The Joint EasyCrypt-F*-CryptoVerif School 2014

Cătălin Hriţcu, Inria Paris-Rocquencourt

1. OVERVIEW

Formal security verification tools are slowly reaching maturity. The Joint EasyCrypt-F*-CryptoVerif School took place between 24 and 28 November 2014 in Paris and taught participants how to use three state-of-the-art security verification tools, as well as gave participants a glimpse of the formal foundations behind these tools. The school brought together over 80 participants from 12 countries, with backgrounds spanning security, cryptography, programming languages, and formal methods. The time of the school was roughly evenly split between lectures and hands-on exercise sessions.[1] For the exercise sessions participants installed and tried EasyCrypt, F*, and CryptoVerif on their own laptops, under the guidance of members from the developer teams of the tools. The school also contained three short invited talks: Hubert Comon on *Unconditional Soundness*, Véronique Cortier on *Type-Based Verification of Electronic Voting Protocols*, and Matteo Maffei on *Logical Foundations of Secure Resource Management in Protocol Implementations*.

2. EASYCRYPT

EasyCrypt is a toolset for reasoning about relational properties of probabilistic computations with adversarial code. Its main application is the construction and verification of game-based cryptographic proofs. EasyCrypt was used to prove the security of complex cryptographic constructions, including the Cramer-Shoup encryption scheme, the Merkle-Damgaard iterative hash function design, and the ZAEP encryption scheme. More recently, it has been used for proving the security of protocols based on garbled circuits, and for the proof of security for authenticated key-exchange protocols and derived proofs under weaker assumptions.

The EasyCrypt lectures were given by Gilles Barthe, François Dupressoir, Benjamin Grégoire, Benedikt Schmidt, Pierre-Yves Strub, who also jointly run the exercise-sessions. The lectures presented (a) the code-based game-playing approach to computer-aided cryptographic proofs and the connection to program verification; (b) the foundations of EasyCrypt: probabilistic relational Hoare logic and program transformations; (c) the ambient (classical higher-order) logic of EasyCrypt and the most widely-used tactics; (d) grouping related concepts and lemmas into theories and structuring proofs with sections; (e) using modules to achieve abstraction and support high-level reasoning steps; (f) high-level cryptographic proof principles; (g) EasyCrypt case studies; (h) verifying "real-world" security at the source-code level; (i) automated proofs and synthesis; and (j) perspectives for the future.

The EasyCrypt exercise sessions involved proving (a) security against chosen plaintext attacks (IND-CPA) for the Bellare and Rogaway 1993 and the Hashed ElGammal encryption schemes; (b) correctness of a stateful random generator that uses a pseudo-random function (PRF); and (c) the PRP (pseudo-random permutation)/PRF switching lemma.

[1] The materials, including slides and exercises, are available on the website of the school: https://wiki.inria.fr/prosecco/The_Joint_EasyCrypt-F*-CryptoVerif_School_2014

3. F*

F* is a new ML-like functional programming language designed with program verification in mind. It has a powerful refinement type-checker that discharges verification conditions using the Z3 SMT solver. F* has been successfully used to verify nearly 50,000 lines of code, ranging from cryptographic protocol implementations to web browser extensions, and from cloud-hosted web applications to key parts of the F* compiler itself. The newest version on F* erases to both F# and OCaml, on which it is based. It also compiles securely to JavaScript, enabling safe interoperability with arbitrary, untrusted JavaScript libraries.

The F* lectures were given by Antoine Delignat-Lavaud, Cédric Fournet, and Nikhil Swamy, and comprised (a) a high-level introduction into proving programs correct with F*; (b) an overview of how F* deals with state and other effects; (c) an overview of type-based verification at scale and miTLS, a verified reference implementation of TLS; and (d) a more in-depth illustration of how to use types for modular verification of cryptographic code.

The F* exercise sessions included (a) many basic examples: proving correctness of a simple model of access control and of simple recursive functions on numbers (factorial, fibonacci) and lists (mapping, finding, and sorting), and proving termination for Ackermann and tail recursive functions, (b) several cryptographic examples: cryptographic capabilities for accessing files, secure RPC using MACs, and verified encryption padding; and (c) a case-study on formalizing the metatheory of the simply-typed λ-calculus, illustrating the use of F* as a proof assistant. The exercise sessions were prepared and run jointly by: Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Cătălin Hriţcu, Chantal Keller, Markulf Kohlweiss, Aseem Rastogi, Pierre-Yves Strub, and Nikhil Swamy.

4. CRYPTOVERIF

CryptoVerif is an automatic protocol prover sound in the computational model of cryptography. It can prove secrecy and correspondences (e.g. authentication). The generated proofs are by sequences of games, as used by cryptographers. CryptoVerif was successfully used for security proofs of various cryptographic schemes and protocols, including Kerberos, OEKE, and the SSH transport layer protocol.

The CryptoVerif lectures and exercise sessions were run by Bruno Blanchet, the main designer and developer of CryptoVerif. The lectures covered (a) the process calculus used by CryptoVerif for expressing games; (b) representing security assumptions on primitives as indistinguishability properties; (c) syntactic transformations; and (d) generating protocol implementations from CryptoVerif specifications. Two illustrative examples were used during the lectures: encrypt-then-MAC and full domain hash (FDH) signatures. In the exercise sessions the students used CryptoVerif to prove (a) various security notions for encrypt-then-MAC (IND-CPA, IND-CCA2, INT-CTXT); (b) the security (IND-CPA) of the Bellare and Rogaway 1993 encryption scheme; and (c) authentication for the fixed version of the Woo-Lam shared-key protocol.

ACKNOWLEDGMENTS

The school was organized by Anna Bednarik, Karthikeyan Bhargavan, Cătălin Hriţcu, and Pierre-Yves Strub, together with a group of student volunteers: Evmorfia-Iro Bartzia, Benjamin Beurdouche, Antoine Delignat-Lavaud, and Jean Karim Zinzindohoué. This school was financially supported by the Prosecco team at Inria Paris-Rocquencourt with funds from the ERC Starting Grant CRYSP. The EPSRC CryptoForma network on formal methods and cryptography sponsored a number of grants for UK attendees. The MSR-Inria Joint Centre sponsored the social dinner. Cryptosense organized and sponsored the reception.