Discovering Security Protocol Attacks by Refuting Incorrect Conjectures

Graham Steel, Alan Bundy,

and Ewen Denney

THE UNIVERSITY OF EDINBURGH

Division of informatics

The Problem

Finding attacks on faulty protocols tricky

Model checking and other state-exploration approaches used

The Problem

Finding attacks on faulty protocols tricky

Model checking and other state-exploration approaches used

But generally use simplifications, e.g.:

- Only two agents (and a spy)

The Problem

Finding attacks on faulty protocols tricky

Model checking and other state-exploration approaches used

But generally use simplifications, e.g.:

- Only two agents (and a spy)
- Defined roles, initiator and responder

The Problem

Finding attacks on faulty protocols tricky

Model checking and other state-exploration approaches used

But generally use simplifications, e.g.:

- Only two agents (and a spy)
- Defined roles, initiator and responder
- Only one nonce/key available

The Problem

Finding attacks on faulty protocols tricky

Model checking and other state-exploration approaches used

But generally use simplifications, e.g.:

- Only two agents (and a spy)
- Defined roles, initiator and responder
- Only one nonce/key available

Result:

Some attacks outside scope, e.g. Paulson attack on simplified Otway Rees.

Some protocols outside scope, e.g. conference key protocols

Inductive Method

Paulson and Bella

Protocols formalised in HOL as traces

A trace is a list of events like

‘A sends message X to B’

Inductive Method

Paulson and Bella

Protocols formalised in HOL as traces

A trace is a list of events like

‘A sends message X to B’

Prove security properties by induction on traces, e.g.

‘If A receives message 3 with nonce N,
and he sent message 1 with nonce N to B,
then message 3 came from the server.’

Inductive Method

Paulson and Bella

Protocols formalised in HOL as traces

A trace is a list of events like

‘A sends message X to B’

Prove security properties by induction on traces, e.g.

‘If A receives message 3 with nonce N,
and he sent message 1 with nonce N to B,
then message 3 came from the server.’

Deal directly with arbitrary number of agents, nonces, keys,. . .

Inductive Method

Paulson and Bella

Protocols formalised in HOL as traces

A trace is a list of events like

‘A sends message X to B’

Prove security properties by induction on traces, e.g.

‘If A receives message 3 with nonce N,
and he sent message 1 with nonce N to B,
then message 3 came from the server.’

Deal directly with arbitrary number of agents, nonces, keys,. . .

BUT: No support for non-theorem detection

Refuting Incorrect Conjectures

Many applications

e.g. spotting incorrect generalisations in ITP, finding bugs in recursive algorithms

Refuting Incorrect Conjectures

Many applications

e.g. spotting incorrect generalisations in ITP, finding bugs in recursive algorithms

Refutation of incorrect inductive conjectures has been studied before

e.g. Protzen (1992), Reif (2000), Ahrendt (2000)

Refuting Incorrect Conjectures

Many applications

e.g. spotting incorrect generalisations in ITP, finding bugs in recursive algorithms

Refutation of incorrect inductive conjectures has been studied before

e.g. Protzen (1992), Reif (2000), Ahrendt (2000)

- but search approaches too naïve for protocol problem

Refuting Incorrect Conjectures

Many applications

e.g. spotting incorrect generalisations in ITP, finding bugs in recursive algorithms

Refutation of incorrect inductive conjectures has been studied before

e.g. Protzen (1992), Reif (2000), Ahrendt (2000)

- but search approaches too naïve for protocol problem

A more sophisticated method is ‘Proof by Consistency’

Proof by Consistency

Developed by Musser (1980), Huet & Hullot (1982), Kapur & Musser (1987), Jouannaud & Kounalis (1986), Bachmair (1988), Ganzinger & Stuber (1993) and others.

Conjecture C is an inductive consequence of E

if and only if:

C is consistent with equations E in standard model.

Proof by Consistency

Developed by Musser (1980), Huet & Hullot (1982), Kapur & Musser (1987), Jouannaud & Kounalis (1986), Bachmair (1988), Ganzinger & Stuber (1993) and others.

