Outline

- Protocol security
- Analysis methods
- Multiset rewriting with $\exists$
  - Rewrite formalism with “choose new value”
  - Protocol modeling within this framework
- Decision problems
- Applications of the MSR framework
  - Kerberos 5 [Butler, Cervesato, Jaggard, Sc.]
  - Contract-signing protocols [Chadha, Kanovich, Sc.]
  [Chadha, Mitchell, Sc., Shmatikov]

Protocol Security

◆ Cryptographic Protocol
  • Program distributed over network
  • Use cryptography to achieve goal

◆ Attacker
  • Read, intercept, replace messages, and remember their contents

◆ Correctness
  • Attacker cannot learn protected secret or cause incorrect protocol completion
Run of protocol

Initiate

Respond

Correct if no security violation in any run
Protocol Analysis Methods

◆ Non-formal approaches (useful, but no tools...)
  - Some crypto-based proofs [Bellare, Rogaway]
  - Communicating Turing Machines [Canetti]

◆ BAN and related logics
  - Axiomatic semantics of protocol steps

◆ Methods based on operational semantics
  - Intruder model derived from Dolev-Yao
  - Protocol gives rise to set of traces
    - Denotation of protocol = set of runs involving arbitrary number of principals plus intruder
Example projects and tools

- **Prove protocol correct**
  - Paulson’s “Inductive method”, others in HOL, PVS,
  - MITRE - Strand spaces
  - Process calculus: Abadi-Gordon, Gordon-Jeffrey

- **Search using symbolic representation of states**
  - Meadows: NRL Analyzer, Millen: CAPSL

- **Exhaustive finite-state analysis**
  - FDR, based on CSP [Lowe, Roscoe, Schneider, ...]
  - Murphi, CASPER, CAPSL, ...

All depend on behavior of protocol in presence of attack
Protocol analysis spectrum

- Hand proofs
- Poly-time calculus
- Multiset rewriting with $\exists$
- Spi-calculus
- Strands
- Paulson
- NRL
- Bolignano
- BAN logic
- FDR
- $\text{Mur}_\varphi$
A notation for inf-state systems

- Many previous models are buried in tools
- Define common model in tool-independent formalism
Notation commonly found in literature

- The notation describes protocol traces
- Does not
  - specify initial conditions
  - define response to arbitrary messages
  - characterize possible behaviors of attacker

\[
\begin{align*}
A & \rightarrow B : \{ A, \text{Nonce}_a \}_{K_b} \\
B & \rightarrow A : \{ \text{Nonce}_a, \text{Nonce}_b \}_{K_a} \\
A & \rightarrow B : \{ \text{Nonce}_b \}_{K_b}
\end{align*}
\]
Non-deterministic *infinite-state systems*

**Facts**

\[ F ::= P(t_1, \ldots, t_n) \]

\[ t ::= x \mid c \mid f(t_1, \ldots, t_n) \]

**States** \( \{ F_1, \ldots, F_n \} \)

- Multiset of facts
  - Includes network messages, private state
  - Intruder will see messages, not private state
Rewrite rules

◆ Transition

- \( F_1, \ldots, F_k \rightarrow \exists x_1 \ldots \exists x_m. \ G_1, \ldots, G_n \)

◆ What this means

- If \( F_1, \ldots, F_k \) in state \( \sigma \), then a next state \( \sigma' \) has
  - Facts \( F_1, \ldots, F_k \) removed
  - \( G_1, \ldots, G_n \) added, with \( x_1 \ldots x_m \) replaced by new symbols
  - Other facts in state \( \sigma \) carry over to \( \sigma' \)

- Free variables in rule universally quantified

◆ Note

- Pattern matching in \( F_1, \ldots, F_k \) can invert functions
- Linear Logic: \( F_1 \otimes \ldots \otimes F_k \rightarrow \exists x_1 \ldots \exists x_m (G_1 \otimes \ldots \otimes G_n) \)
Common Intruder Model

Derived from Dolev-Yao model

- Adversary is nondeterministic process
- Adversary can
  - Block network traffic
  - Read any message, decompose into parts
  - Decrypt if key is known to adversary
  - Insert new message from data it has observed
- Adversary cannot
  - Gain partial knowledge
  - Guess part of a key
  - Perform statistical tests, ...
Formalize Intruder Model

