Navigating the Quantum Frontier: A Study of Post-Quantum Cryptography in TLS Integration

CSCI-530: Computer Security Systems

Author:
Sricharan Koride
Student ID: 2343517466
Master of Science in Computer Science
University of Southern California

Statement:

I have read the Guide to Avoiding Plagiarism published by the student affairs office. I understand what is expected of me with respect to properly citing sources, and how to avoid representing the work of others as my own. The material in this paper was written by me, except for such material that is quoted or indented and properly cited to indicate the sources of the material. I understand that using the words of others, and simply tagging the sentence, paragraph, or section with a tag to the copied source does not constitute proper citation and that if such material is used verbatim or paraphrased it must be specifically conveyed (such as through the use of quotation marks or indentation) together with the citation. I further understand that overuse of properly cited quotations to avoid conveying the information in my own words, while it will not subject me to disciplinary action, does convey to the instructor that I do not understand the material enough to explain it in my own words, and will likely result in a lesser grade on the paper.

Signed: Sricharan Koride

ABSTRACT

This paper explores the integration of post-quantum cryptography (PQC) into Transport Layer Security (TLS) to mitigate the emerging threats posed by quantum computing. Highlighting the vulnerabilities of classical cryptography to quantum algorithms such as Shor's, it emphasizes the urgent need for quantum-resistant solutions. The study evaluates prominent PQC algorithms, including NIST finalists, focusing on their implementation in TLS through hybrid and purely post-quantum approaches. Additionally, the paper outlines future directions for PQC in TLS, including hybrid mechanisms, hardware optimizations, and adapting PQC for resource-constrained environments.
Index Terms— Post-Quantum Cryptography, Quantum Computing, Transport Layer Security (TLS), Hybrid Cryptographic Models, Authenticated Key Exchange (AKE), Quantum Threats, Benchmarking, Performance Analysis.

1. INTRODUCTION

Classical cryptography is widely used in many systems to protect communication. However, advances in quantum computing have begun to threaten the security of traditional cryptographic systems. There are two main kinds of cryptographic techniques: symmetric and asymmetric.

Symmetric cryptography uses a shared secret key to encrypt and decrypt data. In contrast, asymmetric key cryptography employs a public key to encrypt and a private key to decrypt. Asymmetric key cryptography relies on mathematical problems that are computationally hard for conventional computer systems. For example, the Rivest-Shamir-Adleman (RSA) algorithm [39] relies on the integer factorization problem. Key-exchange algorithms that are used to establish shared secrets such as the Elliptic Curve Diffie-Hellman (ECDH) [11] and the Elliptic Curve Digital Signature Algorithm (ECDSA) [21] rely on the discrete logarithm problem. Both of the aforementioned problems are computationally infeasible for traditional computers.

Shor's algorithm [40], in conjunction with a sufficiently powerful quantum computer, can virtually threaten a majority of communication systems that we have in place today. Although no quantum computer exists today that can run Shor's algorithm on a reasonably sized asymmetric key [16], it will manifest as a reality at some point in the future.

To address these threats, cryptographic algorithms that resist quantum attacks are being developed -referred to as post-quantum cryptographic (PQC) algorithms. NIST has finalized three candidates for immediate use after three rounds of scrutiny and has announced that the finalists will be standardized for the fourth round [2, 29, 3, 31, 30]. The IETF is integrating post-quantum cryptography into Internet protocols [36, 18], while ETSI is developing quantum-safe standards and fostering global collaboration [14].

One such protocol is Transport Layer Security (TLS) [37]. It is a fundamental protocol for securing internet communications.

Despite its widespread adoption and critical role in protecting sensitive data, TLS remains vulnerable to quantum threats due to its reliance on cryptographic primitives susceptible to quantum attacks. Addressing this vulnerability is imperative to ensure the continued security of communication systems in the quantum era.

The paper begins with Section 1, which introduces the vulnerabilities of classical cryptography to quantum attacks and the need for post-quantum cryptography (PQC) in TLS. Section 2 provides an overview of classical cryptography, its quantum vulnerabilities, and the foundational concepts of PQC. Section 3 discusses the mathematical foundations, key features, and drawbacks of prominent PQC algorithms, including the NIST finalists. Section 4 examines the critical role of TLS, its susceptibility to quantum threats, and how PQC can address these issues. Section 5 evaluates hybrid and purely post-quantum TLS implementations, analyzing key performance factors and benchmarking results. Finally, Section 6 summarizes the findings, highlights the potential of hybrid and post-quantum TLS approaches, and outlines future research directions.

2. BACKGROUND

2.1 Overview of Classical Cryptography and Vulnerabilities

As discussed previously, cryptography is broadly categorized into symmetric and asymmetric techniques. This paper focuses on asymmetric cryptography due to its reliance on public-key primitives that are particularly vulnerable to quantum attacks. Symmetric cryptography, while relatively more secure, is not immune.

With that caveat, we turn our focus to asymmetric key primitives. As stated earlier, RSA relies on the integer factorization problem. The public key in RSA (N) is a product of two large and secret prime numbers p and q. The security of RSA relies on the inability of traditional computers to find these prime factors given N.

Elliptic curve cryptography [23, 28] relies on the *Elliptic Curve Discrete Logarithm Problem (ECDLP)*, which is a mathematical challenge involving operations on points on a curve. The problem can be described simply: given two points G (a base point) and Q on an elliptic curve, it is very difficult to determine the number k such that $Q = kG$, where k is a secret integer. This difficulty forms the basis for the security in elliptic-curve cryptography.

In ECDH, the ECDLP ensures that two parties can exchange public keys and compute a shared secret securely. Each party multiplies the other's public key with their own private key to derive the same shared point, without an attacker being able to deduce the private keys from the exchanged public keys.

2.2 Impact of Quantum Algorithms on Cryptography

Grover's algorithm can accelerate brute-force attacks to find the symmetric key. It reduces the time complexity of brute-force attacks from $O(N)$ to $O(\sqrt{N})$ [19]. However, the speedup provided by Grover's algorithm is quadratic, which means that its impact can be mitigated by doubling the key length of the cipher [20, 5].

Shor's algorithm fundamentally breaks the security of RSA by efficiently factorizing large numbers, which are the foundation of RSA's public-key encryption. It does this using a quantum Fourier transform to detect periodicity in modular arithmetic [5].

