Research Internship Offer

Formal Proofs of Security Protocols using Oblivious Transfers

David Baelde, Hubert Comon, Caroline Fontaine
LSV, École Normale Supérieure de Paris-Saclay
{baelde,comon,fontaine}@lsv.fr

November 7, 2018

1 Background

Our group is working on formal security proofs of cryptographic protocols. Such protocols are distributed programs that rely on cryptographic primitives such as encryption, hash function, signatures, etc. They are used in many applications: secure transactions on Internet, mobile phones, smartcards, RFID chips, electronic voting, etc.

Caroline Fontaine, who joined our group recently, has proposed a protocol for privacy-preserving fingerprinted image distribution that aims at achieving several goals, including traitor tracing, buyer- and item-unlinkability [6]. The formal proof of such a protocol is challenging for several reasons, some of which are the starting point of the proposed research.

We describe below some of the problems that we wish to consider. Depending on the level of the internship (Bachelor, M1, M2), one or more questions could be addressed. They are listed in increasing order of required knowledge.

The internship will take place at LSV, Cachan. There are three advisors as we are all interested in the problem; we expect frequent interactions.

2 Formalizing oblivious transfer

One of the features of our target case study is the use of *oblivious transfer*. Such a primitive allows a sender S (the Merchant in our application) to repeatedly send a key k out of a set of n keys $k_1, \ldots, k_n$ in such a way that S does not know which of $k_1, \ldots, k_n$ is actually received by the receiver R (the Buyer in our application). On the other side, R gets only one of the keys and has no information on the other keys.

There are many ways to realize an oblivious transfer, starting with [5] in 1985 for instance. More recent versions have been proved to achieve the above functionality under various assumptions.

Currently, there is however no specification of such a primitive in formal verification tools such as Proverif [3] or DeepSec [4].

Task 1. The first step would be to design function symbols and equations for oblivious transfers that are amenable to formal proofs in such verification tools. This is not trivial as, for instance, the index i of the chosen key can be seen as a weak secret (as defined and studied in [2] for example).

This task requires (to get) a minimal knowledge on the formal models that are used in the above-mentioned verification tools.

3 Formalizing an idealized content distribution protocol

The content distribution protocol includes other features as, for instance, commitments and the possibility for the Merchant to get half of the keys that have been served, for further tracking of malicious buyers.

Task 2. Formalize in a tool, e.g. ProVerif, the content distribution protocol and its properties and verify it. This step is not trivial either; one of the expected properties, unlinkability, comes in several flavors that have to be understood and formalized in this context.

This task requires (to get) a deeper knowledge of the formal models underlying the verification tools.

4 Perspectives: Formal computational proofs

The previous steps were only considering an idealized attacker (the so-called "Dolev-Yao attacker"). In our group, we develop an approach, based on [1], that allows in principle to automatically prove security properties against a *computational attacker*, i.e., an arbitrary probabilistic polynomial time Turing machine. The idea is to specify formally (in first-order logic) what an attacker cannot do. Such axioms should be computationally sound, assuming some classical hardness assumptions.

Task 3. Design computationally sound axioms for the primitives that are used in the target protocol. The challenges of Task 1 are combined with the difficulty inherent to the more realistic computational model.

This task requires (to get) knowledge on computational security, typically the method of game hopping proofs.

References

[1] Gergei Bana and Hubert Comon-Lundh. A computationally complete symbolic attacker for equivalence properties. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 609–620. ACM, 2014.

[2] Mathieu Baudet. Deciding security of protocols against off-line guessing attacks. In Vijay Atluri, Catherine A. Meadows, and Ari Juels, editors, *Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS 2005, Alexandria, VA, USA, November 7-11, 2005*, pages 16–25. ACM, 2005.

[3] Bruno Blanchet, Vincent Cheval, Xavier Allamigeon, Ben Smyth, and Marc Sylvestre. Proverif: Cryptographic protocol verifier in the formal model. `http://prosecco.gforge.inria.fr/personal/bblanche/proverif/`.

[4] Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. Deepsec prover. `https://github.com/DeepSec-prover/deepsec`.

[5] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[6] Caroline Fontaine, Sébastien Gambs, Julien Lolive, and Cristina Onete. Private asymmetric fingerprinting: A protocol with optimal traitor tracing using tardos codes. In Diego F. Aranha and Alfred Menezes, editors, *Progress in Cryptology - LATINCRYPT 2014 - Third International Conference on Cryptology and Information Security in Latin America, Florianópolis, Brazil, September 17-19, 2014, Revised Selected Papers*, volume 8895 of *Lecture Notes in Computer Science*, pages 199–218. Springer, 2014.