Integrating Automated and Interactive Protocol Verification

Achim D. Brucker
SAP Research
CEC Karlsruhe, Germany

Sebastian A. Mödersheim
DTU Compute

Integrating Two Approaches

Integrating two approaches:

Automated protocol verification	Interactive theorem proving

Goubault-Larrecq:

h1/paradox	Coq

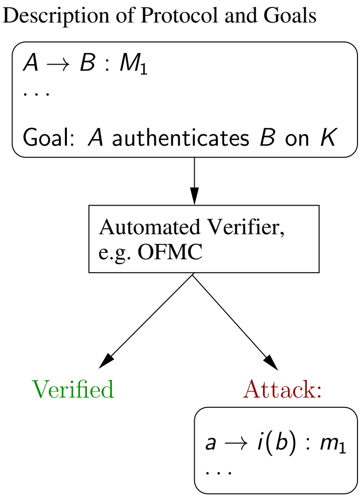
Our work:

Open-source Fixed-point Model Checker	Isabelle/HOL

Goal: best of both worlds

Completely automatic	High reliability

Automated Protocol Verification



- Fully automated
- Advanced verification methods
  - ★ Impose subtle requirements on the specification
  - ★ Implementation may well have bugs

$\Rightarrow$ How to trust the verifier?

Interactive Theorem Proving

- Core (Proof Checker):
  accepts only correct mathematical proofs for a given statement
  On this level: proofs are completely manual.
- Proof Assistance:
  proof strategies for automatically handling large classes of subgoals. Thus interactive theorem proving.
- Programmable:
  development of customized strategies
- High reliability:
  only need to trust the core

- Successfully used for verifying protocols, e.g., by Paulson, Bella

Reference Model (Inductive Model)

- Inductive typed trace-based protocol model, similar to Paulson:

Example (NSL, role $\mathfrak{A}$)

$$\frac{t \in \mathbb{T} \quad NA \notin \mathit{used}\ (t)}{\text{iknows}\ \{NA, A\}_{\text{pk}(B)} \# \text{state}\ \mathfrak{A}\ [A, B, NA] \# \text{secret}\ B\ NA \# t \in \mathbb{T}}$$

$$\frac{t \in \mathbb{T} \quad \text{state}\ \mathfrak{A}\ [A, B, NA] \in [t] \quad \text{iknows}\ \{NA, NB, B\}_{\text{pk}(A)} \in [t]}{\text{iknows}\ \{NB\}_{\text{pk}(B)} \# t \in \mathbb{T}}$$

Reference Model: The Intruder

Standard Dolev-Yao style intruder, for instance:

$$\frac{t \in \mathbb{T} \quad \mathsf{iknows}\ \{m\}_k \in [t] \quad \mathsf{iknows}\ \mathsf{inv}(k) \in [t]}{\mathsf{iknows}\ m \# t \in \mathbb{T}}$$

Often the intruder restricted to a typed model.

Reference Model: Goals

Attack Traces for Secrecy

$$\frac{t \in \mathbb{T} \quad \text{secret } A\ M \in [t] \quad \text{iknows } M \in [t] \quad \text{honest } A}{\text{attack}\#t \in \mathbb{T}}$$

Abstractions

- Abstract fresh data into finitely many equivalence classes, for instance:
  - $NA \mapsto \mathfrak{N}_{\mathfrak{A}}(A, B)$
  - $NB \mapsto \mathfrak{N}_{\mathfrak{B}}(B, A)$
- Consider the set of reachable events $\mathbb{E}$ (not reachable states).

$\Rightarrow$ popular FOL Horn-clause modeling style:

Example

$\mathbb{E}$ is the least set of facts satisfying:

$\Rightarrow$ iknows $\{\mathfrak{N}_{\mathfrak{A}}(A, B), A\}_{\mathrm{pk}(B)}$
$\Rightarrow$ state $\mathfrak{A}$ $[A, B, \mathfrak{N}_{\mathfrak{A}}(A, B)]$
$\Rightarrow$ secret B $\mathfrak{N}_{\mathfrak{A}}(A, B)$
state $\mathfrak{A}$ $[A, B, NA] \wedge$ iknows $\{NA, NB, B\}_{\mathrm{pk}(A)} \Rightarrow$ iknows $\{NB\}_{\mathrm{pk}(B)}$
...
secret A $M \in \mathbb{E} \wedge$ iknows $M \in \mathbb{E} \wedge$ honest $A \Rightarrow$ attack

The Approach of Goubault-Larrecq

A protocol is secure ...