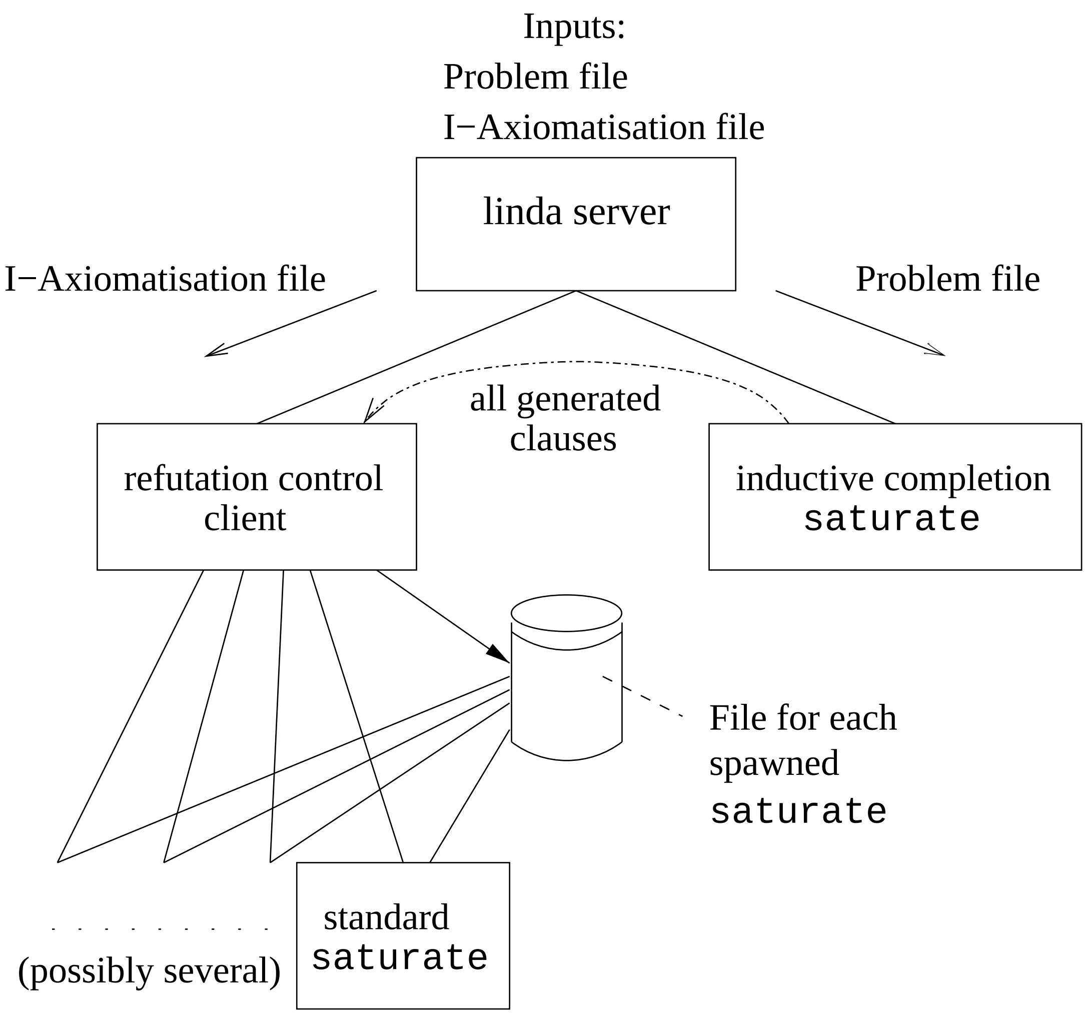
Conjecture C is an inductive consequence of E

if and only if:

C is consistent with equations E in standard model.

Re-cast by Comon and Nieuwenhuis (1999): can handle non-equational case, non-convergent specs., free or non-free constructors, and is refutation complete.

Two stage approach: I-Axiomatisation + First-order consistency


Inputs:
Problem file
I−Axiomatisation file
linda server
I−Axiomatisation file
Problem file
all generated
clauses
refutation control
client
inductive completion
saturate
File for each
spawned
saturate
. . . .
(possibly several)
standard
saturate

Protocol Model

Aim is first-order version of Paulson’s model

Lists for traces, sets for intruder knowledge, arbitrary numbers of agents, nonces, keys, etc.

