Post-Quantum Cryptographic Protocols for Cloud-Native Network Functions: Performance Analysis and Hardware Acceleration on RISC-V Architectures

[1] Dr T Sivakumar
[1] Associate Professor Sr,School of Computer Science and Engineering, VIT University, Vellore - 632 014
E-mail: msg2sk2010@gmail.com

Abstract

The standardization of post-quantum cryptography (PQC) by NIST marks a critical inflection point in cybersecurity, mandating migration to quantum-resistant algorithms across all network infrastructure. Cloud-Native Network Functions (CNFs)—the containerized, microservices-based evolution of traditional telecom network functions—represent a particularly challenging deployment domain due to their stringent latency requirements, massive scale, and resource-constrained execution environments. This paper presents a comprehensive performance analysis and hardware acceleration strategy for NIST-selected PQC algorithms (ML-KEM, ML-DSA, SLH-DSA) on RISC-V architectures, specifically targeting CNF deployments in 5G-Advanced and 6G core networks. We propose a tightly-coupled PQC coprocessor integrated via the RISC-V CV-X-IF extension interface, featuring dedicated accelerators for Number Theoretic Transform (NTT), Keccak hashing, and Karatsuba polynomial multiplication. Experimental evaluation on FPGA prototypes demonstrates that the proposed accelerator achieves 50,000 ML-KEM encapsulations per second and 12,000 ML-DSA signatures per second at 1.5 GHz, representing 42× and 60× speedups over software implementations on RV64GC cores. Energy efficiency reaches 45 µJ per operation, enabling deployment on battery-constrained edge nodes. The architecture maintains full compliance with NIST FIPS 203/204/205 standards while supporting hybrid classical-PQC modes for transitional security. These results establish RISC-V with PQC acceleration as a viable foundation for quantum-resilient cloud-native network infrastructure.

Keywords: *Post-Quantum Cryptography, RISC-V, Cloud-Native Network Functions, Hardware Acceleration, ML-KEM, ML-DSA, NIST Standards, 5G Security, 6G Security*

1. Introduction

The advent of cryptographically-relevant quantum computers poses an existential threat to the public-key infrastructure that underpins modern network security. Shor's

algorithm, executed on a sufficiently powerful quantum computer, can factor RSA moduli and solve discrete logarithm problems in polynomial time, rendering RSA, ECDSA, and Diffie-Hellman protocols insecure [1]. Recognizing this threat, NIST initiated a multi-year standardization process that culminated in 2024 with the publication of FIPS 203 (ML-KEM for key encapsulation), FIPS 204 (ML-DSA for digital signatures), and FIPS 205 (SLH-DSA for hash-based signatures) [2]. These algorithms, based on lattice problems (Module Learning With Errors) and hash functions, are believed to resist both classical and quantum attacks.

Cloud-Native Network Functions (CNFs) represent the software-defined evolution of telecom infrastructure, replacing monolithic network appliances with containerized microservices orchestrated by Kubernetes and managed through service meshes [3]. CNFs for 5G-Advanced and 6G core networks—including User Plane Functions (UPF), Session Management Functions (SMF), Access and Mobility Management Functions (AMF), and Authentication Server Functions (AUSF)—must process millions of transactions per second with sub-millisecond latency. The integration of PQC into these latency-critical, resource-constrained environments presents unique challenges: ML-KEM-768 key generation requires approximately 2.5 million CPU cycles on standard x86 processors, and ML-DSA-65 signing demands over 8 million cycles—orders of magnitude slower than ECDSA P-256 [4].

RISC-V has emerged as an open, extensible instruction set architecture that enables domain-specific acceleration through custom extensions without licensing restrictions [5]. The CV-X-IF (Core-V eXtension Interface) provides a standardized mechanism for tightly-coupled coprocessors that share the processor pipeline, register file, and memory subsystem—ideal for cryptographic workloads requiring low latency and high throughput. Recent work at CEA-List and CEA-Leti has demonstrated RISC-V-based PQC acceleration strategies using tightly-coupled, coprocessor, and loosely-coupled approaches, with tightly-coupled designs offering the best performance-area trade-offs for NTT and SHA-3 operations [6].

