EchoPulse Mutation Overhead (Document C4)

Version: 1.0

Date: May 11, 2025

Author: EchoPulse Initiative

1. Purpose

This document details the computational and memory costs associated with the symbolic mutation process within the EchoPulse Key Encapsulation Mechanism (KEM) protocol. It aims to quantify the timing overhead and RAM usage of both incremental (row-based) mutation and full regeneration of the state transition graph $G(V, E)$. Understanding these costs is crucial for evaluating the long-term performance and resource implications of EchoPulse in sustained key exchange scenarios.

2. Mutation Modes

The EchoPulse protocol employs two primary modes of state transition graph mutation:

* **Single Row Mutation:** This mode involves overwriting a single row of the transition table within $G(V, E)$. It is typically triggered by a scheduled function $\mu(V, t)$, where V represents the graph and t is a temporal parameter (e.g., a counter or timer). This is the default and most frequent mutation operation.

* **Full Graph Mutation:** This mode entails the complete regeneration of the entire 256×16 transition matrix. This operation is less frequent and is typically performed to refresh the overall entropy of the graph using the initial seed and a session counter.

* **Default Interval:** The suggested default interval for triggering a mutation event (typically single row) is every 10 successful key encapsulation operations.

3. Timing Overhead Estimates

The following table provides estimated timing and RAM usage figures for the two mutation modes on representative embedded platforms:

Operation Type	Platform	Estimated Cycles	Time (µs) @ Clock	RAM Usage
Row Mutation	Cortex-M0+	~64	~4.0 @ 16 MHz	~32 B
Row Mutation	Cortex-M4F	~40	~0.5 @ 80 MHz	~32 B
Full Graph Mutation	Cortex-M0+	~16,000	~1000 @ 16 MHz	~8 KB
Full Graph Mutation	RISC-V RV64	~6,000	~50 @ 120 MHz	~8 KB

Note: These are approximate estimates based on typical instruction execution times and memory access latencies for the specified architectures. Actual performance may vary depending on the specific microcontroller implementation and compiler optimizations.

4. Memory Considerations

The RAM usage characteristics for the two mutation modes differ significantly:

* **Row Mutation:** This operation requires a fixed-size buffer of approximately 32 bytes to hold the data for the row being overwritten. The mutation is performed in-place within the existing transition table.

* **Full Graph Mutation:** This operation necessitates temporary staging space to regenerate all 256 rows of the transition matrix before updating the primary $G(V, E)$ structure. This temporary buffer will be of the same size as the transition table itself (approximately 8 KB). However, in typical operation, only one row is actively being rewritten at any given time during scheduled mutations.

5. Security Functionality

The dynamic mutation of the state transition graph serves critical security functions within the EchoPulse protocol:

* **Symbolic Pattern Obfuscation:** Over time, the continuous mutation process obscures any discernible patterns in the symbolic transition paths, making long-term analysis by adversaries significantly more challenging.

* **AI-Based Attack Prevention:** The evolving graph structure hinders the effectiveness of AI-based replay attacks and attempts to infer the underlying transition function or predict future transition paths.

* **Key Derivation Invariance:** It is crucial that the mutation process does not deterministically alter the initial vertex (v_0), the secret key (SK), or the derivation of the public key (PK) for a given session index and seed. The mutation should primarily affect the subsequent graph traversals during encapsulation and decapsulation.

6. Benchmark Trigger Events

The frequency of mutation events can be determined by various strategies:

* **Suggested Default:** A simple and effective default is to trigger a mutation (typically a single row mutation) after every 10 successful key encapsulation operations.

* **Optional Dynamic Strategies:** More advanced strategies could involve triggering mutation based on an entropy threshold of the current graph state or after a certain duration of a communication session. These dynamic approaches can adapt the mutation frequency to the perceived risk or the operational context.

7. Conclusion

The overhead associated with the default row-based mutation in EchoPulse is minimal in terms of both computational cost and RAM usage, making it well-suited for real-time embedded systems with constrained resources. Full graph mutation, while more resource-intensive, is intended for less frequent entropy refreshment and is not part of the routine key exchange cycle's performance profile. The strategic use of mutation provides a significant security advantage against advanced analytical attacks without imposing prohibitive performance penalties.

Document C4 — Transition Mutation Layer — EchoPulse Initiative