Transition to Post-Quantum Cryptography in High-Performance Blockchain Networks: An Analysis of the Solana Architecture

Arthur T. Oliveira[1] **, Ana Cristina dos Santos**[1]

[1]Institute of Technology and Leadership
Av. Professor Almeida Prado, 520, Butantã – 05508-901 – São Paulo – SP – Brazil

arthur.oliveira@sou.inteli.edu.br, anacris.santos@prof.inteli.edu.br

Abstract. *The emergence of quantum computing poses an existential threat to current blockchain cryptographic primitives, particularly elliptic curve-based digital signatures. This work investigates the technical and economic implications of transitioning high-performance blockchain networks to post-quantum cryptography (PQC). We focus on the Solana network as a case study, analyzing how NIST-standardized algorithms (ML-KEM, ML-DSA, Falcon, and SPHINCS+) can be integrated into its unique architecture. Our analysis examines the mathematical foundations of lattice-based and hash-based cryptography, evaluates performance trade-offs in transaction size and throughput, and proposes implementation pathways that preserve network scalability while ensuring quantum resistance.*

1. Introduction

Blockchain technology has evolved from an experimental proposal in 2008 to a fundamental component of the global digital infrastructure. Since the publication of the Bitcoin whitepaper by Satoshi Nakamoto [1], decentralized networks have expanded to support use cases that go beyond financial transfers, including smart contracts, decentralized finance (DeFi), and asset tokenization. According to 2025 data, the crypto-asset market reached a capitalization of approximately USD 3.8 trillion, with the Solana network positioning itself as the sixth largest with USD 116 billion [2].

The security of these networks is based on public-key cryptographic primitives, particularly the Elliptic Curve Digital Signature Algorithm (ECDSA) and its variants like EdDSA. As Narayanan et al. (2016) [3] state, ”cryptography provides a mechanism to securely encode the rules of a cryptocurrency system within the system itself.” However, the advent of cryptographically relevant quantum computers (CRQCs) threatens to compromise this security foundation.

Shor’s algorithm [4], developed in 1994, demonstrated that sufficiently powerful quantum computers can solve the elliptic curve discrete logarithm problem in polynomial time, effectively breaking the security of ECDSA. For blockchain contexts, where ledger immutability is fundamental, this vulnerability is particularly critical. Malicious actors can employ ”Harvest Now, Decrypt Later” strategies, capturing encrypted data today to decrypt it in the future when CRQCs become available [5].

In response to this threat, the National Institute of Standards and Technology (NIST) began a standardization process for Post-Quantum Cryptography (PQC) algorithms in 2016. In August 2024, NIST published the first finalized standards: FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), and FIPS 205 (SLH-DSA) [6]. These algorithms are based on mathematical problems considered intractable even for quantum computers, such as Learning With Errors (LWE) in structured lattices and cryptographic hash functions.

This work investigates the technical feasibility and economic implications of transitioning high-performance blockchain networks, specifically Solana, to post-quantum cryptographic primitives. Solana was chosen as a case study due to its unique architecture based on Proof of History (PoH), processing capacity exceeding 65,000 transactions per second (TPS), and intensive use of Ed25519 for digital signatures [7].

1.1. Objectives

The specific objectives of this work are:

- To analyze the current cryptographic architecture of the Solana network, identifying components vulnerable to quantum attacks;
- To review the mathematical foundations of the NIST-standardized PQC algorithms, with a focus on ML-KEM, ML-DSA, Falcon, and SPHINCS+;
- To evaluate the impact of replacing Ed25519 with PQC algorithms on the network's throughput, transaction size, and operational costs;
- To propose migration strategies that preserve backward compatibility and minimize risks during the transition.

2. Theoretical Foundations

This section establishes the fundamental concepts necessary to understand the transition to post-quantum cryptography in blockchain. We begin with the principles of cryptographic security in decentralized networks, move on to the specific architecture of Solana, and conclude with an introduction to the challenges posed by quantum computing.