The Approach of Goubault-Larrecq

A protocol is secure ...

- iff the fact attack is not contained in the least Herbrand model of the Horn clauses Φ.

The Approach of Goubault-Larrecq

A protocol is secure ...

- iff the fact attack is not contained in the least Herbrand model of the Horn clauses Φ.
- iff $\Phi \wedge \neg$attack is unsatisfiable.

The Approach of Goubault-Larrecq

A protocol is secure ...

- iff the fact attack is not contained in the least Herbrand model of the Horn clauses Φ.
- iff $\Phi \land \neg$attack is unsatisfiable.
- iff resolution on $\Phi \land \neg$attack can produce the empty clause.

The Approach of Goubault-Larrecq

A protocol is secure ...

- iff the fact attack is not contained in the least Herbrand model of the Horn clauses Φ.
- iff $\Phi \wedge \neg$attack is unsatisfiable.
- iff resolution on $\Phi \wedge \neg$attack can produce the empty clause.

Standard Verification (ProVerif,SPASS...)

Resolution until

- empty clause derived: potential attack found
- saturated (no new clauses can be produced): protocol secure.
  - ★ Verifiability: how can we be sure that saturation is correct?

The Approach of Goubault-Larrecq

A protocol is secure ...

- iff the fact attack is not contained in the least Herbrand model of the Horn clauses Φ.
- iff $\Phi \wedge \neg$attack is unsatisfiable.
- iff resolution on $\Phi \wedge \neg$attack can produce the empty clause.

Standard Verification (ProVerif,SPASS...)

Resolution until

- empty clause derived: potential attack found
- saturated (no new clauses can be produced): protocol secure.
  - ★ Verifiability: how can we be sure that saturation is correct?
  - ★ This does not give us proof we can check!

The Approach of Goubault-Larrecq

A protocol is secure …

- iff the fact attack is not contained in the least Herbrand model of the Horn clauses Φ.
- iff $\Phi \wedge \neg$attack is unsatisfiable.
- iff resolution on $\Phi \wedge \neg$attack can produce the empty clause.
- iff $\Phi \wedge \neg$attack has a model

Idea (Goubault-Larrecq)

The Approach of Goubault-Larrecq

A protocol is secure ...

- iff the fact attack is not contained in the least Herbrand model of the Horn clauses Φ.
- iff $\Phi \wedge \neg$attack is unsatisfiable.
- iff resolution on $\Phi \wedge \neg$attack can produce the empty clause.
- iff $\Phi \wedge \neg$attack has a model

Idea (Goubault-Larrecq)

- There may be a finite model!

The Approach of Goubault-Larrecq

A protocol is secure ...

- iff the fact attack is not contained in the least Herbrand model of the Horn clauses Φ.
- iff $\Phi \wedge \neg$attack is unsatisfiable.
- iff resolution on $\Phi \wedge \neg$attack can produce the empty clause.
- iff $\Phi \wedge \neg$attack has a model

Idea (Goubault-Larrecq)

- There may be a finite model!
- A finite model is a proof of satisfiability we can check!

The Approach of Goubault-Larrecq

A protocol is secure ...

- iff the fact attack is not contained in the least Herbrand model of the Horn clauses Φ.
- iff $\Phi \wedge \neg$attack is unsatisfiable.
- iff resolution on $\Phi \wedge \neg$attack can produce the empty clause.
- iff $\Phi \wedge \neg$attack has a model

Idea (Goubault-Larrecq)

- There may be a finite model!
- A finite model is a proof of satisfiability we can check!
- Use the finite model finder h1 or paradox.
  If successful, feed into Coq.

Our Approach

- The Horn clauses are already an abstract representation.
- Can we work on the level of Paulson's inductive trace model instead?

