MSR by Examples

Iliano Cervesato iliano@itd.nrl.navy.mil

ITT Industries, Inc @ NRL - Washington DC

http://www.cs.stanford.edu/~iliano/

Outline

- Security Protocols
- MSR
  - Multiset rewriting
  - The Neuman-Stubblebine Protocol
  - MSR

Security Protocols

Exchange of (encrypted) messages for

- Communicating secrets
- Authentication
- Contract signing
- E-commerce
- …

Neuman-Stubblebine – Phase I

A wants to access service provided by B

$A \rightarrow B: \quad A, n_A$

$B \rightarrow S: \quad B, \{A, n_A, T_B\}_{kBS}, n_B$

$S \rightarrow A: \quad \{B, n_A, k_{AB}, T_B\}_{kAS}, \{A, k_{AB}, T_B\}_{kBS}, n_B$

$A \rightarrow B: \quad \{A, k_{AB}, T_B\}_{kBS}, \{n_B\}_{kAB}$

S is the key distribution center

Ticket

Neuman-Stubblebine – Phase II

$A \rightarrow B:\ n'_A,\ \{A,\ k_{AB},\ T_B\}_{kBS}$

$B \rightarrow A:\ n'_B,\ \{n'_A\}_{kAB}$

$A \rightarrow B:\ \{n'_B\}_{kAB}$

Ticket

A wants to use the service provided by B again

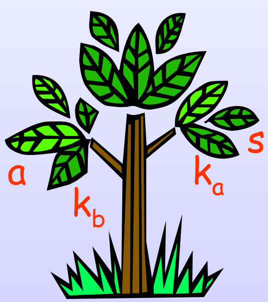
The Dolev-Yao Model of Security

- Symbolic data
  - No bits

k_a

- Black-box cryptography
  - No guessing of keys

- Partially abstract data access
  - Knowledge soup



- Found in most protocol analysis tools
  - Tractability

Why is Protocol Analysis Difficult?

- Subtle cryptographic primitives
  - Dolev-Yao abstraction
- Distributed hostile environment
  - "Prudent engineering practice"
- Inadequate specification languages
  - *… the devil is in details …*

Languages to Specify What?

- Message flow
- Message constituents
- Operating environment
- Protocol goals

Desirable Properties

- Unambiguous
- Simple
- Flexible
  - Adapts to protocol
  - Applies to a wide class of protocols
- Insightful

MSR

- Multiset rewriting with existentials
- Dependent types w/ subsorting New
- Memory predicates New
- Constraints New

Multiset Rewriting

- Multiset: set with repetitions allowed
- Rewrite rule:

$$r: N_1 \rightarrow N_2$$

- Application

$$\overbrace{M', N_1}^{M_1} \xrightarrow{r} \overbrace{M', N_2}^{M_2}$$

- Multi-step transition, reachability

Neuman-Stubblebine – Phase I

$A \rightarrow B$: A, n_A

$B \rightarrow S$: $B, \{A, n_A, T_B\}_{kBS}, n_B$

$S \rightarrow A$: $\{B, n_A, k_{AB}, T_B\}_{kAS}, \{A, k_{AB}, T_B\}_{kBS}, n_B$

$A \rightarrow B$: $\{A, k_{AB}, T_B\}_{kBS}, \{n_B\}_{kAB}$

NS-I: *B*'s point of view

$A \rightarrow B$: A, n_A

$B \rightarrow S$: $B, \{A, n_A, T_B\}_{kBS}, n_B$

$S \rightarrow A$: $\{B, n_A, k_{AB}, T_B\}_{kAS}, \{A, k_{AB}, T_B\}_{kBS}, n_B$

$A \rightarrow B$: $\{A, k_{AB}, T_B\}_{kBS}, \{n_B\}_{kAB}$

NS-I: S's point of view

$A \to B$: A, n_A

$B \to S$: $B, \{A, n_A, T_B\}_{kBS}, n_B$

$S \to A$: $\{B, n_A, k_{AB}, T_B\}_{kAS}, \{A, k_{AB}, T_B\}_{kBS}, n_B$

$A \to B$: $\{A, k_{AB}, T_B\}_{kBS}, \{n_B\}_{kAB}$

NS-I: *A*'s point of view

$A \rightarrow B$: A, n_A

$B \rightarrow S$: $B, \{A, n_A, T_B\}_{kBS}, n_B$

$S \rightarrow A$: $\{B, n_A, k_{AB}, T_B\}_{kAS}, \underbrace{\{A, k_{AB}, T_B\}_{kBS}}_{X}, n_B$

