Impact of Post Quantum Crypto on Networking

Ashkan Hassani, Holger Kinkelin*, Filip Rezabek*
* *Chair of Network Architectures and Services*
School of Computation, Information and Technology, Technical University of Munich, Germany
Email: ashkan.hassani@tum.de, kinkelin@net.in.tum.de, rezabek@net.in.tum.de

***Abstract*—The Transport Layer Security (TLS) 1.3, IPsec, and DNSSec protocols, fundamental components of secure global communication, rely on public-key encryption, digital signatures, and fundamental exchange mechanisms. However, the advent of quantum computing poses a significant risk to these cryptographic systems, as quantum computers have the potential to break these schemes. The development and adoption of Post-Quantum Cryptography (PQC) are critical to address this emerging threat.**

This paper will discuss the transition to PQC, focusing on standardization efforts and the challenges related to networking protocols, public key infrastructure, and hardware limitations. Potential solutions, such as hybrid cryptographic systems, will also examined.

***Index Terms*—Post-Quantum Cryptography, Digital signatures, PKI, TLS, IPsec, DNSSec, certificates, Hybrid cryptography, X.509v3**

1. Introduction

Public key cryptography is the most essential part of keeping communications secure. The security of these encryption methods, like RSA, Diffie-Hellman, and elliptic curve cryptography (ECC), relies on solving complex mathematical problems that are difficult for regular computers to solve. These algorithms underpin protocols like TLS, IPsec, and DNSSec, safeguarding much of today's internet traffic [1].

However, the development of quantum computers threatens to break the mathematical foundations on which current cryptographic systems depend. Algorithms like RSA and DH would no longer be secure against quantum attacks2.

PQC, a newly designed cryptosystem, aims to stay secure even against powerful quantum computers, protecting global communication systems. The main challenge is transitioning to PQC while keeping security, efficiency, and compatibility with current infrastructure [2].

This paper examines why PQC is needed and the standardization work by **NIST**5. It also discusses how the transition to PQC affects networking protocols6, PKI systems8, and IoT devices9, focusing on performance, latency, and computational challenges. Additionally, it covers hybrid cryptographic systems7 for backward compatibility and multivariate-based cryptography for resource-limited environments.

2. The Power of Quantum Computers

Quantum information is based on qubits, analogous to bits in classical computers, but can exist in multiple states at a time, such as superposition. Qubits can become entangled, allowing quantum computers to perform many calculations in parallel. The efficiency of quantum algorithms depends on the availability and fidelity of qubits. In noisy environments, the fidelity of qubits is reduced, which can lead to incorrect calculations. Error correction is a technique to reduce such errors in quantum computations. With lower noise levels and better error-correction capabilities, quantum computers can execute quantum algorithms with the potential to break present cryptographic systems [3], [4].

We can't fully control or scale quantum computers, but we expect significant improvements soon. As these computers get more powerful and we get better at reducing errors and noise, they can run algorithms that could break today's cryptographic systems.

Two important quantum algorithms that demonstrate the power of quantum computers in this context are Shor's and Grover's algorithms. These algorithms can threaten today's cryptosystems and make them vulnerable.

2.1. Shor's Algorithm and Its Impact on Asymmetric Cryptography

In 1994, a scientist named Peter Shor introduced a quantum algorithm that allows quantum computers to solve mathematical problems such as prime factorization and discrete logarithms, which form the basis of widely used RSA, DH, and ECC.

RSA encryption and **digital signatures** are secure because factoring the product of two large prime numbers is infeasible for classical computers. Similarly, **DH key exchange** and **ECC** rely on the difficulty of solving discrete logarithms, either in a finite field or on elliptic curves. The best-known classical algorithms for both problems run in exponential time, ensuring these cryptosystems' security as long as key sizes are sufficiently large.

However, Shor's algorithm solves these problems in **polynomial time**, which means a powerful quantum computer could factor in large integers and break RSA encryption by deriving the private key from the public key. Similarly, quantum computers could break DH and ECC by solving the discrete logarithm problem in polynomial time, compromising key exchange and authentication in these systems [5].

 doi: 10.2313/NET-2025-05-1_03

2.2. Grover's Algorithm and Its Impact on Symmetric Cryptography

While Shor's algorithm severely threatens asymmetric cryptography, Grover's algorithm affects symmetric cryptography to a lesser extent.

