Automated Game-Based Cryptographic Proofs

Gilles Barthe[1], Benjamin Grégoire[2], Sylvain Heraud[2], and Santiago Zanella Béguelin[1]

[1] IMDEA Software Institute, Madrid, Spain,
{Gilles.Barthe,Santiago.Zanella}@imdea.org
[2] INRIA Sophia Antipolis-Méditerranée, France,
{Sylvain.Heraud,Benjamin.Gregoire@inria.fr}

The security of cryptographic schemes is generally proved by reduction, showing that the existence of an efficient adversary attacking the security of a scheme would contradict a security assumption. A typical reduction is structured as a sequence of probabilistic experiments, often called games, where the first experiment in the sequence encodes the security of the scheme against an efficient adversary $\mathcal{A}$, and the last experiment gives an explicit construction that uses $\mathcal{A}$ to efficiently solve a problem assumed to be computationally hard, e.g. computing discrete logarithms. Since relating the probability of success of the initial adversary to the success of the final construction can be involved, intermediate games are introduced to decompose the proof in steps of more manageable complexity. As noticed by Bellare and Rogaway [5] and Halevi [6], game-based cryptographic proofs can be rigorously formalized by taking a code-based approach, representing games as probabilistic programs and justifying proof steps using programming-language techniques.

CertiCrypt [3] is a general framework built on top on the Coq proof assistant to certify the security of game-based cryptographic schemes using a code-based approach. The adoption of programming idioms allows giving precise definitions of games, and paves the way for applying programming language methods to justify proof steps rigorously. Specifically, many proof steps involve establishing observational equivalence between two programs, or proving that they satisfy a relational invariant. These statements are established formally using an equational theory for observational equivalence or a full-fledged relational Hoare logic, and certified program transformations. To date, CertiCrypt has been successfully applied to verify prominent cryptographic constructions, including OAEP [2], FDH [7], and Zero-Knowledge protocols [4].

EasyCrypt [1] is an automated tool that builds machine-checked proofs from *proof sketches*. Proof sketches offer a machine-processable representation of the essence of a security proof, including the sequence of games, relations between the probability of events in those games, and Hoare

logic judgments that justify them. In a nutshell, EasyCrypt implements a verification condition generator that computes for any probabilistic relational Hoare logic judgment a set of verification conditions, expressed in the language of first-order logic, and amenable to automated verification by state-of-the-art tools such as SMT solvers and theorem provers. Moreover, the verification condition generator is *proof-producing*, and generates Coq files that can be machine-checked using the CertiCrypt framework. Additionally, EasyCrypt implements an automated mechanism for proving claims about probabilities. Overall, EasyCrypt demonstrates that provable security can dramatically benefit from automation using state-of-the-art verification technology, and that verifiable game-based proofs can be constructed with only a moderate effort.

References

1. Barthe, G., Grégoire, B., Heraud, S., Zanella Béguelin, S.: Computer-aided security proofs for the working cryptographer. In: Advances in Cryptology – CRYPTO 2011. Lecture Notes in Computer Science, Springer (2011)
2. Barthe, G., Grégoire, B., Lakhnech, Y., Zanella Béguelin, S.: Beyond provable security. Verifiable IND-CCA security of OAEP. In: Topics in Cryptology – CT-RSA 2011. Volume 6558 of Lecture Notes in Computer Science., Berlin, Springer (2011) 180–196
3. Barthe, G., Grégoire, B., Zanella Béguelin, S.: Formal certification of code-based cryptographic proofs. In: 36th ACM SIGPLAN-SIGACT symposium on Principles of Programming Languages, POPL 2009, New York, ACM (2009) 90–101
4. Barthe, G., Hedin, D., Zanella Béguelin, S., Grégoire, B., Heraud, S.: A machine-checked formalization of Sigma-protocols. In: 23rd IEEE Computer Security Foundations symposium, CSF 2010, Los Alamitos, Calif., IEEE Computer Society (2010) 246–260
5. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Advances in Cryptology – EUROCRYPT 2006. Volume 4004 of Lecture Notes in Computer Science., Berlin, Springer (2006) 409–426
6. Halevi, S.: A plausible approach to computer-aided cryptographic proofs. Cryptology ePrint Archive, Report 2005/181 (2005)
7. Zanella Béguelin, S., Grégoire, B., Barthe, G., Olmedo, F.: Formally certifying the security of digital signature schemes. In: 30th IEEE symposium on Security and Privacy, S&P 2009, Los Alamitos, Calif., IEEE Computer Society (2009) 237–250