Introduction to EasyCrypt

Alley Stoughton

Boston University

EasyCrypt

- EasyCrypt is a proof assistant for mechanizing proofs of the security of cryptographic constructions and protocols
- Models cryptography at high level, but associated frameworks that connect it with actual implementations
- Experimental extension of EasyCrypt for differential privacy
- See:
  - EasyCrypt's GitHub: https://github.com/EasyCrypt/easycrypt
  - My installation and configuration instructions: https://alleystoughton.us/easycrypt-installation.html
  - Formosa Crypto Zulip: https://formosa-crypto.gitlab.io/

EasyCrypt's Object Programming Language

- EasyCrypt's programming language is based on probabilistic while language (pWhile):
  - Assignments, random assignments (choosing values from sub-distributions), sequencing, conditionals, while loops
- Modules consist of global variables and procedures
  - Procedures can call other procedures (no recursion)
- Modules may be parameterized, e.g., by adversaries

EasyCrypt's Logics

- EasyCrypt has four logics:
    - a Probabilistic Relational Hoare Logic (pRHL) for proving relations between pairs of procedures
        - Experimental branch with support for proving differential privacy (apRHL)
    - a Probabilisitic Hoare logic (pHL) for proving probabilistic facts about single procedures
    - an ordinary Hoare logic (HL)
    - an ambient higher-order logic for proving mathematical facts and connecting judgements from the other logics
        - Based on higher order classical logic

EasyCrypt's Proofs and Theories

- Proofs are structured as sequences of lemmas
- Lemmas are proved using tactics, as in Coq
  - Simple ambient logic goals can be proved using SMT solvers
- EasyCrypt theories may be used to group definitions, modules and lemmas together
- Theories may be specialized via cloning
  - Any axioms must then be proved

Today

- Today we're going to talk about EasyCrypt's ambient logic
- See my slides on the ambient logic for reference
- But we're going to spend the rest of today's class with live coding