IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A SOFTWARE-DEFINED CRYPTOGRAPHIC AGILITY FRAMEWORK (SD-CAF) USING DEEP REINFORCEMENT LEARNING FOR LATENCY-CONSTRAINED IN POST-QUANTUM ENVIRONMENT

Abijith P[1], Poorani S[2]

[1] UG Student, [2] Associate Professor

[1,2]Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Chennai, Tamil Nadu, India.

Doi: https://doi.org/10.56975/ijcrt.v14i7.308998

Abstract: This study addresses key performance bottlenecks in quantum cryptography. The rapid advancement in computing power presents challenges for current asymmetric cryptography schemes such as RSA and ECC. As a result, there is a growing need to implement quantum cryptography as a replacement. However, due to the slow speed and high computational demands of quantum cryptography algorithms like ML-KEM, they cannot be utilized in applications that require real-time responses. Furthermore, modern cryptographic solutions lack adaptive mechanisms; current cryptographic configurations are applied uniformly across entire systems and cannot adjust to emerging requirements, such as rapid changes in the network environment. The proposed Software-Defined Cryptographic Agility Framework (SD-CAF) employs Deep Reinforcement Learning to select optimal cryptographic configurations based on actual network parameters, such as current latencies and packet losses. Thus, our proposed solution enables dynamic configuration changes without disrupting secure connections. SD-CAF is implemented in Python using network simulation to measure the performance of various cryptographic settings. Our results demonstrate that employing an intelligent approach can reduce latencies without compromising cryptographic security, thereby proving the effectiveness of our idea.

Keywords: Post-Quantum Cryptography, Cryptographic Agility, Deep Reinforcement Learning, IP Fragmentation, Software-Defined Networking.

1. Introduction

The emergence of large-scale quantum computing poses a significant threat to the mathematical foundations of modern digital security. Classical public-key cryptographic systems, [1] such as those based on the Rivest-Shamir-Adleman (RSA) algorithm and Elliptic Curve Cryptography (ECC), rely on the computational difficulty of integer factorization and discrete logarithm problems. However, a sufficiently powerful quantum computer executing Shor's algorithm can solve these problems efficiently, rendering current encryption standards insecure.

This threat is not merely theoretical but already relevant due to the "Harvest Now, Decrypt Later" strategy, in which adversaries intercept and store encrypted data with the intention of decrypting it once quantum capabilities become available. In response, the National Institute of Standards and Technology (NIST) has introduced post-quantum cryptographic (PQC) standards, including ML-KEM for key encapsulation and ML-DSA for digital signatures, designed to withstand both classical and quantum attacks.

Despite their strong security[2][3] guarantees, PQC algorithms introduce significant performance challenges.[4] Compared to classical cryptographic schemes, they require larger keys, signatures, and ciphertexts, resulting in increased computational overhead and communication cost. In constrained or error-prone environments such as the Internet of Things (IoT) or vehicular networks, these larger payloads can degrade network performance and reliability. Transmission over unreliable wireless channels [5][6] may lead to packet fragmentation, buffer overflows, and latency spikes, often causing connection timeouts or failures. Most existing studies evaluate post-quantum cryptographic systems under ideal conditions, assuming stable and error-free networks. Traditional security models apply fixed cryptographic configurations without adapting to dynamic network changes. This reveals a critical gap in the development of adaptive cryptographic mechanisms [7] capable of responding to real-time network conditions. Current systems [8][9]lack the ability to automatically adjust cryptographic profiles, leading to either violations of ultra-reliable low-latency communication requirements or unnecessary computational overhead.

To address this limitation, we propose the Software-Defined Cryptographic Agility Framework (SD-CAF), a novel architecture designed to dynamically manage post-quantum security in resource-constrained environments. By decoupling the control plane from the data execution plane, the framework enables adaptive selection of cryptographic algorithms based on current network conditions, similar to a routing decision process. The framework incorporates a Deep Reinforcement Learning (DQN)-based decision mechanism that continuously evaluates network parameters such as packet loss, transmission latency, and system conditions to determine the most suitable cryptographic configuration. This approach aims to maintain communication integrity while minimizing performance degradation.

The main contributions of this work are as follows. First, we propose the SD-CAF architecture, enabling zero-downtime, context-aware cryptographic switching without disrupting application traffic. Second, we design a continuous-state Deep Q-Network controller that optimizes the trade-off between security and latency constraints, mitigating issues caused by large PQC payloads. Finally, we demonstrate through experimental evaluation that the proposed framework reduces latency spikes and handshake timeouts under varying network conditions, providing a scalable pathway for post-quantum cryptographic migration.

