Mechanized Game-Based Proofs of Security Protocols

Bruno Blanchet
INRIA; École Normale Supérieure, CNRS, Paris, France

Security protocols rely on cryptographic primitives in order to secure communications over insecure networks such as Internet. They are an area of choice for applying formal methods: their design is notoriously error-prone, security errors can have serious consequences, and they are not detected by testing since they appear only in the presence of a malicious adversary. Proving that the protocols are secure is therefore an important challenge. Two main models have been considered in order to obtain such proofs:

- the symbolic model, so-called Dolev-Yao model, in which the cryptographic primitives are black boxes represented by function symbols, the messages are terms over these primitives, and the adversary is restricted to apply only those primitives;
- the computational model, in which the messages are bitstrings, the cryptographic primitives are functions over bitstrings, and the adversary is any probabilistic polynomial-time Turing machine.

The symbolic model facilitates automatic proofs, and was used in most automatic tools for verifying security protocols. The computational model is more realistic, but until recently the proofs in the computational model were manual. In this course, we will present techniques for automating proofs of security protocols in the computational model. We will focus in particular on the tool CryptoVerif [4, 5]. CryptoVerif is an automatic computationally-sound prover for security protocols. It produces proofs presented as sequences of games, like manual proofs of cryptographers [1]; these games are formalized in a probabilistic polynomial-time process calculus. CryptoVerif provides a generic method for specifying security properties of the cryptographic primitives, which can handle shared- and public-key encryption, signatures, message authentication codes, hash functions, Diffie-Hellman key agreements, one-wayness... It can prove secrecy and authentication properties. It produces proofs valid for a number of sessions polynomial in the security parameter, in the presence of an active adversary. It also provides a formula that expresses the probability of success of an attack against the considered protocol as a function of the probability of breaking each primitive. We will illustrate CryptoVerif on two simple examples: the encrypt-then-MAC scheme [2] and the Full Domain Hash (FDH) signature scheme [3].

References

Prerequisites

[1] V. Shoup. *Sequences of Games: a Tool for Taming Complexity in Security Proofs.* Cryptology ePrint Archive, Report 2004/332, 2004.
Available at `http://eprint.iacr.org/2004/332`

[2] M. Bellare, C. Namprempre. *Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm.* In: T. Okamoto (eds), Advances in Cryptology – ASIACRYPT'00, LNCS 1976, pp. 531-545, Springer, 2000.
Available at `http://cseweb.ucsd.edu/ mihir/papers/oem.html`

[3] D. Pointcheval. *Provable Security for Public Key Schemes.* In: Advanced Course on Contemporary Cryptology, Advanced Courses CRM, pp. 133-189. Birkhäuser Publishers, 2005.
Available at `http://www.di.ens.fr/ pointche/pub.php?reference=Po04`

Material covered in the course

[4] B. Blanchet. *A Computationally Sound Mechanized Prover for Security Protocols.* In: IEEE Symposium on Security and Privacy, pp. 140-154, 2006.
Available at `http://www.di.ens.fr/ blanchet/publications/BlanchetOakland06.html`

[5] B. Blanchet, D. Pointcheval. *Automated Security Proofs with Sequences of Games.* In: C. Dwork (Eds), Advances in Cryptology – CRYPTO 2006, LNCS 4117, pp. 537-554, Springer 2006.
Available at
`http://www.di.ens.fr/ blanchet/publications/BlanchetPointchevalCrypto06.html`