Proving Computational Security with a General-Purpose C Verifier

A 5-Minute Talk

François Dupressoir
The Open University

Cédric Fournet
Microsoft Research Cambridge

Andrew D. Gordon
Microsoft Research Cambridge
University of Edinburgh

Security protocols and APIs are difficult to specify and implement. Most of the time, for example for the next version of the TPM, a reference implementation, often written in C, is the only formal specification. In this talk, we show how VCC, a general-purpose C verifier that was previously used to prove symbolic security properties, can be used to prove computational security properties, including, for the first time, computational indistinguishability, of protocols and APIs implemented in C.

To do so, we rely on VCC to prove that the C program has the same observable input-output behaviour as a reference implementation written in F#. We then use the F7 type-checkerto prove perfect security properties of the reference code assuming ideal cryptographic primitives. Finally, we bound the probability that a program that uses ideal cryptographic primitives differs from the same program that uses their concretely secure implementation, relying on standard game-based security assumptions.

Our verification methodology is modular, and can be applied to any system that can be given a deterministic specification. We illustrate it on a C implementation of a simple key management API inspired by the TPM. Our API includes commands for creating sensitive and non-sensitive keys through a KDF, importing and exporting public data to be used as non-sensitive keys, wrapping and unwrapping of keys under a fixed storage key, and encryption and decryption under created and imported keys. We show concrete secrecy of plaintexts encrypted under sensitive keys, and integrity of the unload/load operation by reduction to security assumptions on the authenticated encryption (IND-CPA and INT-CTXT), and on the KDF (IND-PRF).

This work will be presented more in-depth during the joint ASA/FCC session, on the 28th of June 2012 at noon.