A Secrecy Logic and the Post-Compromise Security of an Asymmetric Ratchet

Clément Hérouard
Inria Nancy Grand-Est, Université de Lorraine, LORIA
Nancy, France
clement.herouard@inria.fr

Charlie Jacomme
Université de Lorraine, CNRS, Inria, LORIA
Nancy, France
charlie.jacomme@inria.fr

Joseph Lallemand
Univ. Rennes, CNRS, IRISA
Rennes, France
joseph.lallemand@irisa.fr

Adrien Koutsos
Inria
Paris, France
adrien.koutsos@inria.fr

Abstract

Security proofs for cryptographic protocols are notably complex and error-prone, in particular in the computational model. Computer-aided cryptography aims at increasing the confidence in these proofs, by mechanising them and formally verifying them with automated tools. One such tool is the Squirrel proof assistant. In this paper, we use it to prove the Post-Compromise Security (PCS) of an asymmetric ratchet mechanism used to generate shared keys, such as the one featured in the Signal protocol. While Squirrel is convenient to study stateful protocols such as the ratchet, the analysis his is made particularly challenging by the fact that the usual notion of secrecy for Squirrel, *i.e.* real-or-random secrecy, is not well-suited to that protocol. We instead define predicates and a proof system for *non-deducibility*, a weaker notion of secrecy, which we found to be the notion needed to study the asymmetric ratchet. We establish the soundness of the proof system, implement it within the Squirrel prover, and use it to write a proof of PCS for the asymmetric ratchet, which is the first mechanised such proof.

Keywords

Computer-Aided Cryptography, Formal verification, Security protocols

1 Introduction

Cryptographic protocols are used to ensure the security of our communications, in many contexts such as mobile telephony, payment, web, or electronic voting, and are therefore critical components of our society. Provable security [14] aims at establishing strong guarantees regarding the level of security these protocols achieve, by providing mathematical proofs, typically showing that the security of a protocol cannot be broken without breaking some underlying computational hardness assumption. Computer-aided cryptography [8] aims at further increasing the confidence in cryptographic proofs, by providing formal frameworks to mechanise and machine-check them. This research area has been quite fruitful, with many tools and frameworks under active development such as notably CryptoVerif [11], EasyCrypt [9], Squirrel [2], SSProve [16], or CryptHOL [10].

The Squirrel Prover. Squirrel [2] is an interactive proof assistant dedicated to the verification of cryptographic protocols, which is particularly well-suited to stateful protocols [3]. It relies on a higher-order probabilistic logic [5] to encode the interaction of an adversary and a protocol, and models cryptographic properties using two main built-in security predicates: an indistinguishability atom $\sim$ representing computational indistinguishability, and a reachability atom $[\,\cdot\,]$ representing overwhelming truth (*i.e.* truth up to an at-most negligible probability of error). Squirrel captures cryptographic arguments using high-level reasoning rules that directly operate at the level of its cryptographic predicates. It provides some support for proof automation, both for cryptographic arguments — *e.g.* it can automatically reduce cryptographic indistinguishabilities to hardness assumptions [6] — and for generic logical reasoning using SMTs [4]. The tool has been used to verify a number of protocols, including the YubiKey protocol [3] and the FOO [15] e-voting protocol [7].

Signal and the Double Ratchet. Signal is a widely deployed secure instant messaging protocol, which is used not only by the Signal messenger itself, but also notably by WhatsApp and Facebook Messenger. Signal aims at providing strong guarantees regarding the confidentiality of the messages: *perfect forward secrecy* (PFS), which states that compromising participants at some points in time should not reveal earlier messages, and *post-compromise security* (PCS), according to which it must be possible, after some participants have been compromised, to restore, or *heal*, the security of the conversation for future messages. This property is of particular interest, since it makes real-world attacks more difficult: an attacker would need to compromise the current keys at each step to keep the attack ongoing, as they would otherwise be replaced with healed keys. To achieve this, the keys used to encrypt each message are updated frequently. The Signal protocol relies on two main mechanisms to manage these keys. First, an initial exchange called X3DH (extended triple Diffie-Hellman) allows two agents willing to communicate to establish a shared secret, which will be used to derive the initial keys. Then, the *double ratchet* is in charge of updating the keys and deriving new encryption keys for each message. It is itself composed of two parts, called the *symmetric* and *asymmetric* ratchets.

The symmetric ratchet is used to derive a sequence of so-called *chain keys* from an initial *root key*, from which are then derived *message keys*, used to encrypt successively each message sent from one agent to the other. Basically, each chain key is obtained by applying a Key Derivation Function (KDF) to the previous one.