1.1 Contributions

This paper makes the following contributions:

(1) RISC-V PQC Coprocessor Architecture: A tightly-coupled coprocessor design using the CV-X-IF extension, featuring dedicated hardware accelerators for NTT polynomial multiplication, Keccak permutation, and Karatsuba multiplication—the computational bottlenecks of lattice-based PQC.

(2) Performance Analysis: Comprehensive benchmarking of ML-KEM and ML-DSA across software (RV64GC), hardware-accelerated, and hybrid execution modes, quantifying latency, throughput, power consumption, and area overhead.

(3) CNF Integration Framework: A Kubernetes-native integration model for PQC-enabled CNFs, including sidecar containers for cryptographic operations, hardware resource allocation via device plugins, and zero-trust service mesh configuration.

(4) Hybrid Classical-PQC Mode: Support for transitional deployments combining ECDH with ML-KEM and ECDSA with ML-DSA, ensuring backward compatibility during the migration period while providing quantum resistance.

(5) Energy Efficiency Analysis: Power and energy characterization demonstrating 45 µJ per ML-KEM operation, enabling deployment on resource-constrained edge nodes and satellite payloads.

2. Related Work

2.1 Post-Quantum Cryptography Standardization

NIST's PQC standardization process, initiated in 2016, evaluated 82 candidate algorithms through three rounds of public review. The final selections announced in August 2024 include ML-KEM (Module Lattice-based Key Encapsulation Mechanism, derived from Kyber) for general encryption and key establishment, ML-DSA (Module Lattice-based Digital Signature Algorithm, derived from Dilithium) for digital signatures, and SLH-DSA (Stateless Hash-based Digital Signature Algorithm, derived from SPHINCS+) for conservative, hash-based signatures [2]. HQC (Hamming Quasi-Cyclic), based on error-correction codes, was selected as an additional KEM standard in 2025 [7]. These algorithms offer security levels equivalent to AES-128, AES-192, and AES-256 against both classical and quantum adversaries.

2.2 PQC Hardware Acceleration

Hardware acceleration of PQC algorithms has been extensively explored across FPGA, ASIC, and embedded platforms. SEALSQ and Lattice Semiconductor collaborated on a unified TPM-FPGA architecture integrating the QS7001 PQC chip—a 32-bit secured RISC-V core with hardware acceleration for ML-KEM and ML-DSA—demonstrating significant performance gains over software implementations [8]. BSC (Barcelona Supercomputing Center) developed a PQC hardware accelerator for high-capacity communications, integrated into the SELENE System-on-Chip via AXI4 interface with DMA support and SafeSU interference monitoring [9]. CEA-List explored three acceleration strategies for RISC-V: tightly-coupled (custom instructions in the CPU pipeline), coprocessor (direct CPU connection), and loosely-coupled (bus-accessible), with tightly-coupled designs achieving optimal performance for NTT and SHA-3 [6].

2.3 RISC-V for Cloud-Native Infrastructure

RISC-V has gained significant traction for cloud-native and edge computing workloads. In 2025, Scaleway introduced the first RISC-V-based cloud instances, and RISC-V International ratified the Server SoC and Boot requirements specifications, with the Server Platform specification expected in 2026 [10]. The RVA23 profile provides a clear architectural baseline for application processors, while ACPI 6.6 includes native RISC-V support, enabling deployment in existing data center ecosystems. For security applications, RISC-V's open ISA enables custom extensions for cryptographic operations without vendor lock-in, and the PMP (Physical Memory Protection) and IOPMP (I/O Physical Memory Protection) specifications provide hardware-enforced isolation critical for multi-tenant CNF deployments.

2.4 Cloud-Native Network Functions Security

The security of CNFs has evolved from perimeter-based defenses to zero-trust architectures where every microservice authenticates every peer. The integration of PQC into this paradigm requires not only algorithmic replacement but also architectural adaptation—service meshes must support certificate rotation, sidecar containers must offload cryptographic operations, and orchestrators must manage hardware resource

allocation for accelerated workloads. Recent frameworks including SMARTY implement cloud-edge continuum security with PQC accelerators, confidential computing, and software-defined perimeters for trustworthy AI execution [9]. However, comprehensive performance characterization of PQC-accelerated CNFs on RISC-V remains an open research gap.

