A Formally Verified, Crash-Recoverable State Machine for Zero-Downtime Post-Quantum Cryptographic Key Migration

Author: Thomas W Rodriguez Jr. **Contact:** trodjr@gmail.com **Source & Proofs:** https://github.com/trodjr/higgaion-core-crypto

Abstract

The transition from classical elliptic-curve cryptography to Post-Quantum Cryptography (PQC) presents a critical operational vulnerability for distributed networks and institutional custody providers. Existing migration strategies often enforce conjunctive multi-signature requirements that risk network partitions or rely on irreversible state transitions that lack crash-recovery assurances. We introduce a formally verified 4-state migration engine designed to execute zero-downtime, uncoordinated cryptographic upgrades. By implementing a Disjunctive (OR-mode) Hybrid Verification protocol combined with an inverted "Erasure-Before-WAL" write-ahead logging architecture, the engine guarantees deterministic crash recovery without risking the resurrection of classical key material. The safety, liveness, and memory-safety invariants of this engine have been mathematically verified using a quad-tier formal framework consisting of Coq (Gallina) mechanized proofs, TLA+ model checking, Frama-C WP, and CBMC bounded model checking.

1. Introduction & The "Store Now, Decrypt Later" Threat

The enactment of the National Security Memorandum on Promoting United States Leadership in Quantum Computing (NSM-10) and the release of the Commercial National Security Algorithm Suite 2.0 (CNSA 2.0) mandate that critical infrastructure migrate to PQC algorithms by 2030. For institutional asset custodians and high-value data repositories, the threat model of "Store Now, Decrypt Later" necessitates immediate active migration.

However, the migration of root key material in a distributed multi-node infrastructure introduces severe operational risks. Coordinated upgrades across heterogenous shards often result in split-brain consensus failures. Furthermore, the intermediate state of key transition (where classical keys must be irreversibly destroyed to prevent future extraction) is highly vulnerable to hardware faults or power interruptions.

This paper details a state machine architecture that solves these vulnerabilities, offering a mathematically proven path for safe PQC transition.

2. Disjunctive Hybrid Verification (Zero-Downtime)

Traditional hybrid cryptography enforces a conjunctive (AND-mode) verification policy, requiring both the classical signature (e.g., secp256k1, Ed25519) and the PQC signature (e.g., ML-DSA-87) to be verified. In a distributed network, this requires all validating nodes to be upgraded simultaneously: a logistical impossibility for decentralized systems.

Our architecture introduces **Disjunctive (OR-mode) Hybrid Verification**. The engine generates a composite dual-signature containing both classical and PQC components. * **Legacy Nodes:** Continue to verify only the classical component of the signature. * **Upgraded Nodes:** Verify the ML-DSA-87 PQC signature.

This policy decouples the key-generation timeline from the network-upgrade timeline, permitting uncoordinated, rolling migrations with zero operational downtime.

3. Erasure-Before-WAL: Deterministic Crash Recovery

During the final phase of migration, the classical private key must be destroyed to eliminate the SNDL liability. Standard database Write-Ahead Logging (WAL) protocols dictate that the state transition be written to disk *before* executing the action.

If applied to key migration, standard WAL creates a critical vulnerability: if a crash occurs after the WAL commit but before classical key erasure, the recovery routine could unwittingly resurrect the classical key from system memory limits or backup payloads, violating the security invariant.

To mathematically eliminate this risk, we invert the protocol: 1. **Backup** the classical public key (for ongoing verification needs). 2. **Execute** zero-pass overwrites (`OPENSSL_cleanse()`) on the classical private key memory pages. 3. **Destroy** the classical cryptographic handle context. 4. **Append** the `PQC_ONLY` transition record to the WAL.

If the node suffers a catastrophic power failure between step 2 and step 4, the WAL replays only up to the `FINALIZING` state during recovery. Because the PQC keypair has already been persisted to the dual-compromise keystore, the classical key remains definitively erased, and the node safely resumes the finalization routine.

4. Quad-Tier Formal Verification

In infrastructure managing institutional assets, empirical testing is insufficient. The migration invariants are proven using a quad-tier formal verification pipeline:

1. **Deductive Verification (Coq):** 101 Gallina theorems rigorously prove the transition matrix invariants and keystore derivation formulas, compiling with zero admitted lemmas.
2. **Model Checking (TLA+):** Proves liveness and absence of deadlocks across heterogenous shard states, ensuring no split-brain consensus derivations occur during the OR-mode propagation.
3. **Bounded Model Checking (CBMC):** Analyzes the C implementation loops and pointer arithmetic, mathematically proving the absence of memory leaks and undefined behavior up to an unwind depth of 25.
4. **Frama-C WP:** Ensures rigorous separation of classical and PQC memory domains.

5. Security & Persistence Model

The keystore persistence mechanism utilizes a PQC-hybrid encryption scheme to prevent offline decryption attacks. Stored key material is encrypted utilizing a derived AES-256-GCM key from the concatenation of a `PBKDF2-HMAC-SHA3-256` password derivative and a post-quantum `ML-KEM-1024` shared secret.

This **Dual-Compromise Resistance** requires an adversary to simultaneously break a high-entropy password dictionary attack and the Module-Lattice-Based Key Encapsulation mechanism.

6. Conclusion

The described PQC Migration Engine provides a verifiable, deterministic path for critical infrastructure to comply with NSA CNSA 2.0 timelines without inducing operational instability. By combining disjunctive verification for network availability with inverted WAL mechanics for absolute key destruction guarantees (and backing these algorithms with mechanized Coq proofs), we establish a robust standard for institutional cryptographic upgrades.

Legal Notice & Intellectual Property The overarching PQC crash-recoverable migration state machine, the Erasure-Before-WAL crash recovery architecture, and the Disjunctive (OR-Mode) Hybrid Verification protocols described herein are protected under **U.S. Patent Application No. 64/000,480**.

The cryptographic primitives and mechanised proofs are open-sourced under the MIT License to encourage peer review and academic evaluation. For commercial implementation and Enterprise SDK licensing, please contact the authors.