2.1. Cryptographic Primitives in Blockchain

Blockchain technology relies on three fundamental cryptographic primitives: hash functions, public-key cryptography, and digital signature schemes. According to Narayanan et al. (2016) [3], these primitives work together to ensure three essential properties: ledger immutability, transaction authentication, and distributed consensus.

2.1.1. Cryptographic Hash Functions

A cryptographic hash function $H : \{0,1\}^* \rightarrow \{0,1\}^n$ maps an input of arbitrary size to a fixed-size output n. To be considered cryptographically secure, H must satisfy three properties [8]:

1. **Pre-image resistance**: Given y, it is computationally infeasible to find x such that $H(x) = y$.

2. **Second pre-image resistance**: Given x_1, it is infeasible to find $x_2 \neq x_1$ with $H(x_1) = H(x_2)$.
3. **Collision resistance**: It is infeasible to find any pair (x_1, x_2) with $x_1 \neq x_2$ and $H(x_1) = H(x_2)$.

In Bitcoin and Solana, hash functions like SHA-256 are used to build Merkle Trees, data structures that allow for efficient verification of transaction inclusion in blocks [9].

2.1.2. Public-Key Cryptography

Public-key systems use pairs of mathematically related keys: a private key sk kept secret and a public key pk freely distributed. Security is based on the computational difficulty of deriving sk from pk. In the context of blockchain, most systems use elliptic curve cryptography (ECC), specifically the secp256k1 (Bitcoin) and Curve25519 (Solana) curves [10].

The security of ECC is based on the Elliptic Curve Discrete Logarithm Problem (ECDLP): given a point P on the curve and $Q = kP$, finding the scalar k is computationally infeasible for classical computers when the field size is sufficiently large [11, 12].

2.1.3. Digital Signature Schemes

A digital signature scheme consists of three algorithms [13]:

- KeyGen$(1^\lambda) \rightarrow (sk, pk)$: Generates a key pair with a security parameter λ.
- Sign$(sk, m) \rightarrow \sigma$: Produces a signature σ for the message m using sk.
- Verify$(pk, m, \sigma) \rightarrow \{0, 1\}$: Verifies if σ is a valid signature of m under pk.

Solana uses the Ed25519 scheme, a variant of EdDSA based on Curve25519. Bernstein et al. (2012) [14] demonstrated that Ed25519 offers approximately 128 bits of security against classical attacks, with exceptional verification speeds due to the use of Edwards curve arithmetic.

2.2. Architecture of the Solana Network

Solana represents a significant evolution in blockchain design, engineered to overcome the scalability trilemma identified by Buterin [15]: the difficulty of simultaneously achieving decentralization, security, and scalability. Yakovenko (2018) [7] introduced the concept of Proof of History (PoH) as a solution to this problem.

2.2.1. Proof of History

PoH functions as a verifiable cryptographic clock that establishes a temporal order of events before consensus. Formally, PoH is implemented as a sequential Verifiable Delay Function (VDF) [16]:

$$\text{PoH}(x, i) = H^i(x) = \underbrace{H(H(\ldots H}_{i \text{ times}}(x)\ldots)) \tag{1}$$

where H is a cryptographic hash function (SHA-256 in Solana's case) and i represents the number of iterations. The fundamental property is that each hash can only be computed after the previous one, creating a verifiable proof of elapsed time.

PoH allows the network to achieve a theoretical throughput of up to 710,000 TPS, although the practical value is limited to approximately 65,000 TPS due to hardware and network propagation constraints [17].

2.2.2. Tower BFT

Solana's consensus mechanism, Tower BFT, is an optimized variant of Practical Byzantine Fault Tolerance (PBFT) [18] that leverages PoH as a source of time. Validators vote on specific forks with an exponential lockout scheme: after voting on a fork, a validator is locked for 2^k slots, where k is the number of consecutive votes on that fork [19].

