Symbolic Approach to the Analysis of Security Protocols

Stéphane LAFRANCE
École Polytechnique de Montréal

June 27, 2003
Foundations of Computer Security - FCS'03

(In collaboration with prof. John MULLINS)

- Motivations
- Security Protocol Process Algebra
- Constrained processes
- Symbolic semantics
- Bisimulation equivalence between constrained processes
- Symbolic bisimulation

Motivations

Specification of symbolic values:

- Random numbers, fresh keys, nonces, fake addresses, fake messages,...
- Input action $c(x).P$ in value-passing process algebra:

$$\sum_{a \in \mathcal{M}} c(a).P$$

- Infinite systems.

Specification of Denial of Service in security protocols:

- An unauthenticated intruder sending fake messages to cause DoS;
- DoS through resource exhaustion;
- Specification of function calls and attribution of (memory/CPU) cost on these actions.
- Information flow properties (non-interference)

Message Specification

Message Algebra

$$
\begin{array}{rcllllll}
t & ::= & n & (\textit{number}) & | & id & (\textit{identifier}) & | \quad x \quad (\textit{variable}) \\
 & | & (t,t) & (\textit{pair}) & | & \{t\}_t & (\textit{encryption}) & | \quad [t]_t \quad (\textit{signature}) \\
 & | & h(t) & (\textit{hashing}) & & & &
\end{array}
$$

Decidable Logic for Messages

$$
\begin{array}{rcllllll}
\phi & ::= & \mathbf{0} & (\textit{false}) & | & \mathbf{1} & (\textit{true}) \\
 & | & t == t & (\textit{term equation}) & | & \mathcal{M}(t) & (\textit{message predicate}) \\
 & | & \mathcal{N}(t) & (\textit{number predicate}) & | & \mathcal{K}(t) & (\textit{key predicate}) \\
 & | & \mathcal{I}(t) & (\textit{identifier predicate}) & | & \phi \wedge \phi & (\textit{conjunction}) \\
 & | & \exists_x \ \phi & & & & &
\end{array}
$$

Message Specification

Functions $f \in \mathcal{F}$ with (domain) characterisation formula ϕ_f:

- $\text{extract}^i((a_1, a_2)) = a_i$
  with $\phi_{\text{extract}^i}(x) ::= \exists_{x_1,x_2}\ x == (x_1, x_2)$
- $\text{enc}(k, a) = \{a\}_k$
  with $\phi_{\text{enc}}(x_1, x_2) ::= \mathcal{K}(x_1) \wedge \mathcal{M}(x_2)$
- $\text{dec}(k^{-1}, \{a\}_k) = a$
  with $\phi_{\text{dec}}(x_1, x_2) ::= \mathcal{K}(x_1) \wedge \exists_y\ x_2 == \{y\}_{x_1}$
- $\text{hash}(a) = h(a)$
  with $\phi_{\text{hash}}(x) ::= \mathcal{M}(x)$
- $\text{sign}(k, a) = [a]_k$
  with $\phi_{\text{sign}}(x_1, x_2) ::= \mathcal{K}(x_1) \wedge \mathcal{M}(x_2)$
- $\text{checksign}(k^{-1}, a, [a]_k)$
  with $\phi_{\text{checksign}}(x_1, x_2, x_3) ::= \mathcal{K}(x_1) \wedge \mathcal{M}(x_2) \wedge x_3 = [x_2]_{x_1}$

Message Specification

Generating Functions $new \in \mathcal{F}$ with (image) characterisation formula ϕ_{new}:

- $\mathrm{newMessage}(-)$
  with $\phi_{\mathrm{newMessage}}(x) ::= \mathcal{M}(x)$
- $\mathrm{newNumber}(-)$
  with $\phi_{\mathrm{newNumber}}(x) ::= \mathcal{N}(x)$
- $\mathrm{newId}(-)$
  with $\phi_{\mathrm{newId}}(x) ::= \mathcal{I}(x)$
- $\mathrm{newKey}(-)$
  with $\phi_{\mathrm{newKey}}(x) ::= \mathcal{K}(x)$

Security Protocols Process Algebra - **SPPA**

SPPA Processes:

$$
\begin{array}{rcll}
P & ::= & \mathbf{0} & (\textit{empty agent}) \\
 & | & \overline{c}(t).P & (\textit{output}) \\
 & | & c(x).P & (\textit{input}) \\
 & | & x := f(t).P & (\textit{function call}) \\
 & | & x := new(-).P & (\textit{generating function call}) \\
 & | & P + P & (\textit{sum}) \\
 & | & P \| P & (\textit{parallel composition}) \\
 & | & [t = t']\ P & (\textit{match}) \\
 & | & P \backslash L & (\textit{restriction}) \\
 & | & P/\mathcal{O} & (\mathcal{O}\textit{-observation})
\end{array}
$$

