Cracking the Stateful Nut

Computational Proofs of Stateful Security Protocols using the Squirrel Proof Assistant

CSF'22

David Baelde *Univ Rennes, CNRS, IRISA*
Stéphanie Delaune *Univ Rennes, CNRS, IRISA*
Adrien Koutsos *Inria Paris*
Solène Moreau *Univ Rennes, CNRS, IRISA*

9 August 2022, Haifa, Israel

Context

- **Security protocols** are *distributed* programs which aim at providing some **security properties**.

- Attacks against security protocols can be very **damageable**, e.g. theft or privacy breach.
- Protocol design is **though**, and **errors are hard to spot**.

⇒ well-suited field for **formal verification**.

The Squirrel Prover

The Squirrel Prover:

- Tool for verification of **security protocols** in the **computational model**.
- Implements an **indistinguishability logic**.
  - Inference rules proved valid w.r.t. comp. attacker.
- **Proof assistant**:
  - Users prove goals using sequences of tactics.
  - Logical tactics: apply, rewrite, . . .
  - Crypto. tactics: prf, euf, . . .

- Web-page:

  `https://squirrel-prover.github.io/`

Stateful Protocols

- In **stateful protocols**, agents have a **mutable state** persistent between sessions.
- Used in many **real-world protocols**, e.g.:
  - using integer counters: Yubi-Key, {3,4,5}G-AKA, ...
  - using chains of hashes: OSK, YPLRK, ...
  - using ratcheting/key refresh: Signal, MLS ...

Problem: SQUIRREL did not support mutable state, making stateful protocols out-of-scope.

Our Contributions

- Extend the indistinguishability logic with **mutable state**.
- New **generalized sequent calculus**.
  - Mix reachability and equivalence reasonings.
- **Proof automation**: design a proof system for bi-deduction.
  - **Intuition:** indistinguishability is **preserved by (public) computation**.
  - Allow for **automation** of some proof steps.
- **Implementation** in the SQUIRREL tool.
  - Main case-studies: Yubi-Key, Yubi-HSM.

Indistinguishability Logic

The OSK Protocol

s_T: mutable state of tag T

s_R: mutable state of reader R

s_T and s_R initial value: n_s

n_s, k_H, k_G: random samplings

H, G: keyed hash functions

The OSK protocol:

$$
\begin{array}{ll}
1 : T \longrightarrow R : & s_T := H(s_T, k_H); \\
& \mathrm{out}(G(s_T, k_G)) \\[1em]
2 : R \longrightarrow T : & \mathrm{in}(x); \\
& \text{if } x = G(H(s_R, k_H), k_G) \text{ then} \\
& \quad \mathrm{out}(ok); \\
& \quad s_R := H(s_R, k_H)
\end{array}
$$

Terms represent **probabilistic poly-time computations of bitstrings**.

Used to model both **protocol** and **adversary computations**.

Indistinguishability Logic: Terms

Terms represent **probabilistic poly-time computations of bitstrings**.

Used to model both **protocol** and **adversary computations**.

Names for random samplings of length η (security parameter):
$\mathsf{n_s}, \mathsf{k_H}, \mathsf{k_G}$

Function symbols for honest computations:
$\mathsf{H}(\mathsf{n_s}, \mathsf{k_H})$

Timestamps for time-points of the protocol execution: protocol actions ($\mathsf{Tag}(\mathsf{i})$), variables (e.g. τ), predecessor $\mathsf{pred}(T)$

Indices for session identifiers: variable i

Macros for protocol terms at a given time:
$\mathsf{input}@\tau, \mathsf{output}@\tau, \mathsf{frame}@\tau, \mathsf{s_T}@\tau$

Attacker function symbols for adversary computations:
$\mathbf{att}(\mathsf{frame}@\mathsf{pred}(\tau))$

Indistinguishability Logic: Terms

$$
\begin{array}{lll}
1 : \mathsf{T} \longrightarrow \mathsf{R} : & \mathsf{s_T} := \mathsf{H}(\mathsf{s_T}, \mathsf{k_H}); \\
& \mathsf{out}(\mathsf{G}(\mathsf{s_T}, \mathsf{k_G})) \\
\\
2 : \mathsf{R} \longrightarrow \mathsf{T} : & \mathsf{in}(\mathsf{x}); \\
& \text{if } \mathsf{x} = \mathsf{G}(\mathsf{H}(\mathsf{s_R}, \mathsf{k_H}), \mathsf{k_G}) \text{ then} \\
& \quad \mathsf{out}(\mathsf{ok}); \\
& \quad \mathsf{s_R} := \mathsf{H}(\mathsf{s_R}, \mathsf{k_H})
\end{array}
$$

