Internship Proposal: Provably Correct Rust Implementations of the SPHINCS+ Post-Quantum Cryptographic Primitive

Location of the internship: Prosecco Team, Inria Paris, France
Advisor: Aymeric Fromherz

Context. The security of modern cryptographic constructions typically relies on mathematical problems assumed to be hard to solve, e.g., finding a discrete logarithm (DLOG), or factorizing large numbers into their prime decomposition (FACT). However, such cryptographic schemes are threatened by recent technological advances, namely, ongoing works on quantum computing. Indeed, due to differences in the computing model of classical and quantum computers, both the DLOG and FACT problems would be easily solved by quantum computers. To tackle this issue, many researchers have investigated the development of *post-quantum* cryptographic algorithms, leading to several proposals currently ongoing standardization.

However, while these algorithms have been heavily studied from a theoretical perspective, developing high-performance and secure implementations has been historically tricky. Case in point, the OpenSSL library, one of the most widely-used cryptographic providers, regularly reports security vulnerabilities due to errors in the implementation. In some cases, these vulnerabilities can lead to denial of service attacks or leaks of confidential data, thus drastically impacting both the security of applications and the privacy of users.

Given both the critical aspect and the complexity of highly optimized cryptographic implementations, many previous works investigated the application of formal verification techniques to cryptography [8, 9, 1, 5]. Compared to traditional testing techniques, formal verification approaches aim to soundly analyze all possible execution paths in a program, and therefore allows to statically rule out entire classes of bugs and vulnerabilities. Over the course of this internship, we propose to study the application of such techniques to implementations of post-quantum cryptographic primitives, and particularly the SPHINCS+ signature scheme [2, 3].

Goals. The goals of this internship is to extend and adapt existing formal verification methodologies to apply them to the SPHINCS+ cryptographic primitive. In particular, the work will rely on the Aeneas toolchain [7, 6], which allows to formally reason about Rust implementations through a functional translation to the Lean proof assistant [4]. A successful intern will therefore implement a subset of SPHINCS+ in Rust, identify limitations and extend Aeneas to support the subset of Rust needed for reasoning about their implementation, establish the panic-freedom and functional correctness of the Rust implementation using the Lean proof assistant, and suggest improvements to the Lean proof automation to simplify the verification effort. A possible internship plan is therefore the following:

- Background reading on SPHINCS+ and hash-based signatures, Rust tutorial
- Implementation of a simplified version of SPHINCS+ in Rust, getting familiar with Aeneas and Lean
- Attempt at a first manual proof of a subset of SPHINCS+, identification of pain points for verification
- Familiarization with Lean metaprogramming, extension of existing Lean-based automation for Aeneas

- Iteration and extension of the proof, alongside further development of proof automation
- Benchmarking of the implementation, comparison with reference implementation

Qualifications. This internship of 4 to 6 months would be hosted by the Prosecco Team at Inria Paris, and is particularly well-suited to a student pursuing a master's degree in computer science. The preferred qualifications for the student at the beginning of the internship would be:

- Fluency with the Rust programming language
- Experience with proof assistants, ideally Lean
- Knowledge of type systems and understanding of program semantics
- Familiarity with first-order logic and program specification
- Motivation to work with, and improve experimental research tools

References

[1] José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, Arthur Blot, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Hugo Pacheco, Benedikt Schmidt, and Pierre-Yves Strub. Jasmin: High-assurance and high-speed cryptography. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2017.

[2] Daniel J Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. Sphincs: practical stateless hash-based signatures. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 368–397. Springer, 2015.

[3] Daniel J Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. The sphincs+ signature framework. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 2129–2146, 2019.

[4] Gabriel Ebner, Sebastian Ullrich, Jared Roesch, Jeremy Avigad, and Leonardo de Moura. A metaprogramming framework for formal verification. In *Proceedings of the International Conference on Functional Programming (ICFP)*, 2017.

[5] A. Erbsen, J. Philipoom, J. Gross, R. Sloan, and A. Chlipala. Simple high-level code for cryptographic arithmetic - with proofs, without compromises. In *Proceedings of the IEEE Symposium on Security and Privacy (OAKLAND)*, 2019.

[6] Son Ho, Aymeric Fromherz, and Jonathan Protzenko. Sound borrow-checking for Rust via symbolic semantics. *Proceedings of the ACM on Programming Languages*, 8(ICFP):426–454, 2024.

[7] Son Ho and Jonathan Protzenko. Aeneas: Rust verification by functional translation. *Proceedings of the ACM on Programming Languages*, 6(ICFP):711–741, 2022.

[8] Marina Polubelova, Karthikeyan Bhargavan, Jonathan Protzenko, Benjamin Beurdouche, Aymeric Fromherz, Natalia Kulatova, and Santiago Zanella-Béguelin. HACLxN: Verified generic SIMD crypto (for all your favourite platforms). In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2020.

[9] Jonathan Protzenko, Bryan Parno, Aymeric Fromherz, Chris Hawblitzel, Marina Polubelova, Karthikeyan Bhargavan, Benjamin Beurdouche, Joonwon Choi, Antoine Delignat-Lavaud, Cédric Fournet, Natalia Kulatova, Tahina Ramananandro, Aseem Rastogi, Nikhil Swamy, Christoph M. Wintersteiger, and Santiago Zanella-Béguelin. EverCrypt: A fast, verified, cross-platform cryptographic provider. In *Proceedings of the IEEE Symposium on Security and Privacy (OAKLAND)*, 2020.