Constrained Processes:

$$\langle P,\ \phi \rangle \quad \text{with } fv(P) = fv(\phi).$$

Symbolic Semantics

Input

$$\frac{-}{\langle c(x).P,\ \phi\rangle \xrightarrow{c_{id_P}(x)} \langle P,\ \exists_x \phi \wedge \mathcal{M}(x)\rangle}$$

Output

$$\frac{-}{\langle \overline{c}(t).P,\ \phi\rangle \xrightarrow{\overline{c_{id_P}}(t)} \langle P,\ \phi\rangle}$$

Function

$$\frac{-}{\langle x{:=}f(t).P,\ \phi\rangle \xrightarrow{f_{id_P}} \langle P,\ \exists_x \phi \wedge \phi_f(t) \wedge x{==}f(t)\rangle}$$

Generator

$$\frac{-}{\langle x{:=}new(-).P,\ \phi\rangle \xrightarrow{new_{id_P}} \langle P,\ \exists_x \phi \wedge \phi_{new}(x)\rangle}$$

Match

$$\frac{\langle P,\ \phi\rangle \xrightarrow{\alpha} \langle P',\ \phi'\rangle}{\langle [t{=}t']P,\ \phi\rangle \xrightarrow{\alpha} \langle P',\ \phi' \wedge t{==}t'\rangle}$$

Symbolic Semantics

Sum

$$\frac{\langle P,\ \phi\rangle \xrightarrow{\alpha} \langle P',\ \phi'\rangle \quad \text{and} \quad fv(\phi)\cap fv(\psi)=\emptyset}{\langle P+Q,\ \phi\ \wedge\ \psi\rangle \xrightarrow{\alpha} \langle P',\ \phi'\ \wedge\ \psi\rangle}$$

Parallel

$$\frac{\langle P,\ \phi\rangle \xrightarrow{\alpha} \langle P',\ \phi'\rangle, \quad \alpha\notin C \quad \text{and} \quad fv(\phi)\cap fv(\psi)=\emptyset}{\langle P\|Q,\ \phi\ \wedge\ \psi\rangle \xrightarrow{\alpha} \langle P'\|Q,\ \phi'\ \wedge\ \psi\rangle}$$

Synchronisation

$$\frac{\langle P,\ \phi\rangle \xrightarrow{\overline{c_{id_1}}(t)} \langle P',\ \phi'\rangle \quad \text{and} \quad \langle Q,\ \psi\rangle \xrightarrow{c_{id_2}(x)} \langle Q',\ \psi'\rangle}{\langle P\|Q,\ \phi\ \wedge\ \psi\rangle \xrightarrow{\overline{\delta^c_{id_1}}(t)} \langle P'\|Q,\ \phi'\wedge\psi\rangle \xrightarrow{\delta^c_{id_2}(t)} \langle P'\|Q',\ \phi'\wedge\psi'\wedge x==t\rangle}$$

Restriction

$$\frac{\langle P,\ \phi\rangle \xrightarrow{\alpha} \langle P',\ \phi'\rangle}{\langle P\backslash L,\ \phi\rangle \xrightarrow{\alpha} \langle P'\backslash L,\ \phi'\ \wedge\ \phi^L_\alpha\rangle}$$

