



# 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/>*

# Qualitative (Dolev-Yao) Analysis

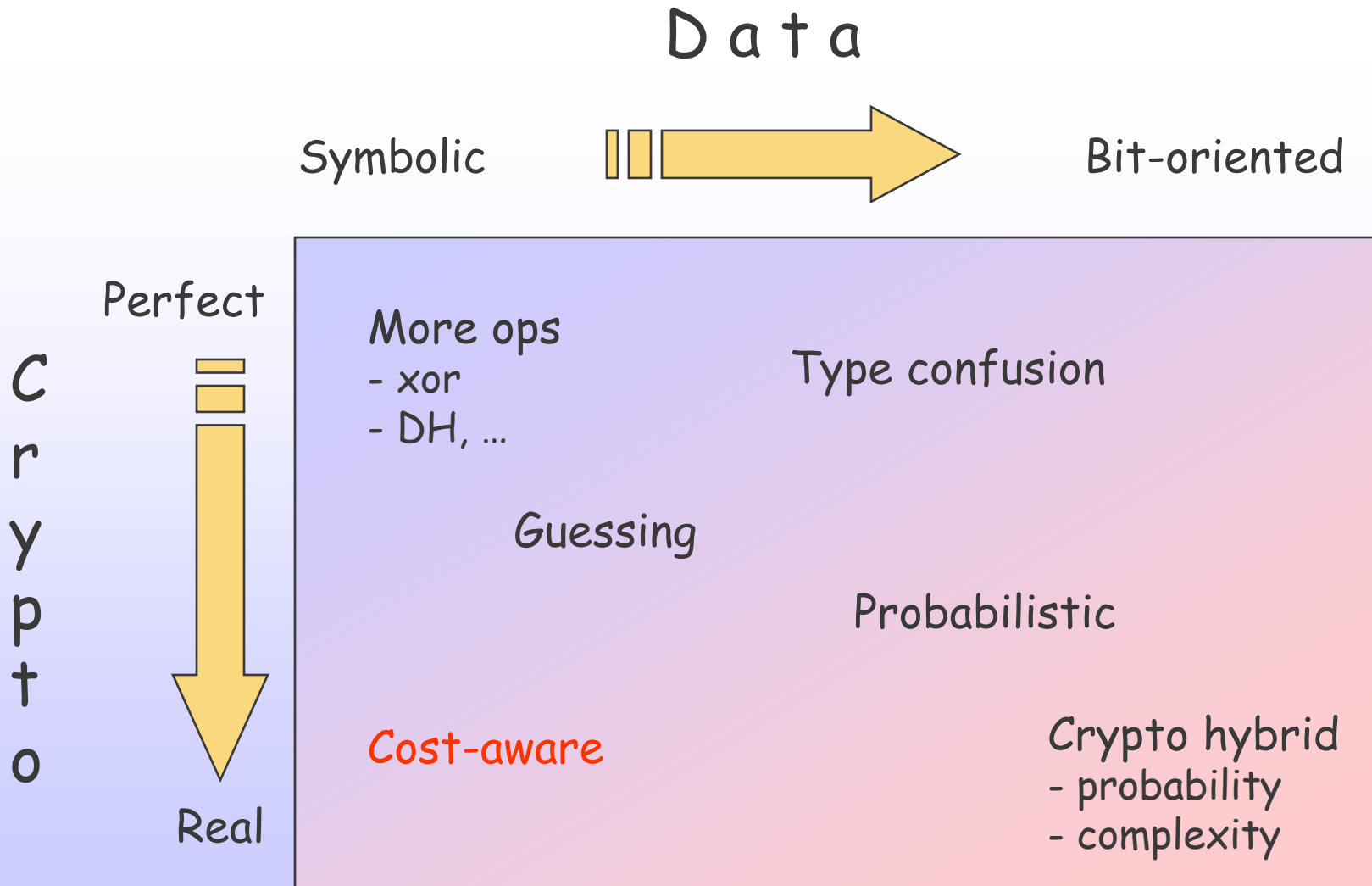
- Classifies protocol operations in
  - Possible (Dolev-Yao)
    - Reception/transmission
    - Crypto with key, ...

} "Easy"  
(polynomial)
  - Impossible
    - Guessing keys
    - Breaking crypto, ...

} "Hard"  
(exponential)
- Security assessed only on possible ops
  - "Easily" achieved by most current tools
  - What next?



# Analysis beyond Dolev-Yao



# Cost-Aware Security Analysis

- Assign cost to operations

[Meadows,01]

- Including non Dolev-Yao

- Discrete logarithm, factoring, ...
    - (Verifiable) guessing
    - Principal subversion, ...

