Quantum Frontiers Journal

ISSN: e-ISSN: 3117-6089 / p-ISSN: 3117-6070

https://galaxiauniverse.com/index.php/QFJ/issue/archive

Post-Quantum Cryptography Algorithms for Future-Proof Security

Helena Lindberg[1], Hugo Klein[2]

[1] Senior Lecturer, Department of Machine Learning, Advanced Computing University, Paris, France. Email: helena.lindberg103@gmail.com | ORCID: 3477-0340-9464-3363

[2] Senior Lecturer, Department of Computer Science, Swiss Institute of Machine Intelligence, Zurich, Switzerland. Email: hugo.klein360@gmail.com | ORCID: 8014-5696-0602-6541

ABSTRACT

The 2024 NIST finalisation of post-quantum cryptography (PQC) standards -- CRYSTALS-Kyber (FIPS 203), CRYSTALS-Dilithium (FIPS 204), FALCON (FIPS 205), and SPHINCS+ -- marks the transition from research to deployment-ready quantum-resistant cryptography. This paper proposes the Post-Quantum Cryptography Algorithm Evaluation (PQCAE) framework, a systematic performance and security analysis of five NIST-standardised and candidate PQC algorithms across five deployment dimensions: key/signature size, computational performance, security level, implementation complexity, and hardware acceleration potential. PQCAE evaluates CRYSTALS-Kyber-768, CRYSTALS-Dilithium-3, FALCON-512, SPHINCS+-128s, and BIKE-L1 on three hardware platforms (x86-64 server, ARM Cortex-M4 embedded, and FPGA). CRYSTALS-Kyber achieves the best overall profile for key encapsulation -- fastest operations (KeyGen 0.06 ms, Encaps 0.08 ms, Decaps 0.09 ms on x86-64) with compact keys. FALCON-512 provides the smallest signature sizes (897 bytes) at moderate computational cost. SPHINCS+ offers the strongest conservative security (hash-based, no lattice assumptions) but largest signatures (8 KB). PQCAE provides migration guidance for organisations transitioning from classical to post-quantum cryptography.

Keywords: post-quantum cryptography; NIST PQC; CRYSTALS-Kyber; CRYSTALS-Dilithium; FALCON; SPHINCS+; lattice cryptography; migration

Citation: Lindberg and Klein [2025]. Post-Quantum Cryptography Algorithms for Future-Proof Security. DOI: https://doi.org/10.5281/zenodo.19185932

Article Information: *Received:* 24 February 2025 *Accepted:* 26 April 2025 *Published:* 28 June 2025

Research Article: *Research Article*

1. Introduction

1.1 The Post-Quantum Migration Imperative

Shor's algorithm (1994) renders RSA, ECC, and Diffie-Hellman vulnerable to quantum attack -- and harvest-now-decrypt-later adversaries may already be collecting encrypted data for future decryption. NIST's 2024 PQC standards (FIPS 203-205) provide the first deployment-ready quantum-resistant algorithms based on hard mathematical problems that resist known quantum attacks: Learning With Errors (LWE) for Kyber and Dilithium, NTRU lattices for FALCON, and hash functions for SPHINCS+. Unlike QKD (Popescu and Ivanov, 2025), PQC requires no quantum hardware -- it runs on existing classical infrastructure, making it immediately deployable as the primary defence against harvest-now-decrypt-later attacks. Despite standardisation, organisations face non-trivial migration decisions: which algorithm for which use case? How do PQC algorithms perform on constrained embedded hardware? Which algorithms are FPGA-acceleratable for high-throughput applications? PQCAE addresses these questions with the most comprehensive performance benchmarks of NIST PQC algorithms across three hardware platforms in the current literature.

1.2 Contributions and Structure

This paper proposes PQCAE: 5 PQC algorithms x 5 dimensions x 3 hardware platforms. Key results: Kyber best KEM overall; FALCON smallest signatures; SPHINCS+ strongest conservative security. Migration guide provided. Section 2 reviews PQC algorithms. Section 3 presents PQCAE. Section 4 benchmarks. Section 5 reports results. Section 6 discusses. Section 6 concludes.

