Process Calculi for Security

Andres Aristizabal Hugo Lopez Carlos Olarte [1]

[1]Pontificia Universidad Javeriana-Cali

Motivation

- Since mobile communications appeared, the need of secure communication increased
- Security Protocols: Important because they help establishing secure channels between communicating systems.
- Process calculi: To model communication systems.
- Security Process Calculi: To model secure communication systems.

Outline

- Syntactic sets:
  - Infinite sets of names N (With elements $n, m, A, ...$)
    - Variables over names: $x, y, ...$
    - Variables over messages : $\psi, \psi_1 ...$
  - Keys
    - Public Keys: Pub(v)
    - Private Keys: Priv(v)
  - Messages
    - Names or Key expressions
    - Composition of messages (M, M') where M, M' are messages.
    - An encryption $\{M\}_k$

Informal meaning of Processes

- *out new*$(\vec{x})M.p$: The process chooses distinct names in $\vec{n}$ and binds them to the variables in $\vec{x}$, The message $M[\vec{n}/\vec{x}]$ is output into the space and the process resumes as $p[\vec{n}/\vec{x}]$
- *in pat* $\vec{x}\vec{\psi}M.p$: This process awaits an input that matches the pattern M (i.e. n, k, N in the store st $M[\vec{n}/\vec{x}, \vec{N}/\vec{\psi}]$) and then resumes as $p[\vec{n}/\vec{x}, \vec{N}/\vec{\psi}]$
- $\|_{i\in I} P_i$: This process is the parallel composition of all the components p_i for i in the indexing set I.
- Free variables:
  - $fv(out\ new\ \vec{x}M.p) = fv(p) \cup fv(M))\backslash\{\vec{x}\}$
  - $fv(in\ pat\ \vec{x}\vec{\psi}M.p) = fv(p) \cup fv(M))\backslash\{\vec{x}, \vec{\psi}\}$
  - $fv(||_{i\in I} P_i) = \bigcup_{i\in I}(P_i)$

An example

- Program of initiator A communicating with B

$$\begin{aligned} Init(A,B) \quad \equiv \quad & out\ new(x)\{x, A\}_{Pub(B)}. \\ & in\{x, y, B\}_{Pub(A)}. \\ & out\{y\}_{Pub(B)} \end{aligned}$$

- Program of responder B

$$\begin{aligned} Resp(B) \quad \equiv \quad & in\{x, Z\}_{Pub(B)}. \\ & out\ new(y)\{x, y, B\}_{Pub(Z)}. \\ & in\{y\}_{Pub(B)} \end{aligned}$$

Some important Characteristics

- *Strong Encryption*: For all messages M, N and keys k, k' if $\{M\}_k = \{N\}_k$ then $M = N$ and $k = k'$
- *Non-Malleability*: An attacker can't, given a cyphertext, produce another cyphertext so that both have related cleartexts.

Transition Semantic

- Configurations: Express the state of execution of the process. Consists of a triple $< p, s, t >$ where p is a closed process term, s is a subset of names N, and t is a subset of closed messages.
- Actions: Input or output actions
- Transitions: The way configurations evolve.

Rules for transitions

Output (Provided names $\vec{n}$ are all distinct and not in s)

$$\langle out\ new(\vec{x})M.p, s, t\rangle \xrightarrow{outnew(\vec{n})M[\vec{n}/\vec{x}]} \langle p[\vec{n}/\vec{x}], s \cup \{\vec{n}\}, t \cup \{M[\vec{n}/\vec{x}]\}\rangle$$

Input (Provided $M[\vec{n}/\vec{x}, \vec{N}/\vec{\psi}]$ in t)

$$\langle in\ pat\vec{x}\vec{\psi}M.p, s, t\rangle \xrightarrow{inM[\vec{n}/\vec{x},\vec{N}/\vec{\psi}]} \langle p[\vec{n}/\vec{x}, \vec{N}/\vec{\psi}], s, t\rangle$$

Parallel Composition (Where P'_i is P'_j for $i = j$, else P_i)

$$\frac{\langle p_j,s,t\rangle \xrightarrow{\alpha} \langle p'_j,s',t'\rangle}{\langle \|_{i\in I} P_i,s,t\rangle \xrightarrow{j:\alpha} \langle \|_{i\in I} P'_i,s',t'\rangle}$$

What can an attacker do?

