An overview of the Jasmin language for High-Assurance & High-Speed Cryptography

Benjamin Grégoire*[1]

[1]Inria – L'Institut National de Recherche en Informatique et e n Automatique (INRIA) – France

Résumé

Implementations of cryptographic libraries face a unique combination of challenges: they must be correct, amenable to auditing, optimized for a wide range of architectures, secure in spite of side-channel attacks, etc. As these goals tend to conflict one against each other, common practice usually focuses on some of them, at the expense of the others. At one end of the spectrum, some implementations are highly efficient but provide little correctness guaranty. At the other end, some formally verified artifacts enjoy machine-checked proofs of correctness and security but tend to execute too slowly.

The field of high-assurance cryptography aims at providing cryptographic implementations that feature high-efficiency together with strong guaranties of correctness and security. This talk will present one line of work in this field: the Jasmin workbench. At its heart, the Jasmin programming language is designed to allow cryptography practitioners to implement efficient primitives that are amenable to formal verification. Indeed various tools allow to automatically or interactively prove a range of properties such as safety, constant-time security, functional correctness, cryptographic security. These tools operate at the level of source code. This is sound as the Jasmin compiler is certified, i.e., formally proved to be correct.

In addition to an overview of the Jasmin infrastructure, this talk will discuss recent achievements as well as open questions.

*Intervenant