[Lowe,02]

- 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 +  $\omega$ -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:

state

$$\begin{array}{ccc} M_1 & \rightarrow & M_2 \\ \underbrace{\phantom{M_1}} & & \underbrace{\phantom{M_2}} \\ M', N_1 & \rightarrow & M', N_2 \end{array}$$



## ... with Existentials

- msets of 1<sup>st</sup>-order atomic formulas
- Rules:

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

- Application

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

$\underline{c}$  not in  $M_1$





# Traces and Scripts

- Traces

- Rewrite sequence  $(r_1, \theta_1), \dots, (r_n, \theta_n)$  from  $M_0$  to  $M_n$ 
  - Rules  $r_i$
  - Substitutions  $\theta_i$

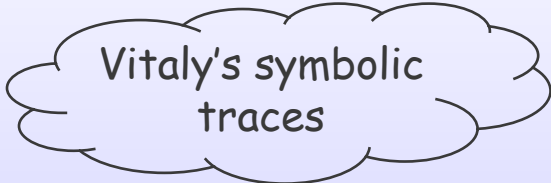
- Scripts

- Parametric traces

- $S, (r, \xi) =$
- $S_1 + S_2$
- $!_n S$

- Normal run:  $S_{NR}$

- Attack scripts:  $S_A$



Vitaly's symbolic traces

# MSR for Security Protocols

- Messages

- $A, k, n, \dots$
- $\{m\}_k, (m, m'), \dots$

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

- Predicates

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

Network messages

Public data

Private data

Intruder info.

Local states

# Example

$$\begin{array}{l} A \rightarrow B: \{n_A, A\}_{k_B} \\ B \rightarrow A: \{n_A, n_B\}_{k_A} \\ A \rightarrow B: \{n_B\}_{k_B} \end{array}$$

- Needham-Schroeder protocol
  - Initiator role

$$\left[ \begin{array}{l} \text{PrvK}_A(k_A, k'_A), \\ \text{PubK}_*(B, k_B) \end{array} \right]$$

$\rightarrow \exists n_A.$

$$\left[ \begin{array}{l} \text{PrvK}_A(k_A, k'_A), \\ \text{PubK}_*(B, k_B), \\ L(k_A, k'_A, k_B, n_A), \\ N(\{n_A, A\}_{k_B}) \end{array} \right]$$

$$\left[ \begin{array}{l} L(k_A, k'_A, k_B, n_A), \\ N(\{n_A, n_B\}_{k_A}) \end{array} \right]$$

$\rightarrow$

$$\left[ N(\{n_B\}_{k_B}) \right]$$



# Preparing for Cost Assignment

- Isolate operations

- Verification

- Success
    - Failure

- Construction

- Split LHS into atomic steps
    - Allow failure

- Apply rule in stages

- Pre-screening

- Detailed verification



# Fine-Grained MSR (1)

- Rules

- Clean-up

$lhs \rightarrow rhs \text{ else } cr$

- 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

- $N^h(x) \rightarrow R(x)$

- $L^v(\underline{x}) \rightarrow R(x)$

- $R(y), R'(op_y(\underline{x})) \rightarrow R''(x)$

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

- $N^h(x) \rightarrow I(x)$
- $M^*(x) \rightarrow I(x)$
- $I(y), I(\text{op}_y(\underline{x})) \rightarrow I(x)$

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

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

## Subversion

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

## Guessing

$$\begin{array}{l} \dots \rightarrow G(x) \\ \dots \rightarrow V_1(m_1) \\ \dots \rightarrow V_2(m_2) \\ G(x), V_1(y), V_2(y) \rightarrow I(x) \end{array}$$




