Theory and Tool Support for the Formal Verification of Cryptographic Protocols

Sébastien Briais

École Polytechnique Fédérale de Lausanne

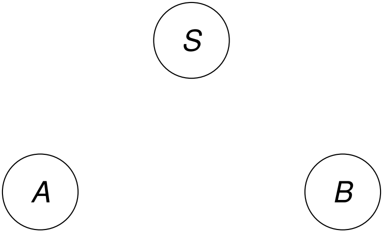
2007, December 17th

Cryptographic protocols are error-prone

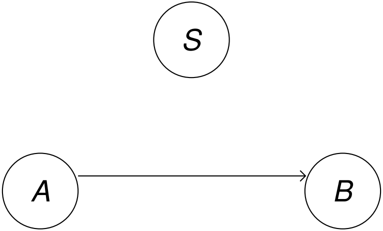
Cryptographic protocols

To secure communication over insecure networks (e.g. Internet).
A communication protocol that uses *cryptography* to achieve security goals.

The Yahalom protocol

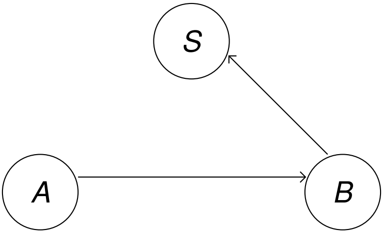


The Yahalom protocol



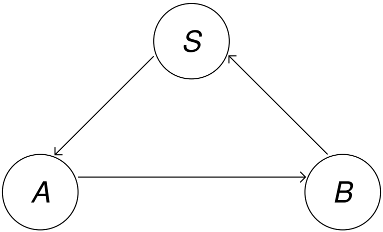
1. $A \rightarrow B : \quad (A \,.\, n_A)$

The Yahalom protocol



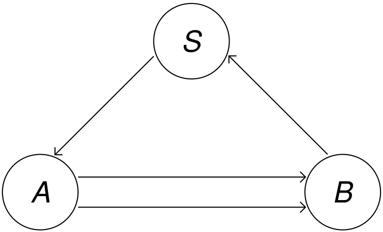
1. $A \longrightarrow B: \quad (A \,.\, n_A)$
2. $B \longrightarrow S: \quad (B \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, (n_A \,.\, n_B)))$

The Yahalom protocol



1. $A \longrightarrow B: \quad (A\,.\,n_A)$
2. $B \longrightarrow S: \quad (B\,.\,\mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A\,.\,(n_A\,.\,n_B)))$
3. $S \longrightarrow A: \quad (\mathrm{Enc}^{\mathrm{s}}_{k_{AS}}((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\,.\,\mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A\,.\,k_{AB}))$

The Yahalom protocol



1. $A \rightarrow B: \quad (A \,.\, n_A)$
2. $B \rightarrow S: \quad (B \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, (n_A \,.\, n_B)))$
3. $S \rightarrow A: \quad (\mathrm{Enc}^{\mathrm{s}}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B)) \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, k_{AB}))$
4. $A \rightarrow B: \quad (\mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, k_{AB}) \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{AB}} n_B)$

Cryptographic protocols are error-prone

Cryptographic protocols

To secure communication over insecure networks (e.g. Internet).
A communication protocol that uses *cryptography* to achieve security goals.

... are error-prone

- Even when assuming perfect cryptographic primitives
- Canonical example: Needham-Schroeder with public key

Cryptographic protocols are error-prone

Cryptographic protocols

To secure communication over insecure networks (e.g. Internet).
A communication protocol that uses *cryptography* to achieve security goals.

... are error-prone

- Even when assuming perfect cryptographic primitives
- Canonical example: Needham-Schroeder with public key

Why is it difficult?

Distributed algorithms that have the obligation to behave robustly in the context of unknown hostile attackers.

The spi calculus approach
Abadi and Gordon

- Cryptographic protocols are described in a precise and concise way.

