Fine-Grained MSR Specifications
for
Quantitative Security Analysis

Iliano Cervesato
iliano@itd.nrl.navy.mil

ITT Industries, inc @ NRL Washington, DC

http://theory.stanford.edu/~iliano/

Protocol Analysis Seminar, NPS, Monterey, CA
February 9, 2004
Qualitative (Dolev-Yao) Analysis

• Classifies protocol operations in
  - Possible (Dolev-Yao)
    - Reception/transmission
    - Crypto with key, ...
  - Impossible
    - Guessing keys
    - Breaking crypto, ...

• Security assessed only on possible ops
  - “Easily” achieved by most current tools
  - What next?
Analysis beyond Dolev-Yao

Data

Symbolic → Bit-oriented

Crypto

Perfect

Real

More ops
- xor
- DH, ...

Type confusion

Guessing

Probabilistic

Cost-aware

Crypto hybrid
- probability
- complexity
Cost-Aware Security Analysis

• Assign cost to operations
  ➢ Including non Dolev-Yao
    ▪ Discrete logarithm, factoring, ...
    ▪ (Verifiable) guessing
    ▪ Principal subversion, ...

[Meadows,01]

• Applications
  ➢ Estimate actual resources needed for attacks
  ➢ Resources limitation (smart cards, PDAs, ...)
  ➢ DoS resistance assessment
  ➢ Comparing attacks or protocols

[Lowe,02]
Outline

• Protocol specification
  ➢ MSR → Fine-Grained MSR
    ▪ Technique applies to other languages
  ➢ Traces and Scripts

• Cost Model
  ➢ Operations → Scripts

• Cost-aware Security
  ➢ Threshold analysis
  ➢ Comparative analysis
MSR

- Executable protocol specification language
  - Theoretical results
    - Decidability
    - Most powerful intruder, ...
  - Practice
    - Kerberos V
    - Implementation underway

- 3 generations already
  - MSR 1: (here)
  - MSR 2: 1 + strong typing
  - MSR 3: 2 + $\omega$-multisets

- Based on Multiset Rewriting
  - Foundations in (linear) logic
  - Ties to Petri nets and process algebra
Multiset Rewriting ...

- **Multiset**: set with repetitions allowed
  - $a, b, c \neq a, a, b, c, c, c$

- **Rewrite rule**:
  - $r: N_1 \rightarrow N_2$

- **Application**:
  - $M_1 \rightarrow M_2$
  - $M', N_1 \rightarrow M', N_2$
... with Existentials

- msets of 1\textsuperscript{st}-order atomic formulas
- Rules:
  \[ r: F(x) \rightarrow \exists n. G(x,n) \]
- Application

\[ M_1 \rightarrow M_2 \]
\[ M', F(t) \rightarrow M', G(t,c) \]
\[ c \text{ not in } M_1 \]
Traces and Scripts

• Traces
  - Rewrite sequence \((r_1, \theta_1), \ldots, (r_n, \theta_n)\) from \(M_0\) to \(M_n\)
    - Rules \(r_i\)
    - Substitutions \(\theta_i\)

• Scripts
  - Parametric traces
    - \(S, (r, \xi)\)
    - \(S_1 + S_2\)
    - \(!_n S\)
  - Normal run: \(S_{NR}\)
  - Attack scripts: \(S_A\)
MSR for Security Protocols

- **Messages**
  - $A, k, n,...$
  - $\{m\}_k, (m,m'),...$

- **Predicates**
  - $N(m)$
  - $M_*(t_1,...,t_n)$
  - $M_A(t_1,...,t_n)$
    - $I(m)$
  - $L^v(t_1,...,t_n)$

Princ., keys, nonces, ... Encryption, concat., ...

Network messages Public data Private data Intruder info. Local states
Example

- Needham-Schroeder protocol
  - Initiator role

\[\begin{align*}
\PrvK_A(k_A, k'_A), & \\
\text{Pub}K_*(B, k_B) & \\
\L (k_A, k'_A, k_B, n_A), & \\
N(\{n_A, A\}_{k_B}) & \\
\end{align*}\]
Preparing for Cost Assignment

- **Isolate operations**
  - Verification
    - Success
    - Failure
  - Construction

- **Apply rule in stages**
  - Pre-screening
  - Detailed verification

  - Split LHS in atomic steps
  - Allow failure
Fine-Grained MSR (1)

- **Rules**
  - Clean-up: \( \text{lhs} \rightarrow \text{rhs} \text{ else cr} \)

- **Predicates**
  - Registers: \( R^v(m) \)
  - Headers: \( N^h(m) \)

- **Phased execution**
  - Select rule based only on predicates
  - Verify if arguments match
    - Allow failure
Fine-Grained MSR (2)

- **Verification rules**
  - $N^h(x) \rightarrow R(x)$
  - $L^v(x) \rightarrow R(x)$
  - $R(y), R'(\text{op}_y(x)) \rightarrow R''(x)$  
    else cr
  - $R(x), R'(x) \rightarrow .$  
    else cr
  - $R(x) \rightarrow R'(m)$
  - ...

