CS 290g Proposal: Verified Synthesis of Crytographic algorithms

Jared Roesch

Spring 2015

Overview

The goal of this project will be to construct a small EDSL (embeded domain specific language) that allows for the verification and synthesis of high performance crypto algorithms.

Verification

By designing our language as an embedded DSL when we type expressions like `e + f` instead of directly executing them for a result we construct a representation of our intended program for manipulation. If we write such a DSL in a dependently typed programming language we can then prove properties about this program's time complexity, correctness or timing.

Program Synthesis

Program synthesis is the task of generating a piece of code from high level constraints, and other user infomation. We will borrow ideas from program synthesis in order to use our specifications to generate low level, high performance code.

Implementation

The implmentation will consist of an EDSL designed in a automatic theorem prover such as Coq or Lean. These are both theorem proving environments and dependently typed programming languages. Lean is a new theorem prover from

MSR while Coq is more established. I will evaluate whether Lean is suitable, and if not use Coq.

The final implementation will transform high level specifications directly into an linkable object file.

Goals

- implement a small framework for verifying crypto algorithms
- implement basic code generator for the verified algorithms
- implement and verify a couple standard crypto operations, and algorithms (CRT, RSA, etc)