Similarly, it compromises elliptic curve cryptography (ECC) by solving the discrete logarithm problem on elliptic curves, which relies on finding a specific multiplier in modular arithmetic. This demonstrates that both RSA and ECC are vulnerable to quantum attacks due to their reliance on problems that Shor's algorithm can solve efficiently with a sufficiently advanced quantum computer [5].

Table 1 summarizes the widely used symmetric and asymmetric cryptographic primitives and the impact of Shor's and Grover's algorithms on them. The security levels shown are against the best pre-quantum and post-quantum attacks known.

Name	Function	Pre-Quantum Attack Complexity	Post-Quantum Attack Complexity
Symmetric Cryptography			
AES-128	Symmetric Encryption	128	64 (Grover)
AES-256	Symmetric Encryption	256	128 (Grover)
Salsa20	Symmetric Encryption	256	128 (Grover)
GMAC	MAC	128	128 (no impact)
Poly1305	MAC	128	128 (no impact)
SHA-256	Hash Function	256	128 (Grover)
SHA3-256	Hash Function	256	128 (Grover)
Public-Key Cryptography			
RSA-3072	Encryption	128	Broken (Shor)
RSA-3072	Signature	128	Broken (Shor)
DH-3072	Key Exchange	128	Broken (Shor)
DSA-3072	Signature	128	Broken (Shor)
256-bit ECDH	Key Exchange	128	Broken (Shor)
256-bit ECDSA	Signature	128	Broken (Shor)

Table 1: Security levels of widely used cryptographic algorithms, pre- and post-quantum. Security levels are computed against the best classical and quantum attacks, with Grover's algorithm halving the security of symmetric systems and Shor's algorithm breaking public-key systems. Attack complexity b means that the best attacks use approximately 2^b operations. Reproduced with permission from Bernstein and Lange [5], © Springer Nature.

2.3 Transition to Post-Quantum Cryptography

To mitigate this imminent threat, many attempts have been made to develop secure alternatives to the current public-key cryptography used in TLS. *Post-Quantum Cryptography (PQC)* refers to cryptographic algorithms that are resistant to cryptanalytic attacks from both classical and quantum computers.

The National Institute of Standards and Technology (NIST) has undertaken a rigorous multi-round process to standardize post-quantum cryptography (PQC) algorithms to counter quantum threats. The process began in 2016, with 69 proposals submitted during the first round. These proposals encompassed various families of cryptographic approaches, including lattice-based, code-based, hash-based, and multivariate polynomial systems. (The description of these families of cryptographic approaches is beyond the scope of this paper; those interested may refer to [5] for a detailed explanation of the same.) After detailed evaluations, 26 candidates advanced to the second round in 2019 [2].

These were divided into two categories: 17 encryption and key-establishment schemes and 9 digital signature schemes. From this pool, 12 candidates advanced to the third round in 2020 [29], comprising 7 encryption/key-establishment schemes and 5 digital signature schemes. Following further scrutiny, NIST identified 4 primary candidates for standardization (CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON, and SPHINCS+) and announced 4 alternate candidates for further analysis in a subsequent fourth round [3, 31].

In August 2024, NIST finalized three post-quantum cryptographic standards [30]: CRYSTALS-Kyber for public-key encryption and key establishment, along with CRYSTALS-Dilithium and FALCON for digital signatures. NIST is continuing its evaluation of alternate candidates to ensure diversity and resilience in cryptographic standards.

Table 2 gives a summary of NIST Round 3 Post-Quantum Cryptography Finalists and Alternate Candidates.

Category	Algorithm Name	Mathematical Foundation
Finalists		
Key Encapsulation Mechanisms (KEM)	CRYSTALS-Kyber	Lattice-based
Digital Signature Algorithms	CRYSTALS-Dilithium	Lattice-based
	FALCON	Lattice-based
	SPHINCS+	Hash-based
Alternate Candidates		
Key Encapsulation Mechanisms (KEM)	Classic McEliece	Code-based
	BIKE	Code-based
	HQC	Code-based
	Saber	Lattice-based
	NTRU	Lattice-based
	FrodoKEM	Lattice-based
	NTRU Prime	Lattice-based
	SIKE* [10]	Isogeny-based
Digital Signature Algorithms	Rainbow* [6]	Multivariate-polynomial-based
	GeMSS	Multivariate-polynomial-based
	Picnic	Hash-based

Table 2: Summary of NIST Round 3 Post-Quantum Cryptography Finalists and Alternate Candidates. Note: Algorithms marked with * were later discontinued due to vulnerabilities identified through cryptanalysis.

2.4 Key Encapsulation Mechanisms

All the efforts examined in this study are based on Key Encapsulation Mechanisms (KEMs) as a replacement to traditional DH Key Exchange. KEM works as specified below in TLS [42]. Fig. 1 illustrates the establishment of a shared secret via KEM in TLS.

- A KEM public-private key pair is generated by the client and the public key is then sent to the server.
- The server *encapsulates* the public key of the client to generate the cipher text and the shared secret. This cipher text is forwarded to the client.
- The client *decapsulates* the ciphertext and obtains the shared secret.

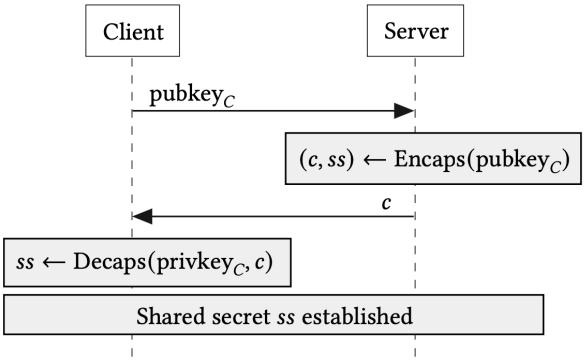

Figure 1: Illustration of the working of a KEM. Image reproduced from Alnahawi et al. [4]

3. POST-QUANTUM CRYPTOGRAPHY ALGORITHMS

3.1 Overview of PQC Algorithms

Post-quantum cryptography (PQC) encompasses diverse algorithms designed to secure systems against quantum attacks.

Table 3 summarizes the key features and drawbacks of the four finalized algorithms—highlighting their practical applications, computational efficiency, and challenges such as key size or computational cost.