While this mechanism already provides PFS (indeed, it should not be possible to invert the KDF to obtain earlier keys from one given chain key), it is not sufficient for PCS, which intuitively requires fresh keying material to be introduced for the healing to be possible.

The asymmetric ratchet mechanism performs this operation. It constructs a sequence of root keys, which are then successively used as starting points for the symmetric ratchet. Essentially, each root key is obtained by applying a KDF H to the concatenation of the previous one r and a newly exchanged shared Diffie-Hellman (DH) secret s. This way, even if the previous key r has been compromised, the new key $\mathsf{H}(\langle r, s\rangle)$ is a fresh key, provided that the DH exchanged has not been tampered with and that s is thus indeed secret. In other words, even if an agent's keys are compromised, the secrecy of the future root keys is healed as soon as the two agents perform an uncompromised DH exchange.

This sort of ratcheting mechanisms have in recent years become a corner-stone of modern secure messaging, and similar constructions are used in various other protocols, *e.g.* Signal's Sparse Post Quantum Ratchet [12] and Apple's PQ3 [1, 17]. No mechanized computational proof of the PCS of Signal, or in fact, to our knowledge, of any such ratcheting mechanism, currently exists – there are only pen-and-paper security proofs.

We propose to tackle this problem by using the Squirrel prover, which is well-suited for stateful protocols, to prove PCS for an asymmetric ratchet, which is the component of the Signal protocol on which that property crucially relies.

Challenges. A major difficulty when analysing the asymmetric ratchet in Squirrel is related to the tool's representation of secrecy. Being a tool built around the notion of indistinguishability, the formulation of secrecy commonly used in Squirrel is real-or-random secrecy, expressing that an adversary cannot distinguish the secret from a freshly sampled nonce. In Squirrel's formalism, the sequence of messages observed by the attacker at a date τ is denoted by $\mathsf{frame}(\tau)$. Secrecy of the root key r can then be expressed using the indistinguishability predicate $\sim$, by the formula $(\mathsf{frame}(\tau), r) \sim (\mathsf{frame}(\tau), n)$, where n is a fresh nonce. However, this formulation of secrecy is too strong when studying PCS. Indeed, when proving PCS, we need to model a scenario where a given root key $r' = \mathsf{H}(\langle r, s\rangle)$ has been compromised by the attacker, where r is the previous root key, and s is produced by an unauthenticated DH exchange. The attacker could, in general, have interfered with the DH exchange, so that s is in fact not secret (*i.e.* it is not a healing step). In such a case, letting the attacker learn r' means that r is no longer a real-or-random secret: the attacker could distinguish it from a random value by re-computing the hash, and comparing with r'. However, this leakage does not let the attacker learn the exact value of r: it is still secret, but only in the weaker sense that its value is not computable by the attacker. Thankfully, this weaker form of secrecy is sufficient to show that a chain key derived from r is itself secret.

To facilitate the study of PCS in Squirrel, we therefore had to propose formal definition for that weaker notion of secrecy.

Secrecy as a first-class notion. Our thesis is that reasoning on secrecy as a first-class security predicate is an useful and effective approach for capturing cryptographic proofs, not only of the asymmetric ratchet, but more generally of protocols with complex key-updates mechanisms. As explained earlier, the apt notion of secrecy is weaker than real-or-random secrecy: the attacker must be unable to use the frame of observed messages to compute the secret root key r. We therefore introduce a definition of secrecy as a predicate $\not\vdash$ called *non-deduction*: $u \not\vdash v$ expresses the fact that no adversarial computation applied to u can produce v with more than negligible probability.

Together with this predicate, we design a proof system, featuring dedicated proof rules that can be used to reason at a logical level about secrecy. This proof system is rather generic, in the sense that it expresses properties of the notion of non-deduction in general, rather than ad-hoc arguments tailored to the proof of the asymmetric ratchet we use it for.

We have integrated the non-deduction predicate, as well as the proof system, into the Squirrel prover, where it combines nicely with a pre-existing predicate for deduction, that already had dedicated built-in reasoning in Squirrel.

As part of the proof system, we propose several proof rules that express reductions in the Random Oracle Model (ROM), which are more powerful than previous rules in the tool.

Contributions. In this paper, we develop a logic for secrecy in the computational model that is less restrictive than other methods based on indistinguishability. In summary, our contributions are as follows:

- We introduce a new predicate for expressing non-deduction, as well as an associated proof system to reason about it.
- We prove the soundness of the proof system.
- We implement the predicate and proof system in the Squirrel prover.
- Finally, we use them to write the first mechanized computational proof of Post-Compromise Security in the asymmetric ratchet of the Signal protocol with two agents, an unbounded number of sessions, and an non-adaptive attacker.