1. Compose different messages into a single tuple
   $Spy_1 \equiv in\ \psi_1.\ in\ \psi_2.\ out\ \psi_1, \psi_2$
2. Decompose a composed message into more components
   $Spy_2 \equiv in\ \psi_1, \psi_2.\ out\ \psi_1.\ out\ \psi_2$
3. Encrypt any message with the keys that are available
   $Spy_3 \equiv in\ x.\ in\ \psi.\ out\ \{\psi\}_{Pub(x)}$
   $Spy_4 \equiv in\ Key(x, y).\ in\ \psi.\ out\ \{\psi\}_{Key(x,y)}$
4. Decrypt messages with available keys
   $Spy_5 \equiv in\ Priv(x).\ in\ \{\psi\}_{Pub(x)}.\ out\ \psi$
   $Spy_6 \equiv in\ Key(x, y).\ in\ \{\psi\}_{Key(x,y)}.\ out\ \psi$
5. Sign with available keys
   $Spy_7 \equiv Priv(x).\ in\ \psi.\ out\ \{\psi\}_{Priv(x)}$
6. Verify signatures with available keys
   $Spy_8 \equiv in\ x.\ in\ \{\psi\}_{Priv(x)}.\ out\ \psi$
7. Create new random values
   $Spy_9 \equiv out\ new(\vec{n})\vec{n}$

The Needham-Schroeder-Lowe Public Key Protocol

NSL Protocol

$A \rightarrow B : \{m, A\}_{Pub(B)}$ (Initiator)
$B \rightarrow A : \{m, n\}_{Pub(A)}$ (Responder)
$A \rightarrow B : \{n\}_{Pub(B)}$

Properties

- *Secrecy*
- *Authenticity*

is there a possible attack?

Modelling NLS in SPL

- Program of initiator A communicating with B

$$\begin{aligned} Init(A,B) \quad \equiv \quad & out\ new(x)\{x, A\}_{Pub(B)}. \\ & in\{x, y, B\}_{Pub(A)}. \\ & out\{y\}_{Pub(B)} \end{aligned}$$

- Program of responder B

$$\begin{aligned} Resp(B) \quad \equiv \quad & in\{x, Z\}_{Pub(B)}. \\ & out\ new(y)\{x, y, B\}_{Pub(Z)}. \\ & in\{y\}_{Pub(B)} \end{aligned}$$

Pi calculus syntax revisited

$L, M, N ::=$ terms	
n	name
(M, N)	pair
$\mathbf{0}$	zero
$suc(M)$	successor
x	variable
$P, Q, R ::=$ terms	
$\overline{M}\langle N \rangle.P$	output
$M(x).P$	Input
$P \mid Q$	composition
$(\nu n).P$	restriction
$!P$	Replication
$[M \ is \ N]P$	match
$\mathbf{0}$	nil
$let \ (x, y) = M \ in \ P$	Pair splitting
$case \ M \ of \ 0 : P \ suc(x) : Q$	integer case

Spi Calculus Syntax

$L, M, N ::=$ terms

$\{M\}_N$ shared-key encryption

$P, Q, R ::=$ terms

$case\ M\ of\ \{x\}_N\ in\ P$ shared-key decryption

Useful shorthands:

$c(x_1, x_2).P \equiv c(y).let\ (x_1, x_2) = y\ in\ P$

$case\ L\ of\ \{x_1, x_2\}_N\ in\ P \equiv\ case\ L\ of\ \{y\}_N\ in\ let\ (x_1, x_2) = y\ in\ P$

Examples

Protocol description

$A \rightarrow S : c_{AB}$ on C_{AS}
$S \rightarrow B : c_{AB}$ on C_{SB}
$A \rightarrow S : M$ on C_{AB}

Spi calculus model

$$
\begin{aligned}
A(M) &\equiv (\nu C_{AB})\overline{C_{AS}}\langle C_{AB}\rangle.\overline{C_{AB}}\langle M\rangle \\
S &\equiv C_{AS}(x).\overline{C_{SB}}\langle x\rangle \\
B &\equiv C_{SB}(x).x(y).F(y) \\
Inst(M) &\equiv (\nu C_{AS}\ C_{SB})(A(M)|S|B)
\end{aligned}
$$