Algorithm and Type	Key Features	Drawbacks
CRYSTALS-Kyber (KEM)	• Lattice-based (Shortest Vector Problem). • IND-CCA secure KEM based on MLWE. • Supports symmetric key establishment for TLS. • High efficiency	• Large public key sizes compared to classical cryptography.
CRYSTALS-Dilithium (Signature)	• Based on MLWE and MSIS. • Constructed using Fiat-Shamir transform. • Superior performance in key/signature generation and verification. • Chosen for Post-Quantum Cryptography (PQC).	• Larger signature sizes than classical alternatives like ECDSA.
Falcon (Signature)	• Based on SIS problem over NTRU lattices. • Compact, fast, scalable, and memory-efficient.	• Computationally intensive key generation and verification processes.
SPHINCS+ (Hash-Based Signature)	• Based on WOTS+ (Winternitz One-Time Signature+). • Balances security and efficiency as a fallback algorithm. • Resistant to attacks that affect lattice-based algorithms.	• High computational cost. • Larger signature sizes compared to lattice-based algorithms.

Table 3: Overview of the four finalized post-quantum cryptographic algorithms, highlighting their key features and drawbacks. Refer to Xu et al. [45] for a more detailed summary.

3.2 Security and Efficiency

PQC algorithms must balance computational efficiency with robust security. As shown in Table 4, NIST has categorized security levels based on equivalence to traditional cryptographic strength, such as AES-128 or SHA-256.

Level	Security Description and Use Cases	Algorithms Supporting This Level
1	• At least as hard to break as AES128 (exhaustive key search). • Suitable for protecting low- to medium-sensitivity data or short-term confidentiality.	• CRYSTALS-Kyber (Kyber512) • CRYSTALS-Dilithium (Dilithium2) • FALCON (Falcon-512) • SPHINCS+ (various parameters)
2	• At least as hard to break as SHA256 (collision search). • Applicable in applications requiring moderate data integrity, like digital signatures.	• CRYSTALS-Dilithium (Dilithium3)
3	• At least as hard to break as AES192 (exhaustive key search). • Used in critical applications requiring high security for long-term protection of sensitive data.	• CRYSTALS-Kyber (Kyber768) • SPHINCS+ (various parameters)
4	• At least as hard to break as SHA384 (collision search). • Rarely targeted for cryptographic implementations; primarily a theoretical security boundary.	• None
5	• At least as hard to break as AES256 (exhaustive key search). • Critical for national security systems, government communications, or any high-value data requiring long-term protection.	• CRYSTALS-Kyber (Kyber1024) • CRYSTALS-Dilithium (Dilithium5) • FALCON (Falcon-1024) • SPHINCS+ (various parameters)

Table 4: NIST Security Strength Categories: Each level describes the computational difficulty of breaking cryptographic primitives, with equivalencies to widely recognized algorithms such as AES and SHA for exhaustive key or collision searches. Additionally, potential use cases for each security level are provided. Reproduced from Doring et al. [12]

4. TLS AND POST-QUANTUM CRYPTOGRAPHY

This paper assumes that readers are already familiar with TLS 1.3 and its operational framework. For a deeper understanding, Alnahawi et al. provide a comprehensive explanation, including its building blocks and various phases [4]. Additionally, a detailed byte-by-byte breakdown of TLS 1.3 is available in the work by Driscoll et al. [13].

TLS comprises two sub-protocols: the handshake and the record protocol. The handshake protocol is responsible for authenticating peers, negotiating cryptographic parameters, and establishing shared secrets. These parameters and shared secrets are utilized by the record protocol to securely transmit sensitive data.

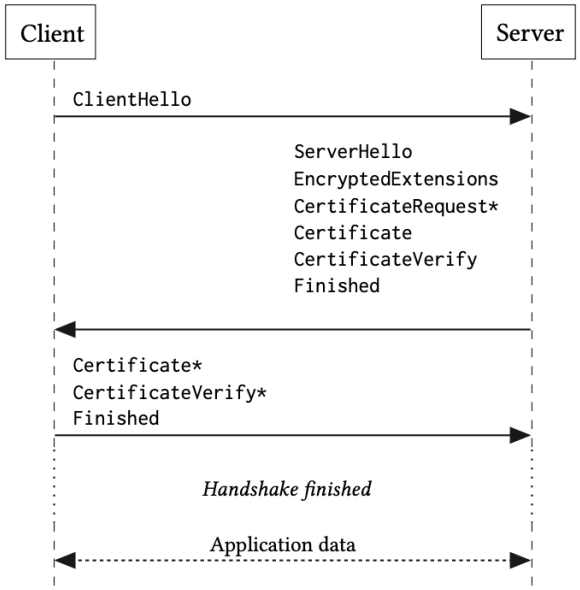

Figure 2: TLS 1.3 handshake protocol illustrating message exchanges between the client and the server. * are optional. Reproduced from Alnahawi et al. [4].

4.1 Prominence of TLS

TLS has seen widespread adoption across various platforms over the years. By 2023, more than 80% of websites utilized HTTPS, a significant increase from less than 50% in 2016 [24]. Furthermore, over 90% of Android applications implemented TLS to secure communication [35]. As of November 2024, Google reported that 99% of incoming emails and 98% of outgoing emails on Gmail were encrypted using STARTTLS [17].

The most commonly implemented variant of the TLS 1.3 handshake protocol features no pre-shared keys (PSKs), a single round-trip (1-RTT), and authentication via X.509 certificates, either with server-only or mutual authentication between the server and client.

4.2 Analysis of Quantum Threats to TLS

4.2.1 Threats to Symmetric Primitives

As discussed in 2.1 and analyzed in detail by Alnahawi et al. in Appendix B of their work, the symmetric primitives employed in TLS 1.3 are generally considered quantum-secure [4].

4.2.2 Threats to Asymmetric Primitives and Their Implications

The primary vulnerabilities in TLS arise from its reliance on asymmetric primitives, as highlighted in 2.

Quantum adversaries pose a critical threat to these primitives. A quantum attacker could exploit the vulnerabilities in DH key exchange to learn the shared secret, thereby compromising the **confidentiality** offered by TLS. This scenario becomes particularly alarming in "Harvest now, decrypt later" attacks, where an attacker records encrypted TLS sessions and decrypts them later, once they possess the necessary quantum computational resources.

Similarly, the vulnerability of ECDSA and RSA endangers the **authenticity** of TLS communications. An attacker could impersonate the Certificate Authority (CA), undermining the trust established within the protocol. However, it is important to note that harvest now, decrypt later attacks cannot target authenticity, as it is established at the conclusion of the handshake protocol.