- **Construction rules**
  - Remain the same
Fine-Grained Intruder

Dolev-Yao style
- \( N^h(x) \rightarrow I(x) \)
- \( M^*(x) \rightarrow I(x) \)
- \( I(y), I(op_x(x)) \rightarrow I(x) \)

Subversion
- \( . \rightarrow X(A) \)
- \( X(A) \rightarrow . \)
- \( X(A), M_A(x) \rightarrow X(A), I(x) \)

Guessing
- \( . \rightarrow G(x) \)
- \( . \rightarrow V_1(m_1) \)
- \( . \rightarrow V_2(m_2) \)
- \( G(x), V_1(y), V_2(y) \rightarrow I(x) \)
Cost

\[ \sum_{\tau^A} v \tau^A \]

- \( \tau \): cost type
  - Time, space, energy, ...
- \( A \): principal incurring cost
- \( v \): amount of cost
  - Physical measurements
  - 0 / \( \infty \) (Dolev-Yao model)
  - Complexity classes
Assigning Cost – Basic Operations

- Network
- Storage
- Operations
  - Construction
  - Successful verification
  - Failed verification
- Subversion
- Guessing
  - Various ways

- Supports very high precision
- Difficulty depends on precision
- Possibly subjective
Assigning Costs – Traces & Scripts

- **Traces:** $\kappa(T)$
  - Add up basic costs
    - **Monotonic costs:** time, energy, ...
    - **Non-monotonic:** space, ...

- **Scripts:** $\kappa(S)$
  - **Interval arithmetic**
    - Script alternative
Quantitative Security Analysis

- A model checking view

\[ C \leq 0 \text{ (Dolev-Yao)} \]

\[ C \leq \kappa_1 \]

\[ C \leq \kappa_2 \]
Threshold Analysis

- $\kappa(S_{NR}) \leq \kappa_{HW/HCI}$ ?
  - Cost of normal run acceptable?
    - PDAs, cell phones, ...

- $\kappa(S_A) \leq \kappa_I$ ?
  - Cost of attack/defense acceptable?
  - Cost of candidate attack vs. resources
    - Non Dolev-Yao operations

- $\min x. \kappa(S_A(x)) \geq \kappa_{I^{++}}$ ?
  - Design protocol
  - Fine-tuning parameters
Comparative Analysis

- $\kappa(S_{A1}) \leq \kappa(S_{A2})$ ?
  - Comparing attacks
    - Protocol can always be attacked

- $\kappa(S^{P1}) \leq \kappa(S^{P2})$ ?
  - Comparing protocols

- $\kappa^{B}(S_{A}) \leq \kappa^{I}(S_{A})$ ?
  - Comparing attack and defense costs
    - Denial of Service
Typical Client/Server Exchange

Client

\[ s^c_q, t^c_q \rightarrow \text{request} \rightarrow s^s_q, t^s_q \]

\[ s^c_c, t^c_c \leftarrow \text{challenge} \leftarrow s^s_c, t^s_c \]

\[ s^c_r, t^c_r \rightarrow \text{response} \rightarrow -(s^s_q + s^s_c), t^s_r \]

\[ s^c_o, t^c_o \leftarrow \text{ok} \leftarrow 0, t^s_o \]

\[ \leq T \]

\[ \leq B \]
### Time DoS

1.  

<table>
<thead>
<tr>
<th>Event</th>
<th>Service Rate</th>
<th>Attack Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>$1/t_{s_q}$</td>
<td>$1/t_{c_q}$</td>
</tr>
</tbody>
</table>

- **Service rate**: $1/t_{s_q}$
- **Service rate**: $1/(t_{s_q} + t_{s_c})$
- **Attack rate**: $1/t_{c_q}$
- **Attack rate**: $1/(t_{s_q} + t_{s_c} + t_{s_r})$

- **Usualy dominated by networking costs**
- **Better attack**
Space DDoS

• Max concurrent requests
  - $B / (s^s_q + s^s_c)$

• Space allocation rate
  - $(s^s_q + s^s_c) / (t^s_q + t^s_c)$

• Space reclamation rate
  - $B / T$

• Max. concurrent attacks
  - $n \leq \frac{B (t^s_q + t^s_c)}{(s^s_q + s^s_c) T}$
    - Use large $B$
    - Keep $T$ small
Conclusions

• Quantitative protocol analysis
  ➢ Cost conscious attacks (non Dolev-Yao)
  ➢ Fine-Grained specification languages (MSR)

• Related work
  ➢ C. Meadows: Cost framework for DoS
  ➢ G. Lowe: guessing attacks
  ➢ D. Tomioka, et al: cost for spi-calculus

• Future work
  ➢ Attack costs: WEP
  ➢ DoS aware protocols: JFK, puzzle-based Client/Server
  ➢ Complexity-based costs
  ➢ Mixing probability