$A \rightarrow B$: $\underbrace{\{A, k_{AB}, T_B\}_{kBS}}_{X}, \{n_B\}_{kAB}$

Ticket

Sending / Receiving Messages

$$\bullet \quad \rightarrow \quad N(A, n_A)$$

Network predicate

N(t): t is a message in transit

$$N(\{B, n_A, k_{AB}, T_B\}_{kAS}, X, n_B) \quad \rightarrow \quad N(X, \{n_B\}_{kAB})$$

Terms

- Atomic terms
  - Principal names A
  - Keys k
  - Nonces n
  - …
- Term constructors
  - (_ _)
  - {_}
  - …

D e f i n a b l e

Nonces

$$\bullet \quad \rightarrow \quad \exists n_A. \quad N(A, n_A)$$

Existential variables

n_A instantiated to a new constant

$$N(\{B, n_A, k_{AB}, T_B\}_{kAS}, X, n_B) \quad \rightarrow \quad N(X, \{n_B\}_{kAB})$$

MSet Rewriting with Existentials

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

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

- Application

$$\overbrace{M', F(\underline{\mathbf{t}})}^{M_1} \xrightarrow{r} \overbrace{M', G(\underline{\mathbf{t}},\underline{\mathbf{c}})}^{M_2}$$

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

Sequencing actions

$\exists \mathcal{L}.$

Fresh!

$$\bullet \;\rightarrow\; \exists n_A. \; \begin{matrix} N(A, n_A) \\ \mathcal{L}(A, n_A) \end{matrix}$$

Role state predicate

$$\begin{matrix} \mathcal{L}(A, n_A) \\ N(\{B, n_A, k_{AB}, T_B\}_{kAS}, X, n_B) \end{matrix} \;\rightarrow\; N(X, \{n_B\}_{kAB})$$

Role state predicates

$$L_l(A, t, \ldots, t)$$

- Hold data local to a role instance
  - Lifespan = role
- Invoke next rule
  - L_l = control
  - $(A, t, \ldots, t)$ = data

Remembering Things

$$\left(\begin{array}{l} \exists L. \\ \quad \bullet \;\to\; \exists n_A. \begin{array}{l} L(A,n_A) \\ N(A,\, n_A) \end{array} \\ \begin{array}{l} L(A,n_A) \\ N(\{B,n_A,k_{AB},T_B\}_{kAS},\, X,\, n_B) \end{array} \;\to\; \begin{array}{l} N(X,\, \{n_B\}_{kAB}) \\ Tkt_A(B,k_{AB},X) \end{array} \end{array}\right)$$

Memory predicate

Memory Predicates

New

$M_A(t, \ldots, t)$

- Hold private info. across role exec.
- Support for <u>subprotocols</u>
  - Communicate data
  - Pass control
- Interface to outside system
- Implements intruder

Role owner

New

Role owner

The principal executing the role

$$
\left(
\begin{array}{llll}
\exists L. & & & \\
\bullet & \rightarrow & \exists n_A. & L(A,n_A) \\
 & & & N(A, n_A) \\
L(A,n_A) & & & N(X, \{n_B\}_{kAB}) \\
N(\{B,n_A,k_{AB},T_B\}_{kAS}, X, n_B) & \rightarrow & & Tkt_A(B,k_{AB},X)
\end{array}
\right)^{\forall A}
$$

What is what?

Types

$\exists L$: princ x nonce.

$\bullet \rightarrow \exists n_A$:nonce. $L(A,n_A)$ $N(A, n_A)$

$\forall$ B:princ.
$\forall$ n_A,n_B: nonce
$\forall$ k_{AB}: shK A B
$\forall$ k_{AS}: shK A S
$\forall$ X: msg

$L(A,n_A)$ $N(\{B,n_A,k_{AB},T_B\}_{kAS}, X, n_B) \rightarrow N(X, \{n_B\}_{kAB})$ $Tkt_A(B,k_{AB},X)$

$\forall A$

Types of Terms

- A: princ
- n: nonce
- k: shK A B
- k: pubK A
- k′: privK k
- … (definable)

Types can <u>depend</u> on term

- Captures relations between objects
  - Static
  - Local
  - Mandatory

Subtyping

τ :: msg

- Allows atomic terms in messages
- Definable
  - Non-transmittable terms
  - Sub-hierarchies

Type of predicates

$\Sigma x{:}\ \tau.\ \underline{\tau}$

- Dependent sums

$\tau^{(x)} \times \underline{\tau}$

x

- Forces associations among arguments
  - E.g.: $\text{princ}^{(A)} \times \text{pubK } A^{(k_A)} \times \text{privK } k_A$

