

EchoPulse Protocol Comparison Table

1. Introduction


This document provides a concise comparison table summarizing key technical and cryptographic differences between the EchoPulse Key Encapsulation Mechanism (KEM) and two established lattice-based post-quantum KEMs standardized by NIST: Kyber512 (from the CRYSTALS-Kyber family, targeting NIST Security Level 1) and FrodoKEM-640-AES (targeting NIST Security Level 1). The purpose of this table is to offer a quick reference for evaluating the trade-offs between these protocols, particularly concerning their underlying security basis, performance characteristics, and suitability for different deployment environments.

2. Comparison Table

Property	EchoPulse	Kyber512	FrodoKEM-640
PQ Security Basis	Symbolic Graph	Lattice (MLWE)	Lattice (LWE)
IND-CCA Security	✓	✓	✓
Payload Size (Ciphertext)	~28 B	768 B	976 B
Public Key Size	~28 B	800 B	9616 B
Memory Usage (Estimated)	30–72 KB	~100 KB	~160 KB
Symbol Mutation / AI Resistance	✓	✗	✗
Side-Channel Protection Potential	✓ (Symbol-CT)	⚠	⚠
Encapsulation Time (MCU-Class)	High (Hash dominated)	Very High	Very High

3. Interpretation Notes

* **Symbol-CT:** Indicates the potential for strong side-channel protection in EchoPulse implementations by employing constant-time lookups for the state transition function (δ) based on the symbolic nature of the operations.

* **  : ** Denotes that while lattice-based schemes have known countermeasures against side-channel attacks, their algebraic structure can present more complex challenges compared to the purely symbolic operations of EchoPulse. The effectiveness depends heavily on specific implementation choices.

* The estimated payload and public key sizes for EchoPulse are based on the example parameters used in previous documents (28-symbol sequences). These can be adjusted based on the desired security level and graph structure. The memory usage for EchoPulse is an estimate and depends on the graph compression techniques employed. The "High" encapsulation time for EchoPulse on MCU-class devices is primarily attributed to the computationally intensive cryptographic hash function.

****4. Conclusion****

The EchoPulse protocol presents a distinct approach to post-quantum key encapsulation, differing significantly from lattice-based schemes like Kyber and FrodoKEM in its underlying security basis. The table highlights the potential advantages of EchoPulse in terms of significantly smaller payload and public key sizes, which can be particularly beneficial for bandwidth-constrained and low-power embedded systems. Furthermore, the integrated symbol mutation framework offers a novel approach towards potential resistance against AI-driven cryptanalysis. While the computational cost on resource-constrained devices, particularly due to hashing, requires careful optimization, the symbolic nature of EchoPulse offers promising avenues for side-channel protection. This comparison underscores the unique trade-offs offered by EchoPulse, warranting further investigation for specific application scenarios where its characteristics might provide advantages over more established PQC KEMs.

Version 1.0 — Protocol Benchmark Layer — EchoPulse Initiative