# Cost

$$\sum v\tau^A$$

- $\tau$ : 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

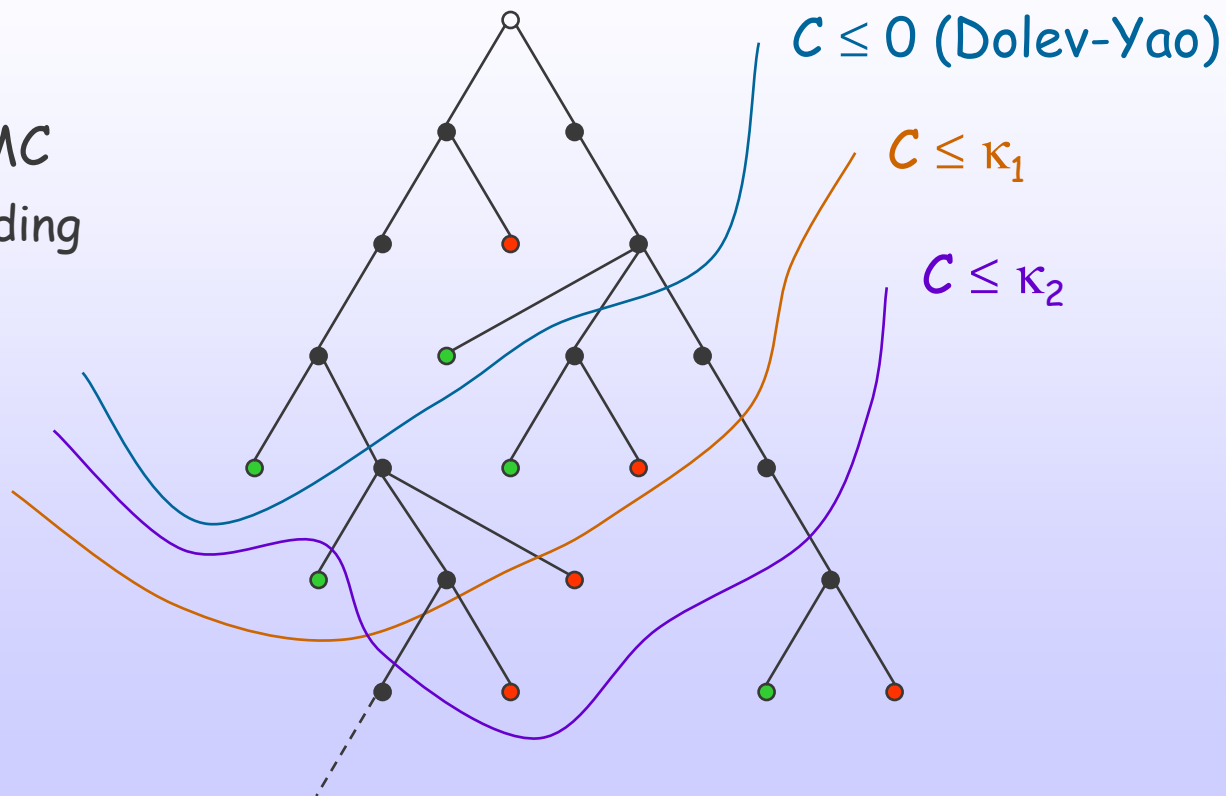
## A model checking view

- Explicit state MC

- Direct

- Symbolic MC

- Via encoding



# 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
      - "Tit for tat" - Carl Gunter

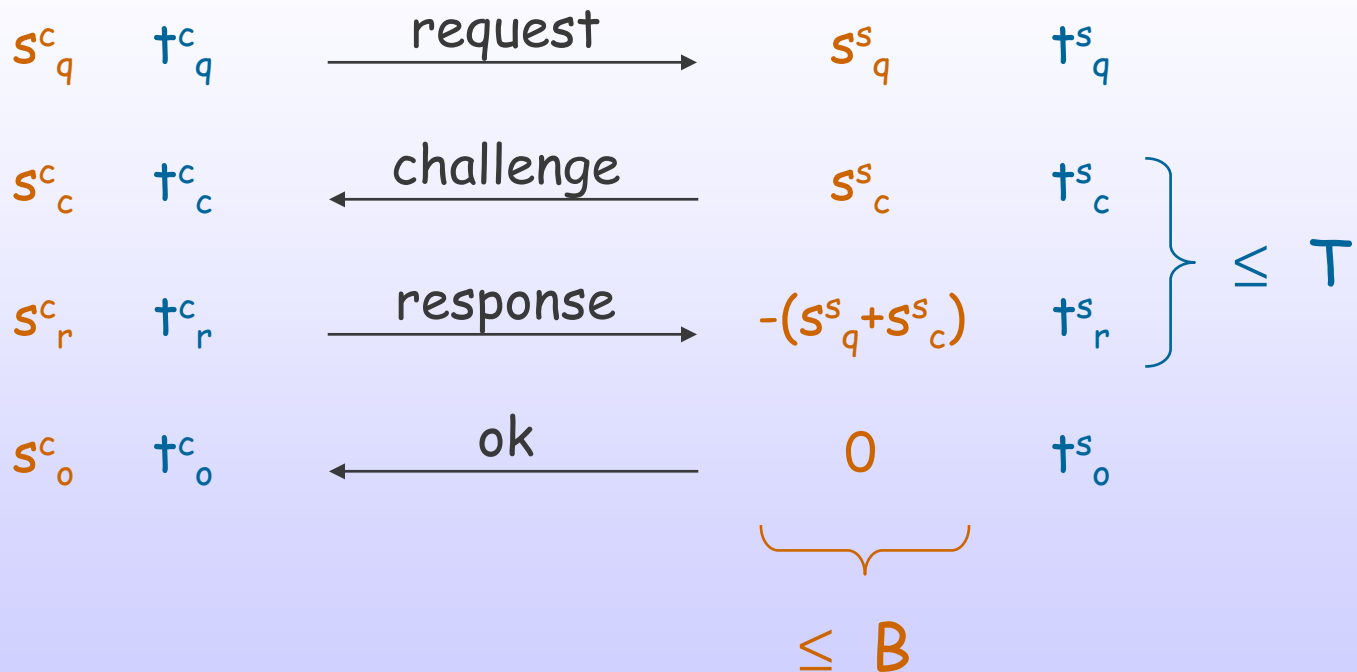


# Typical Client/Server Exchange



Client

Server





# Time DoS

1.  $\varepsilon \xrightarrow{\delta\phi} t_q^s$

- Service rate:  $1/t_q^s$ 
  - Usually dominated by networking costs

2.  $t_q^c \xrightarrow{q} t_q^s$   
 $0 \xleftarrow{c} t_c^s$   
 $\text{-----} \rightarrow$