Type Checking

New

$$\Sigma \vdash P$$

$$\Gamma \vdash t : \tau$$

t has type τ in Γ

P is well-typed in Σ

- Catches:
  - Encryption with a nonce
  - Transmission of a long term key

Data Access Specification – DAS

New

$\Sigma \Vdash P$

r is DAS-valid for A in Γ

$\Gamma \Vdash_A r$

P is DAS-valid in Σ

- Catches
  - A signing/encrypting with B's key
  - A accessing B's private data, …
- Gives meaning to Dolev-Yao intruder

NS-I: *B*'s point of view

$A \rightarrow B$: A, n_A

$B \rightarrow S$: $B, \{A, n_A, T_B\}_{kBS}, n_B$

$S \rightarrow A$: $\{B, n_A, k_{AB}, T_B\}_{kAS}, \{A, k_{AB}, T_B\}_{kBS}, n_B$

$A \rightarrow B$: $\{A, k_{AB}, T_B\}_{kBS}, \{n_B\}_{kAB}$

NS-I: B's role

$\forall B$

$\exists \mathcal{L}$: $princ^{(B)} \times princ \times nonce \times shK\ B\ S \times nonce \times time$.

$\forall A$:princ.
$\forall n_A$: nonce
$\forall k_{BS}$: shK B S
$\forall T_B$: time

$N(A, n_A)$
$Clk_B(T_B)$

$\rightarrow$

$\exists n_B$:nonce.
$N(B, \{A, n_A, T_B\}_{kBS}, n_B)$
$Clk_B(T_B)$
$\mathcal{L}(B, A, n_A, k_{BS}, n_B, T_B)$

$\forall$ ....
$\forall k_{AB}$: shK A B
$\forall n_B$: nonce
$\forall T_{now}$: time
$\forall T_V, T_e$: time

$\mathcal{L}(B, A, n_A, k_{BS}, n_B, T_B)$
$N(\{A, k_{AB}, T_B\}_{kBS}, \{n_B\}_{kAB})$
$Val_B(A, T_B, T_V)$
$(T_e = T_B + T_V)$

$\rightarrow$

$Auth_B(A, k_{AB}, T_B, T_e)$
$Val_B(A, T_B, T_V)$

Constraint

Constraints

$$\chi$$

- Guards over interpreted domain
  - Abstract
  - Modular
- Invoke constraint handler
- E.g.: timestamps
  - $(T_E = T_N + T_d)$
  - $(T_N < T_E)$

NS-I: S's point of view

$A \to B$: A, n_A

$B \to S$: $B, \{A, n_A, T_B\}_{kBS}, n_B$

$S \to A$: $\{B, n_A, k_{AB}, T_B\}_{kAS}, \{A, k_{AB}, T_B\}_{kBS}, n_B$

$A \to B$: $\{A, k_{AB}, T_B\}_{kBS}, \{n_B\}_{kAB}$

NS-I: S's role

Anchored role

$$
\left(
\begin{array}{l}
\forall A,B\text{:princ.} \\
\forall k_{AS}\text{: shK } A\ S \\
\forall k_{BS}\text{: shK } B\ S \\
\forall n_A,n_B\text{: nonce} \\
\forall T_B\text{: time}
\end{array}
\quad
N(B, \{A,n_A,T_B\}_{kBS}, n_B)
\;\rightarrow\;
\begin{array}{l}
\exists k_{AB}\text{: shK } A\ B. \\
N(\{B,n_A,k_{AB},T_B\}_{kAS}, \\
\quad \{A,k_{AB},T_B\}_{kBS}, \\
\quad n_B)
\end{array}
\right)^{S}
$$

Neuman-Stubblebine – Phase II

$A \rightarrow B: \; n'_A \,, \{A,\, k_{AB},\, T_B\}_{kBS}$

$B \rightarrow A: \; n'_B \,, \{n'_A\}_{kAB}$

$A \rightarrow B: \; \{n'_B\}_{kAB}$

NS-II: *A*'s role

$\left(\exists L: princ^{(A)} \times princ^{(B)} \times shK\ A\ B \times nonce. \right)^{\forall A}$

$\forall B{:}princ.$
$\forall k_{AB}{:}\ shK\ A\ B$
$\forall X{:}\ msg$

$Tkt_A(B, k_{AB}, X) \rightarrow \exists n'_A{:}nonce.\ N('n_A, X)\ \ Tkt_A(B, k_{AB}, X)\ \ L(A, B, k_{AB}, n'_A)$

$\forall \ldots.$
$\forall n'_A, n'_B{:}\ nonce$

$L(A, B, k_{AB}, n'_A)\ \ N(n'_B, \{n'_A\}_{kAB}) \rightarrow N(\{n'_B\}_{kAB})$

NS-II: B's role

$\forall B$

$\exists L$: $princ^{(B)}$ x $princ^{(A)}$ x shK A B x nonce.