Key Ideas

In a typed model,

- the fixedpoint $\mathbb{E}$ of the Horn clauses is finite
- and describes an invariant over the traces.

OFMC computes a fixedpoint $\mathbb{E}$ of abstract events.

- If attack $\in \mathbb{E}$, refine abstraction or validate attack.
- If attack $\notin \mathbb{E}$, use fixedpoint for verification in Isabelle.
- Label fresh nonces in the concrete model with their abstraction.

Example

$$\frac{t \in \mathbb{T} \quad NA \notin used\ (t) \quad label(NA) = (\mathfrak{N}_{\mathfrak{A}}, A, B)}{\text{iknows } \{NA, A\}_{\text{pk}(B)} \# \text{state } \mathfrak{A}\ [A, B, NA] \# \text{secret } B\ \ NA \# t \in \mathbb{T}}$$

- Note: the label is merely an annotation, it does not change the model!

Key Ideas

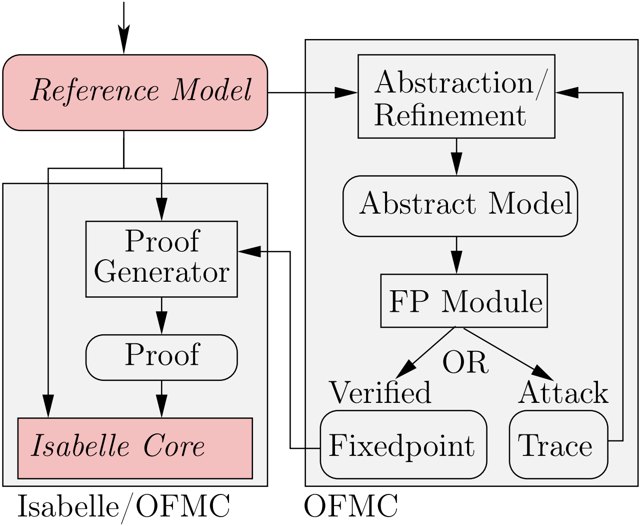
- The concretization of $\mathbb{E}$ is the set of all traces whose abstraction is in $\mathbb{E}$:

Definition (concretization)

$$
\begin{aligned}
[\![l]\!] &= \{(l, n) \mid n \in \mathbb{N}\} \\
[\![f\ t_1\ \dots\ t_n]\!] &= \{f\ s_1\ \dots\ s_n \mid s_i \in [\![t_i]\!]\} \\
[\![\mathbb{E}]\!] &= \cup_{f \in \mathbb{E}} [\![f]\!] \\
\mathbb{T}' &= \{e_1 \# \dots \# e_n \mid e_i \in [\![\mathbb{E}]\!]\}
\end{aligned}
$$

- Verify in Isabelle:
  - ★ All rules are closed under $\mathbb{T}'$, thus $\mathbb{T} \subseteq \mathbb{T}'$.
  - ★ $\mathbb{T}'$ does not contain an attack event, thus $\mathbb{T}$ is safe.
  - ★ …completely automatically.
- If anything goes wrong, this proof generation simply fails. i.e. "we fail to convince Isabelle" in the worst case.

Architecture



Experimental Results

Protocol	FP	time [s]	Protocol	FP	time [s]
Andrew Secure RPC	113	1517	ISO three pass mutual	229	21448
Bilateral-Key Exchange	85	7575	NSCK	135	9471
Denning-Sacco	71	2549	NSL	75	117
ISO one pass unilateral	40	33	Non-Reversible Functions	196	21018
ISO two pass unilateral	56	77	TLS (simplified)	172	26982
ISO two pass mutual	104	442	Wide Mouthed Frog	87	1382

Conclusions and Outlook

- Combinations of the automated and interactive verification seem promising:
  - ★ Fully automated and relatively easy to use
  - ★ High reliability, because one only relies on a small core.
  - ★ May give highest assurance levels of common criteria at relatively low cost.
- Future Plans
  - ★ Improving fixed-point representation to scale better
  - ★ Larger classes of protocols (e.g., algebraic properties)
  - ★ Proof generator based on more specialized routines for efficiency