Examples:

- OSK tag T state updates:

$$\mathsf{s_T}@\tau = \mathsf{H}(\mathsf{s_T}@\mathsf{pred}(\tau), \mathsf{k_H})$$

- Definition of $\mathsf{input}@\tau$:

$$\mathbf{att}(\mathsf{frame}@\mathsf{pred}(\tau))$$

Indistinguishability Logic: Local Formulas

- **Local formulas**: **first-order** formulas built over the atoms:

$$t_1 = t_2, T_1 = T_2, T_1 \leq T_2, \mathsf{happens}(T), \ldots$$

- Example:
  - OSK tag T state updates:

$$\forall \tau. \left(\exists \mathsf{i}.\, \tau = \mathsf{Tag}(\mathsf{i}) \wedge \mathsf{happens}(\tau)\right) \rightarrow \mathsf{s_T}@\tau = \mathsf{H}(\mathsf{s_T}@\mathsf{pred}(\tau), \mathsf{k_H})$$

Indistinguishability Logic: Local Formulas

- ϕ is **valid** w.r.t. $\mathcal{P}$ if it is true with **overwhelming probability**.

- Example of valid formula: w.r.t. any protocol $\mathcal{P}$
  - Random samplings freshness:

$$\mathsf{n}_1 \neq \mathsf{n}_2$$

Indistinguishability Logic: Local Formulas

Local formulas can capture **reachability** security properties.

Example:

- **Authentication** of the OSK protocol:

$$\forall \tau.\ \phi^{\mathsf{R}}_{\mathsf{accept}}[\tau] \rightarrow \exists \mathsf{i}.\ \mathsf{Tag}(\mathsf{i}) \leq \tau \wedge \mathsf{input}@\tau = \mathsf{output}@\mathsf{Tag}(\mathsf{i})$$

Indistinguishability Logic: Global Formulas

Global formulas: **first-order** logic formulas Φ over the atoms:

- $[\phi]_{\mathcal{P}}$ where ϕ is a local formula.
  Valid if the local formula ϕ is valid w.r.t. $\mathcal{P}$.

- $[\vec{u} \sim \vec{v}]_{\mathcal{P}_1,\mathcal{P}_2}$ where $\vec{u}, \vec{v}$ are same-length sequences of terms.
  Valid if no PPTM $\mathcal{A}$ can distinguish between $\vec{u}$ and $\vec{v}$.
  (w.r.t., respectively, $\mathcal{P}_1$ and $\mathcal{P}_2$)

Notations: $\tilde{\forall}$, $\tilde{\nabla}$... to distinguish from local logic constructs.

Indistinguishability Logic: Global Formulas

Global formulas can capture **equivalence** security properties.

Example:

- Strong secrecy of the OSK state: $(\mathcal{P} = \mathsf{OSK})$
  $\tilde{\forall}\tau.[\mathsf{happens}(\tau)]_{\mathcal{P}_1} \,\tilde{\Rightarrow}\, [\mathsf{frame}@\tau, \mathsf{s_T}@\tau \sim \mathsf{frame}@\tau, \mathsf{n_{fresh}}]_{\mathcal{P},\mathcal{P}}$

Indistinguishability Logic: Global Formulas

Example of a valid global formula:

- $[s = t]_{\mathcal{P}_1} \ \tilde{\Rightarrow} \ \left[\vec{u}[s] \sim \vec{v}\right]_{\mathcal{P}_1,\mathcal{P}_2} \tilde{\Rightarrow} \left[\vec{u}[t] \sim \vec{v}\right]_{\mathcal{P}_1,\mathcal{P}_2}$

Indistinguishability Logic: Global Formulas

Example of a valid global formula:

- $[s = t]_{\mathcal{P}_1} \;\tilde{\Rightarrow}\; \left[\vec{u}[s] \sim \vec{v}\right]_{\mathcal{P}_1,\mathcal{P}_2} \tilde{\Rightarrow} \left[\vec{u}[t] \sim \vec{v}\right]_{\mathcal{P}_1,\mathcal{P}_2}$

Global formulas allow to **mix** reachability and equivalence properties.

