



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

$A \rightarrow B: A, n_A$

A wants to access
service provided by B

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

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

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

S is the key
distribution center

Ticket

Neuman-Stubblebine – Phase II


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

Ticket

A wants to use the service
provided by B again

The Dolev-Yao Model of Security

- Symbolic data

- No bits

01001011010... k_a

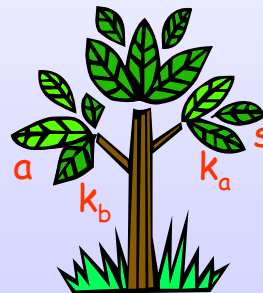
- Black-box cryptography

- 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 
- Memory predicates 
- Constraints 



Multiset Rewriting

- Multiset: set with repetitions allowed
- Rewrite rule:

$$r: N_1 \rightarrow N_2$$

- Application

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

- Multi-step transition, reachability



Neuman-Stubblebine – Phase I

$A \rightarrow B: A, n_A$

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

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

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



NS-I: B's point of view

$A \rightarrow B: A, n_A$

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

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

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



NS-I: S's point of view

$A \rightarrow B: A, n_A$

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

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

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



NS-I: A's point of view

$A \rightarrow B: A, n_A$

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

$S \rightarrow A: \{B, n_A, k_{AB}, T_B\}_{k_{AS}}, \underbrace{\{A, k_{AB}, T_B\}_{k_{BS}}, n_B}_X$

$A \rightarrow B: \underbrace{\{A, k_{AB}, T_B\}_{k_{BS}}, \{n_B\}_{k_{AB}}}_X$

Ticket

Sending / Receiving Messages



• $\rightarrow N(A, n_A)$

Network predicate

$N(t)$: t is a message in transit

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

Terms

- Atomic terms

- Principal names A
- Keys k
- Nonces n
- ...



D
e
f
i
n
a
b
l
e

- Term constructors

- $(_ _)$
- $\{ _ \}$
- ...



