Verified Post-Quantum Cryptography

Milestone 2 - 2022/03/01
Tom Arnold <tca4384@rit.edu>

Problem Statement / Solution

Problem: PQC is important. Can we formally verify a post-quantum cryptosystem?

Solution:

- Implement variants of McEliece post-quantum cryptosystem.
- Formally verify implementation using refinement types (LiquidHaskell).
- Show benefits/tradeoffs of using refinement types in a small-scale project.

Previous & Current Milestone Goals

- Milestone 1
  - ✅ Implement Hamming code version of McEliece.
  - ✅ Verified using refinement types.
- Milestone 2
  - ☐ Implement matrix inversion routine.
  - ☐ Research and implement Goppa code version of McEliece.

Milestone 2 Results

- ☑️ Implemented and verified binary-matrix inversion.
  - Algorithm based on elementary-row operations.
  - Used during McEliece decryption, these were calculated by hand previously.
- ☑️ Implemented Goppa code version of McEliece.
  - Implemented polynomial arithmetic and binary Galois field arithmetic.
  - **Operations**: addition, subtraction, multiplication, division, polynomial and field inversion, & polynomial evaluation.
  - **Operands**: polynomials of binary field elements.
  - **Not fully verified yet**, verification of polynomial arithmetic difficult.

~1000 lines of code written for this milestone.

Roadmap For Milestone 3

- Finish verification of polynomial and field arithmetic.
- Implement the Niederreiter variant of McEliece.