

Verified Cryptographic Protocol Implementations

K. Bhargavan and A. Pironti

The security of web applications relies on cryptographic protocols such as TLS, SSH, Kerberos, and IPSec. Still, leading implementations of these protocols have been found to have serious bugs years after their release. Our goal is to develop and verify reference implementations of cryptographic protocols.

In previous work, cryptographic provers such as ProVerif and CryptoVerif have been used to verify reference implementations of TLS in ML [Bhargavan et al., 2008], and of SSH in Java [Pironti et al., In Press]. However, both these implementations are at the limit of what can be globally verified. The target properties are, in general, undecidable and, even with careful rewriting of the source code, verification may take hours or sometimes not terminate. To obtain security proofs for such implementations, the analysis must often leave out important protocol features, such as recursive certificate chain verification.

More recently, Bengtson et al. [2008] propose a scalable proof technique for F# programs based on cryptographic libraries and refinement types. Their typechecker F7 has been used to verify protocol implementations in both the symbolic model [Bhargavan et al., 2010] and the computational model [Fournet et al., 2011]. They show that refinement types enable faster, compositional verification, but at the cost of writing non-trivial type annotations.

Internship.

In this internship, we propose to write and verify implementations of the SSH protocol. The precise topic will be decided after discussion with the student, but during the course of the internship, the student will learn to do the following:

- Write cryptographic protocol implementations in F#
- Verify protocol implementations in the symbolic and computational model by typing using F7
- Analyze the security of other implementations, written in C and Java
- Find attacks on both the cryptographic protocol and on its implementations

Some prior knowledge of cryptography, verification tools, functional programming, and type systems will be an advantage, but is not mandatory.

The internship will be located at INRIA in Paris. The dates of the internship and its duration are flexible and students of any nationality may apply.

Masters students who are thinking of doing a Ph.D. are encouraged to apply for a six-month internship (i.e. a French M1/M2 stage). Ph.D. students and advanced undergraduate students (with adequate background) may apply for a three-month summer internship. We expect the research carried out during the internship will form the major part of a Masters-level thesis and lead to a conference publication. All interns are funded under the ERC grant CRYSP and successful internships are expected to lead to funded Ph.D. studentships.

Application Details.

To apply, send an email describing your research interests, and including your CV and the names and email addresses of one or two referees (professors or prior employers), to karthikeyan DOT bhargavan AT inria DOT fr. The deadline for applications is **January 5, 2012**. Positions will be kept open until filled.

The intern will have the opportunity to work closely with a team of researchers from INRIA and MSR-INRIA, including B. Blanchet, G. Steel, C. Fournet, and P-Y. Strub.

Related Proposals.

- Security Types for Web Applications
- Building Secure Smartphone Applications

References

- J. Bengtson, K. Bhargavan, C. Fournet, A. D. Gordon, and S. Maffei. 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, 2010. PDF.
- C. Fournet, M. Kohlweiss, and P.-Y. Strub. Modular code-based cryptographic verification. In *Proceedings of the 18th ACM conference on Computer and communications security, CCS '11*, pages 341–350, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0948-6. .
- A. Pironti, D. Pozza, and R. Sisto. Formally-based semi-automatic implementation of an open security protocol. *Journal of Systems and Software*, In Press.