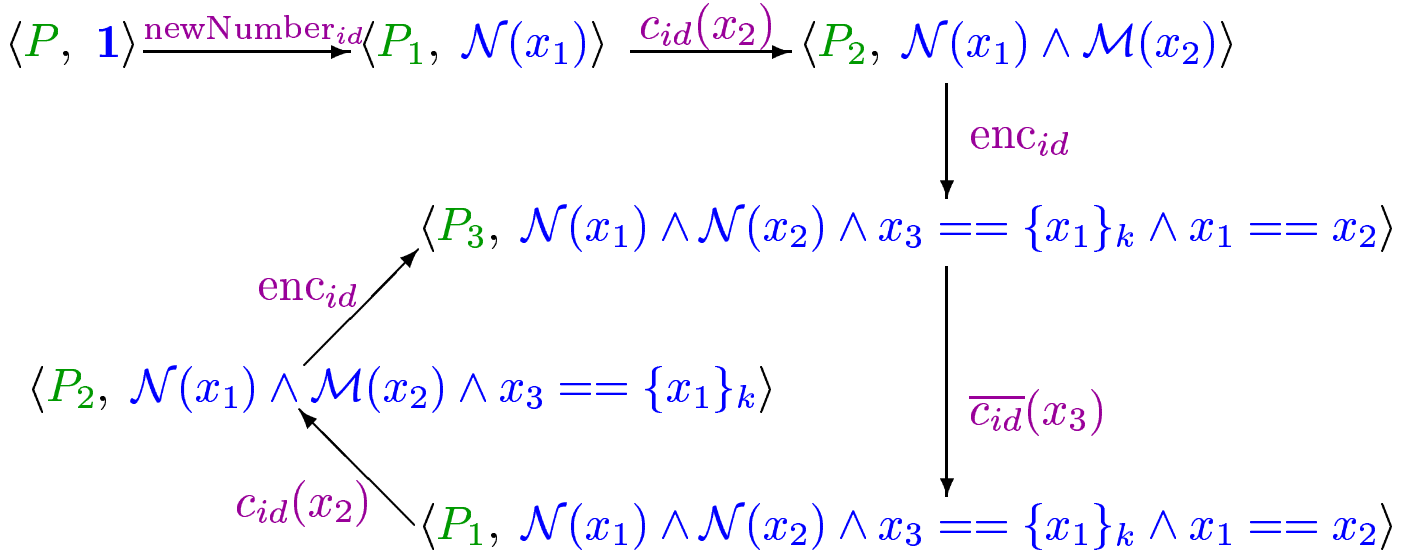
Observation

$$\frac{\langle P,\ \phi\rangle \xrightarrow{\gamma} \langle P',\ \phi'\rangle \quad \text{and} \quad \gamma\in\mathcal{O}^{-1}(\alpha)}{\langle P/\mathcal{O},\ \phi\rangle \xrightarrow{\alpha} \langle P'/\mathcal{O},\ \phi'\rangle}$$

Example

$$
\begin{array}{lcl}
P & ::= & x_1 := \mathrm{newNumber}(-).P_1 \\
P_1 & ::= & c(x_2).P_2 \\
P_2 & ::= & [x_1 = x_2]\ x_3 := \mathrm{enc}(k, x_1).P_3 \\
P_3 & ::= & \overline{c}(x_3).P_1
\end{array}
$$

with $id_P = id$ and $k \in \mathcal{K}$.



Bisimulation Equivalence for Constrained Processes

Set $\mathcal{FR}$ of relations between free variables $R \subseteq fv(P) \times fv(Q)$.

Bisimulation between constrained processes $\langle P,\ \phi \rangle$ and $\langle Q,\ \psi \rangle$ w.r.t. relation R:

$$\mathcal{R} = \{\mathcal{R}^{\varrho}\}_{\varrho}$$

for every valuation $\varrho : \mathcal{V} \to \mathcal{M}$.

Notation: $\langle P,\ \phi \rangle \ \simeq\ \langle Q,\ \psi \rangle$ (w.r.t. some relation R).

Bisimulation Equivalence for Constrained Processes

Each relation $\mathcal{R}^{\varrho} \subseteq \mathcal{D}(\langle P,\ \phi\rangle) \times \mathcal{D}(\langle Q,\ \psi\rangle) \times \mathcal{FR}$ is such that:

- If $\varrho \models \phi$, $\varrho \models \psi$ and ϱ is consistent with R, then $(\langle P,\ \phi\rangle, \langle Q,\ \psi\rangle, R) \in \mathcal{R}^{\varrho}$;
- Whenever $(\langle P_1,\ \phi_1\rangle, \langle Q_1,\ \psi_1\rangle, R_1) \in \mathcal{R}^{\varrho}$,
  1. if $\varrho \models \phi_2$ and

     $$\langle P_1,\ \phi_1\rangle \xrightarrow{\alpha} \langle P_2,\ \phi_2\rangle$$

     then

     $$\langle Q_1,\ \psi_1\rangle \xrightarrow{\alpha'} \langle Q_2,\ \psi_2\rangle$$

     for $\alpha' = \alpha[y_1/x_1] \ldots [y_n/x_n]$ with $(x_i, y_i) \in R_1$, such that $(\langle P_2,\ \phi_2\rangle, \langle Q_2,\ \psi_2\rangle, R_1) \in \mathcal{R}^{\varrho}$ and $\varrho \models \psi_2$;

  2. if $\varrho \models \psi_2$ and

     $$\langle Q_1,\ \psi_1\rangle \xrightarrow{\alpha} \langle Q_2,\ \psi_2\rangle$$

     then

     $$\langle P_1,\ \phi_1\rangle \xrightarrow{\alpha'} \langle P_2,\ \phi_2\rangle$$

     for $\alpha' = \alpha[y_1/x_1] \ldots [y_n/x_n]$ with $(y_i, x_i) \in R_1$, such that $(\langle P_2,\ \phi_2\rangle, \langle Q_2,\ \psi_2\rangle, R_1) \in \mathcal{R}^{\varrho}$ and $\varrho \models \phi_2$.