- Equations to formulate security objectives.
  - secrecy: $P\{^M/_x\} \approx P\{^N/_x\}$ for any M and N
  - authenticity

The spi calculus approach

Abadi and Gordon

- Cryptographic protocols are described in a precise and concise way.

$$
\begin{array}{l}
(\nu k_{AS}, k_{BS}) \\
\quad\quad (\nu n_A)\, \overline{B}\langle (A \,.\, n_A) \rangle . A(x_2).\phi_2 \overline{B}\langle E_2 \rangle . \mathbf{0} \\
\quad \mid (\nu n_B)\, B(x_0).\phi_0 \overline{S}\langle (B \,.\, \mathrm{Enc}^{s}_{k_{BS}}(A \,.\, (\pi_2\,(x_0) \,.\, n_B))) \rangle . B(x_3).\phi_3\, \mathbf{0} \\
\quad \mid (\nu k_{AB})\, S(x_1).\phi_1 \overline{A}\langle E_1 \rangle . \mathbf{0}
\end{array}
$$

- Equations to formulate security objectives.
  - secrecy: $P\{^{M}/_{x}\} \approx P\{^{N}/_{x}\}$ for any M and N
  - authenticity

The spi calculus approach

Abadi and Gordon

- Cryptographic protocols are described in a precise and concise way.

- Equations to formulate security objectives.
  - secrecy: $P\{^M/_x\} \approx P\{^N/_x\}$ for any M and N
  - authenticity

The spi calculus approach

Abadi and Gordon

- Cryptographic protocols are described in a precise and concise way.

- Equations to formulate security objectives.
  - secrecy: $P\{^M/_x\} \approx P\{^N/_x\}$ for any M and N
  - authenticity

The spi calculus approach

Abadi and Gordon

- Cryptographic protocols are described in a precise and concise way.

- Equations to formulate security objectives.
  - secrecy: $P\{^M/_x\} \approx P\{^N/_x\}$ for any M and N
  - authenticity

Testing equivalence

- Usually $\approx$ stands for *testing equivalence*.
- Intuitively, P and Q are testing equivalent *if and only if* they reveal the same information to observers (i.e. attackers).

Testing equivalence

- Usually $\approx$ stands for *testing equivalence*.
- Intuitively, P and Q are testing equivalent *if and only if* they reveal the same information to observers (i.e. attackers).
- Formally, P passes the test (R, β) iff $P \mid R \Downarrow_\beta$, i.e. $P \mid R$ may communicate on channel β.
- $P \simeq Q$ iff they pass the same tests, i.e. for any (R, β),

$$P \mid R \Downarrow_\beta \iff Q \mid R \Downarrow_\beta$$

Testing equivalence

- Usually $\approx$ stands for *testing equivalence*.
- Intuitively, P and Q are testing equivalent *if and only if* they reveal the same information to observers (i.e. attackers).
- Formally, P passes the test (R, β) iff $P \mid R \Downarrow_\beta$, i.e. $P \mid R$ may communicate on channel β.
- $P \simeq Q$ iff they pass the same tests, i.e. for any (R, β),

$$P \mid R \Downarrow_\beta \iff Q \mid R \Downarrow_\beta$$

- Problem: infinite quantification over arbitrary observers R.
- In practise, we define sound approximations that are easier to work with: bisimulations.

Bisimulations

- Behaviour of processes is described with a *Labelled Transitions System*: $P \xrightarrow{\mu} P'$
- Two processes are bisimilar if they can play the same transitions

$$\begin{tikzcd} P \arrow[d, dash] \\ Q \end{tikzcd}$$

Bisimulations

- Behaviour of processes is described with a *Labelled Transitions System*: $P \xrightarrow{\mu} P'$
- Two processes are bisimilar if they can play the same transitions

$$\begin{tikzcd} P \arrow[r, "\mu"] \arrow[d, dash] & P' \\ Q & \end{tikzcd}$$

Bisimulations

- Behaviour of processes is described with a *Labelled Transitions System*: $P \xrightarrow{\mu} P'$
- Two processes are bisimilar if they can play the same transitions

$$\begin{tikzcd} P \arrow[r, "\mu"] \arrow[d, dash] & P' \\ Q \arrow[r, dashed, "\mu"] & Q' \end{tikzcd}$$

Q replies to P

Bisimulations

- Behaviour of processes is described with a *Labelled Transitions System*: $P \xrightarrow{\mu} P'$
- Two processes are bisimilar if they can play the same transitions

$$
\begin{tikzcd}
P \arrow[r, "\mu"] \arrow[d, no head] & P' \arrow[d, dashed, no head] \\
Q \arrow[r, dashed, "\mu"] & Q'
\end{tikzcd}
$$

Q replies to P

Bisimulations

- Behaviour of processes is described with a *Labelled Transitions System*: $P \xrightarrow{\mu} P'$
- Two processes are bisimilar if they can play the same transitions

