Research engineer position

Development of SQUIRREL: an interactive prover for protocol verification

Laboratory, institution and university. The position will be funded by the ERC project POPSTAR, and will take place at IRISA in Rennes or possibly in Paris depending on the choice of the candidate. The candidate will be an employee of CNRS and enjoy associated benefits (unemployment, retirement and health insurance, etc.). The term of the position is one year, renewal is possible.

Contact. Interested applicants are advised to send their application including CV, motivation letter, and references to Stéphanie Delaune (email: stephanie.delaune@irisa.fr), David Baelde (email: baelde@lsv.fr), and Adrien Koutsos (email: adrien.koutsos@inria.fr).

Context. Security protocols are distributed programs that aim at securing all kinds of communications, by using cryptography to ensure security properties, such as confidentiality, authentication or anonymity. Such protocols are widely deployed, e.g. for electronic commerce on the Internet, in banking networks, mobile phones and more recently online elections. As the goal is to provide security guarantees even when communicating over an untrusted network – typically the Internet – it turns out that these protocols are extremely difficult to get right. Formal methods have proved to be a very useful tool to detect errors, and verify the correctness of security protocols. Traditionally two approaches have been used: the computational one, which captures strong notions of security and offers guarantees against all probabilistic polynomial-time attackers, and the symbolic one in which things are modelled more abstractly and which is more amenable to automation. To get an idea of this research area, the interested reader can consult [CK11].

A few years ago, Bana and Comon proposed a new approach to security proofs [BC14]. This approach, which they call computationally complete symbolic attacker (CCSA), uses the symbolic formal setting of first-order logic, but avoids the limitations of the symbolic model. This approach has been demonstrated on various protocols to obtain formal proofs of security, e.g. [CK17]. Until recently, these proofs were only pen-and-paper formal proofs, limiting the scalability and trustworthiness of the CCSA approach.

Recently, a meta-logic over the CCSA logic has been developped [BDJ+21] and implemented as part of a new interactive theorem prover[1]. This prover takes as input protocol specifications written in a dialect of the applied pi-caclulus. It allows users to specify reachability and equivalence properties (encoding security and privacy requirements of the protocol) and to prove them using tactics. The prover features basic automated reasoning capabilities, in an attempt to leave to the user only the high-level aspects of the proof. The SQUIRREL prover is written in OCaml and weighs about 25k lines of code. It currently does not rely on external tools, but is integrated with Proof General for interactive proof development in Emacs. The development of SQUIRREL takes place on Inria's Gitlab and Github, and makes intensive use of testing and continuous integration.

[1] SQUIRREL prover: `https://github.com/squirrel-prover/squirrel-prover`

Mission. The main objective of this position will be to contribute to the development of the SQUIRREL prover. This can be achieved in various ways depending on the skills and expectations of the successful applicant. The list below is non-exhaustive and will be discussed with the applicant.

- *Improvement of the user inteface.* At present, the protocol and properties specification as well as the proofs are in text mode. We would like to develop a web output summarizing the results to allow the average user to access the results produced by the SQUIRREL prover.
- *Graphical representation during proof development.* When performing a proof using SQUIRREL, the current goal and the available hypotheses are shown in text mode. We would like to provide a graphical representation to help the developer to complete its proof. For this, we can take inspiration from the user interface provided in the Tamarin prover[2].
- *Executability of the specification.* A difficulty when it comes to analyzing a protocol is to write a correct specification first. To verify that the specification produced is reasonable, one option is to simulate protocol executions. We would like to add this feature to the SQUIRREL prover.
- *Interfacing with SMT solvers.* We would like to make the tool as automatic as possible by discharging some simple goals using some proof automation. Some dedicated algorithms have already been implemented. However to go further, we would like to discharge some goals to SMT solvers. For this, an interface to e.g. the Why3 SMT solver will be set up.

Requirements. Am engineering degree or a master degree in computer science is required. A PhD is *not* required. We are looking for candidates with good skills in OCaml programming. In particular, the ability to write, understand and debug clean, maintainable OCaml code is mandatory. Some skills on foundations of Computer Science (logic, automated deduction, ...) will be appreciated. Some knowledge in security is an asset but is not mandatory. The knowledge of French language is not compulsory for the position.
Tentative starting date: 1st July 2021.
Monthly gross salary: 2 100 € – 3 000 €, depending on the experience of the candidate.

References

[BC14] Gergei Bana and Hubert Comon-Lundh. A computationally complete symbolic attacker for equivalence properties. In *ACM Conference on Computer and Communications Security*, pages 609–620. ACM, 2014.

[BDJ+21] David Baelde, Stéphanie Delaune, Charlie Jacomme, Adrien Koutsos, and Solène Moreau. An interactive prover for protocol verification in the computational model. In Alina Oprea and Thorsten Holz, editors, *Proceedings of the 42nd IEEE Symposium on Security and Privacy (S&P'21)*, San Francisco, California, USA, May 2021. IEEE Computer Society Press.

[CK11] Véronique Cortier and Steve Kremer, editors. *Formal Models and Techniques for Analyzing Security Protocols*, volume 5 of *Cryptology and Information Security Series*. IOS Press, 2011.

[CK17] Hubert Comon and Adrien Koutsos. Formal computational unlinkability proofs of RFID protocols. In *CSF*, pages 100–114. IEEE Computer Society, 2017.

[2] TAMARIN prover: `http://tamarin-prover.github.io`