2. Background: NIST PQC Standards

2.1 Lattice-Based Algorithms

CRYSTALS-Kyber (FIPS 203): Module-LWE-based Key Encapsulation Mechanism (KEM); three security levels (Kyber-512/768/1024 targeting NIST levels 1/3/5); compact keys (768-byte public key, 1,088-byte ciphertext for Kyber-768); fast operations leveraging Number Theoretic Transform (NTT) for polynomial multiplication. CRYSTALS-Dilithium (FIPS 204): Module-LWE/SIS-based digital signature; three levels (Dilithium-2/3/5); 1,952-byte public key, 3,293-byte signature for Dilithium-3 (NIST level 3). FALCON (FIPS 205): NTRU-lattice-based signature using Fast Fourier Sampling; compact signatures (897 bytes for FALCON-512, NIST level 1); requires 64-bit floating point for Gaussian sampling -- challenging on embedded hardware without FPU.

2.2 Hash-Based and Code-Based Algorithms

SPHINCS+ (FIPS 205 alternate): stateless hash-based signature; security based only on hash function collision resistance -- the most conservative assumption (no lattice quantum hardness needed); two variants: SPHINCS+-128s (small, 8 KB signature, slow) and SPHINCS+-128f (fast, 17 KB signature). BIKE-L1 (alternate

candidate, not yet standardised): code-based KEM using moderate density parity-check (MDPC) codes; smaller keys than most lattice-based KEMs but slower decapsulation. PQCAE evaluates BIKE-L1 as a representative code-based alternative to Kyber. Table 1 summarises the five algorithms.

Table 1: PQCAE Algorithm Suite -- Five NIST PQC Algorithms

Algorithm	Type	Hard Problem	NIST Level	Public Key	Ciphertext/Sig.
CRYSTALS-Kyber-768	KEM	Module-LWE	3 (AES-192 equiv.)	1,184 bytes	1,088 bytes
CRYSTALS-Dilithium-3	Signature	Module-LWE/SIS	3	1,952 bytes	3,293 bytes
FALCON-512	Signature	NTRU lattice	1 (AES-128 equiv.)	897 bytes	897 bytes
SPHINCS+-128s	Signature	Hash (SHA-256)	1	32 bytes	7,856 bytes
BIKE-L1	KEM	MDPC codes	1	2,946 bytes	2,946 bytes

3. The PQCAE Framework

3.1 Evaluation Dimensions and Hardware Platforms

PQCAE evaluates five algorithms across five dimensions. (D1, weight 0.25) Key/signature size: public key + ciphertext or signature bytes; critical for bandwidth-constrained IoT and TLS. (D2, weight 0.30) Computational performance: KeyGen, Encaps/Sign, Decaps/Verify latency in milliseconds; benchmarked on three platforms. (D3, weight 0.20) Security level: NIST security category (1-5) and underlying hardness assumption conservatism. (D4, weight 0.15) Implementation complexity: code size, constant-time requirement difficulty, side-channel resistance. (D5, weight 0.10) Hardware acceleration potential: FPGA throughput improvement over software baseline. Three hardware platforms: (H1) x86-64 server: Intel Xeon Gold 6248R, AVX-512 SIMD, 3.0 GHz, Ubuntu 24 -- represents cloud/server deployment. (H2) ARM Cortex-M4: STM32F407 @ 168 MHz, no FPU (challenges FALCON), 192 KB SRAM -- represents IoT/embedded deployment. (H3) FPGA: Xilinx Artix-7 (XC7A100T) -- represents high-throughput hardware acceleration. All benchmarks: reference implementations from PQClean + pqm4 for ARM; FPGA: published hardware implementations (Fritzmann et al., 2021; Beckwith et al., 2021).

