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/

Seminar Web Page

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

Qualitative (Dolev-Yao) Analysis

- Classifies protocol operations in
  - Possible (Dolev-Yao)
    - Reception/transmission
    - Crypto with key, …
  - "Easy" (polynomial)
  - Impossible
    - Guessing keys
    - Breaking crypto, …
  - "Hard" (exponential)
- Security assessed <u>only</u> 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 [Meadows,01]
  - Including non Dolev-Yao
    - Discrete logarithm, factoring, …
    - (Verifiable) guessing [Lowe,02]
    - Principal subversion, …
- Applications
  - Estimate actual resources needed for attacks
  - Resources limitation (smart cards, PDAs, …)
  - DoS resistance assessment
  - Comparing attacks or protocols

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 Advertisement

- 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 + ω-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 ≠ a,a,b,c,c,c

- Rewrite rule:

$$r: N_1 \rightarrow N_2$$

- Application:

$$\underbrace{M_1}\quad \rightarrow \quad \underbrace{M_2}$$

$$M', N_1 \rightarrow M', N_2$$

state

… with Existentials

- msets of 1st-order atomic formulas
- Rules:

$$r: F(\underline{x}) \rightarrow \exists \underline{n}.\ G(\underline{x},\underline{n})$$

- Application

$$\underbrace{M', F(\underline{t})}_{M_1} \rightarrow \underbrace{M', G(\underline{t},\underline{c})}_{M_2}$$

$\underline{c}$ 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 θ_i
- Scripts
  - Parametric traces
    - S, (r,ξ)
    - $S_1 + S_2$
    - $!_n\ S$
  - Normal run: S_{NR}
  - Attack scripts: S_A

MSR for Security Protocols

- Messages

➢ A, k, n,…	Princ., keys, nonces, …
➢ $\{m\}_k$, (m,m'), …	Encryption, concat., …

- Predicates

➢ N(m)	Network messages
➢ $M_*(t_1,\dots,t_n)$	Public data
➢ $M_A(t_1,\dots,t_n)$	Private data
▪ I(m)	Intruder info.
➢ $L^v(t_1,\dots,t_n)$	Local states

Example

$A \rightarrow B: \{n_A, A\}_{kB}$
$B \rightarrow A: \{n_A, n_B\}_{kA}$
$A \rightarrow B: \{n_B\}_{kB}$

- Needham-Schroeder protocol
  - Initiator role

$$\begin{bmatrix} PrvK_A(k_A, k'_A), \\ PubK_*(B, k_B) \end{bmatrix} \rightarrow \exists n_A. \begin{bmatrix} PrvK_A(k_A, k'_A), \\ PubK_*(B, k_B), \\ L(k_A, k'_A, k_B, n_A), \\ N(\{n_A, A\}_{kB}) \end{bmatrix}$$

$$\begin{bmatrix} L(k_A, k'_A, k_B, n_A), \\ N(\{n_A, n_B\}_{kA}) \end{bmatrix} \rightarrow \begin{bmatrix} N(\{n_B\}_{kB}) \end{bmatrix}$$

Preparing for Cost Assignment

- Isolate operations
  - Verification
    - Success
    - Failure
  - Construction

  - Split LHS in atomic steps
  - Allow failure

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

Fine-Grained MSR (1)

- Rules
  - Clean-up — lhs → rhs 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
  - $N^h(x) \rightarrow R(x)$
  - $L^v(\underline{x}) \rightarrow \mathbf{R(x)}$
  - $\mathbf{R(y)}, R'(op_{\underline{y}}(\underline{x})) \rightarrow \mathbf{R''(x)}$ else cr
  - $R(x), R'(x) \rightarrow .$ else cr
  - $\mathbf{R(x)} \rightarrow R'(m)$
  - …
- Construction rules
  - Remain the same

Fine-Grained Intruder

Dolev-Yao style

$I(g), I(g^x) \rightarrow I(x)$

- $N^h(x) \rightarrow I(x)$
- $M^*(x) \rightarrow I(x)$
- $\mathbf{I(y)}, I(op_{\underline{y}}(\underline{x})) \rightarrow \mathbf{I(x)}$

- $I(x) \rightarrow N^h(x)$
- $. \rightarrow \exists x.\ I(x)$
- $\mathbf{I(x)} \rightarrow I(op(\underline{x}))$

Subversion

- $. \rightarrow X(A)$
- $X(A) \rightarrow .$
- $X(A), M_A(x) \rightarrow X(A), \mathbf{I(x)}$

Guessing

$\ldots \rightarrow G(x)$

$\ldots \rightarrow V_1(m_1)$

$\ldots \rightarrow V_2(m_2)$

$G(x), V_1(y), V_2(y) \rightarrow I(x)$

Cost

$$\sum v\tau^{A}$$

- τ: cost type
  - Time, space, energy, …
- A: principal incurring cost
- v: amount of cost
  - Physical measurements
  - 0 / ∞ (Dolev-Yao model)
  - Complexity classes