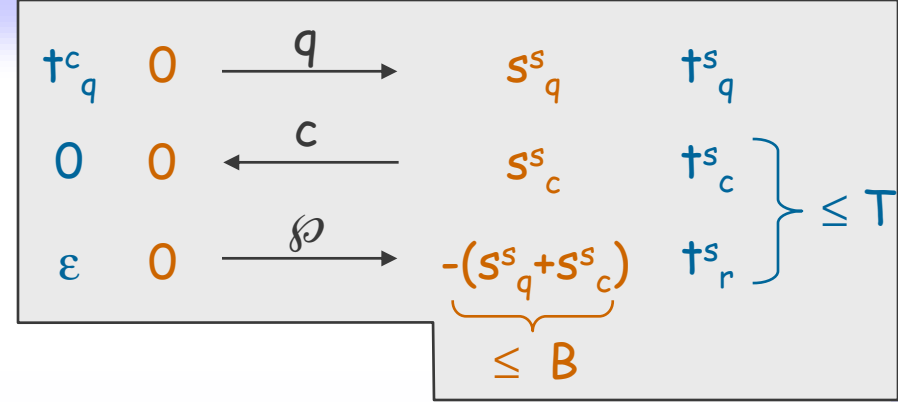
- Service rate
  - $1/(t_q^s + t_c^s)$
- Attack rate
  - $1/t_q^c$

3.  $t_q^c \xrightarrow{q} t_q^s$   
 $0 \xleftarrow{c} t_c^s$   
 $\varepsilon \xrightarrow{\delta\phi} t_r^s$

- Service rate
  - $1/(t_q^s + t_c^s + t_r^s)$
- Attack rate
  - $1/t_q^c$

Better  
attack

# Space DDoS



- Max concurrent requests

➤  $n(B) = B / (s_q^s + s_c^s)$

- Optimal time-out

➤  $t_{\min} < T$

➤  $T \leq (t_q^s + t_c^s) * (n(B) - 1)$

- Example

➤  $s_q^s + s_c^s = 128 \text{ b}$

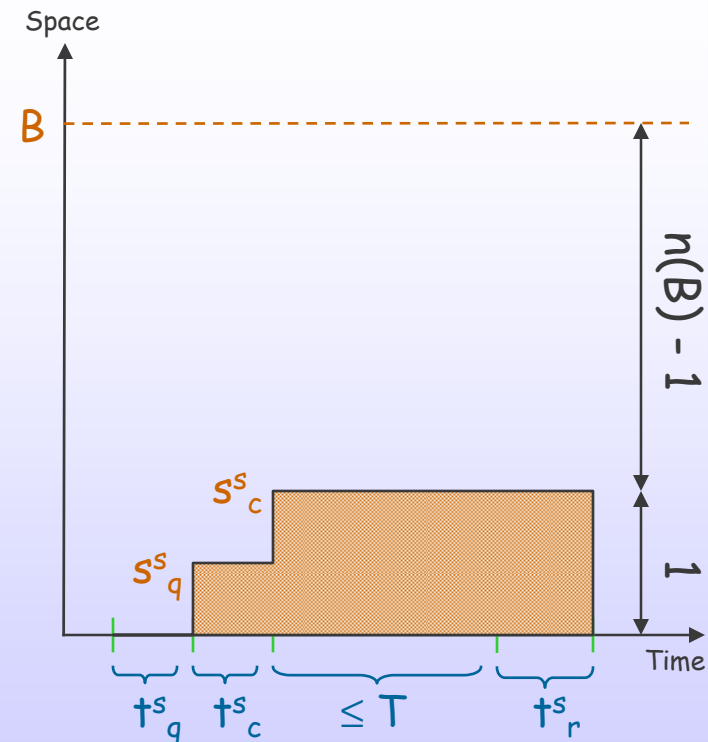
➤  $t_q^s + t_c^s = 100 \text{ ms}$

➤  $t_{\min} = 90 \text{ s}$

➤  $n(B) = 10,000$

$B = 1.28 \text{ Mb}$

$T \leq \sim 16 \text{ min}$



# 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, client puzzles, bins
- Protocol analysis as optimization problem
- Economics of network security
- Complexity-based costs and mixing probability