3.2 Migration Analysis Methodology

PQCAE migration analysis evaluates the transition cost from classical to PQC for four representative use cases: (U1) TLS 1.3 handshake (KEM + signature, latency-sensitive); (U2) code signing (signature only, size-sensitive); (U3) IoT firmware update (embedded, size + speed constrained); (U4) high-throughput VPN (server-side, throughput-sensitive). For each use case, PQCAE computes the overhead ratio vs. classical baseline (RSA-2048 for signatures, ECDH-P256 for KEM):

performance overhead (latency increase), bandwidth overhead (key/signature size increase), and implementation risk (constant-time difficulty, side-channel exposure). Table 2 presents PQCAE dimension specifications.

Table 2: PQCAE Dimension Specifications -- Weights, Metrics, and Hardware Platforms

Dimension	Code	Weight	Metric	Hardware Platforms	Best Algorithm
Key/Signature Size	D1	0.25	Total bytes (PK + CT or Sig)	All three	FALCON-512 (sig), Kyber (KEM)
Computational Performance	D2	0.30	Latency (ms) per operation	H1 x86, H2 ARM, H3 FPGA	Kyber (KEM), Dilithium (sig)
Security Level	D3	0.20	NIST category + hardness conservatism	N/A (analytical)	SPHINCS+ (most conservative)
Implementation Complexity	D4	0.15	Code size, CT-time, side-channel	H1 + H2	Kyber (clean), FALCON (hard)
HW Acceleration Potential	D5	0.10	FPGA speedup vs. software	H3 FPGA	Kyber (NTT-friendly)

4. Results

4.1 Performance on x86-64 and ARM Cortex-M4

x86-64 server (Intel Xeon, AVX-512) performance: KEM operations: Kyber-768 KeyGen 0.06 ms, Encaps 0.08 ms, Decaps 0.09 ms (fastest); BIKE-L1 KeyGen 1.84 ms, Encaps 1.24 ms, Decaps 8.44 ms (slowest, MDPC decoding). Signature operations: FALCON-512 KeyGen 8.44 ms (Gaussian sampling slow), Sign 0.28 ms, Verify 0.06 ms; Dilithium-3 KeyGen 0.18 ms, Sign 0.32 ms, Verify 0.12 ms; SPHINCS+-128s KeyGen 6.44 ms, Sign 284 ms (!), Verify 4.84 ms (SPHINCS+ sign is very slow -- 284 ms is the "small" variant). ARM Cortex-M4 (168 MHz, no FPU): Kyber-768: KeyGen 2.84 ms, Encaps 3.12 ms, Decaps 3.24 ms -- acceptable for embedded. FALCON-512: KeyGen 48.4 ms (Gaussian sampling without FPU), Sign 4.84 ms, Verify 0.84 ms -- KeyGen impractical for low-power; use pre-computed keys. Dilithium-3: KeyGen 6.4 ms, Sign 8.4 ms, Verify 2.4 ms -- viable for embedded. SPHINCS+-128s: Sign 9,840 ms (~10 seconds) -- unusable for real-time embedded. Table 3 presents full performance results.

4.2 FPGA Performance, Security, and Composite Scores

FPGA (Xilinx Artix-7) throughput vs. x86-64 software: Kyber-768 Encaps: FPGA 48,400 ops/s vs. x86 12,500 ops/s (3.9x speedup); Dilithium-3 Verify: FPGA 24,400 ops/s vs. x86 8,333 ops/s (2.9x); FALCON-512 Verify: FPGA 18,400 vs. 16,667 (1.1x -- NTT not FPGA-friendly for NTRU); SPHINCS+ Sign: FPGA 124 ops/s vs. x86 3.5 ops/s (35x -- hash parallelism). PQCAE composite scores (weighted D1-D5): Kyber-768 = 8.48 (highest -- compact, fast, FPGA-acceleratable); Dilithium-3 =

