Internship project
Formalization of multi-terms in the Squirrel Prover

Charlie Jacomme

September 10, 2024

Keywords This project will mix higher-order logic, security protocols modelings, development of new proof techniques, and OCaml implementation inside an interactive prover.

General Context The Squirrel prover [2] is an interactive prover designed to prove the security of cryptographic protocols, by only working inside the context of a logic and abstracting away the usual probabilistic and reductionist arguments. If the original foundations of Squirrel's logic were a first order logic designed by Bana and Comon [4], it was recently extended to a full high-order logic [3]. This recent extension opens up new possibility in terms of expressivity, that we aim to explore here.

The project The higher-order logic behind Squirrel models possible executions of a protocol using recursive function. Without giving the full definitions, for a given protocol, it is possible to reason over all its executions by looking at the term $\mathsf{frame}@\tau$, which recursively unfold as the list of all messages sent over the network and seen by the attacker (see [1, Section 3] for more details).

In the formal description of the logic, everything is defined for a given protocol, which yields a concrete definition of frame based on the concrete inputs and outputs that a protocol may perform. A significant gap exists in the actual interactive prover (`squirrel-prover.github.io`), where several protocols can be defined, each protocol corresponding to a so-called system, and terms and lemmas can relate to two systems (so called bi-system) at once.

The goals of this project are:

- to formalize this gap, casting inside the theory the current behaviors of the Squirrel prover, formally capturing how multiple systems may cohabits inside terms (yielding so-called multi-terms) and formulas, what are the axioms instantiated when doing so, ... This should not require extensions to the logic, but will require a deep understanding of the logic in order to properly express everything inside of it.

- based on the formalization, to develop inside the tool (OCaml programing) supports for more complex handling of multiple systems so that terms can relate to more than two, and develop new reasoning techniques for proofs. Transitivity arguments and functional equalities of systems may be explored.

References

[1] David Baelde, Stéphanie Delaune, Charlie Jacomme, Adrien Koutsos, and Joseph Lallemand. The squirrel prover and its logic. *ACM SIGLOG News*, 11(2):62–83, 2024.

[2] David Baelde, Stéphanie Delaune, Charlie Jacomme, Adrien Koutsos, and Solène Moreau. An Interactive Prover for Protocol Verification in the Computational Model. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 537–554, May 2021. ISSN: 2375-1207.

[3] David Baelde, Adrien Koutsos, and Joseph Lallemand. A Higher-Order Indistinguishability Logic for Cryptographic Reasoning. In *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, June 2023.

[4] Gergei Bana and Hubert Comon-Lundh. A Computationally Complete Symbolic Attacker for Equivalence Properties. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS'14)*, pages 609–620, Scottsdale, Arizona, USA, November 2014. ACM Press.