EchoPulse Security Analysis

1. Overview of Security Goals

The EchoPulse Key Encapsulation Mechanism (KEM) is designed to achieve indistinguishability under chosen-ciphertext attacks (IND-CCA) within a post-quantum security framework that operates on symbolic state transitions within a dynamically evolving graph. A key security goal is to provide inherent resistance against cryptanalytic techniques potentially leveraging artificial intelligence through the continuous mutation of the underlying graph structure. Furthermore, the design emphasizes features that facilitate side-channel attack mitigation, particularly relevant for deployments in resource-constrained embedded systems.

2. Entropy and Key Space Size

The security of EchoPulse relies in part on the size of the effective key space, which is determined by the number of possible private key sequences (SK) and the randomness introduced during the encapsulation process. Given a private key path length of l symbols, drawn from a symbol set Σ of size 256 (28), and an average out-degree d of the state transition graph G(V,E), the total entropy associated with a private key path can be approximated by l×log2(d).

For example, to achieve a security target of approximately 128 bits, with an average out-degree of d≈24 (log2 (24)≈4.6 bits per symbol), a private key path length of l≥28 symbols would be required (28×4.6≈128.8 bits). The total number of possible private key sequences of length l is ∣Σ∣l=256l=(28)l=28l, representing the raw key space size. The effective security level is further influenced by the graph structure and the difficulty of reaching specific states.

3. Indistinguishability

The IND-CCA security of EchoPulse aims to ensure that an adversary, even one capable of making adaptive chosen-ciphertext queries to a decapsulation oracle, cannot distinguish a challenge ciphertext and its corresponding shared secret key from a ciphertext and a randomly chosen key with a probability significantly better than one-half. This security is underpinned by several design elements:

1. **Dynamic Mutation:** The continuous and unpredictable (from the adversary's perspective) evolution of the state transition graph G(V,E) through the deterministic mutation function μ prevents the adversary from building a consistent mapping between symbol sequences and resulting states across multiple sessions.
2. **Complex Transition Function (δ):** A well-designed transition function δ should exhibit properties that obscure the relationship between input states and symbols and the resulting output states, preventing the adversary from predicting state evolutions or identifying exploitable patterns.
3. **Hashing of Final State and Randomness:** The derivation of the shared secret key K=H(encode(venc)∥r) involves hashing the final state venc reached after applying the random symbol sequence r. This use of a cryptographic hash function provides strong diffusion, ensuring that even small changes in venc or r result in seemingly uncorrelated changes in the output key, hindering attempts to gain information about the underlying graph or key paths through observing key outputs.

4. Forward/Backward Secrecy

EchoPulse incorporates mechanisms to enhance forward and backward secrecy:

4. **Session-Specific Graph:** The mutation of the state transition graph based on the session index ensures that the graph used for key exchange in one session is different from the graph used in previous and subsequent sessions. This prevents the reuse of key-generating paths across different communication instances, contributing to forward secrecy (past session keys remain secure even if future keys are compromised).
5. **Computational Hardness of Reverse Traversal:** The design goal of ensuring that the transition function δ is computationally hard to reverse (i.e., finding a preceding state given a current state and a symbol) contributes to backward secrecy. An adversary who has compromised the current state v should not be able to efficiently trace back to the secret key path (SK) or the public key path (PK) used in prior key generation.
6. **Replay Prevention:** The session-indexed mutation of the graph also serves as a strong defense against replay attacks. A captured ciphertext r from a previous session, when used with the mutated graph of a subsequent session, will lead to a different (and incorrect) derived key.

5. Graph Structure Hardness

The security of EchoPulse relies on the computational difficulty of learning the structure of the state transition graph G(V,E) and the behavior of the transition function δ:

7. **One-Way Transitions:** The transition function δ should be designed to be computationally hard to invert, preventing an adversary from easily determining the sequence of symbols that led to a particular state.
8. **Mutation-Induced Obfuscation:** The continuous mutation of the graph invalidates any attempts by an adversary to build a static model of the graph's connectivity or the mapping of symbolic paths to states over time. The correlations between symbol sequences and resulting states change with each mutation.
9. **Graph Complexity:** A sufficiently large number of states and a complex interconnection structure contribute to the difficulty of analyzing the graph and predicting state transitions.

6. Symbol Safety

The design of EchoPulse treats symbols as semantically inert byte values, which contributes to security:

10. **No Algebraic Properties:** The lack of inherent mathematical structure in the symbols prevents attacks that exploit algebraic properties, a common vulnerability in traditional public-key cryptosystems.
11. **Fixed Encoding:** The consistent 1-byte encoding of symbols simplifies implementation and reduces the risk of encoding-related vulnerabilities.
12. **Constant-Time Transitions (Recommended):** Implementing the transition function δ using constant-