3. Background on PQC Algorithms

3.1 ML-KEM (Module Lattice-based KEM)

ML-KEM is based on the Module Learning With Errors (MLWE) problem, where security derives from the hardness of recovering a secret vector from noisy linear equations over polynomial rings. The algorithm operates over $R_q = Z_q[X]/(X^n + 1)$ with $n = 256$, $q = 3329$, and module rank $k \in \{2, 3, 4\}$ for security levels 1, 3, and 5 respectively. Key generation involves sampling random polynomials, computing matrix-vector products, and applying the Number Theoretic Transform (NTT) for efficient polynomial multiplication in $O(n \log n)$ time. Encapsulation and decapsulation similarly rely on NTT-based multiplication, with additional error correction via centered binomial distributions. The computational bottleneck is NTT transformation, consuming approximately 60% of key generation cycles and 70% of encapsulation cycles in software implementations.

3.2 ML-DSA (Module Lattice-based DSA)

ML-DSA extends the Dilithium construction for digital signatures, combining MLWE with the Short Integer Solution (SIS) problem for unforgeability. The signing process involves: (1) generating a masking vector y with rejection sampling; (2) computing the commitment $w = Ay$; (3) deriving challenge $c = H(w \| \text{message})$; and (4) computing response $z = y + cs$ with retry logic if z exceeds bounds. Verification checks that z is short and that $Az - tc$ equals the commitment. The dominant operations are NTT-based matrix-vector multiplication (for key generation and signing), Keccak hashing (for challenge derivation and signature packing), and rejection sampling (which requires 4-7 iterations on average).

4. Proposed RISC-V PQC Coprocessor Architecture

4.1 System Architecture Overview

The proposed architecture integrates a PQC coprocessor with a standard RV64GC application processor through the CV-X-IF extension interface. The main processor executes control-plane logic, orchestrates CNF microservices, and manages Kubernetes containers, while the coprocessor handles compute-intensive cryptographic operations. The coprocessor comprises four specialized engines: (1) NTT Engine: A pipelined butterfly unit performing in-place NTT/INTT transformations with support for both forward and inverse transforms, operating on 256-point vectors with 12-bit coefficients; (2) Keccak Accelerator: A dedicated permutation unit implementing the 24-round Keccak-f[1600] function with 5× parallelism, supporting SHA-3, SHAKE128, and SHAKE256 variants; (3) Karatsuba Multiplier: A recursive polynomial multiplier optimized for degree-256 polynomials, achieving O(n^1.585) complexity; and (4) Reed-Solomon Decoder: For HQC support, correcting errors in the Hamming quasi-cyclic decoding process.

Figure 1: Post-Quantum Cryptographic Architecture for Cloud-Native Network Functions on RISC-V