Generalisation of Bisimulation

Our bisimulation equivalence relation between constrained processes ($\langle P,\ \mathbf{1}\rangle \simeq \langle Q,\ \mathbf{1}\rangle$) is a generalisation of Milner's (strong) bisimulation equivalence for value-passing processes ($P \simeq Q$).

Theorem. *Let P and Q be closed processes. Then, $P \simeq Q$ if and only if $\langle P,\ \mathbf{1}\rangle \simeq \langle Q,\ \mathbf{1}\rangle$.*

Symbolic Bisimulation

Symbolic bisimulation $\langle P,\ \phi\rangle \simeq_{\mathrm{s}} \langle Q,\ \psi\rangle$:

A **sound** and **complete** method for verifying bisimulation between constrained processes.

Formulas occurring in $\langle P,\ \phi\rangle$ and $\langle Q,\ \psi\rangle$:

$$\Phi \ = \ \{\phi' \mid \langle P',\ \phi'\rangle \in \mathcal{D}(\langle P,\ \phi\rangle)\} \ \cup \ \{\psi' \mid \langle Q',\ \psi'\rangle \in \mathcal{D}(\langle Q,\ \psi\rangle)\}$$

Equivalence relation over valuations: $\varrho \equiv \varrho'$ if

1. For every $\phi' \in \Phi$,
$$\varrho \models \phi' \quad \text{iff} \quad \varrho' \models \phi'$$
2. For every $x, y \in fv(\Phi)$,
$$\varrho(x) = \varrho(y) \quad \text{iff} \quad \varrho'(x) = \varrho'(y).$$

Finitely many equivalence classes :

$$[\![\varrho]\!] \ = \ \{\varrho' \mid \varrho' \equiv \varrho\}$$

Symbolic Bisimulation

Symbolic bisimulation between constrained processes $\langle P,\ \phi\rangle$ and $\langle Q,\ \psi\rangle$:
finite family $\mathcal{R} = \{\mathcal{R}^{[\![\varrho]\!]}\}_{\varrho}$, for every valuation ϱ.

Theorem. $\langle P,\ \phi\rangle \simeq \langle Q,\ \psi\rangle$ *if and only if* $\langle P,\ \phi\rangle \simeq_{\mathrm{s}} \langle Q,\ \psi\rangle$.

Related Work

- **M. Hennessy and H. Lin**. Symbolic bisimulations. *Theoretical Computer Science*, 138:353–389, 1995.
- **M. Boreale**. Symbolic trace analysis of cryptographic protocols. In *Proceedings of ICALP'01*, 2001.
- **M. Fiore and M. Abadi**. Computing symbolic models for verifying cryptographic protocols. In *Proceedings of CSFW'01*, 2001.
- **S. Lafrance and J. Mullins**. Using admissible interference to detect denial of service vulnerabilities. In *Proceedings of Sixth International Workshop in Formal Methods*, 2003.

Specification of Denial of Service

Cost function for memory resources:

$$\rho_M : Actions \longrightarrow \mathcal{C}_M$$

where $\rho_M(\alpha)$ stands for the quantity of memory resources required to execute action α.

Server memory capacity M_B:
running simultaneously N actions of memory cost greater than M_B may cause a memory resource exhaustion DoS for the server B.

Intruder memory capacity M_E:
the intruder E may only execute actions of memory cost lesser or equal to M_E.

Specification of Denial of Service

Robustness against DoS

Impassivity:
No intruder (respecting its capacity) may cause **interference** *on the server's actions with memory cost greater than the server's capacity* M_B.

Interference [Focardi, Gorrieri, 1995]:
causing an action which would not have occurred otherwise.

Admissible Interference [Mullins, 2000]:
allow (predetermined) "honest" behaviours for the intruder.

Specification of Denial of Service

We observe only the server's costly actions (with cost over its memory capacity):
observation criterion $\mathcal{O}_{costly}$

Theorem. *Server B satisfies impassivity if and only if*

$$\langle E \parallel B,\ \mathbf{1} \rangle \ \simeq_{\mathcal{O}_{costly}}\ \langle B,\ \mathbf{1} \rangle$$