.012	0.018	0.034	0.036	0.04	0.036
p256_hqc128	0.012	0.018	0.034	0.036	0.042	0.036
p384_hqc192	0.016	0.017	0.067	0.07	0.090	0.070
p521_hqc256	0.021	0.034	0.118	0.12	0.130	0.120
p256_bikel1	0.010	0.017	0.017	0.019	0.021	0.019
p384_bikel3	0.012	0.017	0.034	0.036	0.04	0.036
p521_bikel5	0.016	0.017	0.034	0.036	0.038	0.036

Table 9. Detailed information of measuring handshake times (in s) for packet loss = 1%.

cipher	Min	25th Percentile	Median	75th Percentile	Max	95th Percentile
prime256v1	0.009	0.017	0.018	0.018	1.084	0.219
secp384r1	0.016	0.018	0.018	0.019	1.087	0.036
secp521r1	0.016	0.018	0.034	0.04	1.082	0.270
kyber512	0.008	0.009	0.017	0.019	1.086	0.044
kyber768	0.008	0.009	0.017	0.017	1.283	0.033
kyber1024	0.008	0.010	0.017	0.017	1.314	0.218
hqc128	0.010	0.017	0.034	0.037	1.105	0.038
hqc192	0.016	0.017	0.068	0.071	1.285	0.071
hqc256	0.017	0.019	0.118	0.119	1.485	0.122
bikel1	0.008	0.010	0.017	0.017	1.147	0.032
bikel3	0.009	0.011	0.017	0.018	1.089	0.033
bikel5	0.010	0.018	0.034	0.045	1.085	0.269
p256_kyber512	0.010	0.017	0.017	0.018	1.105	0.044
p384_kyber768	0.010	0.018	0.033	0.018	1.087	0.268
p256_kyber768	0.010	0.017	0.017	0.018	1.089	0.033
p521_kyber1024	0.016	0.018	0.034	0.036	1.085	0.037
p256_hqc128	0.011	0.018	0.034	0.036	1.088	0.037
p384_hqc192	0.016	0.017	0.067	0.072	2.146	0.083
p521_hqc256	0.021	0.034	0.118	0.118	2.094	0.121
p256_bikel1	0.010	0.017	0.017	0.019	1.283	0.020
p384_bikel3	0.012	0.018	0.034	0.034	1.086	0.037
p521_bikel5	0.016	0.017	0.034	0.034	2.093	0.036

Table 10. Detailed information of measuring handshake times (in sec) for packet loss = 3%.

cipher	Min	25th Percentile	Median	75th Percentile	Max	95th Percentile
prime256v1	0.009	0.017	0.018	0.019	3.364	1.03
secp384r1	0.016	0.018	0.018	0.019	3.156	1.028
secp521r1	0.016	0.018	0.034	0.036	2.103	1.029
kyber512	0.008	0.009	0.017	0.017	1.289	1.078

Table 10. *Cont.*

cipher	Min	25th Percentile	Median	75th Percentile	Max	95th Percentile
kyber768	0.008	0.009	0.017	0.017	2.144	0.643
kyber1024	0.008	0.01	0.017	0.018	2.095	1.033
hqc128	0.010	0.017	0.034	0.041	1.305	0.318
hqc192	0.016	0.017	0.067	0.068	1.943	1.079
hqc256	0.017	0.019	0.118	0.119	3.869	1.129
bikel1	0.008	0.010	0.017	0.017	1.3	1.076
bikel3	0.009	0.017	0.018	0.018	2.404	1.079
bikel5	0.011	0.018	0.034	0.035	2.759	1.078
p256_kyber512	0.009	0.017	0.017	0.018	2.401	0.500
p384_kyber768	0.011	0.018	0.033	0.036	2.194	1.077
p256_kyber768	0.010	0.017	0.017	0.018	2.095	1.033
p521_kyber1024	0.012	0.018	0.034	0.035	2.306	1.077
p256_hqc128	0.012	0.018	0.034	0.034	2.096	0.340
p384_hqc192	0.016	0.018	0.068	0.069	2.653	1.086
p521_hqc256	0.021	0.034	0.118	0.12	3.149	1.129
p256_bikel1	0.010	0.017	0.017	0.018	2.099	1.078
p384_bikel3	0.011	0.018	0.034	0.039	2.905	1.03
p521_bikel5	0.016	0.018	0.035	0.04	3.488	1.079

Table 11. Detailed information of measuring handshake times (in sec) for packet loss = 6%.

cipher	Min	25th Percentile	Median	75th Percentile	Max	95th Percentile
prime256v1	0.01	0.017	0.018	0.219	3.362	1.081
secp384r1	0.016	0.018	0.019	0.219	4.984	1.083
secp521r1	0.016	0.018	0.035	0.221	3.668	1.082
kyber512	0.008	0.01	0.018	0.02	3.357	1.081
kyber768	0.008	0.01	0.018	0.018	6.408	1.081
kyber1024	0.008	0.01	0.018	0.018	8.794	1.291
hqc128	0.010	0.018	0.034	0.046	6.908	1.281
hqc192	0.016	0.018	0.068	0.076	8.6	1.131
hqc256	0.017	0.019	0.118	0.136	9.894	1.135
bikel1	0.008	0.01	0.018	0.019	5.506	1.081
bikel3	0.009	0.017	0.018	0.021	5.641	1.081
bikel5	0.011	0.018	0.034	0.037	6.002	1.128
p256_kyber512	0.01	0.017	0.018	0.019	3.466	1.081
p384_kyber768	0.011	0.018	0.034	0.039	4.76	1.082
p256_kyber768	0.01	0.017	0.018	0.019	4.606	1.283
p521_kyber1024	0.012	0.018	0.035	0.041	6.589	1.131
p256_hqc128	0.011	0.018	0.034	0.036	7.521	1.130
p384_hqc192	0.016	0.018	0.068	0.071	7.697	1.131
p521_hqc256	0.021	0.034	0.118	0.128	7.447	1.181
p256_bikel1	0.010	0.017	0.018	0.019	6.309	1.231
p384_bikel3	0.012	0.018	0.034	0.036	7.170	1.081
p521_bikel5	0.016	0.018	0.035	0.036	6.692	1.13