2.2.3. Cryptographic Primitives in Solana

Solana employs Ed25519 extensively for:

- Signing transactions by users
- Authenticating validators
- Deterministic address derivation (Program Derived Addresses)

Each Solana transaction contains one or more 64-byte Ed25519 signatures, which are verified in parallel by validators using GPU optimizations [20]. The efficiency of this verification is critical for the network's high throughput.

2.3. The Quantum Threat

Quantum computing is based on the principles of superposition and quantum entanglement to perform computations that would be infeasible classically. According to Nielsen and Chuang (2010) [21], a quantum computer with n qubits can represent 2^n states simultaneously, offering massive parallelism for certain types of problems.

2.3.1. Shor's Algorithm

Shor's algorithm [4] is a quantum algorithm that solves the problems of integer factorization and discrete logarithms in polynomial time. For an n-bit number, Shor's algorithm has a complexity of $O(n^3)$, compared to the sub-exponential complexity of the best known classical algorithms.

For elliptic curves, the adapted Shor's algorithm can break a 256-bit curve using approximately 2,330 logical qubits with an error rate of 10^{-3} [22]. Conservative projections suggest that CRQCs with these capabilities may be available between 2030 and 2035 [5].

2.3.2. Grover's Algorithm

Grover's algorithm [23] provides a quadratic speedup for unstructured search problems. For hash functions with an n-bit output, Grover's algorithm reduces the complexity of a pre-image search from $O(2^n)$ to $O(2^{n/2})$.

This speedup implies that hash functions offering 256 bits of classical security effectively provide 128 bits of quantum security. Consequently, to maintain equivalent security levels in the post-quantum era, it is necessary to double the output sizes of hash functions [24].

2.4. Principles of Post-Quantum Cryptography

Post-quantum cryptography seeks to develop algorithms that run on classical computers but remain secure against quantum adversaries. NIST has identified five main families of mathematical problems considered resistant to quantum attacks [25]:

1. **Lattices**: Problems based on the difficulty of finding short vectors in high-dimensional lattices.
2. **Multivariate**: Solving systems of multivariate polynomial equations.
3. **Hash-based**: Schemes based exclusively on the properties of hash functions.
4. **Isogeny-based**: Problems related to paths in graphs of isogenies of supersingular elliptic curves.

The algorithms standardized by NIST focus primarily on the first two categories. Table 1 presents the security levels defined by NIST for comparing PQC algorithms.

Table 1. NIST Security Levels for PQC Algorithms

Level	Symmetric Equivalent	Security Bits
1	AES-128 (key search)	≈ 143
2	SHA-256 (collision)	≈ 207
3	AES-192 (key search)	≈ 207
4	SHA-384 (collision)	≈ 272
5	AES-256 (key search)	≈ 272

Source: NIST (2016) [25]

The distinction between levels based on key search (odd) and collision (even) reflects the different threat models for KEMs and signature schemes, respectively. This section has established the necessary foundations to understand both the current blockchain architecture and the challenges of the post-quantum transition.

3. Literature Review of PQC Algorithms

This section presents a technical analysis of the post-quantum cryptography algorithms standardized by NIST. We focus on the schemes most relevant for blockchain applications: ML-KEM for key encapsulation, and ML-DSA, Falcon, and SPHINCS+ for digital signatures.

3.1. ML-KEM (FIPS 203)

The Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM), derived from CRYSTALS-Kyber, was standardized as FIPS 203 in August 2024 [26]. Bos et al. (2018) [27] describe Kyber as an evolution of Learning With Errors (LWE) that balances security and efficiency through structured lattices.

3.1.1. Mathematical Foundations

The security of ML-KEM is based on the Module-LWE (MLWE) problem, a generalization of Ring-LWE. Formally, the MLWE problem is defined over the quotient polynomial ring $R_q = \mathbb{Z}_q[x]/(x^n+1)$, where $n = 256$ and $q = 3329$ are fixed parameters of ML-KEM [28].