Examples (2)

Protocol description

$A \rightarrow S : \{K_{AB}\}_{K_{AS}}$
$S \rightarrow B : \{K_{AB}\}_{K_{SB}}$
$A \rightarrow S : \{M\}_{K_{AB}}$

Spi calculus model

$$
\begin{aligned}
A(M) &\equiv (\nu K_{AB})(\overline{C_{AS}}\langle\{K_{AB}\}_{K_{AS}}\rangle.\overline{C_{AB}}\langle\{M\}_{K_{AB}}\rangle) \\
S &\equiv C_{AS}(x).case\ x\ of\ \{y\}_{K_{AS}}\ in\ \overline{C_{SB}}\langle\{y\}_{K_{SB}}\rangle \\
B &\equiv C_{SB}(x).case\ x\ of\ \{y\}_{K_{SB}}\ in \\
&\qquad C_{AB}(z).case\ z\ of\ \{w\}_y\ in\ F(w) \\
Inst(M) &\equiv (\nu K_{AS}\ K_{SB})(A(M)|S|B)
\end{aligned}
$$

Examples (3)

Protocol description

$A \rightarrow S : A, \{B, K_{AB}\}_{K_{AS}}$
$S \rightarrow B : \{A, K_{AB}\}_{K_{SB}}$
$A \rightarrow B : A, \{M\}_{K_{AB}}$

Spi calculus model

$$
\begin{aligned}
Send(i,j,M) &\equiv (\nu K)(\overline{C_S}\langle(\underline{i}, \{\underline{j}, K\}_{iS})\rangle | \overline{C_j}\langle(\underline{i}, \{M\}_K)\rangle) \\
Recv(j) &\equiv C_j(y_{cipher}).case\ y_{cipher}\ of\ \{x_A, x_{key}\}_{K_{S_j}}\ in \\
&\qquad C_j(z_A, z_{cipher}).[x_A\ is\ z_A] \\
&\qquad case\ z_{cipher}\ of \{z_{plain}\}_{x_{key}}\ in \\
&\qquad F(x_A, \underline{j}, z_{plain}) \\
S &\equiv C_S(x_A, x_{cipher}). \\
&\qquad \Pi_{i \in 1..n}[x_A\ is\ \underline{i}]\ case\ x_{cipher}\ of\ \{x_B, x_{key}\}_{K_{iS}}\ in \\
&\qquad \Pi_{j \in 1..n}[x_B\ is\ \underline{j}]\overline{C_j}\langle\{x_A, x_{key}\}_{K_{S_j}}\rangle
\end{aligned}
$$

Security Properties

Spi calculus model

$$
\begin{aligned}
A(M) &\equiv (\nu C_{AB})\overline{C_{AS}}\langle C_{AB}\rangle.\overline{C_{AB}}\langle M\rangle \\
S &\equiv C_{AS}(x).\overline{C_{SB}}\langle x\rangle \\
B &\equiv C_{SB}(x).x(y).F(y) \\
Inst(M) &\equiv (\nu C_{AS}\ C_{SB})(A(M)|S|B)
\end{aligned}
$$

Protocols Specification ("correct")

$$
\begin{aligned}
A(M) &\equiv (\nu C_{AB})\overline{C_{AS}}\langle C_{AB}\rangle.\overline{C_{AB}}\langle M\rangle \\
S &\equiv C_{AS}(x).\overline{C_{SB}}\langle x\rangle \\
B_{spec}(M) &\equiv C_{SB}(x).x(y).F(M) \\
Inst_{spec}(M) &\equiv (\nu C_{AS}\ C_{SB})(A(M)|S|B_{spec}(M))
\end{aligned}
$$

Security Properties (2)

$\simeq$ relation

$A \simeq B$ means the behaviour of A and B are indistinguishable. (R cannot distinguish running in parallel with P from running in parallel with Q)

Authenticity: B always applies F to the message M that A sends:
$Inst(M) \simeq Inst_{spec}(M)$ for any M
Secrecy: Message M cannot be read in transit:
$Inst(M) \simeq Inst(M')$ if $F(M) \simeq F(M')$, for any M, M'