Table 12. Detailed information of measuring handshake times (in sec) for packet loss = 10%.

cipher	Min	25th Percentile	Median	75th Percentile	Max	95th Percentile
prime256v1	0.01	0.018	0.034	0.272	7.728	1.285
secp384r1	0.016	0.018	0.034	0.272	6.667	1.436
secp521r1	0.017	0.019	0.036	0.273	6.964	1.537
kyber512	0.008	0.01	0.019	0.22	7.265	1.689
kyber768	0.008	0.01	0.018	0.272	8.639	1.446
kyber1024	0.008	0.01	0.019	0.270	8.499	1.701
hqc128	0.010	0.018	0.035	0.270	6.454	1.334
hqc192	0.016	0.018	0.068	0.272	9.851	1.593

Table 12. *Cont.*

cipher	Min	25th Percentile	Median	75th Percentile	Max	95th Percentile
hqc256	0.017	0.021	0.12	0.272	9.978	1.534
bikel1	0.009	0.010	0.018	0.221	6.552	1.483
bikel3	0.009	0.017	0.034	0.319	6.298	1.687
bikel5	0.011	0.018	0.035	0.271	7.398	1.486
p256_kyber512	0.009	0.017	0.033	0.221	6.599	1.284
p384_kyber768	0.011	0.018	0.034	0.221	7.471	1.889
p256_kyber768	0.010	0.018	0.033	0.271	7.522	1.483
p521_kyber1024	0.012	0.018	0.035	0.272	7.463	1.846
p256_hqc128	0.012	0.018	0.036	0.271	8.459	2.091
p384_hqc192	0.016	0.018	0.069	0.271	8.149	1.535
p521_hqc256	0.033	0.035	0.12	0.272	9.492	1.810
p256_bikel1	0.010	0.018	0.033	0.272	6.611	1.922
p384_bikel3	0.016	0.018	0.035	0.525	6.74	1.891
p521_bikel5	0.016	0.018	0.036	0.472	6.812	1.546

While the tables offer a structured overview of the results, visual representations presented next offer a clearer aspect of performance trends.

First, Figure 1 combines data from Tables 2–5 to compare the performance of classical ECDH against purely post-quantum implementations. Based on the relevant outcomes, we obtain the following results:

- In general, most post-quantum algorithms exhibit performance characteristics quite similar to those of their classical counterparts. More precisely, the median values for TLS handshake duration are observed to be higher in the cases of HQC192 and HQC256. However, at both the 50th and 95th percentile levels, the performances of the remaining algorithms appear to be largely comparable.
- For the Level 3 and Level 5 variants of BIKE and HQC, a noticeable increase in the median result emerges once the packet loss ratio exceeds 10%. This observation is not surprising, as higher packet loss typically leads to more frequent retransmissions within the network. The corresponding increase in the 95th percentile value occurs more gradually.
- Overall, the performances of the two code-based schemes remain quite comparable to that of the lattice-based CRYSTALS-Kyber.

Next, Figure 2 combines data from Tables 2, 3, 6, and 7 to allow comparison between classical ECDH and hybrid PQ implementations. Based on these results, the following observations can be made:

- Hybrid variants of post-quantum algorithms exhibit performances comparable to that of classical ciphers, in line with the behavior observed for their pure post-quantum counterparts.
- The key encapsulation mechanisms (KEMs) with the highest median values were p384_hqc192 and p521_hqc256, whereas the remaining algorithms showed largely comparable performances at both the 50th and 95th percentile levels.
- Similar to the pure post-quantum schemes, a distinct increase in median values was observed once the packet loss ratio exceeded 10%.
- In general, the performance of the hybrid variants appeared to be closely aligned with those of the corresponding pure post-quantum schemes.

Last, but not least, Figure 3 allows the comparison of the results from Tables 4–7, focusing on the performances of purely PQ and hybrid implementations for the three key exchange algorithms. This comparison is being split into three subfigures, one for each

algorithm: CRYSTALS-Kyber, HQC, and BIKE. The results presented in Figure 3 suggest the following:

- Overall, the various implementations of post-quantum key encapsulation mechanisms (KEMs) demonstrated broadly comparable performances with respect to both the 50th percentile (median) and the 95th percentile of TLS handshake durations.
- A slight divergence can be observed for the PQC and hybrid Level 3 and Level 5 variants of HQC (i.e., HQC192 and HQC256), which exhibited marginally higher median handshake durations. This pattern is consistent with the trends already identified in Figures 1 and 2 and is further confirmed here.
- In contrast to the other KEMs, the CRYSTALS-Kyber family did not display any spike in median handshake duration, even under packet loss conditions exceeding 10%.

