EchoPulse Formal KEM Definition

1. Key Generation (KeyGen)

The key generation algorithm, denoted as KeyGen(λ), takes as input a security parameter λ and outputs a public key PK and a secret key SK.

1. **Input:** Security parameter λ (implicitly determines the lengths of symbol sequences and other parameters).
2. **Output:** Public key $PK \in \Sigma l'$ and secret key $SK \in \Sigma l$, where Σ is the symbol set, and l, l' are positive integers determined by λ.

The key generation process proceeds as follows:

1. Initialize the state transition graph G(V,E) with $|V|=1024$ states and transitions defined by $\delta: V \times \Sigma \to V$, starting from a fixed initial state $v0 \in V$. The construction of G is deterministic based on a master secret or a public seed (depending on the instantiation).
1. Generate the secret key $SK=(\sigma1,\sigma2,\ldots,\sigma l)$ as a sequence of l symbols chosen randomly from Σ or derived deterministically from a master secret.
1. Compute the private state $vpriv=\delta(\delta(\ldots\delta(v0,\sigma1),\sigma2)\ldots,\sigma l)$.
1. Generate the public key $PK=(\sigma1',\sigma2',\ldots,\sigma l'')$ either independently as a sequence of l′ symbols or derived deterministically from SK through a one-way transformation.
1. Compute the public state $vpub=\delta(\delta(\ldots\delta(v0,\sigma1'),\sigma2')\ldots,\sigma l'')$.
1. Output the public key PK and the secret key SK.

2. Encapsulation (Encaps)

The encapsulation algorithm, denoted as Encaps(PK), takes as input the public key PK and outputs a ciphertext C and a shared secret key K.

3. **Input:** Public key $PK \in \Sigma l'$.
4. **Output:** Ciphertext $C \in \Sigma m$ and shared secret key $K \in \{0,1\}n$, where m is the length of the random symbol sequence and n is the output length of the hash function H.

The encapsulation process proceeds as follows:

2. Generate a random symbol sequence $r=(\rho1,\rho2,\ldots,\rho m)$ of length m, where each ρi is chosen uniformly at random from Σ. Set the ciphertext $C=r$.
2. Compute the public state reached by applying PK from the initial state: $vpub=\delta(v0,PK)$, where $\delta(v0,(\sigma1',\ldots,\sigma l''))=\delta(\ldots\delta(\delta(v0,\sigma1'),\sigma2')\ldots,\sigma l'')$.
2. Compute the encapsulated state venc by applying the random symbol sequence r from the public state vpub: $venc=\delta(vpub,r)$, where $\delta(vpub,(\rho1,\ldots,\rho m))=\delta(\ldots\delta(\delta(vpub,\rho1),\rho2)\ldots,\rho m)$.
2. Compute the shared secret key K by applying a cryptographic hash function H to the concatenation of the encoded encapsulated state venc and the random symbol sequence r: $K=H(encode(venc)\|r)$.
2. Output the ciphertext C and the shared secret key K.

3. Decapsulation (Decaps)

The decapsulation algorithm, denoted as Decaps(C,SK), takes as input a ciphertext C and the secret key SK, and outputs a shared secret key K′.

5. **Input:** Ciphertext $C \in \Sigma m$ and secret key $SK \in \Sigma l$.
6. **Output:** Shared secret key $K' \in \{0,1\}n$.

The decapsulation process proceeds as follows:

3. Let the received ciphertext be $C=r=(\rho1,\rho2,\ldots,\rho m)$.
3. Compute the private state reached by applying SK from the initial state: $vpriv=\delta(v0,SK)$, where $\delta(v0,(\sigma1,\ldots,\sigma l))=\delta(\ldots\delta(\delta(v0,\sigma1),\sigma2)\ldots,\sigma l)$.
3. Compute the decapsulated state vdec by applying the received symbol sequence r from the private state vpriv: $vdec=\delta(vpriv,r)$, where $\delta(vpriv,(\rho1,\ldots,\rho m))=\delta(\ldots\delta(\delta(vpriv,\rho1),\rho2)\ldots,\rho m)$.
3. Compute the derived shared secret key K′ by applying the same cryptographic hash function H to the concatenation of the encoded decapsulated state vdec and the received symbol sequence r: $K'=H(encode(vdec)\|r)$.
3. Output the derived shared secret key K′.

4. Correctness

The EchoPulse KEM is correct if, for all public/secret key pairs (PK,SK) generated by KeyGen(λ), and for all ciphertexts C and shared secrets K generated by Encaps(PK), the decapsulation of C using SK results in the same shared secret K′:

$Decaps(C,SK)=K$

This correctness relies on the deterministic nature of the state transitions and the synchronized state of the graph G(V,E) between the sender and the receiver for the given session. Specifically, if the graph and the initial state v0 are consistent, and no errors occur during the application of the symbol sequences, then venc computed during encapsulation should be equal to vdec computed during decapsulation.

5. Security Model

The security of the EchoPulse KEM aims to achieve indistinguishability under chosen-ciphertext attacks (IND-CCA) or a related post-quantum security notion. The security relies on the following core components:

7. **(a) Hash Function H:** The cryptographic hash function H must provide strong collision resistance and preimage resistance.
8. **(b) Mutation Randomness:** The deterministic but evolving nature of the state transition graph G(V,E) through the mutation function μ, driven by a shared salt and session index, introduces temporal randomness that hinders long-term analysis.
9. **(c) Path Entropy:** The length of the private and public key symbol sequences (l and l′) and the branching factor of the graph must provide sufficient entropy to resist brute-force attacks on the secret