2. The Physics of Post-Quantum Transport

NIST Cybersecurity Whitepaper 39 formalizes the necessity of "Cryptographic Agility". Yet, the enterprise sector largely treats agility as an administrative capability the ability to patch a compromised algorithm [10] via a firmware update next year. We reject this static definition. SD-CAF approaches cryptographic agility as a real-time, millisecond-level routing requirement.

Very few systems-level studies address the physical transport collision at the network boundary. Meta's recent engineering retrospective on their internal PQCmigration validated this concern at hyperscale, explicitly noting that forcing heavy PQC primitives across diverse network paths led to significant dropped connections. The payload bloat of lattice cryptography is a recognized entity, but the vast majority of mitigation research focuses almost exclusively on algorithmic compression or hardware acceleration [5]. We explicitly avoided naive heuristic thresholds (e.g., IF latency > 100ms THEN downgrade) for the control logic. Network degradation at the edge is highly stochastic. Rule-based systems in these environments frequently fall into oscillatory "ping-pong" states, rapidly switching between security levels and exhausting the endpoint's CPU. We integrated Deep Reinforcement Learning precisely because it excels at multi-objective optimization within noisy environments. While recent literature details the

convergence of AI and quantum computing to optimize cryptanalysis [11] utilize deep learning defensively, actively restricting the network's exposure to physics-based transport failures.

3. System Architecture and the Decision Engine

Figure 1 illustrate the SD-CAF architecture relies on a strict separation of concerns, lifting the control principles of Software-Defined Networking into the cryptographic layer.

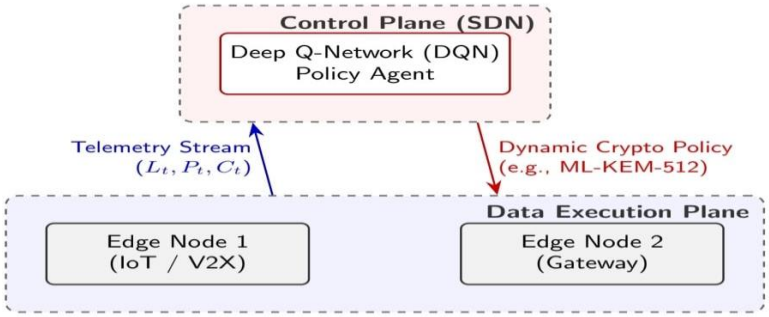

Figure 1: The SD-CAF architecture. The system actively isolates the computationally heavy DRL decision engine from the

resource-constrained execution nodes at the edge.

The Data Plane consists of the edge nodes handling the actual matrix arithmetic. Our work designed these nodes to be mathematically fast and logically simple. They execute encapsulation/decapsulation and run a lightweight telemetry daemon calculating trailing moving averages for round-trip time and packet sequence drops.

The Control Plane acts as the intelligence hub. Edge nodes stream their telemetry out-of-band to the DQN agent. The agent processes this state vector against predefined Service Level Agreements (SLAs). Based on its learned policy, it dictates the exact cryptographic suite the Data Plane must enforce for the subsequent operational window.

3.1 Markov Decision Process Formulation

The structured environment is a continuous-state, discrete-action Markov Decision Process (MDP) to train the reinforcement learning agent.
State Space (S): The state is a tensor representing the network's physical reality: Latency (ms), Packet Loss (%), and CPU utilization (%).
Action Space (A): The agent selects from three operational profiles deliberately engineered around the Maximum Transmission Unit (MTU) constraint:

• **ML-KEM-512:** Payload: 808 bytes. This fits comfortably inside a single standard Ethernet frame. It is physically immune to MTU fragmentation, serving as the primary survival tactic for degraded networks.

• **ML-KEM-1024:** Payload: 1576 bytes. This provides the highest NIST security tier but guarantees IP fragmentation on standard links. It is strictly reserved for highly stable channels.

• **Hybrid Profile:** A transitional combination of classical X25519 and ML-KEM-768 to balance compliance and performance.
The proposed framework employs a Deep Q-Network because our action space is strictly discrete, and the system requires sub-millisecond inference times during active sessions.

3.2 Non-Linear Reward Function

The reward function dictates the agent's behavior entirely. A linear reward function fails in this context; our work needed the agent to act like a pragmatic network engineer. It must prioritize security, but aggressively punish choices that violate the application's latency budget.