Given a module of dimension k over R_q, the MLWE problem consists of distinguishing pairs $(\mathbf{A}, \mathbf{b})$ where $\mathbf{A} \in R_q^{k \times k}$ is uniform and:

$$\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \pmod{q} \tag{2}$$

from completely random pairs, where $\mathbf{s}, \mathbf{e} \in R_q^k$ are vectors with small coefficients sampled from a discrete Gaussian distribution [29].

3.1.2. Parameter Sets

FIPS 203 specifies three parameter sets, presented in Table 2, which correspond to different NIST security levels.

Table 2. Parameters and Sizes for ML-KEM

Variant	k	NIST Level	pk (bytes)	ct (bytes)
ML-KEM-512	2	1	800	768
ML-KEM-768	3	3	1184	1088
ML-KEM-1024	4	5	1568	1568

Source: NIST FIPS 203 (2024) [26]

The choice of parameter k determines the dimension of the underlying MLWE problem. Langlois and Stehlé (2015) [28] proved that the average-case security of MLWE can be reduced to the worst-case hardness of the Shortest Vector Problem (SVP) in ideal lattices with an approximation factor of $\tilde{O}(\sqrt{nk})$.

3.1.3. Fujisaki-Okamoto Transformation

To achieve IND-CCA2 security, ML-KEM applies a variant of the Fujisaki-Okamoto (FO) transformation [30] to the underlying PKE scheme. The FO transformation converts an IND-CPA scheme into an IND-CCA2 KEM through three mechanisms [31]:

1. Random encapsulation: instead of encrypting an external message, a random message m is generated internally.
2. Key derivation: the shared secret K is derived via a hash of m and public parameters.
3. Re-encryption: during decapsulation, the receiver re-encrypts m and compares it with the received ciphertext to detect modifications.

3.2. ML-DSA (FIPS 204)

The Module-Lattice-Based Digital Signature Algorithm (ML-DSA), derived from CRYSTALS-Dilithium, was standardized as FIPS 204 [32]. Ducas et al. (2018) [33] proposed Dilithium as an efficient instantiation of the Fiat-Shamir with aborts framework.

3.2.1. Fiat-Shamir Framework

Dilithium is based on the Fiat-Shamir paradigm [34], transforming an identification protocol into a signature scheme using a cryptographic hash as a random oracle. The underlying protocol is a variant of Lyubashevsky's scheme [**?**] for lattice-based signatures.

Security is based on the Short Integer Solution (SIS) problem over modules. Given a uniform $\mathbf{A} \in R_q^{k \times l}$, the SIS problem consists of finding a short vector $\mathbf{z} \in R_q^l$ such that:

$$\mathbf{A} \cdot \mathbf{z} = \mathbf{0} \pmod{q} \tag{3}$$

with $\|\mathbf{z}\| \leq \beta$ for some bound β [**?**].

3.2.2. Parameter Sets

Table 3 presents the three parameter sets for ML-DSA, with sizes significantly larger than Ed25519 (64-byte signature).

Table 3. Parameters and Sizes for ML-DSA

Variant	(k, l)	Level	pk (bytes)	sig (bytes)
ML-DSA-44	(4,4)	2	1312	2420
ML-DSA-65	(6,5)	3	1952	3293
ML-DSA-87	(8,7)	5	2592	4595

Source: NIST FIPS 204 (2024) [32]

3.3. Falcon (NIST Round 3)

Falcon (Fast Fourier Lattice-based Compact Signatures over NTRU) was selected by NIST for offering significantly smaller signatures than Dilithium. Fouque et al. (2020) [37] describe Falcon as an implementation of the GPV framework [35] over NTRU lattices.