D
e
f
i
n
a
b
l
e



Nonces



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

Existential variables
 n_A instantiated to a new constant

$$N(\{B, n_A, k_{AB}, T_B\}_{KAS}, X, n_B) \rightarrow N(X, \{n_B\}_{KAB})$$

MSet Rewriting with Existentials

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

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

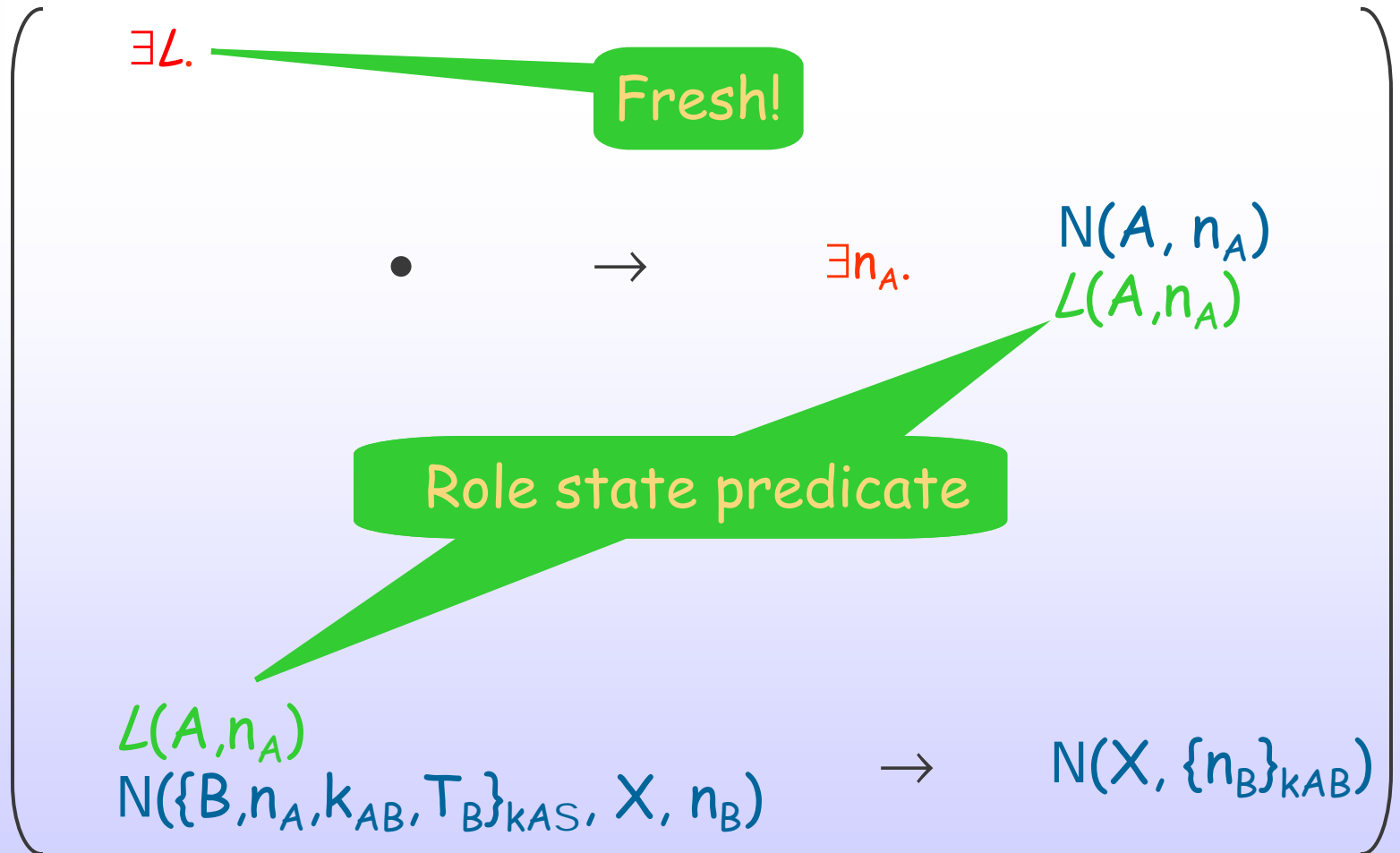
- Application

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

\underline{c} not in M_1



Sequencing actions



Role state predicates

$$L_i(A, t, \dots, t)$$

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



Remembering Things

$$\left(\begin{array}{l} \exists L. \\ \bullet \rightarrow \exists n_A. \begin{array}{l} L(A, n_A) \\ N(A, n_A) \end{array} \\ \\ \begin{array}{l} L(A, n_A) \\ N(\{B, n_A, k_{AB}, T_B\}_{KAS}, X, n_B) \end{array} \rightarrow \begin{array}{l} N(X, \{n_B\}_{KAB}) \\ Tkt_A(B, k_{AB}, X) \end{array} \end{array} \right)$$