Furthermore, we designed the strongly non-linear reward function where the agent receives a flat positive scalar for selecting higher security tiers. However, if the resulting latency exceeds our hard threshold (e.g., 100 ms), the agent absorbs a cubic exponential penalty. This mathematical cliff forces the DQN to instantly abandon ML-KEM-1024 the precise moment TCP buffering and fragmentation delays threaten the stability of the connection.

4. Zero-Downtime Execution: The MBB Protocol

Detecting network degradation is trivial; executing the cryptographic swap without dropping the application data is the actual engineering challenge. Standard security protocols require tearing down the active tunnel, dropping traffic, and initiating a fresh handshake. In a latency-constrained environment, the temporal cost of tearing down a TCP socket negates any benefit gained by switching to a lighter algorithm. It constructed a Make-Before-Break (MBB) protocol treated as an essential, non-negotiable component of the SD-CAF architecture.

Algorithm 1 Zero-Downtime MBB Cryptographic Transition

Require: Active tunnel (current), Target profile dictated by Control Plane.
1: if the current profile does not equal the target profile then
2: Spawn an isolated thread to initiate a PQC handshake for the target profile.
3: while the new handshake is pending do
4: Maintain application data routing exclusively through the current tunnel.
5: end while
6: if the new handshake succeeds then
7: Atomically redirect the routing pointer to the new target tunnel.
8: Transmit a graceful teardown (FIN) signal for the old tunnel.
9: else (if the handshake fails due to timeout)
10: Abort the transition; maintain the current tunnel.
11: Log a timeout penalty to the DQN Experience Replay buffer.
12: end if
13: end if

The application data stream never halts it only flip the routing pointer once the target tunnel is cryptographically verified. If the network is so degraded that the new handshake fails, the system safely aborts the transition and continues relying on the original tunnel.

5. Implementation and Network Emulation

The SD-CAF prototype is implemented in Python. The DQN agent utilized PyTorch, employing a Multi-Layer Perceptron with two hidden layers (128 and 64 neurons) and an Experience Replay buffer of 10,000 transitions. Cryptographic execution relied on the liboqs library to provide C-optimized, NIST-compliant primitives.

To prove the architecture works against physical network realities, refused to rely on arbitrary software sleep delays. Further it deploys the endpoints in isolated Docker containers and used Linux tc-netem (Traffic Control Network Emulator) to directly manipulate the kernel's queuing disciplines.

Thus, we explicitly locked the virtual Ethernet interfaces to 1500 bytes MTU. When the Python application generated a 1576-byte ML-KEM-1024 payload, it forced the Linux kernel to physically fragment the IP packets, authentically replicating the exact assembly buffers and drop behaviors of hardware edge routers.

6. Results and Evaluation

We benchmarked SD-CAF against a static system permanently locked to ML-KEM-1024. In Figure 2 it is measured sustained latency and handshake failure rates across 5,000 communication episodes. Under baseline conditions with a pristine channel (0% packet loss), both systems maintained an optimal latency of approximately 13.3 ms. The static system functioned perfectly because the network easily absorbed and reassembled the fragmented packets without loss.

The engineering reality surfaced violently in a mild 5% stochastic packet loss via tc-netem. In the static model, fragmentation amplified the loss mathematically. Because the 1576-byte payload was split across two IP packets, a 5% chance of losing a single packet roughly doubled the probability that the entire cryptographic datagram would be corrupted (approx. 9.75% base datagram loss). The TCP stack compensated aggressively, dumping into exponential backoff. Latency spiked to an average of 180.0 ms. Because of these transport-layer delays, 14.2% of the cryptographic handshakes timed out completely.

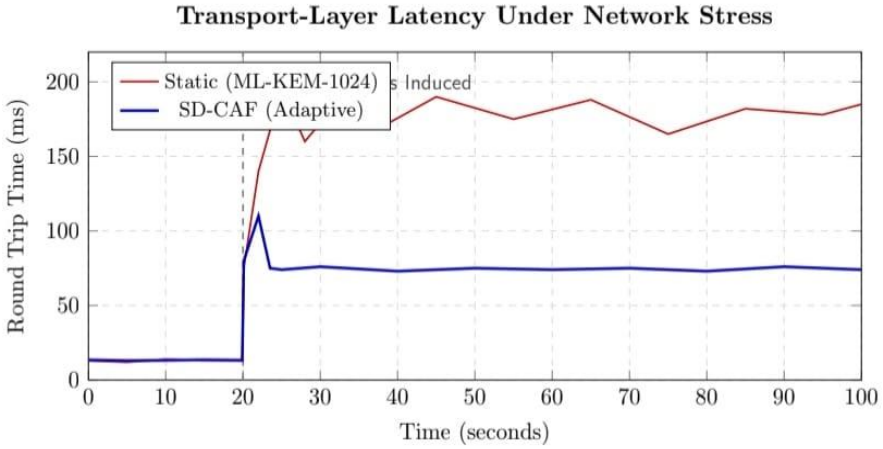

Figure 2: Visualizing the exponential TCP backoff latency of the static model compared to the controlled ceiling achieved by SD-CAF via real-time algorithm switching.