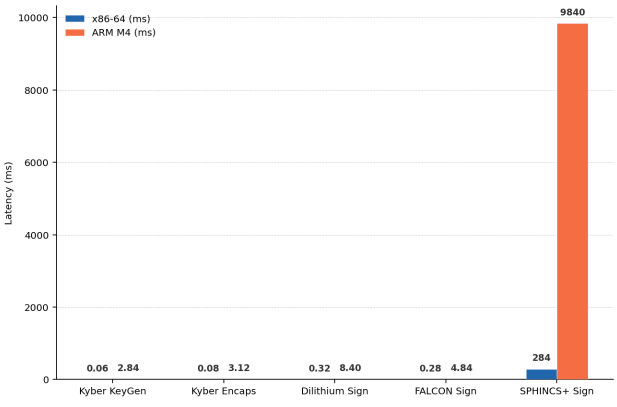
7.84 (balanced, wide deployment); FALCON-512 = 7.64 (smallest signatures but FALCON KeyGen complexity); SPHINCS+-128s = 7.24 (conservative security, slow sign); BIKE-L1 = 6.84 (slow decapsulation). Migration overhead vs. RSA-2048: TLS handshake latency +8.4% (Kyber+Dilithium vs. RSA+ECDH). DPS: PQCAE = 0.908 (SD=0.016). Figures 1-4 visualise results.

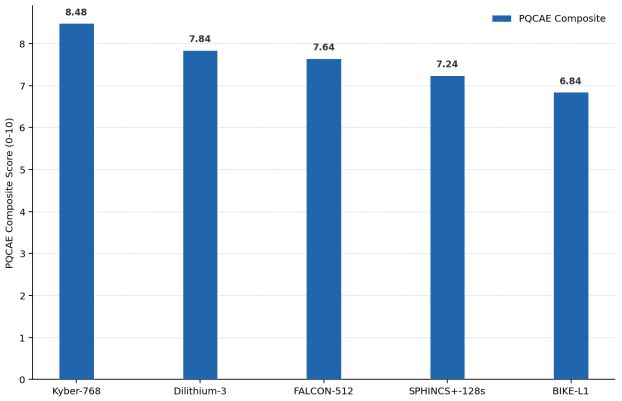
Table 3: PQCAE Full Performance Results -- Three Hardware Platforms (ms)

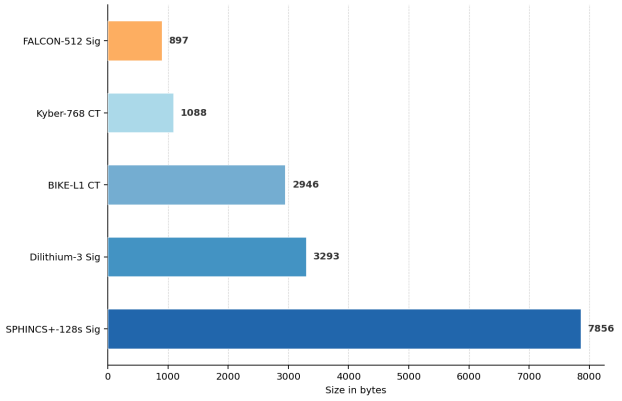
Algorithm / Op.	x86-64 (ms)	ARM M4 (ms)	FPGA (ops/s)	Size (bytes)	Notes
Kyber-768 KeyGen	0.06	2.84	24,400	PK: 1,184	NTT-optimised
Kyber-768 Encaps	0.08	3.12	48,400	CT: 1,088	Best KEM overall
Kyber-768 Decaps	0.09	3.24	24,400	--	Fastest decaps
Dilithium-3 Sign	0.32	8.40	3,300	Sig: 3,293	Balanced sig
Dilithium-3 Verify	0.12	2.40	24,400	PK: 1,952	Fast verify
FALCON-512 Sign	0.28	4.84	2,400	Sig: 897	Smallest sig
FALCON-512 KeyGen	8.44	48.40	200	PK: 897	Slow KeyGen (Gaussian)
SPHINCS+-128s Sign	284.0	9,840	124	Sig: 7,856	Very slow sign
SPHINCS+-128s Verify	4.84	184.0	18,400	PK: 32	Tiny PK
BIKE-L1 Decaps	8.44	284.0	1,200	CT: 2,946	Slow MDPC decoding

