Type-Based Cryptographic Verification for TLS 1.0

K. Bhargavan C. Fournet N. Guts
Microsoft Research-INRIA Joint Centre, Orsay

Internship Proposal 2010
(2-3 months)

Background TLS 1.0 [Dierks and Rescorla, 2008] is one of the most widely deployed protocols for communications security. In recent work [Bhargavan et al., 2008], we developed a first verified reference implementation of TLS, using ML as an implementation language, and using translations to specialized cryptographic analyzers for verification: ProVerif Blanchet [2001] for symbolic cryptography, and CryptoVerif [Blanchet, 2006] for computational cryptography. This implementation is at the limit of what can be verified using global algorithms: whole-program security analysis does not scale well beyond a few thousand lines of code and, even with careful rewriting of the source code, verification may take hours or sometimes not terminate. Besides, our implementation remains much less flexible than the standard, its programming style is deeply influenced by the underlying verification techniques, and the verified properties do not cover important cases, such as mutual certificate-based authentication.

In a parallel line of work, we are investigating compositional proof techniques and tools, based on cryptographic libraries and refinement type-checking with F7 [Bengtson et al., 2008, Bhargavan et al., 2010]. Our results so far confirm that refinement types enable faster, compositional verification in the symbolic model of cryptography, at the cost of writing non-trivial type annotations. More tentatively, experiments with partial type inference suggest that the burden of type annotations can be largely mitigated.

Project In this internship, we propose to validate and improve our type-based tools by using this TLS codebase as a major case study. The internship involves ML programming, experimental verification work with F7, as well as more theoretical developments regarding refinement-type inference and computational soundness of typechecking.

Keywords: Functional programming, Security protocols, Cryptography, Type theory, Program verification.

References

J. Bengtson, K. Bhargavan, C. Fournet, A. D. Gordon, and S. Maffeis. Refinement types for secure implementations. In *21st IEEE Computer Security Foundations Symposium (CSF'08)*, pages 17–32, 2008. PDF.

K. Bhargavan, C. Fournet, R. Corin, and E. Zalinescu. Cryptographically verified implementations for TLS. In *15th ACM conference on Computer and Communications Security (CCS'08)*, pages 459–468. ACM, 2008. PDF.

K. Bhargavan, C. Fournet, and A. D. Gordon. Modular verification of security protocol code by typing. In *ACM Symposium on Principles of Programming Languages (POPL'10)*, pages 445–456. ACM, 2010.

B. Blanchet. An efficient cryptographic protocol verifier based on Prolog rules. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop*, pages 82–96. IEEE Computer Society Press, 2001. ProVerif Homepage.

B. Blanchet. A computationally sound mechanized prover for security protocols. In *IEEE Symposium on Security and Privacy*, pages 140–154. IEEE Computer Society, 2006.

T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2, 2008.