Lov Grover introduced an algorithm that searches for an element in an unstructured database using the principle of quantum computers. This algorithm offers a quadratic speedup from $O(2^n)$ to $O(2^{(n/2)})$ over classical methods, which can also speed up brute force attacks by reducing the security of key to half its length, which speeds up the threat vulnerability of symmetric schemes such as DES and AES.

Doubling the key size can mitigate the impact of Grover's algorithm, as larger keys increase the time required for a successful attack. For example, AES-128 would offer only 64-bit security against a quantum attack, but AES-256 would still provide 128-bit security, making it an effective defense [6].

3. Impact of Quantum attacks on Networking

Key agreements and digital signatures are fundamental for secure communication and supporting protocols like TLS, SSH, IPsec, and digital certificates. Shor's algorithm threatens these public-key cryptosystems by solving the underlying mathematical problems they rely on. This would make it impossible to securely exchange symmetric keys using DH, while digital signatures generated by ECC could be reverse-engineered, exposing private keys from corresponding public keys.

In contrast, symmetric cryptography remains more resilient against quantum attacks. Grover's algorithm weakens symmetric encryption but does not fully compromise them. Recent research from the Center for Computational Quantum Physics in 2023 shows that implementing Grover's algorithm may not be as effective in practice as thought due to noise sensitivity and hardware limitation [7].

4. Urgency of Transition

Quantum computing is advancing rapidly, with record-breaking qubit counts and noise reduction and error correction improvements. While we can not predict when a cryptographically relevant quantum computer (CRQC) will be powerful enough to break current cryptographic algorithms, the uncertainty and rapid development emphasize transitioning to PQC [8]. If public key cryptography used in protocols, like TLS or IPsec, is broken by a quantum computer, it would also render all symmetric keys (used to encrypt data transmissions) vulnerable to decryption. This section investigates several attacks that become feasible with quantum computers.

One area that would be significantly affected is **DNSSec**. Since DNSSec is based on public key cryptography, all authoritative DNS name servers that want to sign their responses must first generate a public-private key pair before any response can be signed. It uses **chain of trust** to ensure that each zone's public key is validated by its parent zone. This hierarchical system of signing and verification up to the root DNS server creates a trust relationship that underpins the integrity of DNSSec.

Once quantum Computers can run cryptoanalytic attacks on public key algorithms, this entire chain of trust will be compromised, and DNSSec can no longer protect against **DNS Spoofing** attack.

The **Harvest now, decrypt later (HNDL)** attack is another particular concern. In this case, the encrypted data is collected today to decrypt in the future once quantum computers have advanced enough to break these cryptographic systems. Sensitive information, such as long-term corporate secrets, is at risk if the transition doesn't happen as soon as possible [9].

5. NIST Standardization

The NIST PQC standardization process for renewing cryptographic standards for key exchange and digital signatures, which started in December 2017, had three rounds of evaluation. Candidates were required to meet submission requirements and minimum acceptability criteria published by NIST [10]. Each round of the standardization process employed three primary evaluation criteria: security, Cost and performance, algorithm, and implementation characteristics.

For security evaluation, NIST analyzed the resistance of algorithms to side-channel attacks, perfect forward secrecy, and multi-key attacks. They also became more focused on real-world implementation and readiness for standardization.

For Cost and performance, the focus was on computational efficiency in key generation speed, memory requirements, and code size. Key considerations included side-channel resistance, constant-time performance, and memory optimization to ensure efficient real-world deployment.

They focused on efficiency and simplicity across multiple platforms for algorithm and implementation. NIST prioritized designs resistant to side-channel attacks and compatible with protocols like TLS and IPsec. The goal was to find algorithms that could be integrated with minimal performance lost [11]–[13].

Table 4. Standardized Algorithms

Old Name	New Name	Base
CRYSTALS–Kyber	**ML-KEM**	**lattice-based**
CRYSTALS–Dilithium	**ML-DSA**	**lattice-based**
FALCON	**FN-DSA**	**lattice-based**
SPHINCS+	**SLH-DSA**	**hash-based**

PQC algorithms are categorized in four categories based on the mathematical problems they use : 1) **Hash-based** 2) **Code-based** 3) **Multivariate-based** 4) **Lattice-based**

A detailed introduction about the pqc family and specifications for algorithms can be found in the corresponding FIPS standards: ML-KEM [14], ML-DSA [15], and SLH-DSA [16]. Each standard gives thorough guidelines on implementing them and what security settings to use.