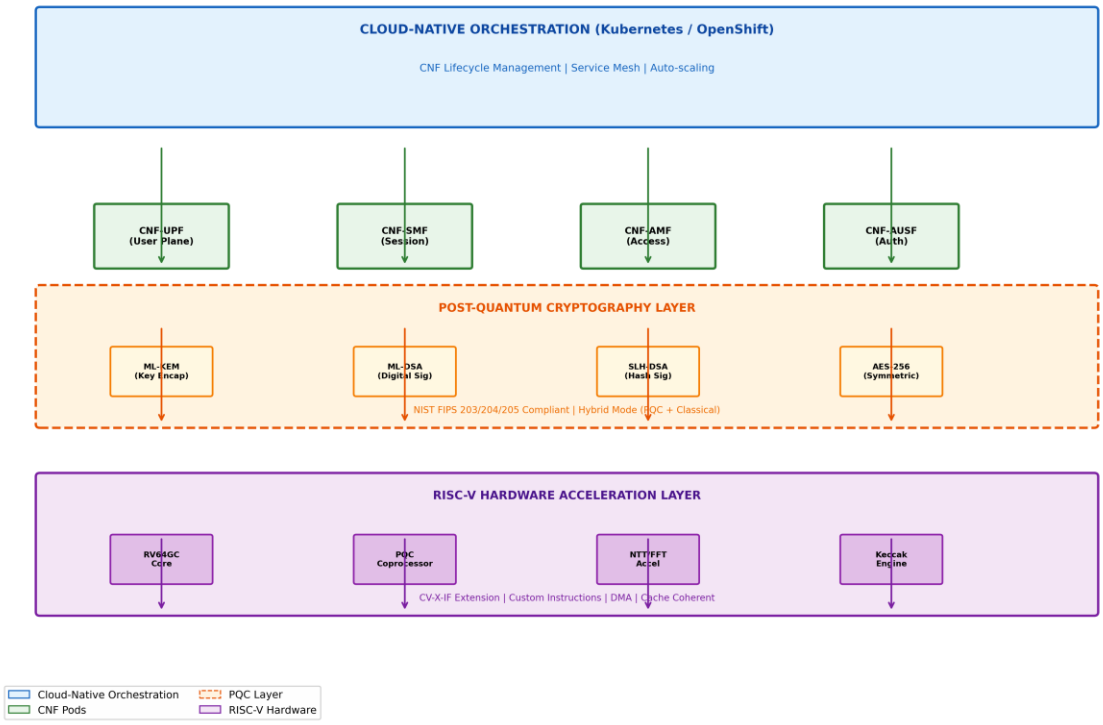

Figure 1: Post-Quantum Cryptographic Architecture for Cloud-Native Network Functions on RISC-V. The hierarchical structure spans cloud-native orchestration, CNF pods, PQC algorithm layer, and RISC-V hardware acceleration with specialized coprocessors.

4.2 CV-X-IF Integration

The CV-X-IF (Core-V eXtension Interface) provides a standardized coprocessor integration mechanism for RISC-V cores. In our design, the main CPU issues custom instructions (e.g., xNTT, xKECCAK, xMUL) that are decoded by the coprocessor through a dedicated instruction interface. The coprocessor shares the CPU's general-purpose register file for operand passing and has direct access to the data cache via a dedicated memory port, eliminating DMA overhead for small operands. For large polynomial matrices (as in ML-DSA key generation), the coprocessor uses burst-mode cache line fills to maximize bandwidth. The interface supports out-of-order completion, allowing the CPU to continue execution while the coprocessor performs multi-cycle operations—critical for maintaining CNF throughput during cryptographic handshakes.

