The ISC Int'l Journal of Information Security

ISeCure

October 2025, Volume 17, Number 3 (pp. 1–12)

http://www.isecure-journal.org

Presented at the ISCISC'2025 in Tehran, Iran.

Recent Trends in Post-Quantum Cryptography Integration and Performance in the Internet Security Stack **

Togu Novriansyah Turnip [1,2,*], and Birger Andersen [1], and Cesar Vargas-Rosales [3]

[1] *Department of Engineering Technology, Technical University of Denmark, Denmark*

[2] *Faculty of Vocational, Del Institute of Technology, Indonesia*

[3] *Tecnologico de Monterrey, School of Engineering and Science, Monterrey, Mexico*

ARTICLE INFO.

Keywords:
Internet Security, IPSec, Post-Quantum Cryptography, SSH, TLS

Type:

doi:

ABSTRACT

The rapid advancement of quantum computing poses a direct threat to classical public-key cryptographic systems at the core of Internet security protocols. Post-quantum cryptography (PQC) has therefore become central to ongoing standardisation and early deployment efforts. This paper presents a comparative analysis of PQC integration into TLS, SSH, and IPsec, examining cross-cutting challenges, protocol-specific trade-offs, and deployment considerations. Our findings show that PQC adoption introduces markedly uneven overheads across protocols: handshake latency may increase by up to 600% in TLS, by 29% in SSH, and by up to 300% in IPsec, while memory requirements in hybrid configurations can exceed 300 KB in resource-constrained environments. We further demonstrate that message fragmentation, certificate chain expansion, and cumulative rekeying costs emerge as protocol-dependent bottlenecks, underscoring that migration strategies must be tailored to the architecture and operational context of each protocol. Beyond performance, we identify interoperability gaps, downgrade vulnerabilities, and side-channel risks as critical obstacles to secure deployment. By combining empirical performance evidence with a structured review of challenges and deployment strategies, our study provides actionable insights for practitioners, informs ongoing standards development, and highlights research priorities essential to building a resilient, quantum-resistant Internet infrastructure.

1 Introduction

The fundamental cryptographic systems that secure digital communications on the Internet are becoming increasingly vulnerable with the upcoming development of quantum computing. Current cryptographic systems, especially those based on ECC (Elliptic Curve Cryptography), RSA

* Corresponding author.

**The ISCISC'2025 program committee effort is highly acknowledged for reviewing this paper.

Email addresses: tnotu@dtu.dk, birad@dtu.dk, cvargas@tec.mx

ISSN: 2008-2045

(Rivest–Shamir–Adleman), and DH (Diffie-Hellman), which depend on the computational difficulty of factoring large numbers or solving discrete logarithms, are seriously threatened by the potential of quantum computers. The current threat requires a shift towards PQC that can resist threats from both traditional and quantum computing. The latest progress in quantum technology has led to a significant increase in research that explores the implementation of PQC into crucial Internet protocols. SSH, TLS, and IPsec are vital for ensuring the confidentiality, integrity, and availability of communications on the Internet. Integrating PQC into these protocols is crucial for ensuring they remain robust against potential quantum attacks. Most of the explored studies have explored the feasibility of migrating from traditional cryptography to PQC. Sikeridis *et al.* [1] demonstrated that PQC can result in a significant increase in handshake latency for TLS by up to 300% and for SSH by up to 50%, depending on the specific post-quantum algorithms used. Furthermore, they examined the impact of the initial TCP window size on the performance of post-quantum TLS and SSH protocols. Based on their findings, a slight increase in size can effectively mitigate the slowing down, resulting in a 50% reduction. In addition, Pazienza, A. *et al.* [2] analysed key exchange protocols in the quantum era, and provided the requirements for a secure key exchange protocol.

The implementation of PQC presents significant challenges. PQC algorithms typically require larger cryptographic keys and more computational resources, leading to increased latency and reduced efficiency in protocol operations. This poses difficulties when integrating PQC into existing internet protocols, as it can substantially affect the performance of network communications and the overall user experience. For example, increased handshake latencies in protocols like TLS and SSH can impact a range of activities, from simple web browsing to secure remote server access. Nethen NV. *et al.* [3] have outlined a process for managing the migration from traditional to PQC cryptography, emphasising the importance of a well-structured transition plan to mitigate these challenges. Furthermore, Crockett, E. *et al.* [4] demonstrated that the adoption of PQC could lead to noticeable increases in handshake latencies for TLS and SSH protocols. This increase in latency could potentially degrade the user experience, highlighting the need for careful consideration and optimisation during the migration to PQC. The National Institute of Standards and Technology (NIST) is actively pursuing the standardisation of quantum-resistant cryptographic schemes, with ongoing projects since 2016 [5]. Furthermore, Open Quantum Safe (OQS) is creating open-source libraries to develop PQC prototyping [6]. While OQS provides quantum-safe cryptography for these protocols through its OpenSSL and OpenSSH forks and other integrations, there has been no mention of a specific implementation for IPsec integration in its documentation. As quantum computing continues to evolve, the necessity for crypto-agility and the early integration of PQC into vital Internet protocols becomes increasingly complex [7]. The implementation of these quantum-resistant algorithms requires an in-depth review of the current protocol stack, as outlined by Illiano *et al.* [8].

In this paper, we review the challenges faced and the solutions proposed in the current state of research on the implementation of PQC in Internet protocols. It also provides an overview of the various studies, findings, and developmental projects that aim to secure the Internet against the quantum computing era. As we dive into the complex challenges of implementing PQC, the survey seeks to provide valuable information on how these critical protocols can be altered to guarantee a secure digital future.

The remainder of this paper is structured as follows. Section 2 reviews recent survey studies related to the integration of post-quantum cryptography and secure communication protocols. Section 3 outlines the internet protocol stack and summarizes current standardisation efforts for PQC algorithms. Section 4 analyses the integration of PQC into TLS, SSH, and IPsec, focusing on implementation challenges, performance impacts, and deployment considerations. Section 5 concludes the paper and highlights key directions for future research.