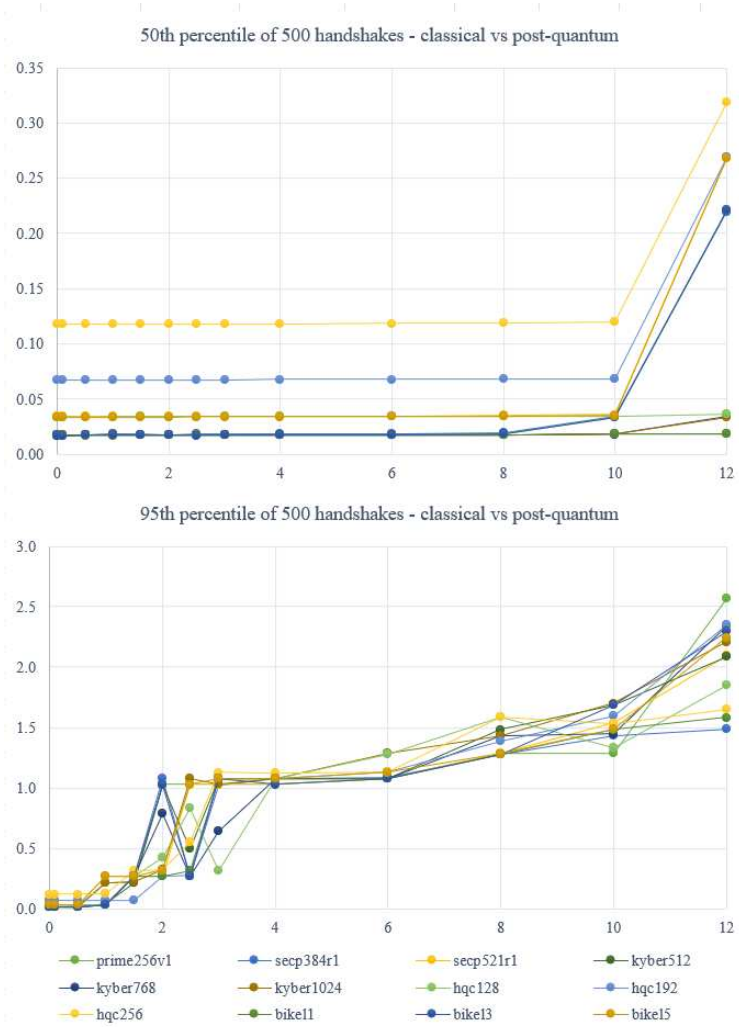

Figure 1. Classical vs. post-quantum implementations.

Next we examine the overall performance in more depth by considering the mean and relevant standard deviation of the measurements; this will shed more light on the actual performance characteristics rather than simply examining the 25%, 50%, 75%, and 95% percentiles. To this end, we next present these results in Table 13 (for packet loss ratio = 0%), in Table 14 (for packet loss ratio = 1%), in Table 15 (for packet loss ratio = 3%), in Table 16 (for packet loss ratio = 6%), and in Table 17 (for packet loss ratio = 10%). These Tables also present, in their last columns, the space defining the confidence internal for a confidence level of 99%; the small widths of the confidence intervals suggest a high degree of certainty for our estimates.

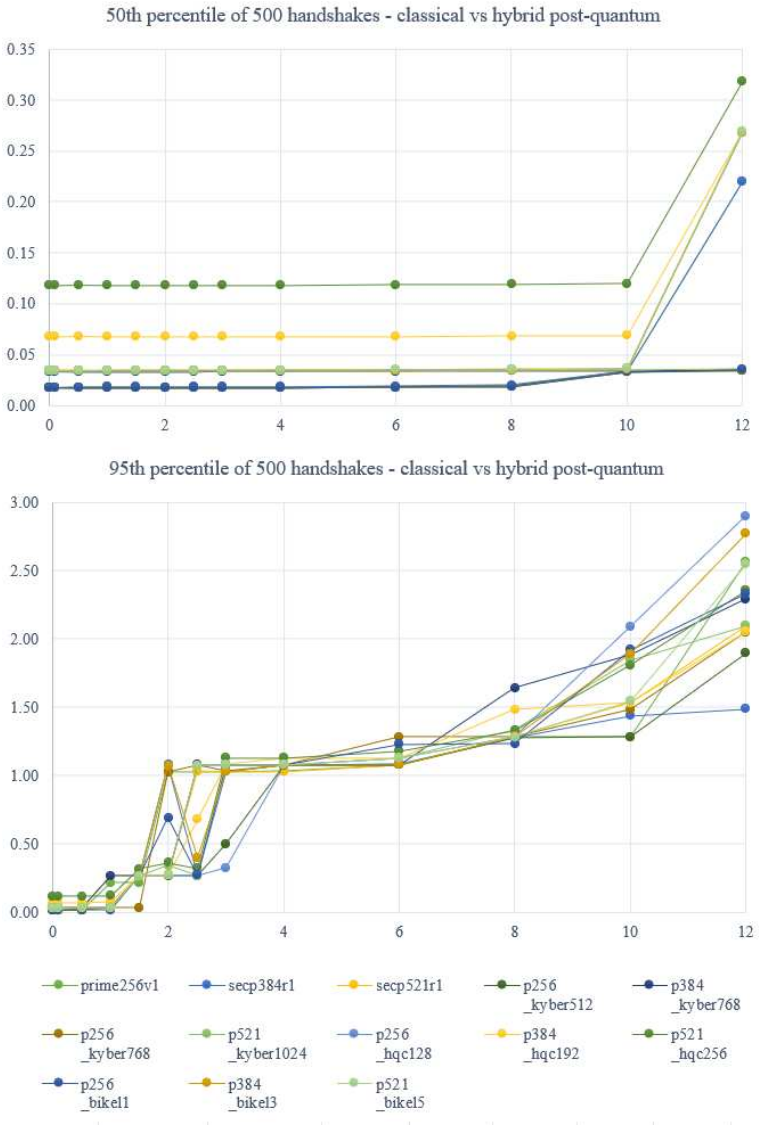

Figure 2. Classical vs. hybrid implementations.