4.3 Implementing PQC in TLS: A Practical Perspective

4.3.1 Integration of New Signature Schemes for Authentication

The TLS implementation abstracts signature schemes as opaque components at the higher protocol level. Consequently, integrating Post-Quantum signature schemes is relatively straightforward. For example, reserved code points in TLS can facilitate the adoption of new signature schemes. The supported TLS cipher suites are specified in the "cipher_suites" field using *code points*, which serve as unique identifiers for cryptographic algorithms employed in TLS [4, 37].

4.3.2 Integration of New Key Exchange Schemes

The process of secret key establishment in KEM closely resembles the traditional DH key exchange. KEMs can seamlessly replace DH key exchange in TLS without requiring significant modifications to the overall protocol flow. Moreover, considering that RSA is technically a KEM, this approach is not novel and has already been implemented in TLS 1.2 [4, 37].

The TLS handshake, a critical component of TLS connection establishment, involves key exchange and authentication, both of which significantly influence its performance. Key exchange can be implemented using a purely Post-Quantum Cryptography (PQC) algorithm or a hybrid approach that combines classical cryptographic systems with PQC algorithms.

4.3.3 OpenSSL and liboqs

OpenSSL's ssl directory [32] serves as the core for implementing secure communication protocols like TLS. It provides the building blocks for establishing secure connections, including ciphersuites, key exchange mechanisms, and record-layer processing. Open Quantum Safe's liboqs library is a crucial component in integrating Post-Quantum Cryptography (PQC) algorithms into OpenSSL. liboqs [25] offers a comprehensive collection of PQC implementations, including key encapsulation mechanisms (KEMs) and digital signature algorithms, that can be integrated into the ssl directory. This integration allows OpenSSL to support PQC ciphersuites within TLS, enabling secure communication that is resilient to attacks from future quantum computers. In fact, these are the frameworks that virtually all integration efforts rely on.

4.4 Hybrid Key Exchange Approaches

Two main methodologies are employed for hybrid key exchange:

- **Concatenation** Merges classical and PQC public keys or ciphertexts into a single entity, treating them as one within the TLS framework.
- **Separation** Modifies TLS data structures to transport classical and PQC cryptographic keys independently, maintaining their distinct identities.

4.5 TLS Authentication

TLS authentication relies on digital signatures and HMACs, with signatures playing a dual role:

- Validating certificates.
- Signing message hashes during the TLS handshake.

Authentication can be achieved using:

- Purely Post-Quantum Cryptography (PQC) approaches.
- Hybrid solutions combining classical cryptographic methods with PQC to enhance security and compatibility.

.

5. PERFORMATIVE COMPARISONS OF PQC IN TLS

5.1 Factors Affecting the Performance of PQC Algorithms

This paper identifies the following key factors as critical to the performance of post-quantum cryptographic (PQC) algorithms. These factors significantly influence their efficiency, scalability, and practicality in real-world deployments. Some of these factors, like network conditions, certificate chain length, security level, TCP window size, server throughput, etc., were taken into consideration in several studies while studying the performance of PQC algorithms in TLS.

- **Key Length** Larger keys in PQC algorithms often lead to higher memory and bandwidth requirements, which can impact performance, especially in resource-constrained environments.
- **Signature Size** Algorithms with large signatures can introduce latency issues, particularly in constrained environments where transmission bandwidth is limited.
- **Security Level** Higher security levels generally require more complex underlying mathematical problems. However, the relationship is not always straightforward, as some algorithms may achieve high security with relatively simpler problems through efficient design.
- **MTU (Maximum Transmission Unit)** Large PQC public keys and certificates can exceed the MTU, resulting in packet fragmentation and increased transmission overhead.
- **Hardness of Problems** The complexity of problems (e.g., lattices, multivariate polynomials) defines the trade-off between security and performance.
- **Certificate Size and Length of Certificate Chain** Large PQC certificates and lengthy certificate chains increase handshake times and memory requirements during TLS negotiations.
- **Network Conditions** High round-trip times (RTTs) and packet loss can exacerbate the performance impact of large keys and signatures, especially in latency-sensitive applications. The sensitivity to network conditions varies depending on the specific PQC algorithm.
- **Computational Complexity** Different PQC algorithms have varying computational costs. Some algorithms, like Kyber, are designed to be computationally efficient, while others, like SPHINCS+, may have higher overhead.
- **TCP Window Size** Optimizing the TCP window size can be beneficial for PQC, especially in high-latency networks. The optimal window size depends on various factors, including network characteristics and the specific PQC algorithm.
- **Signing Times from Server** The time taken by a server to sign data affects overall handshake efficiency, with Dilithium excelling in signing speed.
- **Server Throughput** Server throughput is influenced by various factors, including CPU utilization, memory usage, and network bandwidth. Algorithms with lower computational overhead can generally process more requests per unit time, leading to higher server throughput.

5.2 Integration Efforts Examined

This paper examines three efforts to integrate and measure the performance of PQC in TLS using one of the hybrid approaches described earlier. This is by no means representative of the exhaustive breadth of research that has been done in this regard but it is an attempt at capturing the general trend in this area within the limited scope of this paper. The analysis is limited to NIST finalists, with a brief mention of other candidates in the future scope section.

5.2.1 Study 1: Benchmarking Post-Quantum Cryptography in TLS [33]

The study focuses on implementing and evaluating hybrid key exchange and post-quantum authentication within TLS 1.3, using the OQS-OpenSSL 1.1.1 framework. The methodology maps TLS 1.3's ephemeral key exchange to support post-quantum and hybrid key exchange algorithms by defining new "groups" for each method. In this setup, the client sends public key values (keyshares) in its `ClientHello` messages, and the server responds with encapsulated ciphertexts in the `ServerHello`. The hybrid mode combines two algorithms, concatenating their shared secrets to replace the ECDH shared secret in the TLS key schedule. This implementation modifies OpenSSL's SSL directory and integrates Open Quantum Safe's `liboqs` to support post-quantum Key Encapsulation Mechanisms (KEMs). For authentication, post-quantum signature algorithms are integrated into TLS 1.3 by adding algorithm identifiers and utilizing OpenSSL's `EVP_PKEY` object for managing X.509 certificates, enabling seamless signature negotiation with minimal higher-level changes.