Memory
predicate

Memory Predicates



$$M_A(t, \dots, t)$$

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



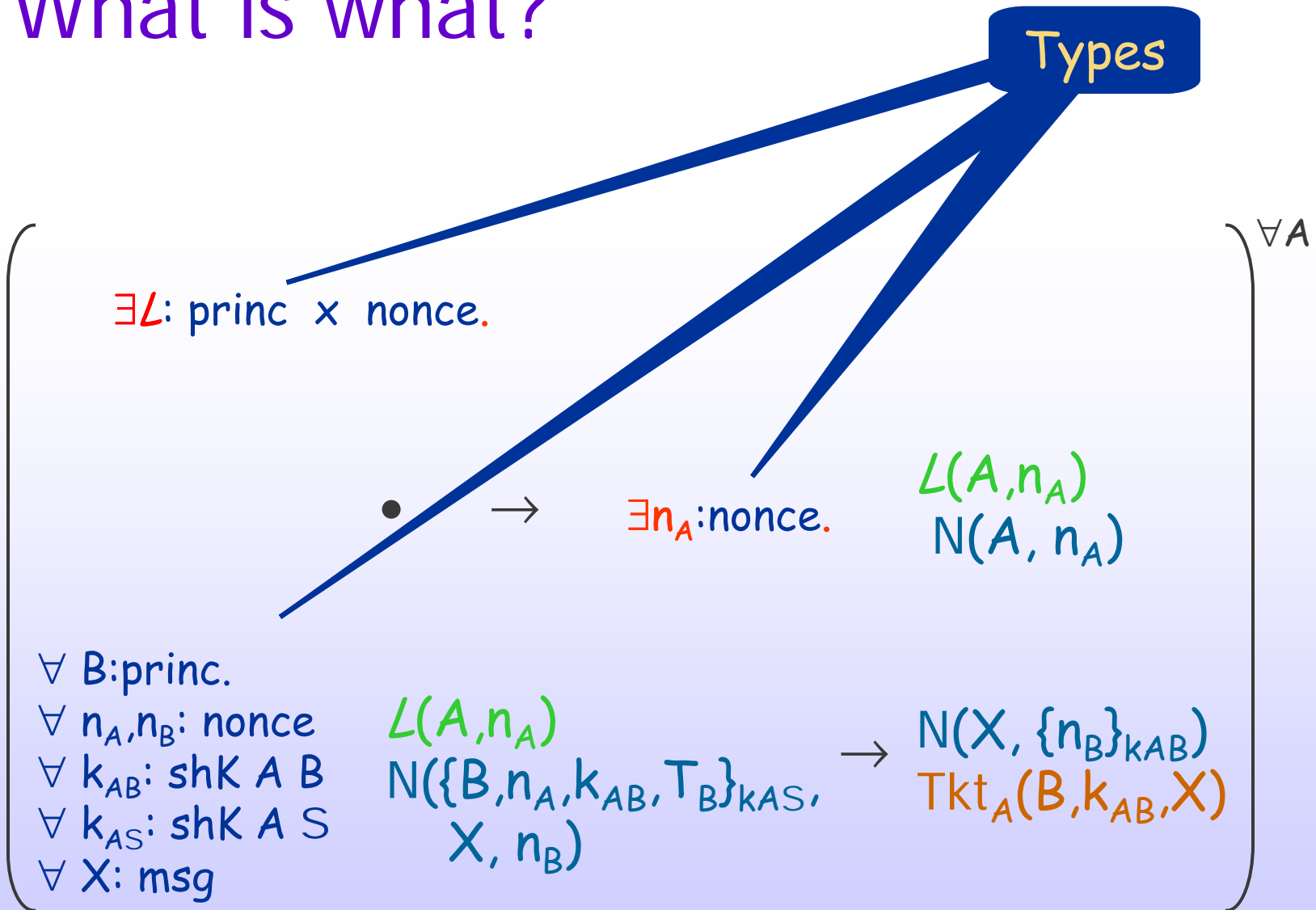
Role owner

New


Role owner
The principal executing the role

$$\left(\begin{array}{l} \exists L. \bullet \rightarrow \exists n_A. \begin{array}{l} L(A, n_A) \\ N(A, n_A) \end{array} \\ \\ \begin{array}{l} L(A, n_A) \\ N(\{B, n_A, k_{AB}, T_B\}_{KAS}, X, n_B) \end{array} \rightarrow \begin{array}{l} N(X, \{n_B\}_{KAB}) \\ Tkt_A(B, k_{AB}, X) \end{array} \end{array} \right) \forall A$$

What is what?



Types of Terms

- 
- A : princ
 - n : nonce
 - k : shK A B
 - k : pubK A
 - k' : privK k
 - ... (definable)

Types can depend
on term

- Captures relations between objects
 - Static
 - Local
 - Mandatory

Subtyping

$\tau :: \text{msg}$

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



Type of predicates

- Dependent sums

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

A thought bubble containing the expression $\sum x. \tau. \underline{\tau}$ is connected by a line to a callout bubble pointing to the $\underline{\tau}$ term in the expression $\tau(x) \times \underline{\tau}$. The callout bubble contains the variable x .

- Forces associations among arguments

➤ E.g.: $\text{princ}^{(A)} \times \text{pubK } A^{(k_A)} \times \text{privK } k_A$



Type Checking



$\Sigma \vdash P$

t has type
 τ in Γ

$\Gamma \vdash t : \tau$

P is well-
typed in Σ

- Catches:

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



Data Access Specification – DAS



r is DAS-valid
for A in Γ

New

$\Gamma \Vdash_A r$

$\Sigma \Vdash P$

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

$A \rightarrow B: A, n_A$

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

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

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



NS-I: B's role

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

$\forall A: \text{princ}.$

$\forall n_A: \text{nonce}$

$\forall k_{BS}: \text{shK } B \ S$

$\forall T_B: \text{time}$

$\exists n_B: \text{nonce}.$

$N(A, n_A)$
 $\text{Clk}_B(T_B)$

\rightarrow

$N(B, \{A, n_A, T_B\}_{k_{BS}}, n_B)$
 $\text{Clk}_B(T_B)$
 $L(B, A, n_A, k_{BS}, n_B, T_B)$