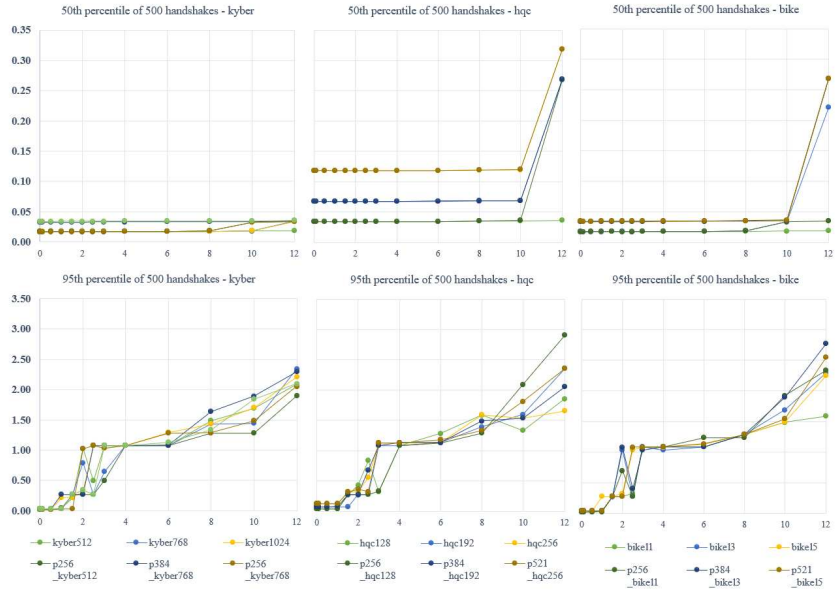

Figure 3. Purely post-quantum vs. hybrid implementations.

Table 13. Packet loss ratio = 0%.

cipher	Mean	std_deviation	Cl_99%
prime256v1	0.017	0.002	±0
secp384r1	0.024	0.010	±0.00008
secp521r1	0.020	0.004	±0.000325841
kyber512	0.011	0.003	±0.000244381
kyber768	0.012	0.003	±0.000244381
kyber1024	0.012	0.004	±0.000325841
hqc128	0.017	0.002	±0.000162921
hqc192	0.018	0.001	±0.0000815
hqc256	0.024	0.007	±0.000570222
bikel1	0.012	0.003	±0.000244381
bikel3	0.016	0.003	±0.000244381
bikel5	0.018	0.001	±0.0000815
p256_kyber512	0.017	0.002	±0.000162921
p384_kyber768	0.018	0.001	±0.0000815
p256_kyber768	0.017	0.002	±0.000162921
p521_kyber1024	0.018	0.001	±0.0000815
p256_hqc128	0.018	0.001	±0.0000815
p384_hqc192	0.018	0.001	±0.0000815
p521_hqc256	0.035	0.001	±0.0000815
p256_bikel1	0.017	0.002	±0.000162921
p384_bikel3	0.018	0.001	±0.0000815
p521_bikel5	0.018	0.001	±0.0000815

Table 14. Packet loss ratio = 1%.

cipher	Mean	std_deviation	Cl_99%
prime256v1	0.045	0.151	±0.012300501
secp384r1	0.037	0.117	±0.009530852
secp521r1	0.038	0.116	±0.009449392
kyber512	0.031	0.132	±0.010752756
kyber768	0.040	0.164	±0.013359485
kyber1024	0.043	0.251	±0.020446528
hqc128	0.035	0.126	±0.010263994
hqc192	0.038	0.135	±0.010997137
hqc256	0.051	0.150	±0.012219041
bikel1	0.043	0.176	±0.014337008
bikel3	0.034	0.132	±0.010752756
bikel5	0.037	0.131	±0.010671296
p256_kyber512	0.053	0.187	±0.015233071
p384_kyber768	0.042	0.153	±0.012463422
p256_kyber768	0.039	0.142	±0.011567359
p521_kyber1024	0.036	0.129	±0.010508375
p256_hqc128	0.047	0.164	±0.013359485
p384_hqc192	0.046	0.166	±0.013522405
p521_hqc256	0.060	0.157	±0.012789263
p256_bikel1	0.046	0.160	±0.013033644
p384_bikel3	0.046	0.160	±0.013033644
p521_bikel5	0.048	0.173	±0.014092627

Table 15. Packet loss ratio = 3%.

cipher	Mean	std_deviation	Cl_99%
prime256v1	0.099	0.290	±0.023623479
secp384r1	0.096	0.251	±0.020446528
secp521r1	0.099	0.251	±0.020446528
kyber512	0.084	0.249	±0.020283608
kyber768	0.089	0.283	±0.023053257
kyber1024	0.091	0.282	±0.022971797
hqc128	0.105	0.413	±0.033643093