2 Related Work

Recent survey studies have extensively explored the broader fields of quantum cryptography and PQC, highlighting various aspects ranging from quantum internet protocols and cryptographic algorithm implementations to security considerations for future networks as illustrated in Figure 1. Illiano *et al.* [8] and Li *et al.* [9] provided comprehensive overviews of quantum internet protocols from a layered architectural perspective but primarily focused on theoretical and conceptual rather than oriented towards protocol-specific practical implementations. Kumar and Garhwal [10], along with Subramani *et al.* [11], reviewed state-of-the-art quantum cryptography techniques, emphasising classical versus quantum methods, yet without addressing specific internet protocol integrations.

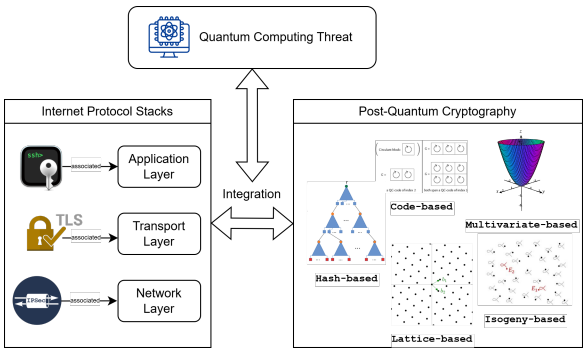

Figure 1. Integration of PQC into Internet Protocol Stacks for Quantum-Safe Communication

While prior studies have examined quantum cryptography integration across various domains, for example, Mehic *et al.* [12] on cellular networks, Fernández-Caramés [13] on IoT security, and Nejatollahi *et al.* [14] on lattice-based algorithm performance, they often neglect the specific implementation and integration challenges of PQC within widely used protocols like TLS, SSH, and IPsec. More recent reviews and focused analyses, including those by Baseri *et al.* [7], Durr-E-Shahwar *et al.* [15], Dam *et al.* [16], Alnahawi *et al.* [17], and Dekkaki *et al.* (2024) [18], have begun to address protocol-level transitions, particularly for TLS, but still lack comprehensive comparative evaluations that span multiple protocols. In contrast, Turnip *et al.* [19] provide a systematic review of hybrid PQC integration for authentication and key agreement in next-generation networks, highlighting the importance of balancing security and performance when deploying PQC in heterogeneous environments. Complementing this, Abbasi *et al.* [20] offer detailed empirical benchmarks across diverse hardware platforms, demonstrating that while PQC integration in TLS 1.3 is feasible for powerful systems, it introduces significant overhead in resource-constrained settings. These recent works bridge critical gaps by explicitly evaluating the protocol-specific challenges, comparative performance, and security implications of PQC deployment across the core layers of the internet security infrastructure.

Therefore, the clear research gap lies in a comprehensive, comparative analysis explicitly targeting the integration of PQC into TLS, SSH, and IPsec protocols. This gap includes limited discussions on cross-protocol implementation challenges, comparative performance evaluations, and a nuanced analysis of security implications specific to these widely adopted protocols. Addressing this research gap would significantly enhance the existing knowledge base, providing researchers and practitioners with comprehensive, actionable insights essential for facilitating practical PQC deployment and guiding future research directions across crucial internet security protocols.

3 Standardisation of PQC Algorithms

PQC algorithms are cryptographic algorithms designed to be secure against the potential threats posed by quantum computers. Quantum computers are expected to solve certain problems exponentially faster than traditional computers. To address these concerns, PQC algorithms that can resist quantum attacks are being developed and standardised. The NIST is a pioneer in standardising PQC algorithms that can withstand the attacks of quantum computers. These efforts involve evaluating and vetting a variety of candidate algorithms to ensure their security and practicality for real-world applications, such as in M2M, vehicular communication, and IoT networks [21–23].

The family of PQC algorithms include lattice-based, hash-based, code-based, multivariate polynomial, and isogeny-based cryptography [24].

(1) **Lattice-based cryptography** Lattice-based cryptography, which includes algorithms like NTRU [25] and Kyber [26], relies on the hardness of problems related to lattice structures, such as the Learning With Errors (LWE) problem. These algorithms are considered promising for their efficiency and security against both traditional and quantum attacks.

(2) **Code-based cryptography** Code-based cryptography, exemplified by the McEliece cryptosystem, relies on the hardness of decoding random linear codes [27]. Despite its large key sizes, it is highly secure and has been studied extensively over several decades.

(3) **Isogeny-based cryptography** Isogeny-based cryptography, such as the Supersingular Isogeny Key Encapsulation (SIKE) protocol [28], uses the hardness of finding isogenies between elliptic curves. These algorithms offer compact key sizes and strong security, making them attractive applications.

(4) **Multivariate-based cryptography** Multivariate polynomial cryptography involves solving systems of multivariate polynomial equations [29]. The Rainbow signature scheme is a notable example, known for its strong security properties and efficiency.

(5) **Hash-based cryptography** Hash-based algorithms, such as eXtended Merkle Signature Scheme (XMSS) [30] and SPHINCS+ [31], use hash functions to create secure digital signatures. These algorithms are highly secure and well-understood.

In 2016, NIST initiated a global call for propos-

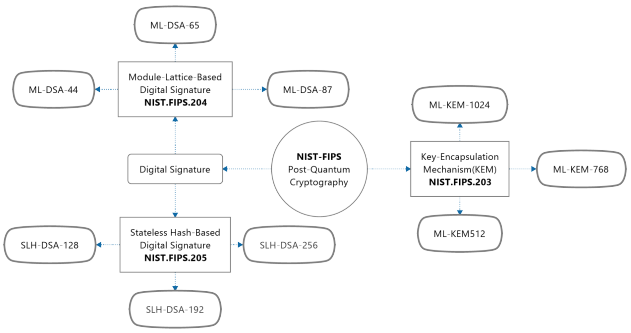

Figure 2. Three Final NIST Post-Quantum Cryptography Standards

als for quantum-resistant cryptographic algorithms, receiving 69 submissions for evaluation [32]. The evaluation process for PQC algorithms includes several rounds, each narrowing down the candidate algorithms based on various criteria such as security, performance, and implementation characteristics. The algorithms considered for standardisation cover a range of cryptographic functionalities, including digital signatures [33] and key exchange mechanisms (KEM) [34]. PQC algorithms often have different performance characteristics compared to traditional algorithms. Factors such as key sizes, computational overhead, and efficiency are critical to practical implementation [35]. The study conducted by Nejatollahi *et al.* [14] found that the majority of the current PQC algorithms are based on lattice schemes. Lattice-based schemes have smaller key sizes compared to some code-based and multivariate structures, which possess larger public key sizes.

