

CV2F*: Combining CryptoVerif and F* for Proving Security of Protocol Implementations

Bruno Blanchet¹, Aymeric Fromherz¹, Charlie Jacomme²,
Benjamin Lipp¹, Emmanuel Mera¹

¹Inria Paris

²Inria Nancy Grand-Est, Université de Lorraine, LORIA

March 2026



CryptoVerif is a **mechanized prover** that:

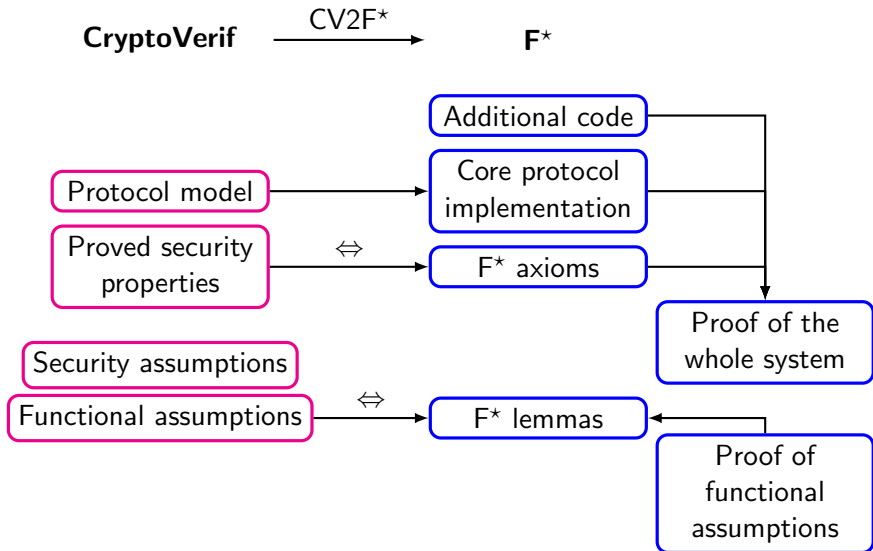
- works in the **computational model**.
- generates **proofs by sequences of games**.
- proves **secrecy**, **authentication**, and **indistinguishability** properties.
- provides a **generic** method for specifying properties of **cryptographic primitives**.
- works for **N sessions** (polynomial in the security parameter), with an **active adversary**.
- gives a bound on the **probability** of an attack (exact security).
- has an **automatic** strategy or can be **manually guided**.

⇒ Suitable for proving security of **models** of protocols

F^{*} is a **proof-oriented** programming language:

- functional language with effects
- dependent types
- SMT-based proof automation

⇒ Suitable for proving **programs**



Case study: simplified Signal + Sesame (in progress)

- in CryptoVerif: simplified Signal (X3DH, no ratcheting)
- in F*: Sesame, session management for Signal

CryptoVerif:

$$\begin{aligned} & \text{event}(\text{Recv}(rpk, spk, m)) \wedge \\ & \text{event}(\text{Honest}(spk)) \Rightarrow \\ & \quad \text{event}(\text{Send}(spk, rpk, m)) \vee \\ & \text{event}(\text{Corrupt}(spk)) \end{aligned}$$

F*:

$$\begin{aligned} & \text{event}(\text{Recv}(ruid, suid, m)) \wedge \\ & \text{event}(\text{Honest}(suid)) \Rightarrow \\ & \quad (\text{event}(\text{Send}(suid, ruidl, m)) \wedge \\ & \quad \quad ruid \in ruidl) \vee \\ & \text{event}(\text{Corrupt}(suid)) \end{aligned}$$

Workshop for the 25 years of ProVerif

June 3, 2026

Inria Paris

<https://proverif.inria.fr/25years.html>