 doi: 10.2313/NET-2025-05-1_03

5.1. Module Lattice-Based Key-Encapsulation Mechanism Standard

CRYSTALS–Kyber, standardized as **ML-KEM** in [14], is a PQ algorithm for key encapsulation mechanism (KEM). It enables the secure establishment of a shared secret key between two parties in a communication, which can be used for symmetric-key cryptography. This algorithm is a lattice-based cryptography, which means it's simple, efficient, and parallelizable. Its security is based on the hardness of the Module Learning with Errors (MLWE) problem, a generalization of Learning with Error (LWE). This emphasizes provable security based on the worst-case hardness of lattice problems [17], [18].

ML-KEM consists three algorithms:

1) **Key generation**: Produces public and private key
2) **Encapsulation**: Encrypts a shared secret key using the public key
3) **Decapsulation**: Decrypts a shared secret key using the private key

ML-KEM has three possible parameter sets, which make it flexible for different use cases, depending on the level of security required [14]:

1) **ML-KEM-512**
   - Encapsulation key size: **800 bytes**
   - Ciphertext size: **768 bytes**
   - Security strength: Equivalent to **AES-128**
2) **ML-KEM-768**
   - Encapsulation key size: **1,184 bytes**
   - Ciphertext size: **1,088 bytes**
   - Security strength: Equivalent to **AES-192**
3) **ML-KEM-1024**
   - Encapsulation key size: **1,568 bytes**
   - Ciphertext size: **1,472 bytes**
   - Security strength: Equivalent to **AES-256**

5.2. Module Lattice-Based Digital Signature Standard

ML-DSA is a PQ digital signature scheme based on CRYSTALS–Dilithium [15]. Digital signatures are most effective when they are bound to a specific identity. The signer must prove they own the matching private key to connect a public key to a verified identity. In PKI, this is achieved by issuing a certificate confirming the identity and the proof of private key possession. ML-DSA provides strong Security and is resistant to attacks that attempt to forge signatures by tricking the signer. This means even if an attacker tricks the signer into signing arbitrary messages; it still wouldn't be possible to create new valid signatures for other messages. This ensures that messages can not be faked or changed, which keeps the integrity and authenticity of digital signatures intact. The Security for lattice-based signatures relies on Module Learning with Errors (MLWE) and short integer solution (SIS) problems. SIS involves finding solutions to specific types of linear equations [15], [19].

ML-DSA consists of three main algorithms: 1) **key generation** 2) **Signing** 3) **Verifying**

ML-DSA comes in three different security levels, which ensure a balance between computational efficiency and cryptographic strength. This flexibility makes ML-DSA suitable for various uses, from securing digital communications to protecting long-term sensitive data [15].

1) **ML-DSA-44**
   - Public key size: **1,312 bytes**
   - Private key size: **2,560 bytes**
   - Signature size: **2,420 bytes**
2) **ML-KEM-65**
   - Public key size: **1,952 bytes**
   - Private key size: **4.032 bytes**
   - Signature size: **3,309 bytes**
3) **ML-KEM-87**
   - Public key size: **2,592 bytes**
   - Private key size: **4,896 bytes**
   - Signature size: **4,627 bytes**

5.3. Stateless Hash-Based Digital Signature Standard

SPHINCS+, standardized in [16], is a stateless hash-based digital signature scheme. It combines two key hash-based schemes:

1) **forest of random subsets (FORS)**: A few-time signature scheme [20]
2) **eXtended Merkle Signature Scheme (XMSS)**: A multi-time signature scheme [21]

An SLH-DSA is generated by first computing a randomized hash of the message. Part of this hash randomly chooses a FORS key, and the rest is signed with that key. The signature includes both the FORS signature and the data needed to verify the FORS public key. This verification data is created using the XMSS signature.