A notable recent advancement is the finalisation of PQC standards by NIST, as illustrated in Figure 2. On 13 August 2024, NIST released the Federal Information Processing Standards (FIPS) documents that define official standards for PQC algorithms [36]:

- **NIST FIPS 203** defines the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM), which is derived from the **CRYSTALS-KYBER** scheme and based on the computational hardness of the Module Learning With Errors (MLWE) problem. It is particularly ideal for applications requiring robust, long-term protection of critical data and communication channels.
- **NIST FIPS 204** defines the Module-Lattice-Based Digital Signature Algorithm (ML-DSA), a cryptographic standard that relies on the MLWE problem for secure digital signatures. ML-DSA, a derivative of the **CRYSTALS-DILITHIUM** scheme, is designed to provide strong defence against quantum computing threats.
- **NIST FIPS 205** provides the Stateless Hash-Based Digital Signature Algorithm (SLH-DSA), derived from the **SPHINCS+** scheme. It offers various security levels, making it suitable for applications requiring robust data integrity and non-repudiation, including electronic communications, software distribution, and data storage.

On March 11, 2025, NIST selected **HQC (Hamming Quasi-Cyclic)** as the fifth algorithm in its standardisation and a backup choice for KEM for general-purpose encryption. HQC relies on error-correcting codes to support the primary lattice-based algorithm ML-KEM by providing a mathematically unique and robust alternative. A draft standard is expected within one year, marking a significant step toward a more agile and future-proof cryptographic landscape.

4 Integrating PQC into SSH, TLS, and IPsec

In this section, we present an overview of the internet protocol stack's challenges on the implementation of PQC. We also discuss the trade-offs of performance and security. Furthermore, we consider and present the practicality and deployment of PQC adoption. Finally, we elaborate on future trends or directions.

4.1 General Challenges of PQC Deployment

The implementation of PQC introduces various performance degradation aspects that need to be carefully considered, especially as we transition into 6G networks and broader IoT integration. These challenges revolve primarily around computational overhead, bandwidth requirements, and latency issues. Each of these areas significantly impacts system performance.

(1) **Performance Overheads and Optimisation**:
PQC algorithms can influence significant computational overhead, leading to increased latency in security protocols such as TLS, SSH, and IPsec [37–41]. This is particularly challenging in IoT devices and embedded systems. PQC algorithms are inherently more complex than traditional cryptography due to the nature of the mathematical problems they solve to ensure quantum resistance, as follows:
- **Algorithm Complexity**: Lattice-based cryptographic systems, one of the leading candidates for PQC [42], require operations on large, high-dimensional lattices. These operations are computationally intensive because they involve complex algebraic structures and operations that are

Table 1. Comparative summary of classical and post-quantum cryptographic algorithm performance across platforms and protocols. Pub: public key; CT: ciphertext; Sig: signature; SK: secret key; DS: digital signature; RRT: Round-Trip Time

Protocol	Algorithm(s)	Architecture	Execution Time (ms)	Handshake Latency (ms)	Parameter Size (B)
TLS 1.3 [1]	ECDH	x86	0.39–5.85	72.59	Pub: 32
TLS 1.3 [1,20,37]	Kyber	x86, ARM	0.23–2.57	73.72	Pub: 699–5122; CT: 699–5193; SK: 881–4864
TLS 1.3 [1,20,39]	NTRU / HRSS	x86, ARM	0.79–11.79	≈74	Pub: 5279–8453
TLS 1.3 [20]	BIKE	ARM Cortex-A53	0.27–11.6	2.8	Pub: 699–5122
TLS 1.3 [37]	FrodoKEM	x86	4.041	RTT-sensitive	Pub: 33–14880
TLS 1.3 [37]	SIKE	x86	∼60–120	Slower than ECDH	Pub: 32–9720
TLS 1.3 [1,20]	Dilithium	x86, ARM	0.54–3.2	73.59	Sig: 64–35664
TLS 1.3 [20]	Falcon	x86, ARM	0.22–10	Varies	Sig: 64–34036
TLS 1.3 [1,38]	SPHINCS+	x86, ARM, RPi3, ESP32	17.91–31.13	Adds certs overhead	Sig: up to 1,405,000
SSH [1]	ECDH+RSA	x86	≈0.5	583.87	Pub: 32
SSH [1]	ECDH+Kyber	x86	0.23–0.35	680.28	Pub: 699–5122
SSH [1,4]	Kyber+SPHINCS+	x86, ARM	0.23–0.35 / 17.91	755.89	Sig: up to 35664
SSH [4]	Dilithium	x86, ARM	0.54–3.2	Adds latency in auth	Sig: 64–35664
IPsec [39]	Kyber	x86	0.08–0.13	4.56–6.6	5817–10521
IPsec [39]	NTRU-HPS	x86	0.11–0.22	4.27–6.41	5279–8453
IPsec [39]	Saber	x86	0.09–0.18	4.41–6.31	5337–9465
IPsec [39]	RLizard	x86	0.16–0.40	7.59–12.83	19641–50457
IPsec [39]	LAC	x86	0.17–0.46	4.65–6.79	5817–10425
IPsec [39]	AKCN	x86	0.22–0.42	5.67–7.35	7449–12444
IPsec [56]	Dilithium	x86	0.54–3.2	IKE_AUTH overhead	Sig: 64–35664
IPsec [56]	Falcon	x86	0.22–10	Adds signature overhead	Sig: 64–34036

not typically found in traditional cryptographic algorithms.