Sequents and Proof Systems

Local and Global Sequents

$$\boxed{\Sigma\,;\ \Theta\ :\ \Gamma \vdash_{\mathcal{P}} \phi} \qquad \text{and} \qquad \boxed{\Sigma\,;\ \Theta \vdash \Phi}$$

local formulas

global formulas

Σ: universally quantified variables

Semantics

$$\Sigma;\ \Theta \vdash \Phi \quad \rightsquigarrow \quad \tilde{\forall}\Sigma.\ (\tilde{\bigwedge}\Theta \mathrel{\tilde{\Rightarrow}} \Phi)$$

$$\Sigma;\ \Theta : \Gamma \vdash_{\mathcal{P}} \phi \quad \rightsquigarrow \quad \tilde{\forall}\Sigma.\ (\tilde{\bigwedge}\Theta \mathrel{\tilde{\Rightarrow}} [\,\bigwedge \Gamma \Rightarrow \phi]_{\mathcal{P}})$$

Proof System: Classical Reasoning

Classical **FO inference rules** are **sound**:

- Purely local (local seq.):

$$\frac{\Sigma;\ \Theta\ :\ \Gamma, \phi_1 \vdash_{\mathcal{P}} \psi \qquad \Sigma;\ \Theta\ :\ \Gamma, \phi_2 \vdash_{\mathcal{P}} \psi}{\Sigma;\ \Theta\ :\ \Gamma, \phi_1 \vee \phi_2 \vdash_{\mathcal{P}} \psi}$$

Proof System: Classical Reasoning

Classical **FO inference rules** are **sound**:

- Purely local (local seq.):

$$\frac{\Sigma;\ \Theta\ :\ \Gamma, \phi_1 \vdash_{\mathcal{P}} \psi \qquad \Sigma;\ \Theta\ :\ \Gamma, \phi_2 \vdash_{\mathcal{P}} \psi}{\Sigma;\ \Theta\ :\ \Gamma, \phi_1 \vee \phi_2 \vdash_{\mathcal{P}} \psi}$$

- Purely global (local and global seq.):

$$\frac{\Sigma;\ \Theta, \Phi_1\ :\ \Gamma \vdash_{\mathcal{P}} \psi \qquad \Sigma;\ \Theta, \Phi_2\ :\ \Gamma \vdash_{\mathcal{P}} \psi}{\Sigma;\ \Theta, \Phi_1 \tilde{\vee} \Phi_2\ :\ \Gamma \vdash_{\mathcal{P}} \psi}$$

$$\frac{\Sigma;\ \Theta, \Phi_1 \vdash \Psi \qquad \Sigma;\ \Theta, \Phi_2 \vdash \Psi}{\Sigma;\ \Theta, \Phi_1 \tilde{\vee} \Phi_2 \vdash \Psi}$$

Proof System: Mixing Local and Global Reasoning

Selected **inference rules** involving **mixed kinds of sequents**:

$$\textsc{Global-Local}\quad \frac{\Sigma;\ \Theta \vdash [\phi]_{\mathcal{P}}}{\Sigma;\ \Theta : \ \ \vdash_{\mathcal{P}} \phi}$$

$$\textsc{Local-Global}\quad \frac{\Sigma;\ \Theta : \ \ \vdash_{\mathcal{P}} \phi}{\Sigma;\ \Theta \vdash [\phi]_{\mathcal{P}}}$$

$$\textsc{Rewrite-Equiv}\quad \frac{\Sigma;\ \Theta \vdash [\phi \sim \psi]_{\mathcal{P},\mathcal{P}'} \qquad \Sigma;\ \Theta : \ \ \vdash_{\mathcal{P}'} \psi}{\Sigma;\ \Theta : \ \ \vdash_{\mathcal{P}} \phi}$$

Proof System: Mixing Local and Global Reasoning

Selected **inference rules** involving **mixed kinds of sequents**:

$$\textsc{Global-Local}\quad \frac{\Sigma;\ \Theta \vdash [\phi]_{\mathcal{P}}}{\Sigma;\ \Theta : \ \vdash_{\mathcal{P}} \phi}$$

$$\textsc{Local-Global}\quad \frac{\Sigma;\ \Theta : \ \vdash_{\mathcal{P}} \phi}{\Sigma;\ \Theta \vdash [\phi]_{\mathcal{P}}}$$

$$\textsc{Rewrite-Equiv}\quad \frac{\Sigma;\ \Theta \vdash \big[(\Gamma \Rightarrow \phi) \sim (\Delta \Rightarrow \psi)\big]_{\mathcal{P},\mathcal{P}'} \qquad \Sigma;\ \Theta : \Delta \vdash_{\mathcal{P}'} \psi}{\Sigma;\ \Theta : \Gamma \vdash_{\mathcal{P}} \phi}$$