- Intercept, decompose and remember messages
  \[ N_1(x) \rightarrow M(x) \quad N_2(x,y) \rightarrow M(x), M(y) \]
  \[ N_3(x) \rightarrow M(x) \]
- Decrypt if key is known
  \[ M(\text{enc}(k,x)), M(k) \rightarrow M(x) \]
- Compose and send messages from “known” data
  \[ M(x) \rightarrow N_1(x), M(x) \]
  \[ M(x), M(y) \rightarrow N_2(x,y), M(x), M(y) \]
  \[ M(x) \rightarrow N_3(x), M(x) \]
- Generate new data as needed
  \[ \exists x. M(x) \]

*Highly nondeterministic, same for any protocol*
Protocol theory

◆ Initialization theory
  • Bounded theory that “precedes” protocol run
  • Example: \( \exists \text{key. Principal(key)} \)

◆ Role generation theory
  • \( \text{Principal(key)} \rightarrow A_0(\text{key}), \text{Principal(key)} \)
  • \( \text{Principal(key)} \rightarrow B_0(\text{key}), \text{Principal(key)} \)

◆ Role theory
  • Finite ordered list of rules
    \( A_i(\ldots), N_j(\ldots) \rightarrow \exists \ldots A_k(\ldots), N_l(x) \) where \( i < k, j < l \)
  • Can also have persistent predicates on left/right
Two-phase intruder theory

Avoid pointless looping by intruder
- $M(x), M(y) \rightarrow N(x,y), M(x), M(y)$
- $N(x,y) \rightarrow M(x), M(y)$

Phase 1: Decomposition
Phase 2: Composition
Thesis: MSR Model is accurate

🔹 Captures “Dolev-Yao-Needham-Millen-Meadows- ...” model
  - MSR defines set of traces protocol and attacker
  - Connections with approach in other formalisms

🔹 Useful for protocol analysis
  - Errors shown by model are errors in protocol
  - If no error appears, then no attack can be carried out using only the actions allowed by the model
Complexity results using MSR

<table>
<thead>
<tr>
<th>Intruder with $\exists$</th>
<th>Bounded # of roles</th>
<th>Bounded use of $\exists$</th>
<th>Unbounded use of $\exists$</th>
</tr>
</thead>
<tbody>
<tr>
<td>？？</td>
<td>NP – complete</td>
<td>DExp – time</td>
<td>Undecidable</td>
</tr>
<tr>
<td>Intruder w/o $\exists$</td>
<td>$\neq$, $=$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>？？</td>
<td>DExp – time</td>
<td>Undecidable</td>
<td></td>
</tr>
<tr>
<td>？？</td>
<td>NP – complete</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All: Finite number of different roles, each role of finite length, bounded message size

Key insight: existential quantification ($\exists$) captures cryptographic nonce; main source of complexity

[Durgin, Lincoln, Mitchell, Scedrov]
Lower bounds from Horn clauses

<table>
<thead>
<tr>
<th>Intruder with $\exists$</th>
<th>Bounded # of roles</th>
<th>Bounded # of $\exists$</th>
<th>Unbounded # of $\exists$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neq, =$</td>
<td>NP-complete:</td>
<td>$??$</td>
<td>Undecidable:</td>
</tr>
<tr>
<td>= only</td>
<td>Provable by bounded-length proof</td>
<td>Dextime: Datalog</td>
<td>Datalog + $\exists$</td>
</tr>
<tr>
<td>Intruder w/o $\exists$</td>
<td>$\neq, =$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>= only</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All: Finite number of different roles, each role of finite length, bounded message size

Need to show that hard instances of Horn clause inference can be represented in the restricted form of a security protocol

[Durgin, Lincoln, Mitchell, Scedrov]
Additional decidable cases

◆ Bounded role instances, unbounded msg size
  • Huima 99: decidable
  • Amadio, Lugiez: NP w/ atomic keys
  • Rusinowitch, Turuani: NP-complete, composite keys
  • Other studies, e.g., Kusters: unbounded # data fields

◆ Constraint systems
  • Cortier, Comon: Limited equality test
  • Millen, Shmatikov: Finite-length runs

All: bound number of role instances
Using MSR for protocol analysis

- Extensions and general properties
  - Add dependent types and subsorting [C]
  - DY intruder is most powerful attacker [C]