- **Processing Time**: This complexity translates into longer processing times for key generation, encryption, and decryption processes [43]. In devices with limited computational resources, such as IoT devices or older infrastructure, this can lead to noticeable delays and reduced operational efficiency [44].
- **Resource Utilisation**: Increased CPU and memory usage is common with PQC, affecting device performance and power consumption [45].
- **Techniques**: Algorithm optimisation, lightweight implementations, and hardware acceleration can mitigate performance issues.

(2) **Compatibility and Interoperability Issues**:
Transitioning to PQC requires maintaining compatibility between new and legacy systems. Protocol modifications may be necessary, affecting interoperability [38, 46–48]. **Techniques**: Implementing hybrid PQC schemes, adopting gradual transition strategies, and developing modular cryptographic libraries.

(3) **Key Size and Bandwidth Requirements**: PQC algorithms often have larger key sizes, impacting data transmission efficiency, especially in bandwidth-constrained environments [2, 49, 50] as shown in Table 1.

- **Transmission Efficiency**: Many PQC schemes utilize keys that are significantly larger than those used in current cryptographic. For example, where a 2048-bit key is standard for RSA, some PQC schemes might require keys many times larger. Transmitting these larger keys over a network consumes more bandwidth [1, 38].
- **Data Overhead**: The increase in packet size due to larger keys and ciphertexts adds to the network load, which can reduce the overall throughput of the network, particularly in scenarios where bandwidth is limited or highly contended, such as in mobile networks or dense IoT deployments [51].

Techniques: Optimizing protocols to handle larger keys, and selecting algorithms suitable for low-bandwidth environments.

(4) **Security and Robustness Concerns**: Ensuring robust security against quantum and tra-

ditional attacks is crucial [39, 52–54]. Hybrid systems may introduce new vulnerabilities or new attack vectors, and cryptographic agility is needed to adapt to evolving threats.
Techniques: continuous cryptographic analysis, development of flexible frameworks for updates, and use of hybrid models during the transition.

4.2 Protocol-Specific Challenges of PQC Integration

The implementation of PQC does not affect all Internet security protocols uniformly. Each protocol faces distinct challenges when integrating post-quantum mechanisms. These challenges stem from their handshake design, transport characteristics, and reliance on different cryptographic primitives.

(1) **TLS:** TLS 1.3 handshakes are particularly sensitive to message size and certificate validation. PQC integration increases the size of public keys, ciphertexts, and digital signatures, which can cause record-layer fragmentation and longer handshake latencies. Certificate chains that incorporate PQC signatures significantly expand handshake messages, potentially leading to transmission inefficiencies and client-side memory pressure. Furthermore, hybrid KEMs in TLS require careful downgrade protection, as mismatched algorithm negotiation between servers and clients can introduce new attack surfaces.
(2) **SSH:** Unlike TLS, SSH relies on repeated key exchanges and session rekeying during long-lived connections. Post-quantum key exchange algorithms such as Kyber introduce additional latency compared to classical ECDH, while signature algorithms like SPHINCS+ substantially increase computational cost and handshake time. Since SSH is widely used in resource-constrained or embedded systems, these overheads can degrade usability. Compatibility with legacy clients that lack PQC support also poses an interoperability barrier, especially in environments where incremental migration is necessary.
(3) **IPsec:** The Internet Key Exchange Protocol (IKEv2), which underpins IPsec, introduces unique challenges for PQC adoption because it relies on UDP. Large PQC key and ciphertext sizes can cause fragmentation of the IKE_INIT message, leading to interoperability failures across middleboxes and NAT devices. Only a restricted set of parameter configurations currently avoids this issue, limiting deployment flexibility. Additionally, hybrid authentication models for IKEv2 remain underexplored, raising concerns about performance overhead, key management complexity, and cross-vendor interoperability in large-scale VPN deployments.

Table 2 includes representative PQC algorithms, their advantages and disadvantages, and is used to evaluate their implementation.

4.3 Practicality and Deployment Considerations

The practicality of deploying PQC varies depending on the protocol and its operational environment. While the challenges described in the previous section and the trade-offs summarised in Table 2 are broadly relevant, the deployment context of each protocol highlights distinct considerations.

For TLS 1.3, the primary concern is large-scale deployment in browsers and web servers, where certificate chains signed with post-quantum signatures significantly increase handshake sizes. This raises the risk of record-layer fragmentation, longer handshake latencies, and higher memory requirements on clients such as smartphones and IoT devices. Sikeridis *et al.* [1] and Paquin *et al.* [44] demonstrate that PQC-based handshakes in TLS can increase latency by up to 600% on constrained devices. Furthermore, Alnahawi *et al.* [17] highlight interoperability risks when large certificate chains are introduced, while Schwabe *text* [55] show that hybrid certificates can partially mitigate these effects. Practical deployment will therefore require parameter tuning and optimized cryptographic libraries such in Table 3.

In the case of SSH, PQC deployment is more closely tied to secure remote access in servers, cloud infrastructures, and embedded systems. Because SSH sessions often involve long-lived connections with repeated key exchanges, the performance impact of post-quantum algorithms compounds over time. Sikeridis *et al.* [1] and Crockett *et al.* [4] show that Kyber and SPHINCS+-based SSH increases handshake latency by nearly 29% and inflates rekeying costs. These overheads pose particular challenges for embedded SSH clients, as demonstrated in Bürstinghaus-Steinbach *et al.* [37]. The incremental deployment strategies using hybrid key exchanges, as suggested by [19, 43], are more practical, allowing gradual migration while maintaining compatibility with legacy clients.

For IPsec, PQC integration is especially challenging due to the reliance on UDP transport. Bae *et al.* [41] demonstrate that large ciphertexts from lattice-based KEMs can fragment IKE_INIT messages, leading to failures when traversing NAT devices or firewalls.

Table 2. Trade-off Aspects from Implementing PQC in TLS, SSH, and IPsec Protocols.

