EchoPulse Mutation Framework

1. Purpose of Mutation

The integration of a dynamic mutation framework for the state transition graph G(V,E) within the EchoPulse Key Encapsulation Mechanism (KEM) serves a critical security function. The primary objective of this mutation is to introduce temporal variability into the graph structure, thereby mitigating the risks associated with long-term predictability. By ensuring that the graph evolves across different communication sessions, the mutation framework aims to prevent an attacker from accumulating sufficient knowledge about a static graph to facilitate cryptanalytic attacks. This dynamic nature specifically hinders attempts to learn the underlying graph structure or the mapping of symbolic paths over extended periods, enhancing the overall resilience of the KEM.

2. Mutation Function Definition

The evolution of the state transition graph is governed by a deterministic mutation function μ. This function takes the current state of the graph G(V,E), a shared secret salt, and the current session index as input and produces a new, mutated graph G′(V,E′):

$\mu(G,salt,session\_index) \rightarrow G'$

The mutation process is designed to be entirely deterministic and reproducible by both communicating parties. Given the same initial graph, shared salt, and session index, both the sender and the receiver will independently compute an identical mutated graph. This deterministic nature is paramount for maintaining synchronization and ensuring correct key derivation. The mutation function's output, G′, represents the state of the graph for the subsequent key exchange session.

3. Mutation Types

The mutation function μ can incorporate several types of transformations to the graph structure:

1. **Edge Relabeling:** The symbols associated with existing edges within the graph can be permuted or remapped based on a deterministic algorithm that utilizes the shared salt and the session index. This alters the symbolic transitions between states without changing the underlying connectivity.
2. **Edge Redirection:** The target vertices of existing edges can be changed to other vertices within the graph. This modification of the transition function δ is performed deterministically based on the salt and session index, effectively altering the paths taken through the state space for a given symbol sequence.
3. **Local Subgraph Restructuring:** Localized subsets of vertices and their connecting edges can be reconfigured. This involves altering the adjacency relationships within these subgraphs according to a deterministic rule derived from the salt and session index. This type of mutation can introduce more significant changes to the graph's topology.

4. Synchronization Strategy

The mutation of the state transition graph occurs periodically, every k communication sessions, where k is a predefined parameter. The specific parameters of the mutation applied at each interval are entirely determined by the shared secret salt established during the initial key agreement (or pre-shared) and the current session index. The combination of the salt and the session index uniquely defines the state of the mutation function and, consequently, the resulting mutated graph. This strategy ensures that both the sender and the receiver can independently compute the correct graph state for each session without requiring any explicit communication or synchronization signals beyond the implicit agreement on the session index.

5. Security Contribution

The dynamic mutation of the state transition graph contributes significantly to the security of the EchoPulse KEM in several ways:

4. **Time-Dependent Graph Structure:** By changing the graph over time, the mutation framework prevents an attacker from relying on a static graph structure for cryptanalysis. Any knowledge gained about the graph in one session becomes less relevant in subsequent sessions.
5. **Replay and Forward Analysis Mitigation:** The evolving graph structure complicates replay attacks, as a captured ciphertext from a previous session would be processed on a different graph by the receiver in a later session, leading to a different (and incorrect) derived key. Similarly, forward analysis of future sessions is hindered as the graph structure is not static or easily predictable.
6. **Symbol Sequence Reuse and Graph Mapping Resistance:** The mutation makes it more difficult for an attacker to exploit any potential weaknesses related to the reuse of specific symbol sequences or to infer a consistent mapping between symbol sequences and resulting states across multiple sessions. The changing graph disrupts such static analyses.

6. Mutation Invariants

To ensure the continued functionality of the EchoPulse KEM, the mutation function μ is designed to preserve certain critical invariants of the state transition graph:

7. **Graph Connectivity:** While the specific edges and their labels may change, the graph remains connected, ensuring that there exists a path between any two states. This is important for the key generation and encapsulation/decapsulation processes.
8. **Total Transition Function:** The transition function $\delta: V \times \Sigma \rightarrow V$ remains total, meaning that for every state $v \in V$ and every symbol $\sigma \in \Sigma$, there is a uniquely defined next state $\delta(v,\sigma) \in V$. This ensures that all symbol sequences can be processed from any starting state.
9. **State Determinism:** The deterministic nature of the state transitions is preserved. For a given state and input symbol, the next state is always uniquely determined, even after mutation. This is crucial for the correct derivation of the shared secret key by both parties.

7. Implementation Considerations

The implementation of the mutation framework needs to be efficient, particularly for low-power embedded devices. Several considerations are important:

10. **Precomputation:** The mutated graph for each session can potentially be precomputed based on the