Limitations. We only study the post-compromise security of a high-level model of an asymmetric ratchet, and do not cover all the features of a real-world ratchet like the one used in Signal. Notably, we do not model Signal session-handling protocol Sesame [18]. Remark that this would likely be impossible, as that recent work indicates that Sesame does not provide PCS [13].

References

[1] Apple. [n. d.]. iMessage with PQ3: The new state of the art in quantum-secure messaging at scale. https://security.apple.com/blog/imessage-pq3/.

[2] David Baelde, Stéphanie Delaune, Charlie Jacomme, Adrien Koutsos, and Solène Moreau. 2021. An Interactive Prover for Protocol Verification in the Computational Model. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*. IEEE, 537–554.

[3] David Baelde, Stéphanie Delaune, Adrien Koutsos, and Solène Moreau. 2022. Cracking the Stateful Nut: Computational Proofs of Stateful Security Protocols using the Squirrel Proof Assistant. In *CSF*. IEEE, 289–304.

[4] David Baelde, Stéphanie Delaune, and Stanislas Riou. 2025. SMT-Based Automation for Overwhelming Truth. In *CSF*. IEEE, 505–520.

[5] David Baelde, Adrien Koutsos, and Joseph Lallemand. 2023. A Higher-Order Indistinguishability Logic for Cryptographic Reasoning. In *LICS*. IEEE, 1–13.

[6] David Baelde, Adrien Koutsos, and Justine Sauvage. 2024. Foundations for Cryptographic Reductions in CCSA Logics. In *CCS*. ACM, 2814–2828.

[7] David Baelde, Adrien Koutsos, and Justine Sauvage. 2026. Leveraging Cryptographic Simulator Synthesis for Formally Verifying the FOO E-Voting Protocol. In *to appear in Usenix 2026 - 35th Usenix security symposium*. Baltimore, United States. https://inria.hal.science/hal-05453231

[8] Manuel Barbosa, Gilles Barthe, Karthik Bhargavan, Bruno Blanchet, Cas Cremers, Kevin Liao, and Bryan Parno. 2021. SoK: Computer-Aided Cryptography. In *SP*. IEEE, 777–795.

[9] Gilles Barthe, Benjamin Grégoire, Sylvain Heraud, and Santiago Zanella-Béguelin. 2011. Computer-Aided Security Proofs for the Working Cryptographer. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings (Lecture Notes in Computer Science, Vol. 6841)*, Phillip Rogaway (Ed.). Springer, 71–90. doi:10.1007/978-3-642-22792-9_5

[10] David A. Basin, Andreas Lochbihler, and S. Reza Sefidgar. 2020. CryptHOL: Game-Based Proofs in Higher-Order Logic. *J. Cryptol.* 33, 2 (2020), 494–566.

[11] Bruno Blanchet. 2008. A Computationally Sound Mechanized Prover for Security Protocols. *IEEE Trans. Dependable Secur. Comput.* 5, 4 (2008), 193–207. doi:10.1109/TDSC.2007.1005

[12] Graeme Connell and Rolfe Schmidt. October 2025. Signal Protocol and Post-Quantum Ratchets. https://signal.org/blog/spqr/.

[13] Cas Cremers, Niklas Medinger, and Aurora Naska. 2025. Impossibility Results for Post-Compromise Security in Real-World Communication Systems. In *SP*. IEEE, 4391–4405.

[14] Whitfield Diffie and Martin E. Hellman. 1976. New directions in cryptography. *IEEE Trans. Inf. Theory* 22, 6 (1976), 644–654.

[15] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. 1992. A Practical Secret Voting Scheme for Large Scale Elections. In *AUSCRYPT (Lecture Notes in Computer Science, Vol. 718)*. Springer, 244–251.

[16] Philipp G. Haselwarter, Exequiel Rivas, Antoine Van Muylder, Théo Winterhalter, Carmine Abate, Nikolaj Sidorenco, Catalin Hritcu, Kenji Maillard, and Bas Spitters. 2023. SSProve: A Foundational Framework for Modular Cryptographic Proofs in Coq. *ACM Trans. Program. Lang. Syst.* 45, 3 (2023), 15:1–15:61.

[17] Felix Linker, Ralf Sasse, and David Basin. 2025. A Formal Analysis of Apple's iMessage PQ3 Protocol. In *USENIX Security Symposium*. USENIX Association.

[18] Trevor Perrin and Moxie Marlinspike. 2017. *The Sesame Algorithm: Session Management for Asynchronous Message Encryption.* https://signal.org/docs/specifications/sesame/