Protocols	PQC Algorithms	Advantages	Disadvantages
TLS 1.3 [1, 38, 44]	Kyber + Dilithium	High security; NIST-standardised; hybrid mode supported by OpenSSL/libOQS	Larger handshake messages; certificate chain bloat; handshake latency up to 600% on constrained devices
TLS 1.3 [37]	SPHINCS+	Stateless hash-based signatures; long-term quantum resistance	Very large signature sizes (up to 30 KB) cause record fragmentation and bandwidth overhead
TLS 1.3 [38]	FrodoKEM, SIKE	Provides forward secrecy (FrodoKEM); compact key sizes (SIKE)	FrodoKEM has high computation cost; SIKE is much slower than ECDH
TLS 1.3 [48]	Hybrid (ECC & Kyber)	Gradual migration strategy; backward compatible	Increased overhead; requires protocol extensions; downgrade resilience challenges
TLS 1.3 [55]	XMSS, Falcon	Smaller signatures than SPHINCS+; efficient verification	More complex implementation; limited deployment maturity
SSH [1, 4]	Kyber + SPHINCS+	Strong PQ handshake; backward compatibility possible via hybrid modes	Increased handshake latency (up to 29%); large signatures affect rekeying performance
SSH [4]	Dilithium	Efficient lattice-based signatures; standardised (ML-DSA)	Larger signature size than ECDSA; slower on embedded clients
SSH [4]	Hybrid (ECDH + PQC KEM)	Allows phased adoption; maintains interoperability with classical clients	Protocol modification required for hybrid KEX; performance overhead accumulates in long-lived sessions
IPsec [41]	Kyber, Saber, NTRU	Strong quantum resistance; relatively efficient KEMs; hardware acceleration possible	IKE_INIT fragmentation over UDP; bandwidth expansion up to 300%; NAT/firewall interoperability issues
IPsec [56]	Falcon, Dilithium	Standardised lattice-based digital signatures; feasible for VPN authentication	Larger key sizes/signatures add overhead in IKE_AUTH; may require protocol redesign in the authentication phase
IPsec [47, 48]	Hybrid (ECC + Kyber)	Smooth migration path; RFC 9370 supports multiple KEX in IKEv2	Protocol redesign complexity; interoperability challenges across vendors

This makes IPsec deployment in enterprise VPNs and backbone networks highly sensitive to parameter selection and message size constraints. Smyslov [48] and RFC 9370 [47] describe hybrid IKEv2 mechanisms that provide a migration path but note the complexity of protocol redesign and the need for cross-vendor coordination. In addition, Lawo [56] reports that using large post-quantum signatures in IKE_AUTH adds both latency and bandwidth overhead, requiring optimisation or redesign at the protocol level.

The practicality of PQC deployment must be evaluated not only in terms of algorithmic efficiency but also in the operational role of each protocol: TLS in browsers and content delivery, SSH in remote administration and embedded systems, and IPsec in VPNs and enterprise security gateways. These contextual differences underscore the importance of protocol-specific performance evaluations and standardisation efforts to ensure secure and efficient migration to post-quantum cryptography. Furthermore, it is necessary to analyse the efficiency of KEMs and digital signature schemes to guarantee that they comply with the practical demands of real-world applications [57].

Besides performance and key management, the resilience of PQC algorithms against different attack vectors is of utmost importance. Recent research has identified flaws in specific lattice-based schemes, specifically related to error sampling and the probability of decryption failure [67]. To tackle these vulnerabilities, ongoing studies and improvements are necessary to strengthen the resilience of PQC algorithms and ensure their ability to withstand attacks from both traditional and quantum computers [68]. Furthermore, the integration of PQC must account for the possibility of side-channel attacks, which could exploit vulnerabilities in the physical implementation of cryptographic algorithms [69].

Table 3. Comparison of PQC Libraries, Protocol Support, and Use Case Recommendations

Library Used	Internet Protocol	Hybrid Support	Use Case Recommendation	References
Open Quantum Safe (liboqs)	TLS 1.3, IPsec	Yes	Research, prototyping, and early adoption	[4, 58]
strongSwan + liboqs	IPsec	Yes	VPNs requiring quantum-safe key exchange	[59]
WolfSSL PQC Edition	TLS, IPsec	Partial	Embedded systems, IoT, lightweight applications	[60]
Bouncy Castle PQC	TLS	No	Java/C# applications, enterprise systems	[61]
Botan	TLS 1.3	Yes	C++ applications, enterprise backends	[62]
PQClean	N/A	No	Baseline for PQC implementations in other libraries	[63]
Amazon s2n-TLS	TLS 1.3	Yes	Cloud services, AWS integrations	[64]
Microsoft PQCrypto	TLS, SSH, IPsec	Yes	Azure cloud infrastructure, hybrid deployments	[65]
Mbed TLS	TLS 1.3	Partial	IoT, embedded devices, constrained environments	[66]

Another crucial factor to consider is the scalability of PQC solutions. When organizations implement PQC, they need to ensure that the solutions can efficiently handle larger amounts of data and fulfill the increasing requirements of users. The issue of scalability is especially significant in cloud computing environments, where there is a rapidly expanding demand for secure storage and processing of data [70]. The development of scalable architecture are capable of performing PQC algorithms while maintaining optimal performance is vital for achieving widespread adoption. Moreover, the interoperability of PQC with existing protocols and standards is crucial to smooth integration into current systems. Furthermore, interoperability concerns the PQC library used during integration. Post-quantum cryptographic libraries such as PQClean and libOQS are well-known for their practicality and open-source. PQClean integrates NIST-submitted algorithms into an API, offering minimal effort for future integrations, whereas libOQS is suitable for higher-level developers. Table 3 is a comparative analysis of libraries, their protocol support, hybrid capabilities, and use cases.

4.4 Future Directions for Development

The continued development and deployment of PQC requires not only algorithmic innovation but also protocol-level engineering and system integration. While NIST's finalisation of ML-KEM, ML-DSA, and SLH-DSA [36] provides a foundation for migration, several protocol-specific research gaps remain open.

For TLS 1.3, the foremost challenge is mitigating handshake latency and certificate chain bloat, which in some cases increases latency by up to 600% [1], [44], [38]. Future work should focus on optimized certificate compression, hybrid certificate chains [46], and parameter tuning to avoid record-layer fragmentation. Research into lightweight post-quantum signature schemes with smaller key and signature sizes offers promising directions [55]. Moreover, large-scale, real-world browser measurements of PQC TLS are urgently needed to complement controlled laboratory tests [44].