$\forall \dots$

$\forall k_{AB}: \text{shK } A \ B$

$\forall n_B: \text{nonce}$

$\forall T_{\text{now}}: \text{time}$

$\forall T_V, T_e: \text{time}$

$L(B, A, n_A, k_{BS}, n_B, T_B)$

$N(\{A, k_{AB}, T_B\}_{k_{BS}},$
 $\{n_B\}_{k_{AB}})$

$\text{Val}_B(A, T_B, T_V)$

$(T_e = T_B + T_V)$

\rightarrow

$\text{Auth}_B(A, k_{AB}, T_B, T_e)$
 $\text{Val}_B(A, T_B, T_V)$

Constraint

$\forall B$



Constraints



χ

- 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

$A \rightarrow B: A, n_A$

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

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

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



NS-I: S's role



Anchored role

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

Neuman-Stubblebine – Phase II

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

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

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



NS-II: A's role



$$\left(\begin{array}{l}
 \exists L: \text{princ}^{(A)} \times \text{princ}^{(B)} \times \text{shK } A \ B \times \text{nonce}. \\
 \\
 \forall B: \text{princ}. \\
 \forall k_{AB}: \text{shK } A \ B \quad \text{Tkt}_A(B, k_{AB}, X) \quad \rightarrow \quad \begin{array}{l}
 \exists n'_A: \text{nonce}. \\
 N(n'_A, X) \\
 \text{Tkt}_A(B, k_{AB}, X) \\
 L(A, B, k_{AB}, n'_A)
 \end{array} \\
 \forall X: \text{msg} \\
 \\
 \forall \dots \\
 \forall n'_A, n'_B: \text{nonce} \quad \begin{array}{l}
 L(A, B, k_{AB}, n'_A) \\
 N(n'_B, \{n'_A\}_{k_{AB}})
 \end{array} \quad \rightarrow \quad N(\{n'_B\}_{k_{AB}})
 \end{array} \right) \forall A$$

NS-II: B's role

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

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

$N(n'_A, \{A, k_{AB}, T_B\}_{k_{BS}})$
 $\text{Auth}_B(A, k_{AB}, T_B, T_e)$
 $\text{Clk}_B(T_{\text{now}})$
 $(T_{\text{now}} < T_e)$

\rightarrow

$\exists n'_B: \text{nonce}.$

$N(n'_B, \{n'_A\}_{k_{AB}})$
 $\text{Auth}_B(A, k_{AB}, T_B, T_e)$
 $\text{Clk}_B(T_{\text{now}})$
 $\mathcal{L}(B, A, k_{AB}, n'_B)$

$\forall \dots$

$\forall n'_B: \text{nonce}$

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

\rightarrow

$\forall B$



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

- $N(t)$ Network
- $L(t, \dots, t)$ Local state
- $M_A(t, \dots, t)$ Memory
- χ Constraints

- $N(t)$ Network
- $L(t, \dots, t)$ Local state
- $M_A(t, \dots, t)$ Memory



Summary: Roles

Role state pred.
var. declarations

- Generic roles

$$\left[\begin{array}{c} \exists L: \tau'_1(x_1) \times \dots \times \tau'_n(x_n) \\ \dots \\ \forall x:\tau. \text{ lhs } \exists y:\tau'. \rightarrow \text{ rhs} \\ \dots \\ \forall x:\tau. \text{ lhs } \exists y:\tau'. \rightarrow \text{ rhs} \end{array} \right] \forall A$$

Role
owner

- Anchored roles

$$\left[\begin{array}{c} \exists L: \tau'_1(x_1) \times \dots \times \tau'_n(x_n) \\ \dots \\ \forall x:\tau. \text{ lhs } \exists y:\tau'. \rightarrow \text{ rhs} \\ \dots \\ \forall x:\tau. \text{ lhs } \exists y:\tau'. \rightarrow \text{ rhs} \end{array} \right] A$$



Summary: Snapshots



$$C = [S]^R_\Sigma$$

Active role
set

State

- $N(t)$
- $L_I(t, \dots, t)$
- $M_A(t, \dots, t)$

Signature

- $a : \tau$
- $L_I : \underline{\tau}$
- $M_{\underline{\quad}} : \underline{\tau}$

Summary: Execution Model



$$P \triangleright C \rightarrow C'$$

1-step
firing

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

Summary: Rule application

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

- Constraint check

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

- Firing

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

Properties

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



Case-Studies

- Full Needham-Schroeder public-key
- Otway-Rees
- Neuman-Stubblebine repeated auth.
- OFT group key management
- Kerberos 5





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