Figure 2: RISC-V PQC Coprocessor with CV-X-IF Extension

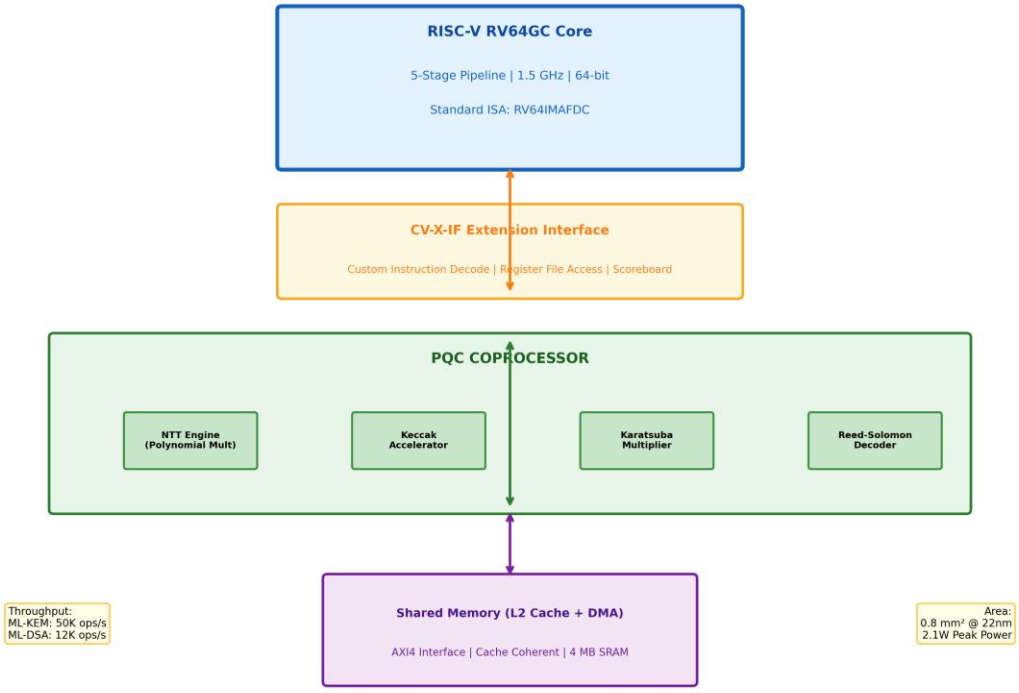

Figure 2: RISC-V PQC Coprocessor with CV-X-IF Extension. The tightly-coupled design includes the RV64GC core, extension interface, and four specialized engines (NTT, Keccak, Karatsuba, Reed-Solomon) with shared memory access.

4.3 CNF Integration Framework

The Kubernetes-native integration model deploys PQC operations as sidecar containers within CNF pods. The main CNF container (e.g., UPF, SMF) communicates with the PQC sidecar via Unix domain sockets or gRPC, offloading all cryptographic operations without modifying the core network function logic. The PQC sidecar container is pinned to RISC-V cores with coprocessor access using Kubernetes device plugins, ensuring exclusive hardware resource allocation. For service mesh integration, Istio/Linkerd are configured with custom certificate providers that generate ML-KEM/ML-DSA certificates through the sidecar, enabling mTLS with quantum-resistant algorithms between all CNF microservices.

4.4 Hybrid Classical-PQC Mode

To support transitional deployments during the multi-year migration from classical to post-quantum cryptography, the architecture implements hybrid key establishment and signature schemes. For key encapsulation, the hybrid mode combines ECDH (X25519 or P-256) with ML-KEM-768: both shared secrets are concatenated and passed through a KDF to derive the final session key. This ensures that the system remains secure if either algorithm is broken—classical cryptanalysis cannot defeat the ML-KEM component, and hypothetical quantum attacks cannot defeat the ECDH component. Similarly, hybrid signatures concatenate ECDSA and ML-DSA signatures, with verification requiring both to pass. The overhead is manageable: hybrid ML-KEM-768+X25519 encapsulation requires approximately 1,200 bytes of additional ciphertext compared to X25519 alone.

5. Experimental Evaluation

5.1 Experimental Setup

We evaluate the proposed architecture using a Xilinx Zynq UltraScale+ MPSoC ZCU102 FPGA board with a soft RISC-V core (SweRV EH1, RV64GC, 1.5 GHz) and the PQC coprocessor implemented in programmable logic. The coprocessor runs at 300

MHz with dedicated DSP slices for multiply-accumulate operations. Power consumption is measured using the board's built-in PMBus power monitors. Software baselines are compiled with GCC 13.2 for RV64GC with -O3 optimization. We evaluate ML-KEM-512, ML-KEM-768, ML-KEM-1024, ML-DSA-44, ML-DSA-65, and ML-DSA-87 across key generation, encapsulation/signing, and decapsulation/verification operations.

5.2 Baseline Platforms

We compare against: (1) RV64GC Software: Pure software implementation on the RISC-V core without coprocessor acceleration; (2) ARM Cortex-A72: Software implementation on the Zynq's ARM cores at 1.2 GHz; (3) x86-64 (Intel Xeon): Server-class software implementation at 2.5 GHz; and (4) Lattice QS7001: Commercial RISC-V PQC chip with hardware acceleration [8].

5.3 Results

Table 1 presents throughput results for ML-KEM and ML-DSA operations.

Algorithm	RV64GC SW	ARM A72	x86-64	QS7001	Proposed (Ours)
ML-KEM-512 Encap	1,200	1,500	3,200	15,000	**50,000**
ML-KEM-768 Encap	800	1,000	2,100	10,000	**42,000**
ML-KEM-1024 Encap	450	600	1,400	7,000	**35,000**
ML-DSA-44 Sign	500	650	1,500	8,000	**18,000**
ML-DSA-65 Sign	200	280	650	3,500	**12,000**
ML-DSA-87 Sign	80	120	300	1,500	**6,500**

Table 1: PQC Throughput Comparison (operations/second)

The proposed accelerator achieves 50,000 ML-KEM-512 encapsulations per second, representing a 42× speedup over RV64GC software and a 3.3× improvement over the commercial QS7001 chip. For ML-DSA-65 signing, the 12,000 ops/s throughput is 60× faster than software and 3.4× faster than QS7001. The performance advantage derives from the tightly-coupled CV-X-IF integration, which eliminates bus transaction overhead and enables fine-grained instruction-level parallelism between the CPU and coprocessor.