Example: Strong Secrecy $\rightarrow$ Weak Secrecy

Example:

Strong secrecy of a state value $\mathsf{s_T}$:

$$\Phi_{\mathcal{S}} \overset{\text{def}}{=} [\mathsf{frame@}\tau, \mathsf{s_T@}\tau \sim \mathsf{frame@}\tau, \mathsf{n_{fresh}}]_{\mathcal{P},\mathcal{P}}$$

implies **weak secrecy** of $\mathsf{s_T}$:

$$\mathsf{input@}\tau \neq \mathsf{s_T@}\tau$$

Example:

Strong secrecy of a state value $\mathsf{s_T}$:

$$\Phi_{\mathcal{S}} \stackrel{\text{def}}{=} [\mathsf{frame}@\tau, \mathsf{s_T}@\tau \sim \mathsf{frame}@\tau, \mathsf{n_{fresh}}]_{\mathcal{P},\mathcal{P}}$$

implies **weak secrecy** of $\mathsf{s_T}$:

$$\mathsf{input}@\tau \neq \mathsf{s_T}@\tau$$

Proof:

$$\dfrac{\tau;\ \Phi_{\mathsf{hap}}, \Phi_{\mathcal{S}} \vdash \left[\left(\mathsf{input}@\tau \neq \mathsf{s_T}@\tau\right) \sim \left(\mathsf{input}@\tau \neq \mathsf{n_{fresh}}\right)\right]_{\mathcal{P},\mathcal{P}} \qquad \tau;\ \Phi_{\mathsf{hap}}, \Phi_{\mathcal{S}} : \ \vdash_{\mathcal{P}} \mathsf{input}@\tau \neq \mathsf{n_{fresh}}}{\tau;\ \Phi_{\mathsf{hap}}, \Phi_{\mathcal{S}} : \ \vdash_{\mathcal{P}} \mathsf{input}@\tau \neq \mathsf{s_T}@\tau}\ \textsc{Rewrite-Equiv}$$

(where Φ_{hap} is $[\mathsf{happens}(\tau)]_{\mathcal{P}}$)

Example: Strong Secrecy → Weak Secrecy

Example:

Strong secrecy of a state value s_T:

$$\Phi_{\mathcal{S}} \stackrel{\text{def}}{=} [\text{frame}@\tau, s_T@\tau \sim \text{frame}@\tau, n_{\text{fresh}}]_{\mathcal{P},\mathcal{P}}$$

implies **weak secrecy** of s_T:

$$\text{input}@\tau \neq s_T@\tau$$

Proof:

$$\dfrac{\tau;\ \Phi_{\text{hap}}, \Phi_{\mathcal{S}} \vdash \left[\left(\text{input}@\tau \neq s_T@\tau\right) \sim \left(\text{input}@\tau \neq n_{\text{fresh}}\right)\right]_{\mathcal{P},\mathcal{P}} \qquad \tau;\ \Phi_{\text{hap}}, \Phi_{\mathcal{S}} : \ \vdash_{\mathcal{P}} \text{input}@\tau \neq n_{\text{fresh}}}{\tau;\ \Phi_{\text{hap}}, \Phi_{\mathcal{S}} : \ \vdash_{\mathcal{P}} \text{input}@\tau \neq s_T@\tau} \ \textsc{Rewrite-Equiv}$$

- 2_{nd} premise: consequence of n_{fresh} freshness
- 1_{st} premise: RHS can be (bi)-deduced from $\Phi_{\mathcal{S}}$!

(where Φ_{hap} is $[\text{happens}(\tau)]_{\mathcal{P}}$)

Bi-Deduction

Bi-Deduction: Intuition

Indistinguishability is **preserved by (public) computation**:

$$\text{if } \left[\vec{u}_1 \sim \vec{u}_2\right] \text{ then } \forall \mathcal{B}.\ \left[\mathcal{B}(\vec{u}_1) \sim \mathcal{B}(\vec{u}_2)\right]$$

As a pseudo-inference rule:

$$\frac{\exists \mathcal{B} \text{ s.t. } \mathcal{B} \text{ computes } \vec{v}_i \text{ from } \vec{u}_i}{\Sigma;\ \Theta, [\vec{u}_1 \sim \vec{u}_2] \vdash [\vec{v}_1 \sim \vec{v}_2]}$$

Bi-Deduction: Example

$$\frac{\exists \mathcal{B} \text{ s.t. } \mathcal{B} \text{ computes } \vec{v}_i \text{ from } \vec{u}_i}{\Sigma;\ \Theta, [\vec{u}_1 \sim \vec{u}_2] \vdash [\vec{v}_1 \sim \vec{v}_2]}$$