In SSH, the key research need is reducing cumulative rekeying overhead during long-lived sessions. PQC signatures such as SPHINCS+ currently add substantial costs [1], [4], [37]. One direction is to design hybrid SSH key exchange extensions that allow selective PQC authentication only during critical session phases, thereby balancing performance with security. Studies should also investigate lightweight lattice-based signatures in embedded SSH clients, where resource limitations are acute. Migration strategies must also address legacy compatibility, as many SSH deployments run on constrained or outdated infrastructure.

For IPsec, the most urgent issue is handling UDP fragmentation of IKE_INIT messages when large PQC keys are introduced [41]. Future development should include formal evaluations of fragmentation behaviour across middleboxes, NAT devices, and firewalls. standardisation efforts such as RFC 9370 [47] and Smyslov [48] propose hybrid IKEv2 exchanges, but more work is needed to verify interoperability across vendors and to optimize message formats. For authentication, reducing Dilithium and Falcon signature overhead in the IKE_AUTH phase [56] is essential. Hardware acceleration of lattice-based primitives

in VPN gateways could also mitigate performance costs.

Beyond protocol-specific concerns, several cross-cutting security priorities demand attention. First, hybrid PQC protocols are vulnerable to **downgrade attacks**, where adversaries attempt to force a session to fall back to classical algorithms, negating quantum resistance. Future research should rigorously analyse downgrade resilience using formal verification tools such as ProVerif and CryptoVerif [43, 67]. Second, PQC implementations remain susceptible to **side-channel attacks**, including timing, power, and cache-based leakage. Ongoing work on side-channel-resistant lattice samplers should be extended to ensure that real-world PQC deployments in TLS, SSH, and IPsec are not compromised by implementation-level weaknesses. Third, **lightweight PQC primitives** are needed for IoT and embedded systems, where CPU, memory, and energy budgets are limited. Current implementations of lattice-based KEMs and hash-based signature schemes, while promising in terms of quantum resilience, often exceed the available memory and processing capacity of microcontrollers. Research should explore parameter tuning, algorithmic simplification, and hardware-accelerated variants suitable for constrained platforms without compromising security [37, 45]. Fourth, **cross-library and cross-protocol interoperability** requires standardised APIs across libOQS, PQClean, and WolfSSL PQC to ensure consistent behaviour across deployment environments [6, 60, 63]. Finally, **long-term deployment and policy challenges** must be addressed. These include lifecycle management of post-quantum keys, migration planning for critical infrastructure, and hybrid trust models for legacy systems that cannot be upgraded immediately [3, 36]. As emphasised by Turnip et al. [19], lessons from PQC deployment in Internet protocols will be essential for shaping 6G-AKA, where cross-layer strategies must ensure privacy, downgrade resilience, and side-channel resistance across heterogeneous environments.

5 Conclusion

Our study demonstrates that adopting PQC introduces sharply different challenges across Internet security protocols. In TLS, the enlargement of certificates and signatures causes significant handshake delays, making latency the dominant bottleneck. In SSH, the reliance on repeated key exchanges and rekeying turns even moderate overhead into a substantial performance issue for long-lived sessions. In IPsec, the use of large ciphertexts creates fragmentation and bandwidth expansion, raising serious interoperability concerns for VPNs and gateways. These findings confirm that the integrating PQC cannot be approached with a uniform strategy but requires protocol-specific solutions that account for unique architectural and operational constraints. The added value of this study lies in moving beyond generic discussion by providing a comparative, protocol-level analysis that identifies where performance and security trade-offs are most critical. This perspective provides actionable guidance for system designers and standardisation bodies as they work toward building secure, efficient, and quantum-resistant Internet infrastructures.

References

[1] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis. Assessing the overhead of post-quantum cryptography in TLS 1.3 and SSH. *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*, pages 149–156, 2020.

[2] A. Pazienza, E. Lella, P. Noviello, and F. Vitulano. Analysis of network-level key exchange protocols in the post-quantum era. *2022 IEEE 15th Workshop on Low Temperature Electronics (WOLTE)*, pages 1–4, 2022.

[3] NV. Nethen, A. Wiesmaier, N. Alnahawi, and J. Henrich. PMMP - PQC migration management process. *arXiv*, 2023.

[4] E. Crockett, C. Paquin, and D. Stebila. Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH. *IACR Cryptol. ePrint Arch.*, 2019.

[5] L. Chen et al. NISTIR 8105: Report on post-quantum cryptography. Technical report, NIST, 2016.

[6] Douglas Stebila and Michele Mosca. Post-quantum key exchange for the internet and the open quantum safe project. In Roberto Avanzi and Howard Heys, editors, *Selected Areas in Cryptography (SAC) 2016*, volume 10532 of *Lecture Notes in Computer Science*, pages 1–24. Springer.

[7] Y. Baseri, V. Chouhan, and A. Hafid. Navigating quantum security risks in networked environments: A comprehensive study of quantum-safe network protocols. *Computers & Security*, 2024.

[8] J. Illiano, M. Caleffi, A. Manzalini, and AS. Cacciapuoti. Quantum internet protocol stack: A comprehensive survey. *Computer Networks, Elsevier BV*, 213, 2022.

[9] Y. Li, H. Zhang, C. Zhang, T. Huang, and F.R. Yu. A survey of quantum internet protocols from a layered perspective. *IEEE Communications Surveys & Tutorials*, 2024.

[10] A. Kumar and S. Garhwal. State-of-the-art survey of quantum cryptography. *Arch. Comput. Methods Eng.*, 2021.

[11] S. Subramani, S. M, K. A, and S.K. Svn. Review

of security methods based on classical cryptography and quantum cryptography. *J. Inf. Technol.*, 2023.

[12] M. Mehic, M. Ferrer, N.A. Malik, et al. Quantum cryptography in 5G networks: A comprehensive overview. *IEEE Commun. Surv. Tutor.*, 2023.