SLH-DSA also has three internal and external functions: 1) **Key generation** 2) **Signature generation** 3) **Signature verification**

The SLH-DSA key generation process requires three unique random values: **PK.seed**, **SK.seed** and **SK.prf**. The security parameter n maybe 16, 24, or 32, depending on the parameter set, which is:

1) $n = 16$
   - SLH-DSA-SHA2-128**s/f**
   - SLH-DSA-SHAKE-128**s/f**
2) $n = 24$
   - SLH-DSA-SHA2-192**s/f**
   - SLH-DSA-SHAKE-192**s/f**
3) $n = 32$
   - SLH-DSA-SHA2-256**s/f**
   - SLH-DSA-SHAKE-256**s/f**

6. Impact of transition to PQC on networking

The transition to PQC is critical for key establishment protocols like TLS, IPsec, and DNSSec. It presents several performance, efficiency, and integration challenges into existing systems.

 doi: 10.2313/NET-2025-05-1_03

6.1. Key establishment and TLS

PQC algorithms tend to have larger key sizes and signatures than classical cryptography systems, which impacts key establishment protocols.

Current systems like TLS and IPsec rely on RSA or ECDH for key exchanges, which use small keys, 32 bytes for X25519 in ECDH. However, PQC algorithms such as ML-KEM need larger key sizes, such as 1,184 bytes for client transmissions and 1,088 bytes for server transmissions. This increase can lead to performance bottlenecks, especially for network handshake latency, bandwidth requirements, and data transmission times. These performance impacts are noticeable in real-time implementation [22], [23].

6.2. DNSSec and PKI

WebPKI and DNSSec will also face difficulties. Current signatures and public keys are small enough to fit within the DNS packet's Maximum Transmission Unit (MTU). However, with PQC algorithms like ML-DSA, the size of signatures and public keys exceeds this limit, causing packet fragmentation, which increases latency and reduces network efficiency.

Recent experiments with new cryptography algorithms for DNSSec, like Falcon-512, SPHINCS+, and XMSS, have shown that while Falcon-512 performs better in terms of packet size and resolver compatibility, it still increases TCP traffic, which might strain DNS infrastructure. Larger signatures can also cause **SERVFAIL** responses in DNS because they exceed the allowed packet size limits.

For PKI, the larger public keys and signatures could slow down certificate validation and put more pressure on servers. This can cause necessary changes to the structure of certificates [24].

6.3. Impact on Hardware

In particular, the hardware that supports the current cryptosystem may not be constrained to handle the computational demands of PQ algorithms. Devices with older hardware, such as the Internet of Things (IoT), legacy systems, and embedded devices, have limited processing power, memory, and battery life, which may need to be improved for large key sizes and more complex calculations.

Also, older network infrastructures may need more power or memory to manage such large keys and complex computations, which could necessitate hardware replacements or firmware upgrades [25].

7. Hybrid cryptography

One potential solution is the use of hybrid cryptographic systems, which allow the simultaneous use of classical and post-quantum cryptography in protocols like TLS 1.3, IPsec, and DNSSec. This ensures that the resulting shared secret will be protected as long as one algorithm remains secure. It offers backward compatibility with existing systems and allows using post-quantum algorithms alongside traditional ones. Devices that support modern cryptography can default to post-quantum algorithms, while legacy devices can continue using classical algorithms until they become insecure. This flexibility allows for a gradual transition to post-quantum security, avoiding the need for an immediate system-wide switch. However, this cryptography also makes challenges about large key sizes and signatures, which can also lead to increased latency and higher bandwidth requirements [25], [26].

8. Transition Strategies for PQC in PKI

Transitioning to PQC in PKI is a complicated process, especially with X.509 certificates, since their role has been crucial in securing web communications using protocols like TLS. There are four main methods for integrating PQC into X.509v3 certificates [27]:

- **Quantum-Safe Certificates**: Replaces traditional public key with quantum-safe. This method is compatible with the current X.509v3 format, but it requires that systems trust quantum-safe algorithms. It also may not support backward compatibility with older systems.
- **Hybrid Certificates**: That combines traditional and post-quantum public keys and signatures in a single certificate, which ensures compatibility with old and new systems. However, they increase network traffic and complexity.
- **Composite Certificates**: It is similar to hybrid certificates but without using X.509. They combine multiple cryptographic algorithms into a single key or signature, requiring the attacker to break multiple algorithms simultaneously. It is suitable for high-security environments but demands more computational power.
- **Parallel Certificate Chains**: It issues two separate certificate chains, one for traditional and one for post-quantum cryptography, for the same entity. First, the traditional system is used. After switching to the new system, the post-quantum system takes over. This way, certificate sizes stay the same, but checking everything takes more time and computing power.