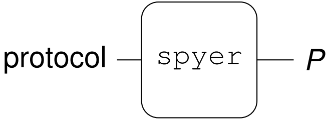
$$
\begin{tikzcd}
P \arrow[r, "\mu"] \arrow[d, dash] & P' \arrow[d, dash, dashed] \\
Q \arrow[r, dashed, "\mu"] & Q'
\end{tikzcd}
$$

Q replies to P

$$
\begin{tikzcd}
P \arrow[r, dashed, "\mu"] \arrow[d, dash] & P' \arrow[d, dash, dashed] \\
Q \arrow[r, "\mu"] & Q'
\end{tikzcd}
$$

P replies to Q

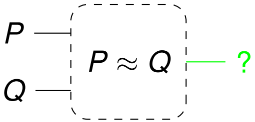
Contributions



From protocol narrations to spi calculus

A formal semantics for protocol narrations.
A rigorous translation to spi calculus.

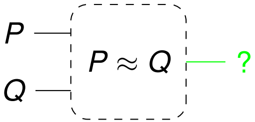
Contributions



Deciding process equivalence

A new notion of bisimulation for the spi calculus.
A symbolic characterisation.

Contributions



Towards a certified tool

Formalization of large parts of the developed theory in Coq.
Dream: Have a correct-by-construction tool.

Contributions

```
1 subgoal

============================
bisimilar P Q
```

Reasoning within Coq

Reason formally about cryptographic protocols in Coq.

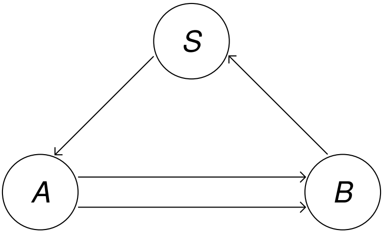
Outline

1. From protocol narrations to spi calculus
2. An open variant of bisimulation for the spi calculus
3. A formalization in Coq

Outline

1. From protocol narrations to spi calculus
2. An open variant of bisimulation for the spi calculus
3. A formalization in Coq

The Yahalom protocol



1. $A \rightarrow B : \quad (A \,.\, n_A)$
2. $B \rightarrow S : \quad (B \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, (n_A \,.\, n_B)))$
3. $S \rightarrow A : \quad (\mathrm{Enc}^{\mathrm{s}}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B)) \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, k_{AB}))$
4. $A \rightarrow B : \quad (\mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, k_{AB}) \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{AB}} n_B)$

The Yahalom protocol in spi-calculus

$$
\begin{array}{l}
(\nu k_{AS}, k_{BS}) \\
\quad (\nu n_A)\, \overline{B}\langle (A \,.\, n_A) \rangle . A(x_2) . \phi_2 \overline{B}\langle (\pi_2\,(x_2) \,.\, \mathrm{Enc}^{s}_{\pi_2\left(\pi_1\left(\mathrm{Dec}^{s}_{k_{AS}} \pi_1 (x_2)\right)\right)} \pi_2 \left(\pi_2 \left(\mathrm{Dec}^{s}_{k_{AS}} \pi_1\,(x_2)\right)\right)) \rangle .\, \mathbf{0} \\
\mid (\nu n_B)\, B(x_0) . \phi_0 \overline{S}\langle (B \,.\, \mathrm{Enc}^{s}_{k_{BS}}(A \,.\, (\pi_2\,(x_0) \,.\, n_B))) \rangle . B(x_3) . \phi_3\, \mathbf{0} \\
\mid (\nu k_{AB}) \\
\quad S(x_1) . \phi_1 \\
\quad \overline{A}\langle (\mathrm{Enc}^{s}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (\pi_1 \left(\pi_2 \left(\mathrm{Dec}^{s}_{k_{BS}} \pi_2\,(x_1)\right)\right) \,.\, \pi_2 \left(\pi_2 \left(\mathrm{Dec}^{s}_{k_{BS}} \pi_2\,(x_1)\right)\right))) \,.\, \mathrm{Enc}^{s}_{k_{BS}}(A \,.\, k_{AB})) \rangle .\, \mathbf{0}
\end{array}
$$