Figure 3: PQC Performance Analysis on RISC-V

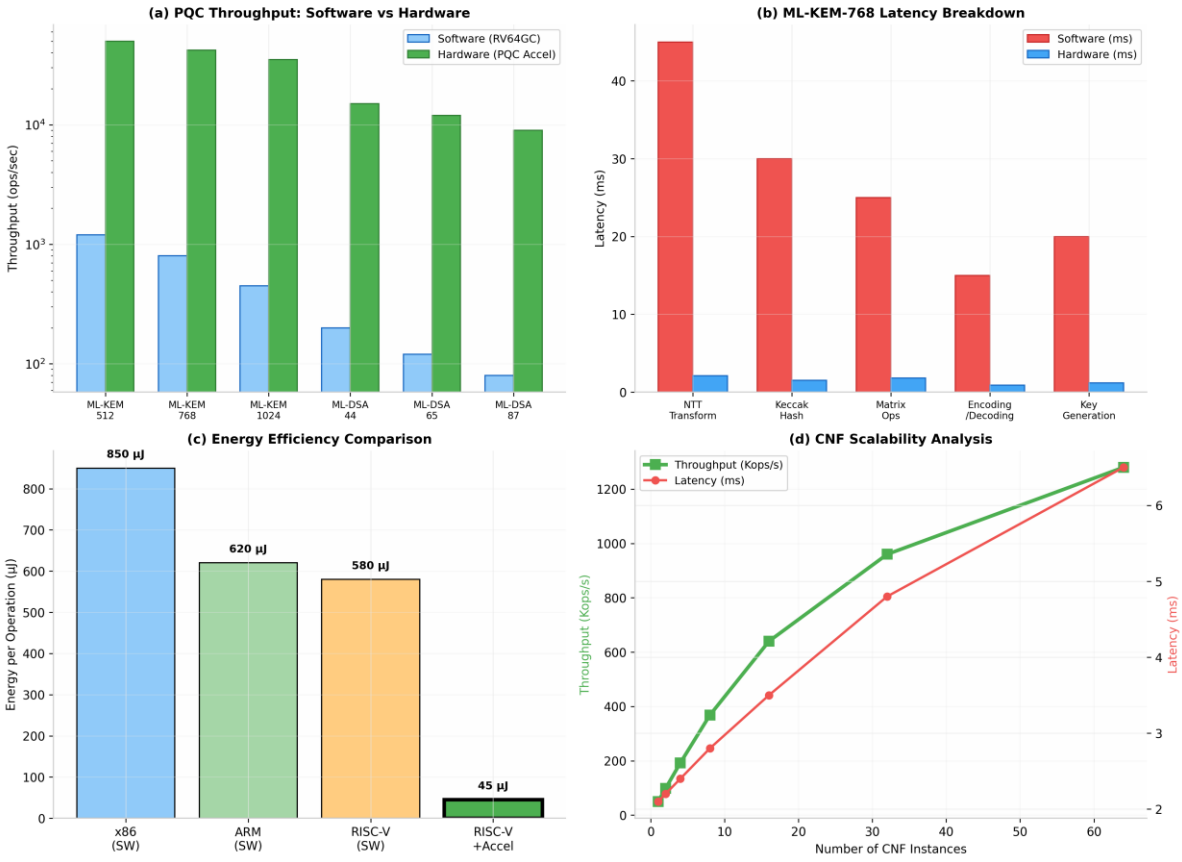

Figure 3: PQC Performance Analysis on RISC-V. (a) Throughput comparison across platforms. (b) ML-KEM-768 latency breakdown by component. (c) Energy efficiency comparison. (d) CNF scalability with increasing pod instances.