[13] T.M. Fernández-Caramés. From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the internet of things. *IEEE Internet of Things Journal*, 7(7):6457–6480, 2020.

[14] H. Nejatollahi, N. Dutt, S. Ray, et al. Post-quantum lattice-based cryptography implementations: a survey. *ACM Comput. Surv. (CSUR)*, 51(6):1–41, 2019.

[15] M. Durr-E-Shahwar, M.A. Imran, A.B. Altamimi, et al. Quantum cryptography for future networks security: A systematic review. *IEEE Access*, 12:24863–24890, 2024.

[16] D.T. Dam, T.H. Tran, V.P. Hoang, et al. A survey of post-quantum cryptography: Start of a new race. *Cryptography*, 7(3):40, 2023.

[17] N. Alnahawi, J. Müller, J. Oupický, and A. Wiesmaier. A comprehensive survey on post-quantum TLS. 2024.

[18] K. Dekkaki, I. Tasić, and M. Cano. Exploring post-quantum cryptography: Review and directions for the transition process. *Technologies*, 12(12):241, 2024.

[19] Togu Novriansyah Turnip, Birger Andersen, and Cesar Vargas-Rosales. Towards 6g authentication and key agreement protocol: A survey on hybrid post quantum cryptography. *IEEE Communications Surveys and Tutorials*, 2025.

[20] Maryam Abbasi, Filipe Cardoso, Paulo Váz, José Silva, and Pedro Martins. A practical performance benchmark of post-quantum cryptography across heterogeneous computing environments. *Cryptography*, 9(2), 2025.

[21] S. Paul and P. Scheible. Towards post-quantum security for cyber-physical systems: Integrating PQC into Industrial M2M communication, 2020. IACR Cryptol. ePrint Arch.

[22] K.A. Shim. A survey on post-quantum public-key signature schemes for secure vehicular communications. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14025–14042, 2022.

[23] S. Paul, F. Schick, and J. Seedorf. TPM-Based post-quantum cryptography: A case study on quantum-resistant and mutually authenticated TLS for IoT environments. *Proceedings of the 16th International Conference on Availability, Reliability and Security*, pages 1–10, 2021.

[24] W. Barker, W. Polk, and M. Souppaya. Getting ready for post-quantum cryptography: Explore challenges associated with adopting and using post-quantum cryptographic algorithms. NIST, 2021.

[25] J. Hoffstein, J. Pipher, and J.H. Silverman. NTRU: a ring-based public key cryptosystem. page 267–288, 1998.

[26] J. Bos et al. Crystals-Kyber: a CCA-secure module-lattice-based KEM. page 353–367, 2018.

[27] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, page 42–44, 1978.

[28] D. Jao and L. de Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In B.-Y. Yang, editor, *Post-Quantum Cryptography, Proc. 4th International Workshop*, page 19–34. Springer, 2011.

[29] T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In C. G. Günther, editor, *Advances in Cryptology, Proc. Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT'88)*, page 419–453. Springer.

[30] M.J. Kannwischer, A. Genêt, D. Butin, J. Krämer, and J. Buchmann. Differential power analysis of XMSS and SPHINCS. pages 168–188, 2018.

[31] SPHINCS+ Homepage. Post-quantum cryptography:SPHINCS+, 2024. Last accessed 23 Aug 2024.

[32] NIST. NIST homepage: Post-quantum cryptography, 2024.

[33] F. Opilka, M. Niemiec, M. Gagliardi, and M.A. Kourtis. Performance analysis of post-quantum cryptography algorithms for digital signature. *Applied Sciences*, 14(12):4994, 2024.

[34] D. Bernstein and T. Lange. Post-quantum cryptography, 2017.

[35] W. Beullens et al. Post-quantum cryptography: current state and quantum mitigation, 2021.

[36] NIST. Federal information processing standards (FIPS) Documents: NIST.FIPS.203, NIST.FIPS.204, NIST.FIPS.205, 2024. Last accessed 23 Aug 2024.

[37] K. Bürstinghaus-Steinbach, C. Krauß, R. Niederhagen, and M. Schneider. Post-quantum TLS on embedded systems: integrating and evaluating Kyber and SPHINCS+ with mbed TLS. page 91–96, 2020.

[38] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis. Post-quantum authentication in TLS 1.3: A performance study. 2020.

[39] J. Henrich, A. Heinemann, A. Wiesmaier, and N. Schmitt. Performance impact of PQC KEMs on TLS 1.3 under varying network characteristics. *Lecture Notes in Computer Science*, 13969:230–249, 2023.

[40] D. Marchsreiter and J. Sepúlveda. Hybrid post-quantum enhanced TLS 1.3 on embedded devices. page 643–650, 2022.

[41] J. Bae, Y. Kim, S. Lee, S. Park, D. Kim, and S. Hong. A performance evaluation of IPsec with post-quantum cryptography. page 210–228, 2023.

[42] G. Alagic et al. Status report on the second round of the NIST post-quantum cryptography standardization process. 2020.

[43] N. Bindel et al. Hybrid key encapsulation mechanisms and authentication protocols for post-quantum cryptography. *Cryptology ePrint Archive*, 2021.

[44] C. Paquin, D. Stebila, and G. Tamvada. Benchmarking post-quantum cryptography in TLS, 2020. Microsoft Research & University of Waterloo.

[45] Z. Liu, T. Pöppelmann, T. Oder, H. Seo, S. S. Roy, T. Güneysu, et al. High-performance ideal lattice-based cryptography on 8-Bit AVR Microcontrollers. *ACM Transactions on Embedded Computing Systems*, 16(4):Article 117, 2017.

[46] S. Paul, Y. Kuzovkova, N. Lahr, and R. Niederhagen. Mixed certificate chains for the transition to post-quantum authentication in TLS 1.3. page 1109–1126, 2022.

[47] C. J. Tjhai, M. Tomlinson, G. Bartlett, S. Fluhrer, D. Van Geest, and O. Garcia-Morchon. Rfc 9370: Multiple key exchanges in the internet key exchange protocol version 2 (IKEv2), 2023.

[48] V. Smyslov. Use of hybrid post-quantum key exchange in internet protocols. *Journal of Network and Systems Management*, 32(1):15–28, 2024.