$\forall$ n'_A: nonce
$\forall$ k_{BS}: shK B S
$\forall$ A:princ.
$\forall$ k_{AB}: shK A B
$\forall$ T_B, T_e: time
$\forall$ T_{now} : time

$N(n'_A, \{A, k_{AB}, T_B\}_{kBS})$
$Auth_B(A, k_{AB}, T_B, T_e)$
$Clk_B(T_{now})$
$(T_{now} < T_e)$

$\rightarrow$

$\exists n'_B$:nonce.
$N(n'_B, \{n'_A\}_{kAB})$
$Auth_B(A, k_{AB}, T_B, T_e)$
$Clk_B(T_{now})$
$L(B, A, k_{AB}, n'_B)$

$\forall$ ....
$\forall$ n'_B: nonce

$L(B, A, k_{AB}, n'_B)$
$N(\{n'_B\}_{kAB})$

$\rightarrow$

Summary: Rules

$\forall x_1 : \tau_1.$ … $\forall x_n : \tau_n.$ lhs $\rightarrow$ $\exists y_1 : \tau'_1.$ … $\exists y_{n'} : \tau'_{n'}.$ rhs

lhs:

- N(t) Network
- L(t, …, t) Local state
- M_A(t, …, t) Memory
- χ Constraints

rhs:

- N(t) Network
- L(t, …, t) Local state
- M_A(t, …, t) Memory

Summary: Roles

Role state pred. var. declarations

- Generic roles

$$\left(\begin{array}{lllll} \exists L: \tau'_1{}^{(x_1)} \times \ldots \times \tau'_n{}^{(x_n)} \\ \ldots \\ \forall \mathbf{x}:\tau. & lhs & \exists \mathbf{y}:\tau'. & \rightarrow & rhs \\ \ldots & & \ldots & & \ldots \\ \forall \mathbf{x}:\tau. & lhs & \exists \mathbf{y}:\tau'. & \rightarrow & rhs \end{array}\right)^{\forall A}$$

Role owner

- Anchored roles

$$\left[\begin{array}{lllll} \exists L: \tau'_1{}^{(x_1)} \times \ldots \times \tau'_n{}^{(x_n)} \\ \ldots \\ \forall \mathbf{x}:\tau. & lhs & \exists \mathbf{y}:\tau'. & \rightarrow & rhs \\ \ldots & & \ldots & & \ldots \\ \forall \mathbf{x}:\tau. & lhs & \exists \mathbf{y}:\tau'. & \rightarrow & rhs \end{array}\right]^{A}$$

Summary: Snapshots

$$C = [S]^{R}_{\Sigma}$$

Active role set

State

- $N(t)$
- $L_l(t, \ldots, t)$
- $M_A(t, \ldots, t)$

Signature

- $a : \tau$
- $L_l : \underline{\tau}$
- $M\_ : \underline{\tau}$

Summary: Execution Model

$$P \triangleright C \to C'$$

1-step firing

- Activate roles
- Generates new role state pred. names
- Instantiate variables
- Apply rules
- Skips rules

Summary: Rule application

$$r = \quad F, \chi \rightarrow \exists \underline{n}:\underline{\tau}.\ G(\underline{n})$$

- Constraint check

$$\Sigma \models \chi \qquad \text{(constraint handler)}$$

- Firing

$$[\underbrace{S_1}_{S,\ F}]^{R(r,\rho)^A}_{\Sigma} \rightarrow [\underbrace{S_2}_{S,\ G(\underline{c})}]^{R\rho^A}_{\Sigma,\ \underline{c}:\underline{\tau}} \qquad \underline{c} \text{ not in } S_1$$

Properties

- Decidability of type checking
- Type preservation
- Access control preservation
- Decidability of DAS verification
- Completeness of Dolev-Yao intruder

Completed Case-Studies

- Full Needham-Schroeder public-key
- Otway-Rees
- Neuman-Stubblebine repeated auth.
- OFT group key management

Part III

The Intruder

Execution with an Attacker

$$P,\ P_I \triangleright C \rightarrow C'$$

- Selected principal(s): I
- Generic capabilities: P_I
  - Well-typed
  - AC-valid
- Modeled completely within MSR

The Dolev-Yao Intruder

- Specific protocol suite P_{DY}
- Underlies every protocol analysis tool
- Completeness still unproved

Capabilities of the D-Y Intruder

- Intercept / emit messages
- Decrypt / encrypt with known key
- Split / form pairs
- Look up public information
- Generate fresh data

Future work

- Experimentation
  - Clark-Jacob library
  - Fair-exchange protocols
  - More multicast
- Pragmatics
  - Type-reconstruction
  - Operational execution model(s)
  - Implementation
- Automated specification techniques