Table 15. *Cont.*

cipher	Mean	std_deviation	Cl_99%
hqc192	0.098	0.259	±0.021098211
hqc256	0.119	0.317	±0.025822906
bikel1	0.107	0.380	±0.030954904
bikel3	0.092	0.273	±0.022238654
bikel5	0.108	0.305	±0.024845383
p256_kyber512	0.102	0.286	±0.023297638
p384_kyber768	0.092	0.272	±0.022157194
p256_kyber768	0.105	0.291	±0.023704939
p521_kyber1024	0.113	0.389	±0.031688046
p256_hqc128	0.090	0.262	±0.021342591
p384_hqc192	0.101	0.280	±0.022808876
p521_hqc256	0.113	0.267	±0.021749893
p256_bikel1	0.099	0.273	±0.022238654
p384_bikel3	0.102	0.292	±0.0237864
p521_bikel5	0.098	0.268	±0.021831353

Table 16. Packet loss ratio = 6%.

cipher	Mean	std_deviation	Cl_99%
prime256v1	0.195	0.388	±0.031606586
secp384r1	0.207	0.464	±0.037797566
secp521r1	0.196	0.415	±0.033806013
kyber512	0.140	0.347	±0.028266715
kyber768	0.187	0.523	±0.042603723
kyber1024	0.210	0.623	±0.05074975
hqc128	0.222	0.627	±0.051075591
hqc192	0.220	0.501	±0.040811597
hqc256	0.203	0.523	±0.042603723
bikel1	0.197	0.462	±0.037634646
bikel3	0.198	0.494	±0.040241375
bikel5	0.213	0.480	±0.039100931
p256_kyber512	0.179	0.449	±0.036575662
p384_kyber768	0.211	0.616	±0.050179528
p256_kyber768	0.199	0.514	±0.04187058
p521_kyber1024	0.236	0.686	±0.055881747
p256_hqc128	0.263	0.674	±0.054904224
p384_hqc192	0.190	0.469	±0.038204868
p521_hqc256	0.212	0.515	±0.04195204
p256_bikel1	0.189	0.515	±0.04195204
p384_bikel3	0.198	0.495	±0.040322835
p521_bikel5	0.213	0.571	±0.046513816

Table 17. Packet loss ratio = 10%.

cipher	Mean	std_deviation	Cl_99%
prime256v1	0.397	0.744	±0.060606443
secp384r1	0.370	0.678	±0.055230065
secp521r1	0.377	0.688	±0.056044667
kyber512	0.324	0.666	±0.054252542
kyber768	0.385	0.813	±0.066227202
kyber1024	0.372	0.804	±0.065494059
hqc128	0.386	0.842	±0.068589549
hqc192	0.423	0.901	±0.073395706
hqc256	0.376	0.752	±0.061258125
bikel1	0.363	0.847	±0.068996851
bikel3	0.363	0.711	±0.057918254
bikel5	0.397	0.828	±0.067449106
p256_kyber512	0.317	0.645	±0.052541876
p384_kyber768	0.381	0.923	±0.075187832

Table 17. *Cont.*

cipher	Mean	std_deviation	CI_99%
p256_kyber768	0.436	0.997	±0.081215892
p521_kyber1024	0.420	0.941	±0.076654116
p256_hqc128	0.349	0.728	±0.059303078
p384_hqc192	0.383	0.791	±0.064435076
p521_hqc256	0.354	0.704	±0.057348032
p256_bikel1	0.398	0.864	±0.070381675
p384_bikel3	0.425	0.882	±0.07184796
p521_bikel5	0.388	0.754	±0.061421045

The information from Tables 13–17 is also illustrated in Figure 4. This analysis shows how important the network quality is, which seems to have a more prevalent role in the overall performance than the underlying cryptographic algorithms; more precisely, not only the mean value but also the standard deviation increase as the packet loss ratio deteriorated. However, for any fixed value of packet loss ratio, the mean values do not have significant differences among the various combinations of algorithms. However, the standard deviation seems to be higher in hybrid and purely post-quantum secure implementations, especially when using post-quantum secure algorithms with high security levels.