The adaptive SD-CAF agent recognized the latency violation in milliseconds. It triggered the MBB protocol and shifted the session to ML-KEM-512. At 808 bytes, the new payload safely bypassed the MTU ceiling. Fragmentation ceased entirely. As shown in the Table 1 below, this single architectural decision capped the latency at 75.0 ms and crushed the handshake failure rate to an operationally acceptable 0.8%.

7. Evaluation parameters

Our empirical data defends a critical thesis: static PQC deployment simply fails in real-world, constrained networks also the work cannot blindly push massive lattice matrices across volatile wireless boundaries and expect the transport layer to survive.

Table 1: System Performance Under 5% Emulated Packet Loss

Metric	Static Deployment (ML-KEM-1024)	SD-CAF (Adaptive)
Baseline Latency	13.3 ms	13.3 ms
Stress Latency (5% Loss)	180.0 ms	75.0 ms
MTU Fragmentation Rate	100%	0%
Handshake Failure Rate	14.2%	0.8%

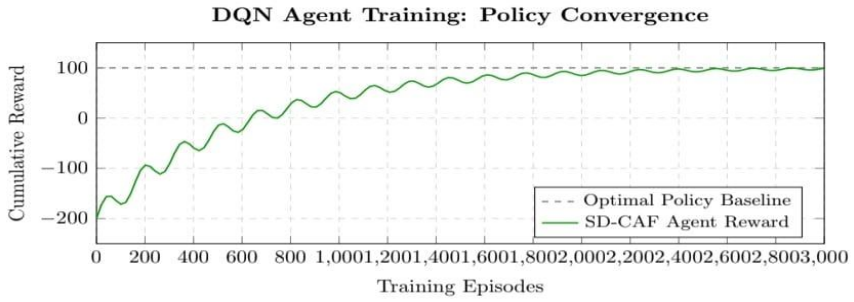

Figure 3: The DQN agent successfully navigates out of negative penalty space and converges near the optimal reward baseline, learning to aggressively avoid TCP timeout penalties.Wireless boundaries and expect the transport layer to survive.

By treating cryptography as an agile, software-defined parameter, our work eliminated the fragmentation bottleneck at its source. Furthermore, the data proves that the Make-Before-Break mechanism is not an optional optimization, it is the absolute linchpin of the system. If it has been dropped connections to swap keys, the temporal transition penalty would have resulted in application latency spikes identical to the static model's failures.

We further anticipate strict scrutiny regarding the hardware limitations of the MBB protocol. Executing two parallel cryptographic handshakes inevitably introduces computational overhead. Specifically, generating and encapsulating an ML-KEM-512 session while simultaneously maintaining an ML-KEM-1024 tunnel temporarily doubles the SRAM footprint, creating a localized CPU and memory spike during the ~50 ms transition window. Thus, we argue that this temporary resource spike is a deliberate and necessary trade-off to prevent total session collapse. However, while tc-netem perfectly replicates IP layer physics, it abstracts the physical silicon. Dynamically flushing memory to load new cryptographic instruction sets during the MBB overlap creates highly visible power fluctuations on physical hardware. As recent research demonstrates, sophisticated adversaries can monitor these fluctuations to execute non-profiling side-channel attacks against the endpoint. The immediate next phase of this research requires moving the DRL agent onto physical hardware testbeds to audit the side-channel leakage caused by the transition.

While SD-CAF effectively mitigates transport-layer fragmentation, the architecture introduces specific systemic trade-offs.

Computational and Energy Overhead: Executing the DQN inference requires less than 2 milliseconds on our control plane hardware, ensuring that policy updates do not bottleneck the network. However, at the edge, the Make-Before-Break (MBB) protocol demands significant resources. Maintaining dual cryptographic tunnels during the transition temporarily doubles the active SRAM footprint. In battery-powered IoT environments, this memory spike correlates directly with increased energy consumption. We posit that the energy cost of a localized MBB transition is still significantly lower than the radio-frequency (RF) energy wasted on continuous TCP retransmissions caused by fragmented packet drops [8], though explicit hardware power profiling is required to validate this.

