Equivalence proofs for low-level cryptographic implementations

City and country Nancy, France
Team and research laboratory Team Pesto, at Loria lab (Inria Nancy, CNRS, Université de Lorraine)
Name and address of the advisor Vincent Laporte, Vincent.Laporte@inria.fr
Remuneration Interns get a remuneration in accordance with their status
This internship is supported by the "Cybersécurité" PÉPR
Keywords formal proof, program verification

Context

The Jasmin programming language (Almeida et al. 2017) has been designed for writing high-quality low-level implementations. It has been used in particular to implement high-speed cryptographic primitives. Its strong connection to verification tools, either automated or interactive, enables to formally prove many properties: safety, termination, security, correctness, ...

The compiler that translates this language into assembly is certified using the Coq proof assistant: properties that are proved at the source level are preserved and still hold at the assembly level. These guarantees are achieved without any compromise on efficiency, as witnessed by many case studies such as the indifferenciability proof of the fasted known implementation of SHA-3 for AVX2 (Almeida et al. 2019).

High-speed implementations are usually specific to the target architecture. Since recently, the Jasmin compiler also targets the ARMv7 architecture. Therefore Jasmin programs, such as the LibJade[1] cryptography library, are being ported to that architecture.

Objective of the internship

As Jasmin implementations get adapted to different hardware, they need to be verified again. In spite of the many differences that appear at this implementation level — size of pointers, availability of instruction-set extensions, scheduling of instructions, data layout, etc. — different implementations share a common functional specification. This suggests on one hand that the text of the specification can be reused and on the other hand that the verification can be carried on by means of equivalence proofs. Prior work has shown that equivalence proofs are a powerful tool for verifying Jasmin implementations (Almeida et al. 2020). One aim of this work is to investigate whether this proof methodology is suitable for verification across target architectures. An other aim is of course to actually verify cryptographic primitives implemented for ARMv7.

During the internship, the candidate will get familiar with the Jasmin programming language and the EasyCrypt proof assistant, study a few cryptographic primitives written in Jasmin and targeting the ARMv7 architecture, and show their functional correctness by means of equivalence proofs. More specifically, the task of the intern would be to:

- Verify the functional correctness of reference and optimized implementations targeting ARMv7 using the EasyCrypt proof assistant.
- Study to which extent the availability of an already verified reference implementation targeting an other architecture reduces the proof effort.

Bibliographic references

Almeida, José Bacelar, Manuel Barbosa, Gilles Barthe, Arthur Blot, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Hugo Pacheco, Benedikt Schmidt, and Pierre-Yves Strub. 2017. "Jasmin: High-Assurance and High-Speed Cryptography." In *CCS 2017 - Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1–17. Dallas, United States. https://hal.archives-ouvertes.fr/hal-01649140.

Almeida, José Bacelar, Manuel Barbosa, Gilles Barthe, Benjamin Grégoire, Adrien Koutsos, Vincent Laporte, Tiago Oliveira, and Pierre-Yves Strub. 2020. "The Last Mile: High-Assurance and High-Speed Cryptographic

[1] https://github.com/formosa-crypto/libjade

Implementations." In *2020 IEEE Symposium on Security and Privacy* (*SP*), 965–82. https://doi.org/10.1109/SP40000.2020.00028.

Almeida, José Bacelar, Cécile Baritel-Ruet, Manuel Barbosa, Gilles Barthe, François Dupressoir, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Alley Stoughton, and Pierre-Yves Strub. 2019. "Machine-Checked Proofs for Cryptographic Standards: Indifferentiability of Sponge and Secure High-Assurance Implementations of SHA-3." In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 1607–22. CCS '19. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3319535.3363211.

Expected ability of the student

The candidate should be familiar with formal semantics of programming languages. No knowledge in security and cryptography is required. Prior experience with a proof assistant such as Coq would be appreciated but is not mandatory.