[49] I. Tzinos, K. Limniotis, and N. Kolokotronis. Evaluating the performance of post-quantum secure algorithms in the TLS protocol. *Journal of Surveillance, Security and Safety*, 3(2):54–67, 2022.

[50] A. A. Giron, J. P. A. do Nascimento, R. Custódio, L. P. Perin, and V. Mateu. Post-quantum hybrid KEMTLS performance in simulated and real network environments. *Lecture Notes in Computer Science*, 13963:229–246, 2023.

[51] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange - a new hope, 2015. Cryptology ePrint Archive, Report 2015/1092.

[52] M. Sosnowski, F. Wiedner, E. Hauser, L. Steger, D. Schoinianakis, S. Gallenmüller, and G. Carle. The performance of post-quantum TLS 1.3. page 503–516, 2023.

[53] J. Zheng, H. Zhu, Y. Dong, Z. Song, Z. Zhang, Y. Yang, and Y. Zhao. Faster post-quantum TLS 1.3 based on ml-kem: Implementation and assessment. *Lecture Notes in Computer Science*, 14028:125–142, 2024.

[54] S. Fluhrer, D. McGrew, P. Kampanakis, and V. Smyslov. Post quantum preshared keys for IKEv2, 2019.

[55] P. Schwabe, D. Stebila, and T. Wiggers. Post-quantum TLS without handshake signatures. page 1461–1480, 2022.

[56] D. Lawo. Wireless and fiber-based post-quantum-cryptography-secured IPsec tunnel. *Future Internet*, 16(8):300, 2024.

[57] S. Farooq, A. Altaf, F. Iqbal, et al. Resilience optimization of post-quantum cryptography key encapsulation algorithms. *Sensors*, 23(12):5379, 2023.

[58] OpenSSL. OpenSSL documentation. OpenSSL Software Foundation, 2024. Last accessed 23 Aug 2024.

[59] strongSwan Project. *strongSwan Documentation*, 2024. Accessed: 2025-05-26.

[60] WolfSSL. Post-quantum cryptography in wolfSSL. WolfSSL Inc., 2024. Last accessed 23 Aug 2024.

[61] The Legion of the Bouncy Castle Inc. *Bouncy Castle Documentation*, 2024. Accessed: 2025-05-26.

[62] Jack Lloyd and Contributors. *Botan C++ Cryptography Library Documentation*, 2024. Accessed: 2025-05-26.

[63] Matthias J. Kannwischer, Peter Schwabe, Douglas Stebila, and Thom Wiggers. Improving software quality in cryptography standardization projects. Cryptology ePrint Archive, Paper 2022/337, 2022.

[64] Amazon Web Services. *AWS KMS – Post-Quantum TLS Support*, 2024. Accessed: 2025-05-26.

[65] Microsoft Research. Post-quantum cryptography, 2024. Accessed: 2025-05-26.

[66] ARM mbedTLS. mbedTLS documentation. ARM mbedTLS, 2024. Last accessed 23 Aug 2024.

[67] A. Khalid, C. Rafferty, J. Howe, S. Brannigan, W. Liu, and M. O'Neill. Error samplers for lattice-based cryptography - challenges, vulnerabilities, and solutions. pages 503–506, 2018.

[68] W. Liu, Z. Ni, J. Ni, et al. High performance modular multiplication for SIDH. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(10):3118–3122, 2020.

[69] D. Bellizia, N. Mrabet, A. Fournaris, et al. Post-quantum cryptography: Challenges and opportunities for robust and secure HW design, 2021. Preprint.

[70] Y. Yang, Q. Huang, and F. Chen. Secure cloud storage based on RLWE problem. *IEEE Access*,

7:27604–27614, 2019.

Togu Novriansyah Turnip received his MIM degree from the National Taiwan University of Science and Technology in 2016. He is currently pursuing his PhD degree at DTU Department of Engineering Technology and Didactics, Energy Technology and Computer Science, Denmark. He is a member of the Public Key Infrastructure (PKI) consortium and the Institute for Systems and Technologies of Information, Control and Communication (INSTICC) community. His research interests include integrating Post-Quantum Cryptography (PQC) into network security protocols, such as TLS, IPsec, SSH, and 6G-AKA, emphasizing formal verification, benchmarking, and hybrid key exchange design.

Birger Andersen is a professor (docent) at the Technical University of Denmark, Ballerup Campus (DTU). Before he was affiliated with Copenhagen University College of Engineering where he was Head of Department of Information Technology until end 2010. From 2006 he has been the head of Center for Wireless Systems and Applications (CWSA), the first research center at Copenhagen University College of Engineering, which he also founded. He holds a Master in Computer Science (1988), a BBA (1986) and a PhD in Computer Science (1992). His teaching and research interests are network security/privacy/reliability and wireless communication. From 2017, he has been coordinating two agronomic projects funded by Climate-KIC/EIT. From 2013-2015, he was heading two EU funded projects in farm management and automation. From 2006 to 2010 he has been heading a project in software defined radio in the Center for Software Defined Radio at Aalborg University.

CÉSAR VARGAS-ROSALES received the MSc and PhD degrees in electrical engineering and communications and signal processing from Louisiana State University. He is the coauthor of the book Position Location Techniques and Applications (Academic Press/Elsevier). His research interests include personal communications, 5G/6G, cognitive radio, MIMO systems, intrusion/anomaly detection in networks, localisation, interference, network and channel coding, quantum information processing, quantum communications, and optimum receiver design. He is a member of Mexican National Researchers System (SNI), Mexican Academy of Science (AMC), and the Academy of Engineering of Mexico. He is an Associate Editor of IEEE ACCESS and the International Journal of Distributed Sensor Networks. He is a Distinguished Lecturer of the IEEE Communications Society, from 2021 to 2023, the IEEE Communications Society Monterrey Chapter Chair, and the Faculty Advisor of the IEEE-HKN Lambda-Rho Chapter with Tecnologico de Monterrey. He was the Technical Program Chair of the IEEE Wireless Communications and Networking Conference (IEEE WCNC).