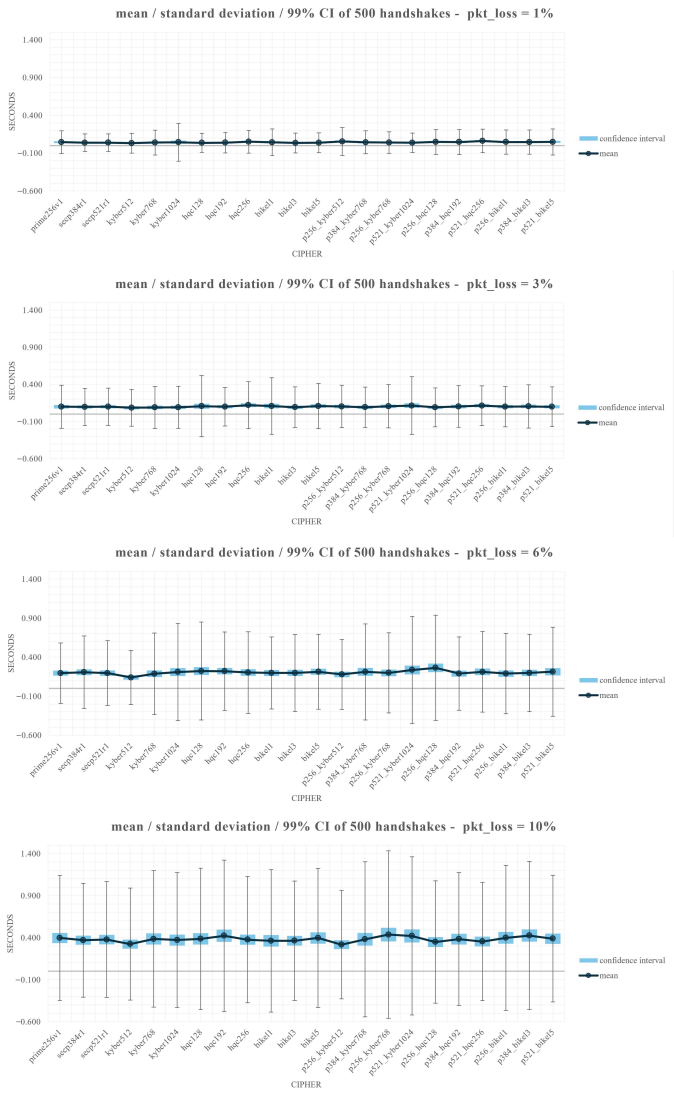

Figure 4. The mean values in conjunction with the standard deviations and the confidence intervals (CI = 99%) for various packet loss ratios.

5. Discussion

Evident from the previous analysis, our extensive set of experiments was sufficient for providing useful insights regarding the performance of various post-quantum secure TLS implementations, taking into account the current PQC standards. Indeed, we conclude that both post-quantum and hybrid quantum-secure key exchange algorithms demonstrate performances that are quite similar to that of classical key exchange methods. More precisely, as shown in Figures 1 and 2, the results across the various algorithms were almost identical. A noticeable increase in median (i.e., the 50th percentile) handshake times was observed once the packet loss ratio reached 10% or higher, whereas the 95th percentile values rose more gradually. Apparently, this is due to the fact that a higher packet loss inevitably leads to more retransmissions, thereby extending the duration of the TLS handshakes.

Moreover, Figure 3 highlights that the 95th percentile handshake durations of the post-quantum and hybrid variants remained similar across the three algorithms under consideration, with CRYSTALS-Kyber demonstrating slightly superior performance at higher packet loss ratios. On the contrary, HQC192 and HQC256 consistently exhibited the slowest median handshake times among both the PQC and hybrid KEMs (Figures 1 and 3), followed by the BIKE L5 variant. At the median level, Figure 3 also confirms the superior performance of CRYSTAL-Kyber relative to the other two ciphers. For packet loss ratios below 10%, PQC and hybrid BIKE achieved similar results, although the median duration of handshakes for the BIKE L3 and L5 variants increased substantially with greater packet loss. Hybrid HQC consistently had longer median TLS handshake times across all packet loss conditions, with a pronounced spike occurring beyond the 10% threshold.

Despite these differences, the 95th percentile values remained closely aligned for all algorithms, with CRYSTALS-Kyber maintaining slightly better behavior. This suggests that, for the majority of the TLS handshakes, all ciphers performed somehow similarly (the median is affected by a small number of handshakes having extended duration). Overall, the findings illustrate that CRYSTALS-Kyber and BIKE provide the most stable performances across the tested metrics, with HQC performing slightly slower. In particular, HQC-128 maintained handshake durations of around 0.04 s, while CRYSTALS-Kyber and BIKE L1 variants began near 0.017 s; despite this proportional increase, the absolute increase in delay is minimal and practically negligible.

Looking at the mean values (Tables 13–17), we conclude that CRYSTALS-Kyber behaves better than classical elliptic curve algorithms, with the exception being the cases of large packet loss ratios and high security levels; therefore, its version with $L5$ security level is about 15% faster than the classical elliptic curve algorithms with the curve $P-521$ for packet loss ratio 0%. However, for higher packet loss ratios, it behaves worse than its elliptic curve counterpart, and as network quality deteriorates, the performance of CRYSTALS-Kyber at security level $L5$ degrades (for instance, for packet loss ratio = 6%, it is about 7% slower). However, the versions of KRYSTALS-Kyber at security levels $L1$ and $L3$ are almost always better than their elliptic curve counterparts, regardless of the network quality. On the other hand, BIKE is a little bit worse than KRYSTALS-Kyber at all security levels, though this is observation is generally more prevalent in higher packet loss ratios levels (for instance, for packet loss ratio = 10%, the BIKE at security level $L5$ is about 6.8% slower than CRYSTALS-Kyber at the same security level). The cryptographic standard HQC performs somehow similar to the BIKE, being better than BIKE in principle for better network conditions and worse in other cases, depending though on the security level; for example, for packet loss ratio = 6%, the versions $L1$ and $L3$ of HQC are worse than the corresponding versions of BIKE by about 3% and 10%, respectively, but it is about 4.5% faster than BIKE if we consider the $L5$ versions; in fact, at this security level and for

this packet loss ratio, HQC behaves better than CRYSTALS-Kyber, though still a little bit worse than its elliptic curve counterpart by about 3.5%. Interestingly enough, for higher packer loss ratios, HQC at security level *L*5 does not behave worse than its elliptic curve counterpart (indeed, for packet loss ratio=10%, HQC is only slightly better at about 0.2%).

