# Automated reasoning in a meta-logic for cryptographic protocols

David Baelde Adrien Koutsos

ENS Paris-Saclay & Inria Paris

Security protocols are small concurrent programs that rely on cryptographic primitives to achieve various security and privacy goals: secrecy, authentication, anonymity, untraceability, etc.

There is a long tradition of using formal methods to analyze security protocols. In the Dolev-Yao approach [DY81], messages are modelled as formal terms and attacker capabilities are modelled as inference rules or an equational theory over messages. The resulting models, which are called *symbolic* models, are amenable to automatic verification. Techniques from rewriting, logic, constraint solving, etc. have been adapted to this setting, resulting in successful verification systems such as Proverif [Bla01], Tamarin [Mei+13], Akiss [Cha+16], or Deepsec [CKR18].

The symbolic models are approximations of the cryptographer's standard model, where the attacker does not have a fixed set of capabilities, but can be an arbitrary probabilistic polynomial-time (PPTIME) Turing machine. Recently, several techniques have been proposed to mechanize the cryptographic proofs and more generally verify protocols in this standard model (Cryptoverif [Bla06], Easycrypt [Bar+11], CryptHOL [BLS20], etc.)

Bana and Comon have proposed in [BC14] a new approach to prove indistinguishabilities between sequences of cryptographic messages, in the computational model. It is based on first-order logic, and involves axioms that must be sound wrt. particular models where function symbols are interpreted as PPTIME Turing machines. In particular, cryptographic assumptions translate to axiom schemes that are used to prove indistinguishabilities involving the corresponding cryptographic primitives. This logic can be used to verify protocols whose runs are bounded, by enumerating all traces and showing indistinguishabilities of the corresponding sequences of messages for each trace; this approach has enabled the analysis of various kinds of protocols [BC14; CK17; Kou19; BCE18].

A team of researchers including Baelde and Koutsos have recently proposed to elaborate on the logic of Bana and Comon by designing a meta-logic, which allows to reason on arbitrary execution traces of a protocol rather than on fixed sequences of messages. Compared to previous methods, this can provide guarantees for protocols with unbounded executions. This meta-logic and its proof systems have been implemented in a dedicated proof assistant, called Squirrel [Bae+20]. The proof technique supported by our meta-logic is fundamentally different from the ones provided by the above-mentioned tools providing guarantees in the computational model. It completely hides probabilities from the user, and does not proceed by game transformations or using a Hoare logic. Instead, Squirrel reasons directly on protocol traces, in a way that is similar e.g. to what is done in the Tamarin prover for a symbolic model. This approach has given promising results on a first set of case studies [Bae+20]. One of the next challenges will be to tackle richer and larger protocols, which will likely require more automation.

The intern will work on automated reasoning in our meta-logic. Although the meta-logic has a probabilitistic semantics, it is possible to use standard techniques from predicate calculus in that context. More specifically, the intern will investigate the use of SMT solvers to automatically discharge some goals. A first goal will be to replace current automated reasoning procedures by the use of SMT solvers, which should then be able to take into account a large class of axioms used in pratice. Depending on the performance of this method, and on the intern's taste, several additional problems can be considered:

- Getting accustomed with our system will necessarily involve the development of simple proofs on some example protocols. More complex case studies will be considered once automation has been improved; in particular, reasoning on protocols with states should benefit a lot from automation. Ambitious case studies include the 5G AKA protocol, Signal, MLS... but simplifications of these protocols would already be difficult challenges for the internship.

- The primary goal of the internship is to use SMT solvers to automate our meta-logic; the method needs to be sound but will likely be incomplete. To go further, on the theoretical side, completeness and decidability issues can be considered for various fragments of our meta-logic.

- It is also possible to design more aggressive automated reasoning. In the first stage, the aim will be to automate "low-level" reasoning, leaving the "high-level" use of the more complex cryptographic axioms to the user. A possible

second stage will be to consider the inclusion of these cryptographic axioms as part of the SMT-based automation.

- Finally, the intern will naturally be involved in the ongoing development of our meta-logic (whose semantics might change slightly) and proof assistant (which needs to be robustly engineered).

Pre-requisites in logic, operational semantics, automated deduction, etc. are more important than in security protocols. This internship can lead to a PhD.

# References


[Bae+20] David Baelde, Stéphanie Delaune, Charlie Jacomme, Adrien Koutsos, and Solène Moreau. *The Squirrel Prover (paper and source code for anonymous submission)*. 2020. URL: `https://github.com/squirrel-submission-sp21/squirrel-prover`.

[BCE18] Gergei Bana, Rohit Chadha, and Ajay Kumar Eeralla. "Formal analysis of vote privacy using computationally complete symbolic attacker". In: *European Symposium on Research in Computer Security*. Springer. 2018, pp. 350–372.

[BC14] Gergei Bana and Hubert Comon-Lundh. "A Computationally Complete Symbolic Attacker for Equivalence Properties". In: *ACM Conference on Computer and Communications Security*. ACM, 2014, pp. 609–620.

[Bar+11] Gilles Barthe, Benjamin Grégoire, Sylvain Heraud, and Santiago Zanella Béguelin. "Computer-Aided Security Proofs for the Working Cryptographer". In: *CRYPTO*. Vol. 6841. Lecture Notes in Computer Science. Springer, 2011, pp. 71–90.

[BLS20] David A. Basin, Andreas Lochbihler, and S. Reza Sefidgar. "CryptHOL: Game-Based Proofs in Higher-Order Logic". In: *J. Cryptology* 33.2 (2020), pp. 494–566.

[Bla06] Bruno Blanchet. "A Computationally Sound Mechanized Prover for Security Protocols". In: *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2006, pp. 140–154.

[Bla01] Bruno Blanchet. "An Efficient Cryptographic Protocol Verifier Based on Prolog Rules". In: *CSFW*. IEEE Computer Society, 2001, pp. 82–96.

[Cha+16] Rohit Chadha, Vincent Cheval, Ştefan Ciobâcă, and Steve Kremer. "Automated verification of equivalence properties of cryptographic protocols". In: *ACM Transactions on Computational Logic (TOCL)* 17.4 (2016), pp. 1–32.

[CKR18] Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. "The DEEPSEC Prover". In: *CAV (2)*. Vol. 10982. Lecture Notes in Computer Science. Springer, 2018, pp. 28–36.

[CK17] Hubert Comon and Adrien Koutsos. "Formal Computational Unlinkability Proofs of RFID Protocols". In: *CSF*. IEEE Computer Society, 2017, pp. 100–114.

[DY81] Danny Dolev and Andrew Chi-Chih Yao. "On the Security of Public Key Protocols (Extended Abstract)". In: *FOCS*. IEEE Computer Society, 1981, pp. 350–357.

[Kou19] Adrien Koutsos. "The 5G-AKA authentication protocol privacy". In: *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2019, pp. 464–479.

[Mei+13] Simon Meier, Benedikt Schmidt, Cas Cremers, and David A. Basin. "The TAMARIN Prover for the Symbolic Analysis of Security Protocols". In: *CAV*. Vol. 8044. Lecture Notes in Computer Science. Springer, 2013, pp. 696–701.