The network emulation framework employed in these experiments uses Linux namespaces and virtual Ethernet (veth) devices to isolate and simulate network conditions like delays, packet loss, and rate limits via the `netem` kernel module. Tools such as NetMirage and Mininet were used for complex setups, emulating large-scale autonomous systems. The experiments, conducted in a client-server topology, focused on the impact of network characteristics on post-quantum cryptography in TLS 1.3, providing insights into the performance and practical feasibility of PQC under real-world conditions.

Results:

- **Kyber512-90s:** Performs better than ECDH in high packet loss conditions and is comparable in low-loss, low-latency networks. Kyber introduces negligible computational overhead, making it a strong PQC alternative.
- **Dilithium2:** Suffers from large signature sizes (2044 bytes vs. ECDSA's 64 bytes), leading to higher transmission times. Performance decreases significantly under high packet loss, where ECDSA outperforms it. However, Dilithium2 remains competitive in typical network conditions with moderate overhead.

Strengths:

- **Comprehensive Evaluation of Realistic Network Conditions**:
  - Simulated varying round-trip times (RTTs) to account for geographical distances and real-world latency impacts.
  - Considered packet loss rates up to 20% to evaluate TLS handshake and web page retrieval performance under poor network conditions.
- **Realistic Application Scenarios**:
  - Incorporated larger payloads (1 KB to 1 MB) and multi-region setups to ensure scalability.
  - Highlighted practical use cases for PQC in TLS, focusing on both cryptographic and application-layer performance.
- **Application-Layer Focus**:
  - Assessed the impact of PQC algorithms on web page retrieval times, offering a user-centric perspective.
  - Expanded beyond typical cryptographic benchmarks that often focus only on handshake times or computational overhead.
- **Holistic View on Key Exchange**:
  - Evaluated the interaction between classical ECDH and hybrid PQC key exchange mechanisms.
  - Provided insights into transitional strategies for organizations adopting quantum-resilient systems.

Weaknesses:

- **Single-Machine Emulation**:
  - While practical, the use of a single-machine setup with network namespaces limits the ability to capture the complexities of distributed, real-world deployments.
  - Factors such as multi-hop routing, cross-region routing inefficiencies, and real network variability are not addressed.
- **No Exploration of Hybrid Authentication**:
  - The study focuses only on hybrid key exchange, neglecting hybrid authentication schemes.
  - Since hybrid authentication is a recommended strategy during the transitional phase to post-quantum cryptography, this omission leaves a critical gap in understanding performance and feasibility.
- **Exclusion of Purely PQC Key Exchange**:
  - By focusing on hybrid key exchange (e.g., ECDH combined with post-quantum algorithms), the study does not explore purely post-quantum key exchange mechanisms.
  - This limits insights into the performance of standalone PQC algorithms, such as Kyber or FrodoKEM, which will be critical in the long-term post-quantum era.
- **Limited Generalizability**:
  - The reliance on emulation, rather than a distributed or cloud-based deployment, may not fully capture the intricacies of real-world implementations.
  - Server-specific performance bottlenecks, cross-region networking, or varying infrastructure capabilities are not addressed.

5.2.2 Study 2: Post-Quantum Cryptography in Use: Empirical Analysis of the TLS Handshake Performance [12]

The selection of cryptographic algorithms focuses on classical options like RSA and ECC, which are widely used for digital signatures and key exchange, as well as Post-Quantum Cryptography (PQC) algorithms classified into Key Encapsulation Mechanisms (KEM) for key exchange and Signature Algorithms (SIG) for authentication. The analysis includes finalists and alternates from the NIST PQC competition, such as CRYSTALS-Dilithium, Falcon, and SPHINCS+ for signatures, and CRYSTALS-Kyber, Saber, and SIKE for key exchange. Criteria for selection emphasize algorithm popularity, variety of mathematical foundations, and compatibility with the testing environment. Security levels target NIST-defined levels 1, 3, and 5, which represent common usage scenarios.

Performance measurements for TLS handshakes utilize a controlled setup involving an Intel NUC7PJYH with Ubuntu 20.04.3 and libraries like liboqs and OpenSSL. The testing process focuses on full TLS 1.3 handshakes, using selected algorithms to generate key pairs and establish secure connections between a web server and client, ensuring minimal network impact by testing on the same machine. Results evaluate the handshake performance and security of the chosen algorithms under realistic configurations.

Results:

- **CRYSTALS-KYBER (Kyber512/768/1024):** Demonstrated superior performance across all security levels, with connection rates ranging from 950 to 1400 connections per second. Showed minimal computational overhead and scalability, maintaining high throughput even at higher security levels.
- **Dilithium (Dilithium2/3/5):** Performed well in signing tasks, with connection rates close to 900 connections per second at higher security levels but faced transmission overhead due to large signature sizes.
- **SPHINCS+ (SHAKE-256):** Struggled across all scenarios due to large signatures and high computational demands, with connection rates consistently below 100 connections per second.

Strengths:

- **Security Level Coverage**:
  - By including multiple security levels (1, 3, and 5), the study addresses varying real-world requirements.
  - Provides insights into how algorithms scale with increased security demands.
- **Exploration of Pure Post-Quantum (PQC) Mechanisms**:
  - Unlike many studies, it thoroughly explores *pure PQC key exchange and authentication* instead of focusing solely on hybrid approaches.
  - Offers valuable data for transitioning fully to quantum-resilient protocols.
- **Standalone and Combined Performance Metrics**:
  - Distinguishes between standalone performance (raw algorithmic efficiency) and combined performance (e.g., KEMs and signature algorithms in the handshake).
  - This layered approach helps identify bottlenecks at both algorithmic and protocol levels.

Weaknesses:

- **Lack of Realistic Network Simulation**:
  - The study does not emulate *real-world network conditions* such as latency, packet loss, or bandwidth constraints.
  - These conditions are critical for understanding performance in practical deployments.
- **No Hybrid Alternatives**:
  - While pure PQC mechanisms are explored, hybrid approaches (e.g., combining classical and PQC algorithms for key exchange or authentication) are absent.
  - Hybrid approaches are particularly important during the transition phase to full PQC systems.
- **No Information Retrieval Metrics**:
  - The study does not account for *real-world application scenarios* like webpage retrieval or data transfer times.
  - This omission limits the practical applicability of the findings, especially for TLS use in content delivery networks or web services.