Example:

$$\tau;\ \Phi_{\mathsf{hap}}, \left[\mathsf{frame}@\tau, \mathsf{s_T}@\tau \sim \mathsf{frame}@\tau, \mathsf{n_{fresh}}\right] \vdash \left[\left(\mathsf{input}@\tau \neq \mathsf{s_T}@\tau\right) \sim \left(\mathsf{input}@\tau \neq \mathsf{n_{fresh}}\right)\right]$$

Proved by bi-deduction with:

$$\mathcal{B}(\mathsf{frame}@\tau, \mathsf{x}) \stackrel{\mathsf{def}}{=} \left(\mathbf{att}(\mathsf{frame}@\mathsf{pred}(\tau)) = \mathsf{x}\right)$$

Bi-Deduction Judgements and Proof System

- The **bi-deduction** rule:

$$\textsc{Bi-Deduce}\quad \frac{\Sigma;\ \#(\vec{u}_1, \vec{u}_2) \triangleright \#(\vec{v}_1, \vec{v}_2)}{\Sigma;\ \Theta, [\vec{u}_1 \sim \vec{u}_2] \vdash [\vec{v}_1 \sim \vec{v}_2]}$$

- We designed a **proof system** for bi-deduction, e.g.:

$$\textsc{FA}\quad \frac{\Sigma;\ \#(\vec{u}_1, \vec{u}_2) \triangleright \#(\vec{v}_1, \vec{v}_2)}{\Sigma;\ \#(\vec{u}_1, \vec{u}_2) \triangleright \#(\mathsf{f}(\vec{v}_1),\ \mathsf{f}(\vec{v}_2))}$$

Bi-Deduction: Implementation

Fully-automated procedure for **bi-deduction** implemented in Squirrel:

- **soundness** follows from our bi-deduction **proof system**;
- integrated in the `apply` tactic (for global sequents);
- extension with fully-automated **inductive reasoning** using *abstract interpretation*.

Case-Studies

Case-Study: Yubi-Key

Security analysis of the Yubi-Key protocol (used for 2FA).

- Yubi-Keys are physical authentication devices with a single button, which generated a OTP (one-time password).
- Uses **counters** for *protection against replay-attack*:
  - OTPs include the encrypted Yubi-Key counter;
  - the counter is incremented after each sessions.
- We prove **injective authentication**:
  - successful login must be preceded by a button press;
  - each counter value is accepted at most once.

Case-Study: Yubi-HSM

Also studied the Yubi-HSM protocol:

- Yubi-HSM = Yubi-Key + keys stored in a HSM (server side).
- We prove **injective authentication**

Case-Study: Yubi-HSM

Also studied the Yubi-HSM protocol:

- Yubi-HSM = Yubi-Key + keys stored in a HSM (server side).
- We prove **injective authentication** in two steps:
  - equivalence of Yubi-HSM with an idealized version;
  - proof of injective authentication, using Rewrite-Equiv to switch from the real to the ideal protocol.

Conclusion

Conclusion

Our Contributions

- Extend the indistinguishability logic with **mutable state**.
- **Generalized sequent calculus**.
  - Mix reachability and equivalence reasonings.
- **Proof automation**: design a proof system for bi-deduction.
  - Allow for **automation** of some proof steps.
- **Implementation** in SQUIRREL + case-studies: Yubi-{Key,HSM}.

Future Works

- More complex protocols and security properties.
- More automation, e.g. using SMT solvers.
- Systematic translation of crypto. assumptions as inference rules.

Thank you for your attention

Proof System: Local $\neq$ Global

Local hypothesis $\neq$ **global hypothesis**:

- **Global hypothesis**: property of a bitstring distribution
- **Local hypothesis**: property of a bitstring

Global hyp. are stronger than local hyp.:

$$\frac{\Sigma;\ \Theta : \phi,\ \Gamma \vdash_{\mathcal{P}} \psi}{\Sigma;\ [\phi]_{\mathcal{P}}, \Theta : \Gamma \vdash_{\mathcal{P}} \psi}$$

$\phi \to \psi$ true with overwh. prob.

ϕ true with overwh. prob.
implies
ψ true with overwh. prob.

But the converse does not generally hold.

Counter-example:

$\mathsf{n} = 0 \to \mathsf{n} = 1$	$[\mathsf{n} = 0] \tilde{\Rightarrow} [\mathsf{n} = 1]$
not valid	valid