Verified Post-Quantum Cryptography

•••

Project Overview

Tom Arnold <tca4384@rit.edu>

Background

- New cryptosystems being developed to withstand attacks from quantum computers.
    - Shor's integer factoring algorithm can break RSA
    - NIST Post Quantum Cryptography competition
- Secure cryptosystem can be broken by implementation details.
    - Timing attacks, memory corruption, bad parameters, etc.
- Formal verification techniques are becoming practical.
    - MSR-INRIA Project Everest => miTLS, fully verified TLS stack.

Problem Statement

- Can we implement a formally verified post-quantum cryptosystem?
- Classic McEliece is one of the NIST PQC finalists. This is a high-security/performance variant of the McEliece cryptosystem from the 70s.
- **Implement variants of McEliece, formally verify them, analyze security.**

Related Work

- Classic McEliece - PQC finalist
  - https://classic.mceliece.org
- FStar - Programming language for formal verification.
  - https://fstar-lang.org
- HACL - High Assurance Cryptographic Library
  - https://github.com/project-everest/hacl-star
  - Implements formally verified cryptographic primitives.
  - Big numbers, AES, SHA, etc.

Overview of Solution / Approach

- Use the FStar language from MSR-Inria to implement variants of McEliece cryptosystem.
  - McEliece - Code-based system from the 70s
  - Niederreiter - High performance variant
- Leverage FStar standard library and HACL for crypto primitives (Galois fields, matrices, buffers)

Experimental Plan

- Analyze and compare McEliece variants
    - Implementation complexity
    - Performance
    - Key size, security
- Are there any tradeoffs from formal verification?

See https://tom9729.bitbucket.io/csci788/ for more information on the project.