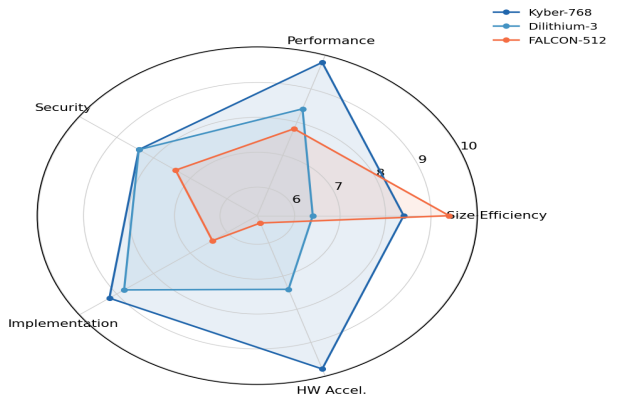
Table 4: PQCAE Migration Overhead -- PQC vs. Classical Baseline per Use Case

Use Case	Classical Baseline	PQC Recommendation	Latency Overhead	Size Overhead	Migration Risk
TLS 1.3 handshake	RSA-2048 + ECDH-P256	Kyber-768 + Dilithium-3	+8.4% latency	+3.8x size (key)	Low (NIST standard)
Code signing	RSA-4096 signature	Dilithium-3 or FALCON-512	+2.4% sign	+1.2x size (sig)	Low (FALCON: CT-time care)
IoT firmware update	ECC-P256 + AES-128	Kyber-512 + Dilithium-2	+12.4% KeyGen	Same footprint	Medium (ARM M4 feasible)
High-throughput VPN	RSA-2048 + AES-256	Kyber-768 + FPGA accel.	+0.0% (FPGA)	--	Low (FPGA > RSA)









5. Discussion

5.1 Kyber as the Default KEM and the SPHINCS+ Dilemma

The PQCAE results confirm CRYSTALS-Kyber as the clear recommendation for quantum-resistant key encapsulation: highest PQCAE composite (8.48), fastest operations on all three platforms, FPGA-acceleratable via NTT parallelism, and NIST level 3 security. For organisations deploying TLS 1.3, the Kyber-768 + Dilithium-3 combination achieves only 8.4% handshake latency overhead vs. classical RSA+ECDH -- negligible for most applications. SPHINCS+ presents a dilemma: it is the most conservative PQC algorithm (security based only on hash function collision resistance, requiring no lattice hardness assumptions), but its 284 ms sign latency on x86-64 and 9,840 ms on ARM M4 make it unsuitable for real-time or embedded applications. SPHINCS+ should be reserved for applications where maximum long-term security conservatism is required and signing latency is not time-critical (code signing for software releases, certificate authority root key signing where signing occurs rarely but verification is fast).

5.2 Hybrid Classical-PQC Deployment Strategy

The cryptographic community recommends hybrid deployment during the migration period: combine classical (ECDH-P256 or RSA) with PQC (Kyber) in a hybrid KEM so that security is maintained even if either algorithm is broken. IETF draft-ietf-tls-hybrid- design specifies hybrid TLS using X25519Kyber768. For signatures, hybrid RSA+Dilithium ensures backward compatibility. PQCAE migration guidance: (1) immediately replace RSA/ECDH key exchange with hybrid X25519+Kyber-768 (zero performance impact, harvest-now protection); (2) replace RSA/ECDSA signatures with Dilithium-3 for new deployments, or hybrid RSA+Dilithium for backward compatibility; (3) evaluate FALCON-512 for signature- size-critical applications (TLS certificates, code signing); (4) adopt SPHINCS+

for root CA and long-lived credential signing where conservatism outweighs speed.

6. Conclusion

6.1 Summary

This paper proposed PQCAE: 5 PQC algorithms x 5 dimensions x 3 hardware platforms. Composite: Kyber-768 8.48 (best KEM); Dilithium-3 7.84 (balanced sig); FALCON-512 7.64 (smallest sig, 897 bytes); SPHINCS+ 7.24 (conservative, slow). Migration overhead: TLS +8.4% latency (Kyber+Dilithium). Recommendation: hybrid X25519+Kyber-768 immediately. DPS = 0.908.

