Research engineer and postdoc positions

Development of Squirrel,
a proof assistant for security protocols

Summary. We are looking for people to join the Squirrel project (described below) as postdocs or research engineers, for one or more years. Funding will be provided by the cybersecurity PEPR and possibly some other projects. The candidate will be an employee of CNRS and enjoy associated benefits: unemployment, retirement and health insurance, etc.
Tentative starting date: September 1st, 2022.
Monthly gross salary: 2 100 € – 3 000 €, depending on the experience of the candidate.

Location. This announcement primarily concerns job positions at IRISA, Rennes. Similar positions might be available at Inria Paris or LMF / Université Paris-Saclay. This may be discussed with the contacts given below, but you may also contact directly Adrien Koutsos `adrien.koutsos@inria.fr` or Caroline Fontaine `caroline.fontaine@lsv.fr`.

Contact. Please send your applications including CV, motivation letter, and references to Stéphanie Delaune and David Baelde by email: `stephanie.delaune@irisa.fr`, `david.baelde@irisa.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. The goal, *i.e.*, providing security guarantees even when communicating over an untrusted network such as internet, is extremely difficult to achieve. 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 (at least on the symbolic approach), 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 proof assistant: Squirrel[1]. This work has brought the first mechanized proofs of security protocols using the CCSA methodology. The approach is subject to active research since then, and has notably been extended to support post-quantum security [CFJ22] and protocols with states [BDKM22].

The Squirrel proof assistant 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 38k 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 Gitlab and Github, and makes intensive use of testing and continuous integration.

[1] See `https://squirrel-prover.github.io/`.

Mission. The main objectives will be to contribute to the development of the Squirrel prover. This could be on practical and/or theoretical aspects, and can be achieved in various ways depending on the skills and expectations of the successful applicants. The list below is non-exhaustive and will be discussed with the applicant.

- *User inteface.* At present, the protocol and properties specification as well as the proofs are in text mode. We are experimenting with a web-based graphical output, and further web integration could be considered.
- *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. Another possibility would be to translate specification between Squirrel and other protocol verification tools.
- *Proof automation.* We have been experimenting with the use of SMT solvers to automate some proofs. Some native developments (e.g. metavariables, more powerful tactics, user-defined tacticals) would also be useful in this respect.
- *Case studies and extensions.* The job will include working *with* the tool and not only *on* its code base. This will be crucial to understand the requirements. Further, the application might join one of several ongoing formalization efforts, which include proof developments but also the extension of the prover with new tactics, reflecting new security assumptions or new reasoning techniques.
- *Maintenance and documentation.* Last but not least, part of the mission could be to maintain the code base, which has been rapidly changing over the past two years and need some refactoring in several places. The overall goal will be to improve robustness and ease future developments. User documentation is also lacking; automated documentation might be considered.

Requirements. An engineering degree or a master degree in computer science is required. A PhD is *not* required but of course it depends on the tasks chosen in the non-exhaustive list above. For tasks related to programming, 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. To work on more theoretical aspects, skills on foundations of Computer Science (logic, automated deduction, proof assistants. . .) will be needed. Some knowledge in security is an asset but is not mandatory. The knowledge of French language is not compulsory for the position.

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.

[BDKM22] David Baelde, Stéphanie Delaune, Adrien Koutsos, and Solène Moreau. Cracking the Stateful Nut. In *CSF 2022 - 35th IEEE Computer Security Foundations Symposium*, Haifa, Israel, August 2022.

[CFJ22] Cas Cremers, Caroline Fontaine, and Charlie Jacomme. A logic and an interactive prover for the computational post-quantum security of protocols. *IACR Cryptol. ePrint Arch.*, page 401, 2022. Accepted to S&P'22.

[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.