5.2.3 Study 3: Assessing the Overhead of Post-Quantum Cryptography in TLS 1.3 and SSH [41]

The experimental setup described focuses on integrating post-quantum (PQ) cryptographic algorithms into SSH and TLS, using libraries like liboqs for implementations. Tests were conducted on a local host and three remote servers running Ubuntu 18.04, with varying network distances (close, intermediate, and long ranges) characterized by average round-trip times (RTTs) of 37 ms, 67 ms, and 163 ms, respectively. TLS utilized 1-RTT handshakes, and SSH employed key-based authentication. The local client was equipped with Intel i7-8665u hardware, while remote servers used Google Cloud instances with Intel Skylake Xeon processors.

For PQ schemes, the study chose optimized representatives from the NIST competition, focusing on algorithms suitable for TLS and SSH. The signature algorithms tested included Dilithium (lattice-based) and SPHINCS+ (hash-based), while key exchange algorithms included Kyber (lattice-based), NewHope, and NTRU-HRSS. Conventional security controls used RSA 2048 and ECDH with the NIST P-256 curve. These algorithms were chosen for their performance, computational cost, and security levels, aiming to balance practicality and cryptographic robustness in real-world scenarios.

Results

- **Handshake Latency** Kyber exhibits minimal latency increases (˜1–4 ms). SPHINCS+ shows significant latency increases, up to 190%.
- **Hybrid Key Exchange** ECDH + Kyber demonstrates negligible latency increases compared to ECDH-only handshakes.
- **TCP Congestion Window Optimization** Kyber benefits significantly from TCP window size optimization, reducing latency by up to 50%.

Strengths

- Comprehensive evaluation of **Kyber** and **SPHINCS+**, both critical to the NIST PQC standardization process.
- Focused analysis of **hybrid key exchange** (ECDH + Kyber) provides practical insights for transition strategies.

Weaknesses

- The study does not explore client authentication or the impact of SPHINCS+ in hybrid authentication scenarios.
- SPHINCS+ results highlight the need for further optimization of hash-based signature schemes in TLS.

6. CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This paper highlights several viable approaches to prepare TLS for the challenges posed by quantum attackers. While hybrid approaches introduce some overhead, their performance remains comparable to purely post-quantum solutions, offering a practical balance between efficiency and robust security during the transition to the quantum era. Notable attempts are being done by Meta [27], Google [8] and Cloudflare [7] in this regard.

From the above discussion we can say that integrating post-quantum cryptography (PQC) into Transport Layer Security (TLS) presents several challenges that must be addressed for successful deployment, some of which are also discussued by Zacharopoulos [47]:

1. **Algorithm Maturity:** Many PQC algorithms are relatively new and lack extensive cryptanalysis. This immaturity raises concerns about their long-term security, as evidenced by vulnerabilities discovered in candidates like Rainbow and SIKE during the NIST standardization process. Even the fourth round standardized algorithms like FALCON and CRYSTALS-Dilithium have been the subject of multiple attacks like side channel attacks [22, 44] and the BEARZ attack [26]. Organizations adopting these algorithms need to have a very robust implementation to secure them from such attacks.
2. **Performance Overhead:** PQC algorithms often introduce significant performance costs, particularly due to increased message sizes. This overhead can strain network resources and degrade performance, especially for devices with limited bandwidth or processing capabilities.
3. **Implementation Complexity:** Deploying PQC within existing TLS frameworks requires substantial modifications to accommodate new algorithms. This complexity can lead to integration challenges and potential security vulnerabilities if not managed carefully.
4. **Interoperability Issues:** Ensuring seamless interoperability between systems using different PQC algorithms is challenging, potentially causing compatibility issues and hindering widespread adoption.
5. **Regulatory and Compliance Concerns:** The evolving nature of PQC standards may conflict with existing regulations and compliance requirements, creating legal and operational uncertainties for organizations.

Addressing these drawbacks is crucial for the successful implementation of PQC in TLS, requiring ongoing research, standardization efforts, and careful consideration of practical deployment challenges.

6.2 Directions for the future

1. **Reducing Round Trips in TLS Handshake:** Optimizing the TLS handshake to minimize the number of round trips required for connection establishment could significantly enhance performance, particularly in latency-sensitive or high-delay environments. Future work could focus on combining post-quantum cryptography (PQC) with low-latency optimizations such as Turbo TLS [1].

2. **PQC in Satellite Communications:** Exploring the integration of PQC in satellite-based communications offers a promising direction, considering the unique challenges posed by long distances and high round-trip times. This could involve testing PQC algorithms under extreme latency and packet loss conditions to assess their viability for space-based networks [46].

3. **Quantum Key Distribution (QKD) Integration:** Combining QKD with PQC in TLS protocols represents a significant step toward achieving quantum-resilient security. Research efforts, such as those by Rubio Garcia et al. [15], who proposed a hybrid TLS solution integrating classical, quantum, and post-quantum cryptography, and Ricci et al. [38], who explored the practical implementation of hybrid keys combining these cryptographic paradigms, provide a foundation for future advancements. QKD can be effective in preventing SNDH attacks as it relies on unbreakable principles of quantum physics.

4. **Hardware-Based Optimization:** Leveraging advanced hardware features, such as Intel's AVX-512 instruction set, can drastically enhance the performance of resource-intensive post-quantum cryptographic algorithms. For instance, the work by Jieyu Zheng et al. demonstrates how AVX-512 optimizations for ML-KEM achieve significant speedups, highlighting the potential of hardware-aware optimizations in reducing computational overhead [48].

5. **Impact of Certificate Chain Length and Signature Algorithms:** The impact of certificate chain length and the choice of post-quantum signature algorithms on handshake latency and scalability requires further analysis. This includes understanding how long certificate chains or large signatures influence performance in constrained or high-latency environments. This is explored by Manohar Raavi et al. [34].

6. **Exploration of Non-Traditional TLS Modes:** Beyond client-server authentication, the performance and security of TLS modes using pre-shared keys (PSKs) or session resumption with PQC need further evaluation. These alternative modes could prove more efficient in specific use cases while ensuring quantum safety.

7. **Performance of Multiple PQC Algorithms:** Comprehensive evaluation of hybrid and standalone configurations involving three or more PQC algorithms can offer deeper insights into their relative strengths and weaknesses. This analysis could extend to their interoperability, scalability, and impact on application-layer metrics like throughput and latency.

8. **Integration of PQC in Embedded Systems:** Research is needed to assess the feasibility of implementing PQC algorithms in resource-constrained embedded systems, focusing on optimizing computational efficiency, memory usage, and power consumption without compromising security. This is critical for applications in IoT, automotive, and industrial control systems where embedded devices are prevalent. This is explored by Kevin B¨urstinghaus-Steinbach et al. [9] and George Tasopoulos et al. [43].

