Security Properties of the Graph Mutation Function μ(G,t) in the EchoPulse Protocol

1. Abstract

This document presents a formal analysis of the graph mutation function μ within the EchoPulse Key Encapsulation Mechanism (KEM). Unlike traditional algebraic primitives, EchoPulse's security fundamentally relies on dynamic graph transformations to ensure path unpredictability and resistance against replay attacks. We formally define μ and its impact on the symbolic graph G, introduce quantitative metrics to characterize its security-relevant properties such as Mutation Diffusion Factor (MDF) and Symbol Remapping Entropy (SRE), and theoretically argue its contribution to replay resistance. The analysis substantiates how μ's evolution over time continuously alters the underlying symbolic structure, making pre-computation or reuse of cryptographic material infeasible across different mutation instances, thereby reinforcing the Symbolic Graph Path Unpredictability (SGPU) assumption and contributing to IND-CCA2 security.

2. Formal Definition of μ (Symbolic Security Modeler)

The graph mutation function μ is a deterministic, time-dependent transformation applied to the EchoPulse symbolic graph. It is a critical component influencing the graph's dynamic behavior and thus, the unpredictability of traversed paths.

Graph State: A graph G at any given time t is formally defined by its state space V and its transition function $\delta_G: V \times \Sigma \to V$. Thus, $G_t = (V, \delta_t)$. We assume V and Σ remain constant.

Mutation Function Definition:

$\mu: G_{base} \times time\_seed \times session\_index \to G_{mutated}$

More precisely, given a static base graph $G_{base} = (V, \delta_{base})$ and a time input $t \in N$ (often derived from session_index or a time_seed), μ produces a new transition function δ_t.

So, $\mu(G_{base}, t) = G_t = (V, \delta_t)$, where δ_t is the mutated transition function at time t.

Modeling the Transformation:

The function μ operates by systematically altering a subset of the mappings within the transition function δ. For each state $v \in V$ and symbol $s \in \Sigma$, the mapping $\delta(v,s) \to v'$ is subject to mutation.

We model μ as a permutation-based or substitution-based algorithm that re-maps transition targets.

Let M_t be the "mutation matrix" at time t, derived deterministically from t and potentially a public seed.

$\delta_t(v,s) = Algorithm(\delta_{base}(v,s), M_t, hash(v,s,t))$.

This could involve:

Direct Re-mapping: For a subset of (v,s) pairs, $\delta_t(v,s)$ is set to a new, deterministically derived $v'' \in V$.

Permutation: Permuting the target states v' for a fixed v across all symbols $s \in \Sigma$.

Composition: $\delta_t(v,s) = \delta_{base}(\delta_{base}(v,s), mutated\_symbol)$.

Properties of μ:

Deterministic: For any fixed G_{base} and t, $\mu(G_{base}, t)$ always yields the identical G_t. This is crucial for consistent decapsulation.

Time-Dependent: $G_{t_0} = G_{t'}$ for $t_0 = t'$ (with high probability, for non-negligible $t - t'$ differences, unless μ has extremely short, pathological cycles). This property is fundamental to replay resistance.

polo it: The computation of δ_t from δ_{base} and t should be efficient (polynomial