Hence, in light of the above findings, and with reference to the research questions posed in Section 1, the following conclusions can be drawn:

Q1 In terms of performance, both purely post-quantum and hybrid implementations of quantum-secure key exchange algorithms behave well compared to their classical counterparts.

Q2 Across all parameters, CRYSTALS-Kyber and BIKE achieve the best performance, while HQC illustrates similar behavior, differing slightly mainly in terms of median and mean TLS handshake times.

Q3 Overall, the 95th percentile handshake durations for both the post-quantum and hybrid variants of the three algorithms under consideration were comparable to a great extent. In contrast, median values were primarily affected by the security level of the algorithm and the packet loss ratio, rather than by the use of a hybrid variant. In particular, the median values for BIKE and HQC L3 and L5 variants increased significantly under packet loss ratios greater than 10%.

Q4 The network conditions do affect the overall performance to a great extent, and this is especially the case for specific variants of HQC and BIKE (namely, for variants at high security levels). However, interestingly enough, this reduction in performance is more significant for classical algorithms than for the corresponding post-quantum secure variants. For instance, for packet loss ratio equal to 10%, the mean value for the handshake time is better for purely post-quantum ciphers such as CRYSTALS-Kyber compared to the corresponding elliptic curve implementations, and the same holds even for some HQC implementations. Therefore, at least for packet loss ratios no more that 12%, we do not observe a behavior that is significantly worse in post-quantum implementations than in the classical ones.

The above results are in line with the analysis performed by the NIST within the standardization process. More precisely, as explicitly stated in the NIST's report on the status of the fourth round of the relevant standardization procedure [15], it is difficult to compare the performance between HQC and BIKE; for example, with respect to the TLS 1.3 handshake performance, the NIST states that HQC outperforms BIKE under ideal network conditions, but the situation is different in the case of nonzero packet loss rates. NIST concludes that it cannot be a definitive assessment of which algorithm is better than the other in terms of performance, but both of them are expected to be acceptable for most general applications. Indeed, this inherent difficulty in comparing HQC and BIKE is also present in our results (see, for example, Table 4 as well as Tables 13–17), with the BIKE being better for most times in non-ideal networks.

6. Conclusions

This paper aims to contribute to the public discussion with respect to adopting post-quantum secure algorithms for real-time applications. To this end, an extensive set of experiments was carried out in terms of measuring the performance of post-quantum secure TLS 1.3 implementations, taking into account the most recent post-quantum secure cryptographic standards, and the experiments were based on a large number of parameters, namely, various network conditions on various security levels of the underlying algorithms, and both pure post-quantum and hybrid implementations were considered in comparison to their classical counterparts. The results further confirm the necessity of enhancing

security protocols and systems with post-quantum secure primitives; this transition is essential for security while having minimal impact on performance.

Several extensions are envisioned to further generalize the findings of this work. For example, the current setup, based on a single virtual machine with artificial packet loss injection, does not capture the full diversity of real-world networks and hardware, thus being a limitation of our research. Therefore, extending the experiments to multiple platforms and real-network scenarios (e.g., RTT = 50–100 ms) is a key direction for future research. Similarly, in such an upgraded experimental environment, more than 500 handshakes would be more ideal for our measurements. Additionally, the various post-quantum digital signatures should also be considered in terms of how they affect overall performance in terms of taking into account the time needed for verifying the certificate's digital signature.

Apparently, post-quantum cryptography is a highly evolving field with several open research directions. More precisely, post-quantum secure cryptographic primitives shall be considered for all widely deployed protocols and systems (e.g., IPSec, S/MIME, PGP, etc.), putting emphasis not only on the performance but also on establishing provable overall security. Moreover, PQC is currently being considered mainly in the context of enhancing security, while the domain of Privacy Enhancing Technologies (PETs) has received significantly less attention. Advanced cryptographic approaches (such as homomorphic encryption, zero-knowledge proofs, attribute-based encryption, etc.) that suffice to alleviate privacy concerns in various applications still rely on computational assumptions that will be vulnerable in the post-quantum era; hence, further research—in light of the relevant standardization procedure—is needed in this area.

Author Contributions: Conceptualization, K.L.; methodology, K.S. and K.L.; software, K.S.; experiments: K.S.; writing, K.S. and K.L.; supervision, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments that helped to greatly improve the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
CL	Confidence Level
ECC	Elliptic Curve Algorithm
ECDH	Elliptic Curve Diffie–Hellman
ESDSA	Elliptic Curve Digital Signature Algorithm
KEM	Key Encapsulation Mechanism
NIST	National Institute of Standards and Technology
OQS	Open Quantum Safe
PET	Privacy Enhancing Technology
RTT	Round Trip Time

TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTLB	Time To Last Byte
PQC	Post-quantum Cryptography

