COL876: SPECIAL TOPICS IN FORMAL METHODS

Formal verification of security protocols

Lecture 9, 18 September 2023

RECAP

- Overarching theme: “Dolev-Yao model” = “intruder is network”
- Saw two ways of formalizing security protocol execution
- As a transition system over knowledge states
- As a labelled transition system over tuples involving a multiset of processes, a substitution, and fresh names

A DIFFERENT PERSPECTIVE

- In all this, what are we really interested in?
- Most of the time, just intruder knowledge
- Why maintain anything that detracts from that?
- Other agents' knowledge states, remaining processes etc
- Somehow capture intruder knowledge as a function of the current state of execution?

INTRUDER KNOWLEDGE

- Predicate K(t) means "Intruder knows t"
- Can always recast our derivation system as a system over K(t) rather than t itself
- $X \vdash K(t_1)$ and $X \vdash K(t_2) \implies X \vdash K((t_1, t_2))$ etc
- Okay, but what about the actual execution?

INTRUDER KNOWLEDGE

- Any send puts a term out onto the channel
  - the intruder picks it up
- Any receive picks up a term from the channel
  - the intruder should have been able to generate said term
- Can think of a protocol description as a sequence of receives and sends
  - each receive implies a corresponding send
  - can cast these as implications over intruder knowledge!

EXAMPLE

$$A \rightarrow B : A, \mathsf{enc}(m, \mathsf{pk}(B))$$
$$B \rightarrow A : \mathsf{enc}(m, \mathsf{pk}(A))$$

- The first send can be modelled as follows

$$\{\} \implies K((A, \mathrm{enc}(m, \mathrm{pk}(B))))$$

- The second one can be modelled as follows

$$K((A, \mathrm{enc}(m, \mathrm{pk}(B)))) \implies K(\mathrm{enc}(m, \mathrm{pk}(A)))$$

BAN LOGIC [1990]

- Convert a protocol into a series of derivation rules over intruder knowledge
- Combine with background theory (term derivation system)
- Check for a derivation of the intruder's knowing a secret!
- So why not just do this?

BAN LOGIC [1990]

- Convert a protocol into a series of derivation rules over intruder knowledge
- Hard to do correctly!
- Need extra operators to capture freshness etc
- Ideal: implications between receives and sends without converting entire protocol into intruder knowledge

MULTISET REWRITING IN TAMARIN

- States: Multisets of "facts"
- Special facts: Fr(t), In(t), Out(t), K(t)
- Rules l—[a]—r move the system from one state to another
- A fact is not "persistent" by default (gets consumed by a rule!)

MULTISET REWRITING IN TAMARIN

- Rules l—[a]—r move the system from one state to another
- Transition corresponding to this rule: S —[a]—> $(S \setminus l\sigma) \cup (r\sigma)$
- Execution is a path through states
  - For each n, Fr(n) only appears once to the RHS of a transition
- Trace corresponding to an execution, each transition of which is labelled by a_i : $[a_1 a_2 \ldots a_n]$

MULTISET REWRITING IN TAMARIN

$$A \rightarrow B : A, \mathsf{enc}(m, \mathsf{pk}(B))$$
$$B \rightarrow A : \mathsf{enc}(m, \mathsf{pk}(A))$$

What does A do? Assume a PKI in place, then, for the first action:

Choose fresh m

Choose a B

Construct and send enc(m, pk(B))

MULTISET REWRITING IN TAMARIN

$$A \rightarrow B : A, \mathsf{enc}(m, \mathsf{pk}(B))$$
$$B \rightarrow A : \mathsf{enc}(m, \mathsf{pk}(A))$$

```
rule Register_pk:
  [ Fr(~ltk) ] - -> [ !Ltk($A, ~ltk), !Pk($A, pk(~ltk)) ]

rule init1:
  let t = enc(m, pk(~ltk)) in
  [ Fr(~m), !Ltk($B, ~ltk)] - -[ FirstSend(~m, $B) ]-> [ Out(t) ]
```