- Relate to other models
  - Strand space model [CDLMS]
  - Linear logic provability [CDKS]

- Prove protocols correct
  - Kerberos 5 [Butler, Cervesato, Jaggard, Sc.]
  - Contract signing [Chadha, Kanovich, Sc.]
    [Chadha, Mitchell, Sc., Shmatikov]
Kerberos Authentication Protocol

Protocol goals
• Repeatedly authenticate a client to multiple servers
• Minimize use of client’s long term key(s)
• Does not guard against DOS attacks

Kerberos 4 - 1989

Kerberos 5
• Specified in RFC 1510 (1993)
• Subsequent revisions by working group

A real world protocol
• Windows 2000 (RFC 1510 + extensions)
• User login, file access, printing, etc.
Previous formal analysis

◆ Kerberos 4
  • Analyzed using inductive approach
    (Bella & Paulson)

◆ Kerberos 5
  • Simplified version analyzed with Murϕ
    (Mitchell, Mitchell, & Stern)
Kerberos 5 Analysis: Goals

- Give precise statement and formal analysis of a real world protocol
  - Find a real world protocol - Kerberos 5
  - Pick favorite formalization method - MSR
- Identify and formalize protocol goals
- Give proofs of achieved protocol goals
  - Gain experience in reasoning with MSR
- Note any anomalous behavior
  - Suggest possible fixes, test these
Kerberos 5

- **Client**: \( C \) wants ticket for end server \( S \)
  - Tickets are encrypted - unreadable by \( C \)
- **\( C \) first obtains long term (e.g., 1 day) ticket from a Kerberos Authentication Server \( K \)
  - Makes use of \( C \)'s long term key
- **\( C \) then obtains short term (e.g., 5 min.) ticket from a Ticket Granting Server \( T \)
  - Based on long term ticket from \( K \)
  - \( C \) sends this ticket to \( S \)
Protocol Messages

C \quad \text{Please give me ticket for T} \quad K

C \quad \text{Ticket for C to give to T} \quad K

C \quad \text{Ticket from K, one for S?} \quad T

C \quad \text{Ticket for C to give to S} \quad T

C \quad \text{Ticket from T} \quad S

C \quad \text{Confirmation (optional)} \quad S
Overview of Results

◆ Formalized Kerberos 5 at different levels of detail
◆ Observed anomalous behavior
  • Some properties of Kerberos 4 do not hold for Kerberos 5
  • Proved authentication properties that do hold for Kerberos 5
◆ Proofs of properties which do hold
  • Methods adapted from Schneider
◆ Interactions with Kerberos working group
Rank and Corank

- Inspired by work of Schneider
- Define functions on MSR facts
  - k-Rank - encryptions by $k$
    - Data origin authentication
  - E-Corank - level of protection by keys in $E$
    - Secrecy
- Proofs
  - State desired property
  - Find applicable (co)rank functions
  - Determine effect of MSR rules on these functions

(End glimpse of Kerberos 5 analysis)
Conclusions

 Thesis

• Protocol analysis requires precise definition of possible runs under attack
• Multiset rewriting with ∃
  - Provides natural, usable formalism
  - Captures set of runs
  - Exhibits uniformity of DY attacker
  - Related to linear logic, other protocol notations
    • Can use proof-theoretic results from LL
    • Can approximate MSR model by finite-state analysis
Conclusions

Results

• Decision problems
  - NP-complete with bounded role instances
  - Dexp-time complete with bounded nonces ($\exists$)
  - Undecidable even if everything else bounded

• Applications
  - Metatheory
    • Two attackers no better than one
    • Correctness of model checking optimizations
  - Protocol analysis
    • Contract signing, Kerberos v5
Future directions

◆ Specification language
  • MSR defines traces, execution tree
  • Need to specify correctness formally

◆ Programming language?
  • Separate commands done from those that remain
  • Distinguish local knowledge from global state

◆ Quantification over protocols
  • Every protocol satisfying $\varphi$ also satisfies $\psi$.
  • Composition: Properties of $\text{Compose}(A, B)$ from properties of $A$ and $B$
Multiset Rewriting and Security Protocol Analysis

Andre Scedrov
University of Pennsylvania