Assigning Cost – Basic Operations

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

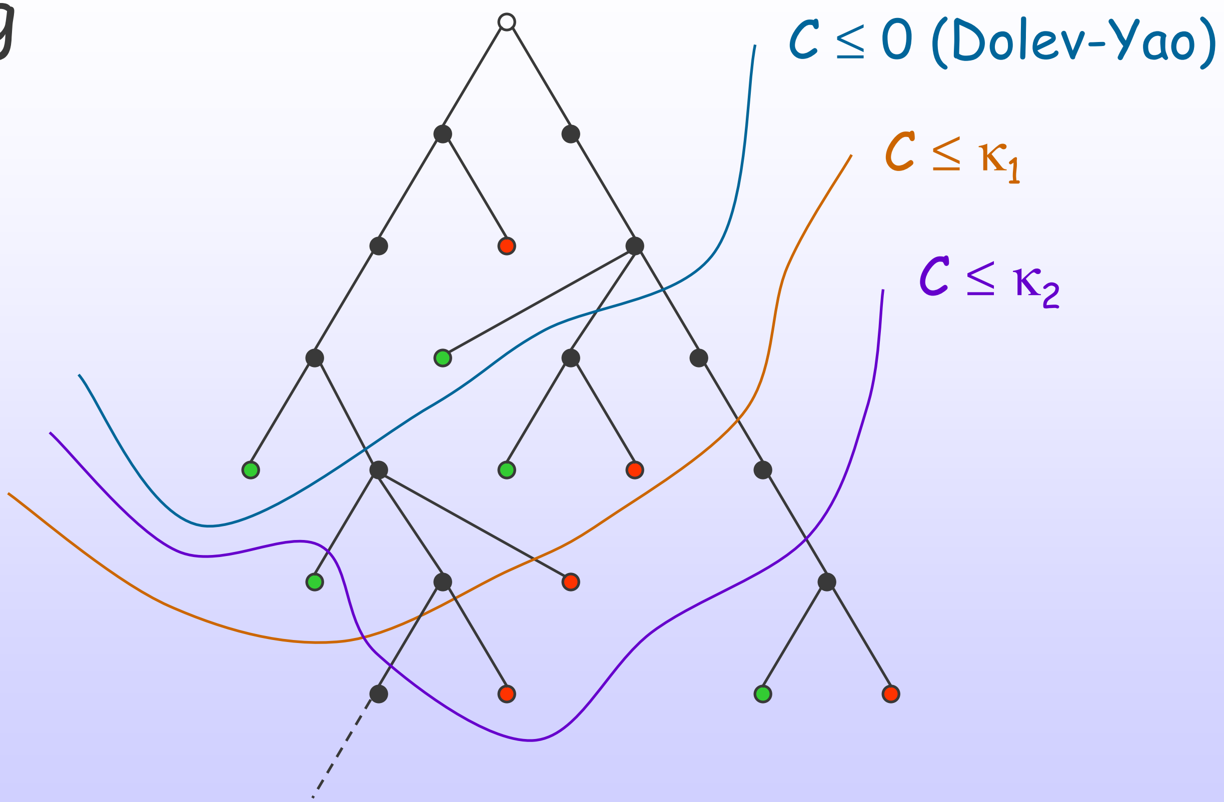
- Supports very high precision
- Difficulty depends on precision
- Possibly subjective

Assigning Costs – Traces & Scripts

- Traces: κ(T)
  - Add up basic costs
    - Monotonic costs: time, energy, …
    - Non-monotonic: space, …
- Scripts: κ(S)
  - Interval arithmetic
    - Script alternative

Quantitative Security Analysis

- A model checking view



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			Server	
s^c_q	t^c_q	request →	s^s_q	t^s_q
s^c_c	t^c_c	← challenge	s^s_c	t^s_c
s^c_r	t^c_r	response →	$-(s^s_q+s^s_c)$	t^s_r
s^c_o	t^c_o	← ok	0	t^s_o

t^s_c, t^s_r: $\leq T$

$-(s^s_q+s^s_c)$, 0: $\leq B$

Time DoS

1. $\varepsilon \xrightarrow{\wp} t^s_q$

- Service rate: $1/t^s_q$
  - Usually dominated by networking costs

2. $t^c_q \xrightarrow{q} t^s_q$

   $0 \xleftarrow{c} t^s_c$

   $\dashrightarrow$

- Service rate
  - $1/(t^s_q + t^s_c)$
- Attack rate
  - $1/t^c_q$

3. $t^c_q \xrightarrow{q} t^s_q$

   $0 \xleftarrow{c} t^s_c$

   $\varepsilon \xrightarrow{\wp} t^s_r$

- Service rate
  - $1/(t^s_q + t^s_c + t^s_r)$
- Attack rate
  - $1/t^c_q$

Better attack

Space DDoS

t^c_q	0	$\xrightarrow{q}$	s^s_q	t^s_q
0	0	$\xleftarrow{c}$	s^s_c	t^s_c
ε	0	$\xrightarrow{\wp}$	$-(s^s_q+s^s_c)$	t^s_r

$t^s_c,\ t^s_r \;\} \le T$

$\underbrace{-(s^s_q+s^s_c)}_{\le B}$

- 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
  - $$n \le \frac{B\,(t^s_q + t^s_c)}{(s^s_q + s^s_c)\,T}$$
  - Use large B
  - Keep T small

Conclusions

- Quantitative protocol analysis
  - Cost conscious attacks (non Dolev-Yao)
  - Fine-Grained specification languages (MSR)
  - Paper: http://theory.stanford.edu/~iliano/forthcoming/
- 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