Automated Proofs for Asymmetric Encryption

Joudicaël Courant, Marion Daubignard, Cristian Ene, Pascal Lafourcade, and Yassine Lakhnech*

Université Grenoble 1, CNRS, Verimag

Abstract. Chosen-ciphertext security is by now a standard security property for asymmetric encryption. Many generic constructions for building secure cryptosystems from primitives with lower level of security have been proposed. Providing security proofs has also become standard practice. There is, however, a lack of automated verification procedures that analyze such cryptosystems and provide security proofs. This paper presents an automated procedure for analyzing generic asymmetric encryption schemes in the random oracle model. It has been applied to several examples of encryption schemes among which the construction of Bellare-Rogaway 1993, of Pointcheval at PKC'2000 and REACT.

1 Introduction

Our day-to-day lives increasingly depend upon information and our ability to manipulate it securely. This requires solutions based on cryptographic systems (primitives and protocols). In 1976, Diffie and Hellman invented public-key cryptography, coined the notion of one-way functions and discussed the relationship between cryptography and complexity theory. Shortly after, the first cryptosystem with a reductionist security proof appeared (Rabin 1979). The next breakthrough towards formal proofs of security was the adoption of computational security for the purpose of rigorously defining the security of cryptographic schemes. In this framework, a system is provably secure if there is a polynomial-time reduction proof from a hard problem to an attack against the security of the system. The provable security framework has been later refined into the exact (also called concrete) security framework where better estimates of the computational complexity of attacks are achieved. While research in the field of provable cryptography has achieved tremendous progress towards rigorously defining the functionalities and requirements of many cryptosystems, little has been done for developing computer-aided proof methods or more generally for investigating a proof theory for cryptosystems as it exists for imperative programs, concurrent systems, reactive systems, etc...

In this paper, we present an automated proof method for analyzing generic asymmetric encryption schemes in the random oracle model (ROM). Generic encryption schemes aim at transforming schemes with weak security properties, such as one-wayness, into schemes with stronger security properties, especially security against chosen ciphertext attacks. Examples of generic encryption

* Grenoble, email:name@imag.fr This work has been partially supported by the ANR projects SCALP, AVOTE and SFINCS.
schemes are \[11, 23, 21, 7, 19, 18, 17\]. The paper contains two main contributions. The first one is a compositional Hoare logic for proving IND-CPA-security. That is, we introduce a simple programming language (to specify encryption algorithms that use one-way functions and hash functions) and an assertion language that allows to state invariants and axioms and rules to establish such invariants. Compositionality of the Hoare logic means that the reasoning follows the structure of the program that specifies the encryption oracle. The assertion language consists of three atomic predicates. The first predicate allows us to express that the value of a variable is indistinguishable from a random value even when given the values of a set of variables. The second predicate allows us to state that it is computationally infeasible to compute the value of a variable given the values of a set of variables. Finally, the third predicate allows us to state that the value of a variable has not been submitted to a hash function.

Transforming the Hoare logic into an (incomplete) automated verification procedure is quite standard. Indeed, we can interpret the logic as a set of rules that tell us how to propagate the invariants backwards. We have done this for our logic resulting in a verification procedure implemented in less than 250 lines of CAML. We have been able to automatically verify IND-CPA security of several schemes among which \[7, 18, 17\]. Our Hoare logic is incomplete for two main reasons. First, IND-CPA security is an observational equivalence-based property, while with our Hoare logic we establish invariants. Nevertheless, as shown in Proposition 1, we can use our Hoare logic to prove IND-CPA security at the price of completeness. That is, we prove a stronger property than IND-CPA. The second reason, which we think is less important, is that for efficiency reasons some axioms are stronger than needed.

The second contribution of the paper presents a simple criterion for plaintext awareness (PA). Plaintext awareness has been introduced by Bellare and Rogaway in \[5\]. It has then been refined in \[1\] such that if an encryption scheme is PA and IND-CPA then it is IND-CCA. Intuitively, PA ensures that an adversary cannot generate a valid cipher without knowing the plaintext, and hence, the decryption oracle is useless for him. The definition of PA is complex and proofs of PA are also often complex. In this paper, we present a simple syntactic criterion that implies plaintext awareness. Roughly speaking the criterion states that the cipher should contain as a sub-string the hash of a bitstring that contains as substrings the plaintext and the random seed. This criterion applies for many schemes such as \[7, 17, 18\] and easy to check. Although (or maybe because) the criterion is simple, the proof of its correctness is complex.

Putting together these two contributions, we get a proof method for IND-CCA security.

An important feature of our method is that it is not based on a global reasoning and global program transformation as it is the case for the game-based approach \[6, 20\]. Indeed, both approaches can be considered complementary as the Hoare logic-based one can be considered as aiming at characterizing, by means of predicates, the set of contexts in which the game transformations can be applied safely.