3.3.1. NTRU Lattices

NTRU lattices were introduced by Hoffstein, Pipher, and Silverman (1998) [36] as a special class of lattices over rings. In the context of Falcon, work is done over the quotient ring $\mathbb{Z}[x]/(\phi)$, where $\phi = x^n + 1$ with $n = 2^\kappa$ being a power of two.

The fundamental NTRU equation is given by:

$$f \cdot G - g \cdot F = q \pmod{\phi} \tag{4}$$

where $f, g, F, G \in \mathbb{Z}[x]/(\phi)$ are polynomials with short integer coefficients, and q is a prime (in Falcon, $q = 12289$) [37].

3.3.2. Fast Fourier Sampling

Falcon's central innovation is the Fast Fourier Sampling (FFS) algorithm, proposed by Ducas and Prest (2016) [38]. FFS reduces the complexity of Gaussian sampling from $O(n^2)$ to $O(n \log n)$ by exploiting the recursive structure of NTRU lattices through the Fast Fourier Transform.

Table 4 shows that Falcon offers the smallest signatures among all NIST final candidates.

Table 4. Parameters and Sizes for Falcon

Variant	n	Level	pk (bytes)	sig (bytes)
Falcon-512	512	1	897	666
Falcon-1024	1024	5	1793	1280

Source: Falcon Specification v1.2 (2020) [37]

3.4. SPHINCS+ (FIPS 205)

The Stateless Hash-Based Digital Signature Algorithm (SLH-DSA), derived from SPHINCS+, was standardized as FIPS 205 [39]. Bernstein et al. (2019) [40] present SPHINCS+ as an evolution of hash-based schemes from Lamport [41] and Merkle [42].

3.4.1. Hierarchical Architecture

SPHINCS+ uses a hypertree of height h divided into d layers, each with Merkle Trees of height h/d. The fundamental innovation is the use of Few-Time Signatures (FTS) through the FORS (Forest of Random Subsets) scheme instead of one-time schemes [43].

The security of SPHINCS+ relies exclusively on the collision and second pre-image resistance properties of hash functions. This gives the scheme a more conservative security foundation than lattice-based algorithms, at the cost of substantially larger signatures [44].

3.4.2. Performance Trade-offs

Table 5 illustrates the characteristic trade-off of SPHINCS+: large signatures but compact public keys and fast verification.

Table 5. Parameters for SPHINCS+ ("s" Variant – Small Signature)

Variant	Level	pk (bytes)	sig (bytes)
SPHINCS+-128s	1	32	7856
SPHINCS+-192s	3	48	16224
SPHINCS+-256s	5	64	29792

Source: NIST FIPS 205 (2024) [39]

3.5. Comparative Analysis for Blockchain

Table 6 provides a consolidated comparison of PQC algorithms with Ed25519, highlighting the fundamental trade-offs for blockchain applications.

Table 6. Comparison of Digital Signature Schemes

Scheme	pk	sig	Total	Factor
Ed25519	32	64	96	1.0×
Falcon-512	897	666	1563	16.3×
ML-DSA-44	1312	2420	3732	38.9×
SPHINCS+-128s	32	7856	7888	82.2×

Source: Sizes in bytes. Factor calculated as $(\text{pk} + \text{sig})$ relative to Ed25519.

It is observed that Falcon offers the best compromise between post-quantum security and size overhead. With signatures 10.4× larger than Ed25519 (666 vs 64 bytes), Falcon is significantly more compact than ML-DSA (37.8×) and SPHINCS+ (122.8×). For high-performance blockchains like Solana, where every byte directly impacts throughput and storage costs, this difference is critical [45].

The choice between algorithms depends on the application's priorities. Bindel et al. (2019) [45] argue that for systems with high signature frequency and bandwidth constraints, Falcon is preferable despite its implementation complexity. On the other hand, ML-DSA offers simpler implementations and less variability in signing times, making it suitable for contexts where latency predictability is a priority.