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Professor Clifford Neuman for his invaluable guidance and support throughout the duration of this research as part of the CSCI 530 Security Systems course. His insightful lectures, thought-provoking discussions, and expert advice have greatly enriched my understanding of cryptographic systems and their role in securing the quantum future.

I am also thankful for the resources and academic environment provided by the University of Southern California.

References

[1] Carlos Aguilar-Melchoro et al. "TurboTLS: TLS Connection Establishment with 1 Less Round Trip". In: *Computer Security – ESORICS 2024*. Springer Nature Switzerland, 2024, pp. 24–44. ISBN: 9783031708909. DOI: 10.1007/978-3-031-70890-9_2. URL: http://dx.doi.org/10.1007/978-3-031-70890-9_2.

[2] Gorjan Alagic et al. *Status report on the first round of the NIST post-quantum cryptography standardization process*. Jan. 2019. DOI: 10.6028/nist.ir.8240. URL: http://dx.doi.org/10.6028/nist.ir.8240.

[3] Gorjan Alagic et al. *Status report on the third round of the NIST Post-Quantum Cryptography Standardization process*. Sept. 2022. DOI: 10.6028/nist.ir.8413-upd1. URL: http://dx.doi.org/10.6028/nist.ir.8413-upd1.

[4] Nouri Alnahawi et al. "A Comprehensive Survey on Post-Quantum TLS". In: *IACR Communications in Cryptology* (July 2024). ISSN: 3006-5496. DOI: 10.62056/ahee0iuc. URL: http://dx.doi.org/10.62056/ahee0iuc.

[5] Daniel J. Bernstein and Tanja Lange. "Post-quantum cryptography". In: *Nature* 549.7671 (2017), pp. 188–194. ISSN: 1476-4687. DOI: 10.1038/nature23461. URL: https://doi.org/10.1038/nature23461.

[6] Ward Beullens. "Breaking Rainbow Takes a Weekend on a Laptop". In: *Advances in Cryptology – CRYPTO 2022*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Cham: Springer Nature Switzerland, 2022, pp. 464–479. ISBN: 978-3-031-15979-4.

[7] Cloudflare Blog. *Post-Quantum for All*. Accessed: 2024-12-06. Oct. 2023. URL: https://blog.cloudflare.com/post-quantum-for-all/.

[8] Google Security Blog. *Post-Quantum Cryptography Standards*. Accessed: 2024-12-06. Aug. 2024. URL: https://security.googleblog.com/2024/08/post-quantum-cryptography-standards.html.

[9] Kevin Bürstinghaus-Steinbach et al. "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. ACM, Oct. 2020, pp. 841–852. DOI: 10.1145/3320269.3384725. URL: http://dx.doi.org/10.1145/3320269.3384725.

[10] Wouter Castryck and Thomas Decru. "An Efficient Key Recovery Attack on SIDH". In: *Advances in Cryptology – EUROCRYPT 2023*. Ed. by Carmit Hazay and Martijn Stam. Cham: Springer Nature Switzerland, 2023, pp. 423–447. ISBN: 978-3-031-30589-4.

[11] W. Diffie and M. Hellman. "New directions in cryptography". In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[12] Ronny Doring and Marc Geitz. "Post-Quantum Cryptography in Use: Empirical Analysis of the TLS Handshake Performance". In: *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, Apr. 2022, pp. 1–5. DOI: 10.1109/noms54207.2022.9789913. URL: http://dx.doi.org/10.1109/noms54207.2022.9789913.

[13] M. Driscoll. *The Illustrated TLS 1.3 Connection: Every byte explained*. https://tls13.ulfheim.net. Accessed: 2024-11-25. 2018.

[14] European Telecommunications Standards Institute (ETSI). *ETSI Security Conference 2024*. https://www.etsi.org/events/2445-etsi-security-conference-2024. Accessed: 2024-11-25. Oct. 2024.

[15] Carlos Rubio Garcia et al. "Quantum-Resistant TLS 1.3: A Hybrid Solution Combining Classical, Quantum and Post-Quantum Cryptography". In: *2023 IEEE 28th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, Nov. 2023, pp. 246–251. DOI: 10.1109/camad59638.2023.10478407. URL: http://dx.doi.org/10.1109/camad59638.2023.10478407.

[16] Craig Gidney and Martin Ekerå. "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits". In: *Quantum* 5 (Apr. 2021), p. 433. ISSN: 2521-327X. DOI: 10.22331/q-2021-04-15-433. URL: http://dx.doi.org/10.22331/q-2021-04-15-433.

[17] Google Transparency Report. *Safer Email: Transparency Report*. https://transparencyreport.google.com/safer-email/overview. Accessed: 2024-11-26. 2024.

[18] IETF PQUIP Working Group. *Post-Quantum Cryptography for Engineers*. https://www.ietf.org/archive/id/draft-ietf-pquip-pqc-engineers-06.html. Accessed: 2024-11-25. Nov. 2024.

[19] Lov K. Grover. "Quantum Mechanics Helps in Searching for a Needle in a Haystack". In: *Physical Review Letters* 79.2 (July 1997), pp. 325–328. ISSN: 1079-7114. DOI: 10.1103/physrevlett.79.325. URL: http://dx.doi.org/10.1103/physrevlett.79.325.

[20] Samuel Jaques et al. "Implementing Grover oracles for quantum key search on AES and LowMC". In: *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II 30*. Springer. 2020, pp. 280–310.

[21] Don Johnson, Alfred Menezes, and Scott Vanstone. "The Elliptic Curve Digital Signature Algorithm (ECDSA)". In: *International Journal of Information Security* 1.1 (2001), pp. 36–63. ISSN: 1615-5262. DOI: 10.1007/s102070100002. URL: https://doi.org/10.1007/s102070100002.

[22] Emre Karabulut and Aydin Aysu. "FALCON Down: Breaking FALCON Post-Quantum Signature Scheme through Side-Channel Attacks". In: *2021 58th ACM/IEEE Design Automation Conference (DAC)*. 2021, pp. 691–696. DOI: 10.1109/DAC18074.2021.9586131.

[23] Neal Koblitz. "Elliptic Curve Cryptosystems". In: *Mathematics of Computation* 48.177 (1987), pp. 203–209. DOI: 10.1090/S0025-5718-1987-0866109-5. URL: https://www.ams.org/journals/mcom/1987-48-177/S0025-5718-1987-0866109-5/.