9. Multivariate-based Public Key Cryptography as a Solution for IoT

Multivariate-based Public Key Cryptography (MPKC) is a potential solution for IoT devices, which often operate with limited computational power and memory. The security of MPKC is based on solving multivariate systems of quadratic equations over a finite field. This problem is expected to remain secure against quantum attacks since no algorithm is currently known to solve it in polynomial time. It is fast and well-suited for devices that require quick computations. It also has a low computational overhead, and the short signature size makes it appropriate for limited bandwidth and storage applications [27].

10. Conclusion

Quantum computers are advancing quickly and could soon break the security systems we use today. Although

we do not know precisely when this will happen, we must switch to PQC. However, this change is complex and fast. It involves significant challenges in terms of performance, infrastructure, and integration.

Hybrid cryptography, which allows both classical and post-quantum algorithms, X.509v3 certificates with flexible methods for integration of PQC into existing PKI, and MPKC for IoT devices with limited resources are potential solutions. These solutions help keep security strong without needing to replace everything right away. While PQC is critical for future security, gradual implementation with compatible technologies such as blockchain or AI-driven security systems is essential.

References

[1] B. Westerbaan, "The state of the post-quantum Internet," *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 2024, https://blog.cloudflare.com/pq-2024/.

[2] S. J. Lily Chen, "Report on Post-Quantum Cryptography. NISTIR 8105," p. 15, 2016, https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf.

[3] A. Faz-Hernandez, "The Quantum Menace," 2019, https://blog.cloudflare.com/the-quantum-menace/.

[4] cloudflare, "What is quantum computing?" https://www.cloudflare.com/learning/ssl/quantum/what-is-quantum-computing/.

[5] P. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," 1999, https://fab.cba.mit.edu/classes/862.22/notes/computation/Shor-1999.pdf.

[6] e. s. mina zicu, "Threats to Modern Cryptography: Grover's Algorithm," 2020, https://www.preprints.org/manuscript/202009.0677/v1.

[7] "Grover's Algorithm Offers No Quantum Advantage," 2023, https://arxiv.org/pdf/2303.11317.

[8] "Quantum Threat Timeline Report," 2022, https://globalriskinstitute.org/publication/2022-quantum-threat-timeline-report/.

[9] "Harvest now, decrypt later," https://en.wikipedia.org/wiki/Harvest_now,_decrypt_later.

[10] "Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process," https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[11] "Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process. NISTIR 8240," 2019, https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf.

[12] "Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. NISTIR 8309," 2020, https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf.

[13] "Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. NISTIR 8413," 2022, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=934458.

[14] "Federal Information Processing Standards Publication Module-Lattice-Based Key-Encapsulation Mechanism Standard, FIPS 203," 2024, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf.

[15] "Federal Information Processing Standards Publication Module-Lattice-Based Digital Signature Standard, FIPS 204," 2024, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf.

[16] "Federal Information Processing Standards Publication Stateless Hash-Based Digital Signature Standard," 2024, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf.

[17] D. S. Adeline Langlois, "Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography," 2014, https://doi.org/10.1007/s10623-014-9938-4.

[18] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," 2005, https://doi.org/10.1145/1060590.1060603.

[19] "Short integer solution problem," https://en.wikipedia.org/wiki/Short_integer_solution_problem.

[20] VegeBun, "How do hash-based post-quantum digital signatures work?" 2022, https://research.dorahacks.io/2022/12/16/hash-based-post-quantum-signatures-2/.

[21] S. A. Huelsing, D.Butin, "XMSS: eXtended Merkle Signature Scheme," 2018, https://doi.org/10.17487/RFC8391.

[22] S. G. K. B. Muruganantham, P. Shamili, "Quantum cryptography for secured communication networks," 2020, https://core.ac.uk/download/pdf/329118533.pdf.

[23] "Post-quantum cryptography is too damn big," 2024, https://dadrian.io/blog/posts/pqc-signatures-2024/.

[24] P. van Dijk, "More PQC in PowerDNS: A DNSSEC Field Study," 2024, https://blog.powerdns.com/2024/07/15/more-pqc-in-powerdns-a-dnssec-field-study.

[25] "Design choices for post-quantum TLS," 2024, https://educatedguesswork.org/posts/pq-rollout/.

[26] B. Westerbaan, "NIST's pleasant post-quantum surprise," 2022, https://blog.cloudflare.com/nist-post-quantum-surprise/.

[27] j. W. Congli Wang, weijia Xue, "Integration of Quantum-Safe Algorithms into X.509v3 Certificates," 2009, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10176713.