Free constructors, so can define equality completely

This allows us to keep it Horn

- by defining both $member(x,l) = true$ and $member(x,l) = false$.

Early Results

Clark and Jacob attack

$$1.\ A \longrightarrow B : \{\!|\, N_A \,|\!\}_{K_{AB}}$$
$$2.\ B \longrightarrow A : \{\!|\, s(N_A) \,|\!\}_{K_{AB}}$$

$$1.\ A \longrightarrow C_B : \{\!|\, N_A \,|\!\}_{K_{AB}}$$
$$1'.\ C_B \longrightarrow A : \{\!|\, N_A \,|\!\}_{K_{AB}}$$
$$2'.\ A \longrightarrow C_B : \{\!|\, s(N_A) \,|\!\}_{K_{AB}}$$
$$2.\ C_B \longrightarrow A : \{\!|\, s(N_A) \,|\!\}_{K_{AB}}$$

Very simple, but note A is initiator in $1.$, responder in $2'$.

Early Results

Clark and Jacob attack

$$
\begin{aligned}
&1.\ A \longrightarrow B : \{\!|\, N_A \,|\!\}_{K_{AB}} \\
&2.\ B \longrightarrow A : \{\!|\, s(N_A) \,|\!\}_{K_{AB}}
\end{aligned}
$$

$$
\begin{aligned}
&1.\ A \longrightarrow C_B : \{\!|\, N_A \,|\!\}_{K_{AB}} \\
&1'.\ C_B \longrightarrow A : \{\!|\, N_A \,|\!\}_{K_{AB}} \\
&2'.\ A \longrightarrow C_B : \{\!|\, s(N_A) \,|\!\}_{K_{AB}} \\
&2.\ C_B \longrightarrow A : \{\!|\, s(N_A) \,|\!\}_{K_{AB}}
\end{aligned}
$$

Very simple, but note A is initiator in $1.$, responder in $2'$.

Good results on other non-theorems from the literature (see paper)

Continuing Work

Aim:

Find attacks in cases where sufficient model has too large a branching rate for model checking

Continuing Work

Aim:

Find attacks in cases where sufficient model has too large a branching rate for model checking

Will test system first on standard protocols

Continuing Work

Aim:

Find attacks in cases where sufficient model has too large a branching rate for model checking

Will test system first on standard protocols

Exploit ability to attack protocols with many participants

- e.g. ELK group protocol, CLIQUE suite, Cocaine auction, etc.

Continuing Work

Aim:

Find attacks in cases where sufficient model has too large a branching rate for model checking

Will test system first on standard protocols

Exploit ability to attack protocols with many participants

- e.g. ELK group protocol, CLIQUE suite, Cocaine auction, etc.

Develop formalism

- would like to be able to accept exact conjectures used in Isabelle/HOL approach

Conclusions

Comon-Nieuwenhuis method for proof by consistency fully implemented

- first in `Saturate`, and now in SPASS

Conclusions

Comon-Nieuwenhuis method for proof by consistency fully implemented

- first in `Saturate`, and now in SPASS

System applied to inductive security protocol model

- can refute incorrect conjectures and extract the attacks

Conclusions

Comon-Nieuwenhuis method for proof by consistency fully implemented

- first in `Saturate`, and now in SPASS

System applied to inductive security protocol model

- can refute incorrect conjectures and extract the attacks

Aim to find attacks that model checking hasn't found

- e.g. perhaps on conference key protocols, anonymous auction protocols, etc.

Conclusions

Comon-Nieuwenhuis method for proof by consistency fully implemented

- first in `Saturate`, and now in SPASS

System applied to inductive security protocol model

- can refute incorrect conjectures and extract the attacks

Aim to find attacks that model checking hasn't found

- e.g. perhaps on conference key protocols, anonymous auction protocols, etc.

More information:

`http://www.dai.ed.ac.uk/~grahams/fcs/`