SPHINCS+ occupies a specific niche: applications that require the most conservative security foundation possible and can tolerate large signatures. Hülsing et al. (2013) [44] highlight that security based solely on hash properties eliminates risks associated with potential mathematical discoveries that could compromise lattice-based problems.

This review establishes that, for the specific context of high-performance blockchain, Falcon emerges as the most promising candidate to replace Ed25519 in Solana. The following sections will analyze in depth the architectural implications and mathematical foundations of this transition.

References

[1] Nakamoto, S. (2008). "Bitcoin: A Peer-to-Peer Electronic Cash System". https://bitcoin.org/bitcoin.pdf

[2] CoinMarketCap. (2025). "Cryptocurrency Prices". Accessed on: Oct 08, 2025. https://coinmarketcap.com/

[3] Narayanan, A., Bonneau, J., Felten, E., Miller, A., Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies*. Princeton University Press.

[4] Shor, P. W. (1994). "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*, p. 124-134.

[5] Mosca, M. (2018). "Cybersecurity in an Era with Quantum Computers: Will We Be Ready?" *IEEE Security & Privacy*, 16(5), 38-41.

[6] National Institute of Standards and Technology. (2024). "Post-Quantum Cryptography: FIPS Approved". Accessed on: Oct 08, 2025. https://csrc.nist.gov/news/2024/postquantum-cryptography-fips-approved

[7] Yakovenko, A. (2018). "Solana: A new architecture for a high performance blockchain v0.8.13". https://solana.com/solana-whitepaper.pdf

[8] Katz, J., Lindell, Y. (2020). *Introduction to Modern Cryptography*. 3rd ed. CRC Press.

[9] Merkle, R. C. (1980). "Protocols for Public Key Cryptosystems". In: *IEEE Symposium on Security and Privacy*, p. 122-134.

[10] Bernstein, D. J. (2006). "Curve25519: new Diffie-Hellman speed records". In: *Public Key Cryptography - PKC 2006*, Springer LNCS 3958, p. 207-228.

[11] Koblitz, N. (1987). "Elliptic curve cryptosystems". *Mathematics of Computation*, 48(177), 203-209.

[12] Miller, V. S. (1985). "Use of elliptic curves in cryptography". In: *CRYPTO 1985*, Springer LNCS 218, p. 417-426.

[13] Goldwasser, S., Micali, S., Rivest, R. L. (1988). "A digital signature scheme secure against adaptive chosen-message attacks". *SIAM Journal on Computing*, 17(2), 281-308.

[14] Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., Yang, B.-Y. (2012). "High-speed high-security signatures". *Journal of Cryptographic Engineering*, 2(2), 77-89.

[15] Buterin, V. (2017). "On Sharding Blockchains". Accessed on: Oct 08, 2025. https://github.com/ethereum/wiki/wiki/Sharding-FAQ

[16] Boneh, D., Bonneau, J., Bünz, B., Fisch, B. (2018). "Verifiable Delay Functions". In: *CRYPTO 2018*, Springer LNCS 10991, p. 757-788.

[17] Solana Labs. (2024). "Solana Network Performance Metrics". Accessed on: Oct 08, 2025. https://explorer.solana.com/

[18] Castro, M., Liskov, B. (1999). "Practical Byzantine fault tolerance". In: *OSDI 1999*, p. 173-186.

[19] Solana Labs. (2021). “Tower BFT - Solana’s High-Performance Implementation of PBFT”. https://solana.com/news/tower-bft

[20] Solana Labs. (2021). “Pipelining in Solana - The Transaction Processing Unit”. https://solana.com/news/pipelining-in-solana

[21] Nielsen, M. A., Chuang, I. L. (2010). *Quantum Computation and Quantum Information*. Cambridge University Press.

[22] Roetteler, M., Naehrig, M., Svore, K. M., Lauter, K. (2017). “Quantum resource estimates for computing elliptic curve discrete logarithms”. In: *ASIACRYPT 2017*, Springer LNCS 10625, p. 241-270.