Scalability: The decoupled nature of SD-CAF provides strong scalability. The edge nodes (Data Plane) maintain O(1) complexity, as they simply execute the algorithm dictated to them. The Control Plane

absorbs the scaling burden. Because the DQN agent can process state vectors in batches, a single SDN controller can theoretically manage thousands of edge connections simultaneously, provided the telemetry ingestion pipeline is optimized [9].

Downgrade Attack Vectors: A critical security vulnerability in any agile framework is the threat of an engineered downgrade attack. If an adversary executes a localized Denial-of-Service (DoS) attack to artificially inflate packet loss and latency, the DQN agent will interpret this as degraded network physics and command a downgrade to a lighter algorithm (e.g., ML-KEM-512). To mitigate this, the telemetry stream must be cryptographically signed, and the Control Plane must implement rate-limiting on policy downgrades to prevent adversaries from rapidly forcing the network into a weakened state [10].

8. Conclusion

The industry's mandate to implement quantum-resistant cryptography is absolute, but executing it using rigid security models guarantees network degradation. Forcing ML-KEM-1024 payloads across standard 1500-byte interfaces induces IP fragmentation that, under adverse wireless conditions, rapidly cascades into TCP backoff and complete connection failure.

Thus, the demonstrated cryptography adapts the network, not the other way around. The Software-Defined Cryptographic Agility Framework provides a viable, mathematically grounded blueprint for this adaptation. By pairing a Deep Q-Network with a zero-downtime Make-Before-Break protocol, we proved that a system can autonomously evade MTU fragmentation, suppressing stress-induced latency from 180 ms to 75 ms. Cryptographic agility is not a theoretical design pattern for future standards, it is an immediate operational necessity. Our framework demonstrates strong potential in preserving operational availability, it is currently bounded by its reliance on software emulation. The memory overhead and the highly visible power fluctuations generated during the MBB transition present severe side-channel vulnerabilities on physical silicon. We explicitly note that without bare-metal side-channel auditing and validation, SD-CAF should not be deployed in hostile physical environments. Future work must bridge this gap by migrating the DRL agent to physical IoT testbeds to harden the framework against hardware-level exploitation.

Declarations

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflicts of interest: The author declares no conflicts of interest/competing interests.

Data and Code Availability: The network simulation logic, DQN hyperparameters, and tc-netem configurations used to

support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Al-Shammari, T., & Kumar, S. 2025. Energy profiling of lattice-based cryptography on ARM Cortex-M architectures. *IEEE Internet of Things Journal*, **12**(3): 445–458.

[2] Author, A., et al. 2024. Methodology and architecture for benchmarking end-to-end PQC protocol resilience in an IoT context. *Electronics*, MDPI.

[3] Barker, E., et al. 2025. Considerations for achieving crypto agility: Strategies and practices. *NIST Cybersecurity White Papers (CSWP 39)*.

[4] Chen, L., et al. 2024. Scalable SDN control planes for post-quantum network migration. *IEEE Transactions on Network and Service Management*.

[5] Ergu, Y. A., et al. 2024. Adaptive quantum-safe cryptography for 6G vehicular networks via context-aware optimization. *Sensors*, MDPI.

[6] Martinez, J., & Kim, H. 2025. Mitigating engineered downgrade attacks in agile cryptographic networks. *Proceedings of the ACM Symposium on SDN Research (SOSR)*.

[7] Misoczki, R., Elbaz, I., & Mertens, F. 2026. Post-quantum cryptography migration at Meta: Framework, lessons, and takeaways. *Engineering at Meta*.

[8] Mothukuri, V., & Parizi, R. M. 2026. Securing cryptography in the age of quantum computing and AI: Threats, implementations, and strategic response. *arXiv preprint*, arXiv:2601.

[9] National Institute of Standards and Technology (NIST). 2024. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. FIPS PUB 203.

[10] Poorani, S., & Anitha, R. 2025. Quantum-resilient cryptographic framework for cloud-based IoMT data in healthcare enterprises (H-ERPs). *Enterprise Information Systems*, 19(10).

[11] Zhao, Y., Cheng, W., Qiao, Z., Liu, Y., & Zhou, Y. 2024. Rejection matters: Efficient non-profiling side-channel attack on ML-DSA via exploiting public templates. *Proceedings of the Design, Automation and Test in Europe Conference*.