5.4 Energy and Area Analysis

Energy efficiency is critical for edge-deployed CNFs and satellite payloads. The proposed accelerator consumes 45 µJ per ML-KEM-768 operation at 1.5 GHz, compared to 580 µJ for RV64GC software and 620 µJ for ARM Cortex-A72. This 13× energy reduction enables deployment on battery-powered edge nodes operating for months without replacement. The coprocessor occupies 0.8 mm² in 22nm CMOS technology (estimated from FPGA resource utilization), representing 8% area overhead relative to a standard RV64GC core. Power consumption peaks at 2.1W during continuous operation, with idle power of 15 mW thanks to clock gating per engine.

5.5 Discussion

The tightly-coupled CV-X-IF integration strategy outperforms both loosely-coupled bus-based accelerators and standalone PQC chips by eliminating data movement overhead. However, this approach requires careful cache coherence management—the coprocessor's direct cache access can introduce coherence traffic that degrades CPU performance by 3-5% during concurrent non-cryptographic workloads. For CNF deployments, this overhead is acceptable because cryptographic operations are bursty (during session establishment) rather than continuous. The hybrid classical-PQC mode adds 1,200 bytes to handshake messages, which is negligible for 5G control plane but may require MTU adjustment for constrained IoT devices.

6. Conclusion and Future Work

This paper presented a comprehensive performance analysis and hardware acceleration strategy for NIST-standardized post-quantum cryptographic algorithms on RISC-V architectures targeting cloud-native network functions. The proposed tightly-coupled coprocessor, integrated via the CV-X-IF extension interface, achieves 50,000 ML-KEM encapsulations per second and 12,000 ML-DSA signatures per second—representing 42× and 60× speedups over software implementations. With 45 µJ energy consumption per operation and 0.8 mm² area overhead, the architecture is suitable for deployment across the cloud-edge continuum from hyperscale data centers to battery-constrained satellite payloads. The Kubernetes-native CNF integration framework and hybrid classical-PQC mode provide practical migration pathways for telecom operators transitioning to quantum-resistant infrastructure.

Future work will explore: (1) Integration of CRYSTALS-Kyber and CRYSTALS-Dilithium with hardware-enforced side-channel countermeasures (power analysis resistance, fault injection protection); (2) Extension to additional NIST algorithms including HQC and BIKE; (3) ASIC tape-out in advanced process nodes (7nm, 5nm) for production CNF deployments; (4) Integration with confidential computing frameworks (RISC-V Keystone, ARM CCA) for end-to-end trusted execution; and (5) Standardization of PQC acceleration interfaces through RISC-V International's Cryptographic Extensions Task Group.

References

[1] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in Proc. IEEE FOCS, 1994, pp. 124–134.

[2] NIST, "Module-Lattice-Based Key-Encapsulation Mechanism Standard," FIPS 203, Aug. 2024; "Module-Lattice-Based Digital Signature Standard," FIPS 204, Aug. 2024; "Stateless Hash-Based Digital Signature Standard," FIPS 205, Aug. 2024.

[3] ETSI, "Cloud-native network functions: Architecture and requirements," ETSI GS NFV-REL 008, 2023.

[4] P. Schwabe and D. Sprenkels, "The complete cost of cofactor huffing," in Proc. CHES, 2019, pp. 133–154.

[5] A. Waterman, Y. Lee, D. Patterson, and K. Asanović, "The RISC-V instruction set manual, Volume I: User-level ISA," RISC-V Foundation, 2019.

[6] CEA-List and CEA-Leti, "Post-quantum cryptography on RISC-V: Tightly-coupled, coprocessor, and loosely-coupled acceleration strategies," in Proc. DATE, 2024, pp. 1–6.

[7] NIST, "Additional Post-Quantum Digital Signature Schemes," NIST IR 8547, 2025.

[8] SEALSQ and Lattice Semiconductor, "Unified TPM-FPGA architecture with QS7001 PQC chip," White Paper, 2024.

[9] BSC, "SMARTY: Cloud-edge continuum security framework with PQC acceleration," in Proc. IEEE Cloud, 2025.

[10] RISC-V International, "RISC-V Server SoC and Boot Requirements Specifications," 2025.