[23] Grover, L. K. (1996). “A fast quantum mechanical algorithm for database search”. In: *STOC 1996*, p. 212-219.

[24] Bernstein, D. J., Buchmann, J., Dahmen, E. (Eds.). (2009). *Post-Quantum Cryptography*. Springer.

[25] National Institute of Standards and Technology. (2016). “Report on Post-Quantum Cryptography”. NISTIR 8105.

[26] National Institute of Standards and Technology. (2024). “FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard”.

[27] Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., Seiler, G., Stehlé, D. (2018). “CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM”. In: *IEEE EuroS&P 2018*, p. 353-367.

[28] Langlois, A., Stehlé, D. (2015). “Worst-case to average-case reductions for module lattices”. *Designs, Codes and Cryptography*, 75(3), 565-599.

[29] Lyubashevsky, V., Peikert, C., Regev, O. (2010). “On ideal lattices and learning with errors over rings”. In: *EUROCRYPT 2010*, Springer LNCS 6110, p. 1-23.

[30] Fujisaki, E., Okamoto, T. (1999). “Secure integration of asymmetric and symmetric encryption schemes”. In: *CRYPTO 1999*, Springer LNCS 1666, p. 537-554.

[31] Hofheinz, D., Hövelmanns, K., Kiltz, E. (2017). “A modular analysis of the Fujisaki-Okamoto transformation”. In: *TCC 2017*, Springer LNCS 10677, p. 341-371.

[32] National Institute of Standards and Technology. (2024). “FIPS 204: Module-Lattice-Based Digital Signature Standard”.

[33] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D. (2018). “CRYSTALS-Dilithium: A lattice-based digital signature scheme”. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1), 238-268.

[34] Fiat, A., Shamir, A. (1986). “How to prove yourself: Practical solutions to identification and signature problems”. In: *CRYPTO 1986*, Springer LNCS 435, p. 186-194.

[35] Gentry, C., Peikert, C., Vaikuntanathan, V. (2008). “Trapdoors for hard lattices and new cryptographic constructions”. In: *STOC 2008*, p. 197-206.

[36] Hoffstein, J., Pipher, J., Silverman, J. H. (1998). “NTRU: A ring-based public key cryptosystem”. In: *ANTS III*, Springer LNCS 1423, p. 267-288.

[37] Fouque, P.-A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z. (2020). "Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU". Submission to the NIST PQC Standardization Process.

[38] Ducas, L., Prest, T. (2016). "Fast Fourier-based sampling for lattices". *IACR Cryptology ePrint Archive*, 2016/046.

[39] National Institute of Standards and Technology. (2024). "FIPS 205: Stateless Hash-Based Digital Signature Standard".

[40] Bernstein, D. J., Hülsing, A., Kölbl, S., Niederhagen, R., Rijneveld, J., Schwabe, P. (2019). "The SPHINCS+ signature scheme". In: *CCS 2019*, p. 2129-2144.

[41] Lamport, L. (1979). "Constructing digital signatures from a one-way function". Technical Report CSL-98, SRI International.

[42] Merkle, R. C. (1989). "A certified digital signature". In: *CRYPTO 1989*, Springer LNCS 435, p. 218-238.

[43] Aumasson, J.-P., Bernstein, D. J., Durumeric, Z., Hülsing, A., Kölbl, S., Rijneveld, J. (2022). "The SPHINCS+ Signature Framework". Internet-Draft draft-irtf-cfrg-sphincs-plus-06.

[44] Hülsing, A., Rijneveld, J., Buchmann, J. (2013). "W-OTS+ - Shorter Signatures for Hash-Based Signature Schemes". In: *PQCrypto 2013*, Springer LNCS 7932, p. 173-188.

[45] Bindel, N., Herath, U., McKague, M., Stebila, D. (2019). "Transitioning to a quantum-resistant public key infrastructure". In: *PQCrypto 2019*, Springer LNCS 11505, p. 384-403.