Lecturer: Nadim Kobeissi **Website:** `https://appliedcryptography.page`

Project E: Post-Quantum Cryptography Migration

Overview

In this project, you will explore the practical challenges of migrating existing systems to post-quantum cryptography. You'll work with NIST's standardized post-quantum algorithms, integrating ML-KEM (Kyber) for key encapsulation and ML-DSA (Dilithium) for digital signatures. This project guides you through hybrid approaches that combine classical and post-quantum algorithms, ensuring security against both current and future quantum threats. You will start by choosing a target protocol that you wish to devise a post-quantum migration strategy for. Then, you'll analyze performance impacts, message size increases, and integration challenges when replacing pre-quantum primitives (such as RSA or ECDH) with quantum-resistant alternatives. By completing this project, you'll gain practical experience in implementing post-quantum designs, developing migration strategies for existing systems, and evaluating the trade-offs between different post-quantum algorithm choices for various use cases.

Learning Objectives

After completing this project, you should be able to:

- Integrate NIST standardized post-quantum algorithms (ML-KEM and ML-DSA) through the use of existing implementations and cryptographic libraries.
- Design hybrid cryptographic schemes that provide security against both classical and quantum adversaries.
- Analyze performance and compatibility trade-offs in post-quantum migration.
- Develop practical migration strategies for transitioning existing systems to post-quantum security.

Background

The advent of quantum computers poses a significant threat to current public-key cryptography. Post-quantum cryptography provides security against both classical and quantum adversaries:

- ML-KEM (Kyber) provides quantum-resistant key encapsulation based on lattice problems.
- ML-DSA (Dilithium) offers quantum-resistant digital signatures also based on lattice cryptography.
- Hybrid approaches, such as X-Wing[1] combine classical and post-quantum algorithms to hedge against implementation vulnerabilities.
- Migration strategies must consider performance impacts, message size increases, and backward compatibility.

Requirements

Your post-quantum migration project must implement the following core functionality:

1. **Protocol Selection and Analysis:**
    - Choose a target protocol (e.g., TLS, SSH, VPN, or custom protocol) for migration.
    - Document the current cryptographic primitives used in the protocol.

[1] Manuel Barbosa, Deirdre Connolly, João Diogo Duarte, Aaron Kaiser, Peter Schwabe, Karolin Varner and Baas Westerban, *X-Wing: The Hybrid KEM You've Been Looking For*, IACR Communications in Cryptology, 2024.

- Analyze the protocol's security requirements and constraints.

2. **Post-Quantum Integration:**
   - Integrate ML-KEM (Kyber) for key encapsulation mechanisms.
   - Integrate ML-DSA (Dilithium) for digital signature operations.
   - Create hybrid schemes combining classical and post-quantum algorithms.
3. **Migration Strategy:**
   - Design a backward-compatible migration path.
   - Implement protocol negotiation for algorithm selection.
   - Ensure graceful fallback to classical algorithms when necessary.
4. **Performance Analysis:**
   - Measure and compare key generation, encapsulation, and signature times.
   - Analyze message size increases and bandwidth impacts.
   - Evaluate memory usage and computational requirements.
   - Document performance trade-offs for different security levels.

Implementation Guidelines

Step 1: Protocol Analysis

Begin by thoroughly analyzing your chosen protocol:

- What cryptographic primitives does it currently use? (RSA, ECDH, ECDSA, etc.)
- What are the performance requirements and constraints?
- What are the message size limitations?
- How does the protocol handle algorithm negotiation?

Document your findings and justify your choice of target protocol.

Step 2: Post-Quantum Integration

Implement the post-quantum algorithms:

- Integrate ML-KEM (Kyber) for key exchange operations.
- Integrate ML-DSA (Dilithium) for signature operations.
- Design and implement hybrid modes that combine classical and post-quantum algorithms.

Step 3: Migration Design

Develop a comprehensive migration strategy:

- Protocol negotiation mechanisms for algorithm selection.
- Backward compatibility with non-post-quantum implementations.
- Phased migration approach with risk mitigation.

Step 4: Performance Evaluation

Conduct thorough performance analysis:

- Benchmark all cryptographic operations.
- Measure protocol overhead and latency impacts.
- Analyze scalability under various load conditions.

Step 5: Security Analysis

Evaluate the security of your migration:

- Analyze the security levels provided by different parameter sets.
- Consider cryptographic agility and future algorithm updates.
- Evaluate risks during the migration period.

Deliverables

Submit the following:

1. Source code for your post-quantum migration implementation.
2. Design document including:
   - Detailed protocol analysis and migration strategy.
   - Description of hybrid cryptographic schemes implemented.
   - Justification for algorithm choices and parameter selections.
3. Performance analysis report with:
   - Benchmarking results comparing pre- and post-quantum implementations.
   - Analysis of message size and bandwidth impacts.
   - Recommendations for different use cases.
4. Security analysis discussing:
   - Security guarantees of the migrated protocol.
   - Potential vulnerabilities during migration.
   - Long-term cryptographic agility considerations.

Evaluation Criteria

Your project will be evaluated based on:

- Correctness of post-quantum algorithm integration.
- Quality and practicality of the migration strategy.
- Thoroughness of performance analysis.
- Security considerations and risk mitigation.
- Quality of code and documentation.

Resources

- NIST Post-Quantum Cryptography standardization documents and FIPS 203 (ML-KEM), FIPS 204 (ML-DSA).
- Reference implementations: Kyber reference implementation, Kyber Go implementation, Dilithium reference implementation, and Open Quantum Safe library.
- Course materials on post-quantum cryptography.

Submission Guidelines

- Submit your code as a ZIP archive or through a Git repository.
- Include all documentation in PDF or Markdown format.
- Presentations: Prepare a 10-minute presentation demonstrating your post-quantum migration strategy and implementation.