References

1. Albrecht, M.R.; Paterson, K.G. Analysing Cryptography in the Wild—A Retrospective. *IEEE Secur. Priv.* **2024**, *22*, 12–18. [CrossRef]
2. National Institute of Standards and Technology, National Cybersecurity Center of Excellence (NCCoE). Migration to Post-Quantum Cryptography: Quantum Readiness—Cryptographic Discovery (NIST SP 1800-38B), Preliminary Draft. NIST Special Publication (SP) 1800-38B, National Institute of Standards and Technology. 2023; Preliminary Draft. Available online: https://www.nccoe.nist.gov/sites/default/files/2023-12/pqc-migration-nist-sp-1800-38b-preliminary-draft.pdf (accessed on 31 August 2025).
3. European Commission. Recommendation (EU) 2024/1101 on a Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography. 2024. Official Journal of the European Union. Available online: https://eur-lex.europa.eu/eli/reco/2024/1101/oj (accessed on 31 August 2025).
4. Ivezic, M. White House—Quantum Related National Security Memorandum. 2022. Available online: https://postquantum.com/quantum-policy/white-house-quantum-security-memo/ (accessed on 31 August 2025).
5. National Institute of Standards and Technology. Post-Quantum Cryptography Standardization. Available online: https://csrc.nist.gov/pqc-standardization (accessed on 9 September 2025).
6. Bernstein, D.J.; Hülsing, A.T.; Lange, T. *Post-Quantum Cryptography—Integration Study*; Technical Report; European Union Agency for Cybersecurity (ENISA): Chalandri, Greece, 2022. [CrossRef]
7. McKinseyI&Company. What is Quantum Computing? 2025. Available online: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-quantum-computing (accessed on 31 August 2025).
8. Shor, P. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134. [CrossRef]
9. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 22–24 May 1996; STOC '96, pp. 212–219. [CrossRef]
10. Rescorla, E. *The Transport Layer Security (TLS) Protocol*; Version 1.3; RFC 8446; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2018. [CrossRef]
11. Alnahawi, N.; Müller, J.; Oupický, J.; Wiesmaier, A. A Comprehensive Survey on Post-Quantum TLS. *IACR Commun. Cryptol.* **2024**, *1*. [CrossRef]
12. National Institute of Standards and Technology (NIST). Security (Evaluation Criteria)—Post-Quantum Cryptography Standardization. 2017. Available online: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria) (accessed on 10 September 2025).
13. National Institute of Standards and Technology. Module-Lattice-Based Key-Encapsulation Mechanism Standard. Federal Information Processing Standards Publication 203. 2024. Available online: https://csrc.nist.gov/pubs/fips/203/final (accessed on 3 October 2025).
14. National Institute of Standards and Technology. Round 2 Additional Digital Signature Schemes. Computer Security Resource Center, National Institute of Standards and Technology. 2022. Available online: https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures (accessed on 3 October 2025).
15. Alagic, G.; Bros, M.; Ciadoux, P.; Cooper, D.; Dang, Q.; Dang, T.; Kelsey, J.; Lichtinger, J.; Liu, Y.K.; Miller, C.; et al. Status Report on the Fourth Round of the NIST Post-Quantum Cryptography Standardization Process. 2025. Available online: https://csrc.nist.gov/pubs/ir/8545/final (accessed on 3 October 2025).
16. Kwiatkowski, K.; Sullivan, N.; Langley, A.; Levin, D.; Mislove, A. Measuring TLS Key Exchange with Post-Quantum KEM 2019. Available online: https://csrc.nist.gov/CSRC/media/Events/Second-PQC-Standardization-Conference/documents/accepted-papers/kwiatkowski-measuring-tls.pdf (accessed on 31 August 2025).
17. Paquin, C.; Stebila, D.; Tamvada, G. Benchmarking Post-Quantum Cryptography in TLS. *Cryptology ePrint Archive*. 2019. Paper 2019/1447. Available online: https://eprint.iacr.org/2019/1447 (accessed on 31 August 2025).
18. Tzinos, I.; Limniotis, K.; Kolokotronis, N. Evaluating the performance of post-quantum secure algorithms in the TLS protocol. *J. Surveill. Secur. Saf.* **2022**, *3*, 101–127. [CrossRef]
19. 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, Taipei, Taiwan, 5–9 October 2020.

20. 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. Cryptology ePrint Archive. 2024. Paper 2024/176. Available online: https://eprint.iacr.org/2024/176 (accessed on 31 August 2025).
21. 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*; Su, C., Gritzalis, D., Piuri, V., Eds.; Springer: Cham, Switzerland, 2022; pp. 432–451.
22. Montenegro, J.A.; Rios, R.; Lopez-Cerezo, J. A performance evaluation framework for post-quantum TLS. *Future Gener. Comput. Syst.* **2025**, *175*, 108062. [CrossRef]
23. Henrich, J.; Schmitt, N.; Alnahawi, N.; Heinemann, A. A Lot of Data and Added Complexity. How Does PQC Affect the Performance of My TLS Connection? In Proceedings of the 28th International Conference (ISC 2025), Seoul, Republic of Korea, 20–22 October 2026; pp. 107–128.
24. Open Quantum Safe Project. Open Quantum Safe. 2025. Open-Source Project Supporting the Transition to Quantum-Resistant Cryptography; Part of the Linux Foundation Post-Quantum Cryptography Alliance. Available online: https://openquantumsafe.org/ (accessed on 31 August 2025).
25. Open Quantum Safe Project oqs-provider: OpenSSL 3 Provider Enabling Quantum-Safe Cryptography. Open-Source Project Supporting the Transition to Quantum-Resistant Cryptography; Part of the Linux Foundation Post-Quantum Cryptography Alliance. 2025. Available online: https://github.com/open-quantum-safe/oqs-provider (accessed on 31 August 2025).
26. Souvatzidaki, K. tls_timer.py. 2025. Available online: https://github.com/k-souvatzidaki/post-quantum-tls/blob/master/tls_timer.py (accessed on 10 November 2025).