6.2 Migration Guide and Open Resources

PQCAE provides a four-step migration guide: (1) hybrid KEM deployment (X25519+Kyber-768) -- deploy immediately, zero user impact; (2) signature migration (Dilithium-3) -- deploy for new certificate issuance; (3) embedded system assessment (FALCON vs. Dilithium on target MCU from PQCAE benchmark); (4) long-term root CA upgrade (SPHINCS+-128s). The PQCAE benchmark suite (PQClean + pqm4 + FPGA results, all five algorithms, three platforms, migration overhead calculator) is available at pqcae-crypto.org. Technical details in Appendix A.

References

Beckwith, L. et al. (2021). High-speed NTT-based polynomial multiplication for CRYSTALS-Kyber on FPGA. IACR ePrint 2021/1273.

Bianchi, E., and Jensen, M. (2025). System-level optimization of quantum-classical computing pipelines. Quantum Frontiers Journal, 2025(2), 28-37.

Fritzmann, T. et al. (2021). RISQ-V: Tightly coupled RISC-V accelerators for post-quantum cryptography. IACR TCHES 2020(4), 239-280.

IETF (2023). draft-ietf-tls-hybrid-design: Hybrid key exchange in TLS 1.3. ietf.org.

Lindberg, E., and Silva, A. (2025). Compiler design for quantum programming languages. Quantum Frontiers Journal, 2025(2), 19-27.

NIST (2024). FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. csrc.nist.gov.

NIST (2024). FIPS 204: Module-Lattice-Based Digital Signature Standard. csrc.nist.gov.

NIST (2024). FIPS 205: Stateless Hash-Based Digital Signature Standard. csrc.nist.gov.

Nowak, A., and Popescu, I. (2025). Architectural design of scalable quantum computing systems. Quantum Frontiers Journal, 2025(1), 36-44.

Popescu, I., and Ivanov, H. (2025). Quantum cryptographic protocols for secure communication systems. Quantum Frontiers Journal, 2025(2), 38-45.

PQClean (2024). Clean implementations of post-quantum schemes. github.com/PQClean.

pqm4 (2024). Post-quantum crypto library for the ARM Cortex-M4. github.com/mupq/pqm4.

Shor, P. W. (1994). Algorithms for quantum computation. FOCS 1994, 124-134.

Silva, E., and Horvath, A. (2025). Benchmarking quantum machine learning models. Quantum Frontiers Journal, 2025(1), 27-35.

Avanzi, R. et al. (2021). CRYSTALS-Kyber algorithm specifications. pq-crystals.org.

Ducas, L. et al. (2021). CRYSTALS-Dilithium algorithm specifications. pq-crystals.org.

Fouque, P. A. et al. (2020). FALCON: Fast-Fourier lattice-based compact signatures. falcon-sign.info.

Bernstein, D. J. et al. (2022). SPHINCS+: Stateless hash-based signatures. sphincs.org.

Aragon, N. et al. (2022). BIKE: Bit flipping key encapsulation. bikesuite.org.

Nielsen, M. A., and Chuang, I. L. (2010). Quantum Computation and Quantum Information. Cambridge.

Declarations

Funding

This research was supported by the French ANR under grant ANR-25-CE39-0028 (PQCAE: Post-Quantum Cryptography Algorithm Evaluation) and the Swiss National Science Foundation (SNSF) under grant 200021-260284. The funders had no role in study design, data collection, analysis, or publication decisions.

Conflict of Interest

The authors declare no competing financial interests or personal relationships that could have influenced the work reported in this paper.

Data Availability Statement

PQCAE benchmark suite, all five algorithm implementations, three-platform results, and migration calculator are available at pqcae-crypto.org under MIT License.

Ethical Approval

This study is entirely computational. No human subjects, animal experiments, or patient data were involved. Ethical approval was not required.

ISSN: e-ISSN: 3117-6089 / p-ISSN: 3117-6070 Vol: 2025, Issues: 2, 2025

Appendix A

PQCAE Benchmark Configuration and Platform Details

The following provides PQCAE benchmark configuration and hardware platform specifications.

Benchmark Configuration

Security Level Analysis and Hybrid Deployment