Cryptographic Game-style language in EasyCrypt

Jakob Nussbaumer
Universität Bonn

Michael Nüsken
b-it, Universität Bonn

29th Crypto Day, 6/7 September 2018

Mathematical proofs are difficult to verify by a human and even those verifications are error prone. This issue is everlasting and problematic in the field of cryptography. For this reason automatic provers and computer-aided toolsets are on the rise to achieve verification of proofs. Our goal is to have a computer-checked security proof for IPsec. Something similar was done in the project called miTLS (Microsoft, Inria & the Joint Centre, 2014), where a security proof for the TLS handshake (Bhargavan, Fournet, Kohlweiss, Pironti, Strub & Zanella-Béguelin, 2014) was created. Inspired by this we use the same computer-aided toolset, EasyCrypt (IMDEA Software Institute, Inria & École Polytechnique, 2018), which is suited for cryptographic proofs.

The current problem with such toolsets, or automatic provers, lies not only in learning a new computer language. Cryptography is such a new field that a standard, universal language is yet to be established. Therefore every scientific group uses their own style for writing proofs. That means if one wants to implement their own handwritten proof, one first has to translate it into the style the creators use. This resulted into a new goal: having a working example in EasyCrypt, which can be consequently used as a template. As an example we picked the proof that the ElGamal encryption scheme is indistinguishable under chosen message attacks if the Decisional Diffie-Hellman problem is hard. By using this we first present our cryptographic language of games, which is a slightly altered version of (Katz & Lindell, 2016). In this language a security proof is done by letting an imaginary adversary play games. If one can show that a game cannot be won, the underlying system is secure. Normally one has a starting game G_0, which is transformed into a new game G_1. If the difference between G_0 and G_1 is small enough, it is enough to prove that G_1 cannot be won. This technique is called gamehopping and can be applied to a long chain of games. Such a gamehop is used in proofing the security of the ElGamal encryption scheme. We show the implementation of this proof in EasyCrypt. This results in a template one can use for generating other gamehops.

References

Karthikeyan Bhargavan, Cédric Fournet, Markulf Kohlweiss, Alfredo Pironti, Pierre-Yves Strub & Santiago Zanella-Béguelin (2014). *Proving the TLS Handshake Secure (as it is).* URL https://eprint.iacr.org/2014/182.

IMDEA Software Institute, Inria & École Polytechnique (2018). *EasyCrypt: Computer-Aided Cryptographic Proofs.* URL https://www.easycrypt.info/trac/.

Jonathan Katz & Yehuda Lindell (2016). *Introduction to Modern Cryptography.* Chapman & Hall/CRC Cryptography and Network Security Series. AAA. ISBN 9781466570269.

Microsoft, Inria & the Joint Centre (2014). *miTLS: A Verified Reference Implementation of TLS.* URL https://mitls.org/.

Complete work: https://crypto.bit.uni-bonn.de/teaching/18ss/lab/

crypto day matters 29 (2018)
Beiträge vom 29. Krypto-Tag, 06.-07.09.2018, Bosch, Renningen.
doi:10.18420/cdm-2018-29-22