Holistic Analysis of Mix Protocols

Giampaolo Bella, Denis Butin and David Gray
Introduction

- Security protocols often analysed in isolation
- Real word: protocol sequencing / stacking / interleaving
- Inductive Method: protocol verification through theorem proving
- Scales up to protocol composition
- Example: Certification + Authentication protocols
Motivation

- Formal analysis of *isolated* protocols mature
- Protocol composition much less studied . . .
- . . . despite specific attacks!
Scyther (Cas Cremers)

Composition derived from isolated analysis under certain conditions

Else, brute force composition analysis possible, but search space may become too large
Method: the Inductive approach

- Mathematical induction on protocol steps
- Dolev-Yao threat model
- Tool support: Isabelle/HOL interactive theorem prover
Running example

- Generic certification protocol with a CA
- Mutual authentication: Needham-Schroeder Public Key with Lowe’s fix
- Sequential composition
Certification guarantees

- A message sent by the CA contains two well-formed certificates
- Those certificates contain the public key of the mentioned agent
- If an agent obtains a well-formed certificate, CA generated it
Derived authentication protocol guarantees

The mix protocol resulting from combining the certification and authentication protocol enjoys the following additional guarantees:

▶ A honest initiator sends the responder a message containing a confidential nonce
▶ That message is encrypted with the responder’s public key
Derived authentication protocol guarantees (cont’d)

- A honest responder replies to the initiator with a message containing a different, confidential nonce.
- That message is encrypted with the initiator’s public key.
Formalisation paradigm

\[
\begin{align*}
\text{NS2: } & \left[ evs2 \in \text{ns\_public} ; \text{Nonce NB} \notin \text{used evs2} ; evscb \in \text{cert} ; \\
& \text{Gets } B \left( Crypt \ (pubEK \ B) \ \{ \text{Nonce NA, Agent A} \} \right) \ \in \ set \ evs2 ; \\
& Crypt \ (priSK \ CA) \ \{ \text{Key K, Agent A} \} \ \in \ \text{parts}(\text{knows} \ B \ evscb) \right] \\
\implies & \text{Says } B \ A \ \left( Crypt \ K \ \{ \text{Nonce NA, Nonce NB,} \right. \\
& \text{Agent B} \} \left) \ # \ evs2 \in \text{ns\_public} \right.
\end{align*}
\]
Summary

- Arbitrary mix protocols holistic analysis possible in Isabelle/HOL
- Demonstrated on a certification + authentication sequence example
- More work than automated provers, but increased flexibility
Future Work

- Tackle protocol mixes problematic for Scyther
- Several protocols at once
- More intricate protocol interactions
Principles of the inductive method

- Number of agents is unbounded, session interleaving is allowed: replay attack weakness detected
- Cryptographic keys: type key, different subtypes for private / public / encryption / signature
- Events: *Says* (models sending), *Gets* (reception), *Notes* (knowledge)
- Trace: history of network events. Inductive reasoning over traces.
- Focus is *not* security of *algorithms*: treated as black boxes in Isabelle
Message set operators

- Fundamental operators, constantly used in security statements
- **parts**: decompose into atomic message components, even ciphertext for which decrypting key unavailable
- **analz**: like parts, but leaving undecryptable ciphertext untouched
- **synth**: build up messages from message components. Includes encryption if encrypting key available
Formal protocol model

- Every protocol step modeled as inductive rule with pre- and postconditions
- Protocol model is set of all admissible traces under those rules
- Empty trace modeled by \textit{Nil} event
- Threat model (DY) represented by \textit{Fake} event
- Agents’ knowledge derived from traces