[24] Let's Encrypt. *Let's Encrypt Statistics: HTTPS Adoption*. https://letsencrypt.org/stats/. Accessed: 2024-11-26. 2023.

[25] *liboqs*. https://openquantumsafe.org/liboqs/. Accessed: 2024-12-06.

[26] Sarah McCarthy et al. "BEARZ Attack FALCON: Implementation Attacks with Countermeasures on the FALCON signature scheme". In: *IACR Cryptology ePrint Archive*. 2019. URL: https://api.semanticscholar.org/CorpusID:174800723.

[27] Meta Engineering. *Post-Quantum Readiness for TLS at Meta*. https://engineering.fb.com/2024/05/22/security/post-quantum-readiness-tls-pqr-meta/. Accessed: 2024-11-25. 2024.

[28] Victor S. Miller. "Use of Elliptic Curves in Cryptography". In: *Advances in Cryptology — CRYPTO '85 Proceedings*. Ed. by Hugh C. Williams. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 417–426. ISBN: 978-3-540-39799-1.

[29] Dustin Moody et al. *Status report on the second round of the NIST post-quantum cryptography standardization process*. July 2020. DOI: 10.6028/nist.ir.8309. URL: http://dx.doi.org/10.6028/nist.ir.8309.

[30] National Institute of Standards and Technology (NIST). *NIST Releases First 3 Finalized Post-Quantum Encryption Standards*. https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards. Accessed: 2024-11-25. 2024.

[31] National Institute of Standards and Technology (NIST). *PQC Candidates to be Standardized and Round 4 Announcement*. https://csrc.nist.gov/news/2022/pqc-candidates-to-be-standardized-and-round-4. Accessed: 2024-11-25. 2022.

[32] *OpenSSL*. https://openssl-library.org/. Accessed: 2024-12-06.

[33] Christian Paquin, Douglas Stebila, and Goutam Tamvada. "Benchmarking Post-quantum Cryptography in TLS". In: *Post-Quantum Cryptography*. Ed. by Jintai Ding and Jean-Pierre Tillich. Cham: Springer International Publishing, 2020, pp. 72–91. ISBN: 978-3-030-44223-1.

[34] Manohar Raavi et al. "Security Comparisons and Performance Analyses of Post-quantum Signature Algorithms". In: *Applied Cryptography and Network Security*. Springer International Publishing, 2021, pp. 424–447. ISBN: 9783030783754. DOI: 10.1007/978-3-030-78375-4_17. URL: http://dx.doi.org/10.1007/978-3-030-78375-4_17.

[35] Abbas Razaghpanah et al. "Studying TLS Usage in Android Apps". In: *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*. CoNEXT '17. Incheon, Republic of Korea: Association for Computing Machinery, 2017, pp. 350–362. ISBN: 9781450354226. DOI: 10.1145/3143361.3143400. URL: https://doi.org/10.1145/3143361.3143400.

[36] Sriharsha Reddy et al. *Post-Quantum Cryptography Recommendations for Internet Applications*. https://www.ietf.org/archive/id/draft-reddy-uta-pqc-app-03.html. Accessed: 2024-11-25. Nov. 2024.

[37] Eric Rescorla. *The transport layer security (TLS) protocol version 1.3*. Tech. rep. 2018.

[38] Sara Ricci et al. "Hybrid Keys in Practice: Combining Classical, Quantum and Post-Quantum Cryptography". In: *IEEE Access* 12 (2024), pp. 23206–23219. ISSN: 2169-3536. DOI: 10.1109/access.2024.3364520. URL: http://dx.doi.org/10.1109/access.2024.3364520.

[39] R. L. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Commun. ACM* 21.2 (Feb. 1978), pp. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342. URL: https://doi.org/10.1145/359340.359342.

[40] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Review* 41.2 (Jan. 1999), pp. 303–332. ISSN: 1095-7200. DOI: 10.1137/s0036144598347011. URL: http://dx.doi.org/10.1137/s0036144598347011.

[41] Dimitrios Sikeridis, Panos Kampanakis, and Michael Devetsikiotis. "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. ACM, Nov. 2020. DOI: 10.1145/3386367.3431305. URL: http://dx.doi.org/10.1145/3386367.3431305.

[42] Goutam Tamvada and Sofía Celi. *Deep dive into a post-quantum key encapsulation algorithm*. https://blog.cloudflare.com/post-quantum-key-encapsulation/. Accessed: 2024-12-02. Nov. 2022.

[43] George Tasopoulos et al. "Performance Evaluation of Post-Quantum TLS 1.3 on Resource-Constrained Embedded Systems". In: *Information Security Practice and Experience*. Springer International Publishing, 2022, pp. 432–451. ISBN: 9783031212802. DOI: 10.1007/978-3-031-21280-2_24. URL: http://dx.doi.org/10.1007/978-3-031-21280-2_24.

[44] Ruize Wang et al. "Single-Trace Side-Channel Attacks on CRYSTALS-Dilithium: Myth or Reality?" In: *IACR Cryptol. ePrint Arch.* 2023 (2023), p. 1931. URL: https://api.semanticscholar.org/CorpusID:266438362.

[45] Guobin Xu et al. "An Overview of Quantum-Safe Approaches: Quantum Key Distribution and Post-Quantum Cryptography". In: *2023 57th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, Mar. 2023, pp. 1–6. DOI: 10.1109/ciss56502.2023.10089619. URL: http://dx.doi.org/10.1109/ciss56502.2023.10089619.

[46] Shengbo Xu et al. "A Preliminary Study of PQC Implementations for Satellite Communication Networks". In: ().

[47] Dimitris Zacharopoulos. *On the Drawbacks of Post-Quantum Cryptography in TLS*. https://pkic.org/2024/09/27/on-the-drawbacks-of-post-quantum-cryptography-in-tls. Affiliation: HARICA, Accessed: 2024-11-25. 2024.

[48] Jieyu Zheng et al. "Faster Post-quantum TLS 1.3 Based on ML-KEM: Implementation and Assessment". In: *Computer Security – ESORICS 2024*. Springer Nature Switzerland, 2024, pp. 123–143. ISBN: 9783031708909. DOI: 10.1007/978-3-031-70890-9_7. URL: http://dx.doi.org/10.1007/978-3-031-70890-9_7.