Technical Overview: EchoPulse – A Novel Post-Quantum Key Encapsulation Mechanism

1. Purpose & Strategic Positioning

The advent of quantum computing poses a significant threat to currently deployed public-key cryptography, including widely adopted standards like RSA and ECC. While significant research efforts are focused on Post-Quantum Cryptography (PQC) algorithms based on lattices (e.g., Kyber), codes (e.g., Classic McEliece), and isogenies (e.g., SIKE), these approaches often exhibit limitations in resource-constrained environments. Schemes like Kyber and Dilithium, while promising for general-purpose computing, can suffer from large key and ciphertext sizes, as well as computationally intensive operations that may strain the capabilities of Internet of Things (IoT) devices, embedded systems, and military field equipment. EchoPulse aims to address these limitations by introducing a fundamentally new Key Encapsulation Mechanism (KEM) based on symbolic state transitions within a dynamically evolving graph, rather than traditional algebraic structures. This novel approach seeks to provide a lightweight and efficient alternative for secure key exchange in resource-limited scenarios while offering inherent resistance to quantum attacks.

2. Why a New KEM is Needed

The impending post-quantum era necessitates the development and standardization of cryptographic primitives resilient to attacks from quantum computers. While the NIST PQC standardization process has identified several promising candidates, a critical gap remains in providing solutions tailored for highly constrained devices. Military and embedded systems often operate under strict limitations regarding processing power, memory footprint, and communication bandwidth. Existing lattice-based KEMs, for instance, can involve large polynomial manipulations and significant storage requirements for keys and ciphertexts, potentially exceeding the capabilities of these resource-scarce platforms. Similarly, code-based schemes might present challenges in terms of key generation complexity and key sizes. The need for a KEM that offers robust post-quantum security with minimal computational overhead and compact payloads is paramount for securing a significant portion of the digital ecosystem.

3. EchoPulse Rationale

EchoPulse departs from traditional PQC paradigms by employing symbolic state transitions within a finite, directed graph. Instead of relying on mathematical operations within algebraic structures, the core of EchoPulse lies in the evolution of a system state through a sequence of discrete symbols. The secret and public keys are represented as specific sequences of these symbols, defining distinct paths within the graph originating from a common initial state. Key encapsulation involves a sender-generated random symbol sequence that triggers further state transitions based on the receiver's public key state. The resulting final state, combined with the random sequence, is then hashed to derive the shared secret key.

To enhance security and resilience against long-term analysis, EchoPulse incorporates an adaptive mutation mechanism. The underlying state transition graph is not static but evolves deterministically over time. This evolution is synchronized between communicating parties using a shared salt and a session index, ensuring that both sender and receiver operate on the same graph structure for each key exchange.

EchoPulse is designed with a target memory footprint of approximately 5 KB, encompassing the storage for the state transition graph (e.g., 1024 nodes, each requiring 1 byte for state representation) and a working buffer of around 1 KB. The design aims to achieve an entropy level of approximately 128 bits per traversed path within the graph, contingent on the graph's structural complexity and the distribution of transitions.

4. Key Innovations

EchoPulse introduces several key innovations:

1. **Symbol Set (Σ):** A finite set of abstract symbols forms the basis of state transitions, replacing algebraic elements.
2. **Transition Graph G(V,E):** A directed graph where vertices represent states and labeled edges (symbols from Σ) define transitions between states.
3. **Encapsulation via H(V‖r):** The shared secret key is derived by hashing the final state reached after applying the random symbol sequence (r) to the public key state, concatenated with r.
4. **Mutation Function μ(V,t):** A deterministic function that modifies the structure of the transition graph over time (t), synchronized using a shared salt and session index.
5. **Sub-1KB Payloads:** The design targets a total key exchange payload (public key and ciphertext) significantly below 1 KB, crucial for bandwidth-constrained environments.

5. Applications & Use Cases

The lightweight and potentially highly configurable nature of EchoPulse makes it suitable for various applications, including:

6. **Military Field Devices:** Secure communication and key exchange in tactical environments with limited computational resources and network bandwidth.
7. **IoT Nodes:** Providing a post-quantum secure mechanism for device authentication and secure data transmission in resource-constrained IoT ecosystems.
8. **Edge Authentication:** Securing communication between edge computing devices and central servers.
9. **Symbolic Crypto Research:** Offering a new paradigm for cryptographic design, potentially leading to further innovations in post-quantum security.

6. Conclusion

EchoPulse represents a departure from traditional post-quantum cryptographic constructions. It is not derived from lattices, codes, or isogenies, but instead offers a symbolic reinvention of the Key Encapsulation Mechanism logic based on state transitions within a dynamically evolving graph. Its design prioritizes low resource consumption and compact payloads, making it a promising blueprint for securing resource-constrained devices in