Verified Post-Quantum Cryptography

Tom Arnold
tca4384@rit.edu

24 February 2021

Abstract

New cryptosystems are being developed and standardized to resist attacks from quantum computers. Formal verification can help verify the correct implementation of these systems. We present a study of formal verification applied to the domain of post-quantum cryptography by implementing the McEliece cryptosystem in LiquidHaskell and verifying it using refinement types.

1 Introduction

Widely used public key cryptosystems are vulernable to attacks from quantum computers. These cryptosystems are based on one-way functions for which there is no efficient solution to calculate the inverse of the function; the security of the system is built on this fact. Quantum computing has introduced new algorithms which can efficiently compute solutions to some of these problems. As a result, cryptographers are working to implement new cryptosystems based on problems that are not efficiently solvable by quantum or digital computers [1].

The McEliece cryptosystem is one such system that dates back to the 70s when it was proposed by Robert McEliece. The system is based on coding theory and to break it an attacker would have to solve the general decoding problem which is NP-complete [5]. A modern variant of McEliece called Classic McEliece is one of the finalists in the NIST Post-Quantum standardization effort [2].

Formal verification can be used to prove the absense of bugs in a way that automated or manual testing cannot. Typically such verification is only done for software which must be held to a high standard. Cryptography is one such field where even simple errors can have severe real-world consequences [3].

In the following paper we show how a post-quantum cryptosystem (McEliece) can be implemented and formally verified using refinement types with LiquidHaskell [4]. We also examine the effort involved in performing such verification so as to better understand the costs and benefits involved.

References

[1] Risse, T. (2011). How SAGE Helps To Implement Goppa Codes and The McEliece Public Key Crypto System.

[2] Computer Security Division. Post-quantum cryptography: CSRC. Retrieved February 24, 2022, from https://csrc.nist.gov/Projects/post-quantum-cryptography

[3] Jean-Karim Zinzindohoué, Karthikeyan Bhargavan, Jonathan Protzenko, and Benjamin Beurdouche. 2017. HACL*: A Verified Modern Cryptographic Library. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). Association for Computing Machinery, New York, NY, USA, 1789–1806. DOI:https://doi.org/10.1145/3133956.3134043

[4] Vazou, N., Seidel, E.L., Jhala, R., Vytiniotis, D., Peyton-jones, S. (2014). Refinement types for Haskell. ICFP 2014.

[5] McEliece, R.J. (1978). A public key cryptosystem based on algebraic coding theory.