The Yahalom protocol in spi-calculus

$$
\begin{array}{l}
(\nu k_{AS}, k_{BS}) \\
\quad (\nu n_A)\, \overline{B}\langle (A \,.\, n_A)\rangle.A(x_2).\color{red}{\phi_2}\color{black}\overline{B}\langle(\pi_2\,(x_2)\,.\,\mathrm{Enc}^{s}_{\pi_2\left(\pi_1\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1(x_2)\right)\right)}\pi_2\left(\pi_2\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x_2)\right)\right))\rangle.\,\mathbf{0} \\
\mid (\nu n_B)\, B(x_0).\color{red}{\phi_0}\color{black}\overline{S}\langle(B\,.\,\mathrm{Enc}^{s}_{k_{BS}}(A\,.\,(\pi_2\,(x_0)\,.\,n_B)))\rangle.B(x_3).\color{red}{\phi_3}\color{black}\,\mathbf{0} \\
\mid (\nu k_{AB}) \\
\quad S(x_1).\color{red}{\phi_1}\color{black} \\
\quad \overline{A}\langle(\mathrm{Enc}^{s}_{k_{AS}}((B\,.\,k_{AB})\,.\,(\pi_1\left(\pi_2\left(\mathrm{Dec}^{s}_{k_{BS}}\pi_2\,(x_1)\right)\right)\,.\,\pi_2\left(\pi_2\left(\mathrm{Dec}^{s}_{k_{BS}}\pi_2\,(x_1)\right)\right)))\,.\,\mathrm{Enc}^{s}_{k_{BS}}(A\,.\,k_{AB}))\rangle.\,\mathbf{0}
\end{array}
$$

The Yahalom protocol in spi-calculus

$$
\begin{array}{l}
(\nu k_{AS}, k_{BS}) \\
\quad (\nu n_A)\, \overline{B}\langle (A \,.\, n_A) \rangle . A(x_2) . \phi_2 \overline{B}\langle (\pi_2\,(x_2) \,.\, \mathrm{Enc}^{s}_{\pi_2\left(\pi_1\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x_2)\right)\right)} \pi_2\left(\pi_2\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x_2)\right)\right)) \rangle . \mathbf{0} \\
\ \mid (\nu n_B)\, B(x_0) . \phi_0 \overline{S}\langle (B \,.\, \mathrm{Enc}^{s}_{k_{BS}}(A \,.\, (\pi_2\,(x_0) \,.\, n_B))) \rangle . B(x_3) . \phi_3\, \mathbf{0} \\
\ \mid (\nu k_{AB}) \\
\qquad S(x_1) . \phi_1 \\
\qquad \overline{A}\langle (\mathrm{Enc}^{s}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (\pi_1\left(\pi_2\left(\mathrm{Dec}^{s}_{k_{BS}}\pi_2\,(x_1)\right)\right) \,.\, \pi_2\left(\pi_2\left(\mathrm{Dec}^{s}_{k_{BS}}\pi_2\,(x_1)\right)\right))) \,.\, \mathrm{Enc}^{s}_{k_{BS}}(A \,.\, k_{AB})) \rangle . \mathbf{0}
\end{array}
$$

$$
\begin{array}{lcl}
\phi_0 & = & [A = \pi_1\,(x_0)] \\
\phi_1 & = & [\pi_1\left(\pi_2\left(\mathrm{Dec}^{s}_{k_{BS}}\pi_2\,(x_1)\right)\right) : \boldsymbol{M}] \wedge [B = \pi_1\,(x_1)] \wedge [A = \pi_1\left(\mathrm{Dec}^{s}_{k_{BS}}\pi_2\,(x_1)\right)] \\
\phi_2 & = & [B = \pi_1\left(\pi_1\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x_2)\right)\right)] \wedge [n_A = \pi_1\left(\pi_2\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x_2)\right)\right)] \\
\phi_3 & = & [A = \pi_1\left(\mathrm{Dec}^{s}_{k_{BS}}\pi_1\,(x_3)\right)] \wedge [n_B = \mathrm{Dec}^{s}_{\pi_2\left(\mathrm{Dec}^{s}_{k_{BS}}\pi_1(x_3)\right)} \pi_2\,(x_3)]
\end{array}
$$

State explicitly the assumptions

A protocol narration does not explicitly state the initial knowledge and what is to be generated freshly during a protocol run.

State explicitly the assumptions

A protocol narration does not explicitly state the initial knowledge and what is to be generated freshly during a protocol run.

$$
\begin{array}{l}
A, S \textbf{ share } k_{AS} \\
B, S \textbf{ share } k_{BS} \\
A \textbf{ generates } n_A \;;\; B \textbf{ generates } n_B \;;\; S \textbf{ generates } k_{AB} \;; \\
A \rightsquigarrow B : (A \,.\, n_A) \;; \\
B \rightsquigarrow S : (B \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, (n_A \,.\, n_B))) \;; \\
S \rightsquigarrow A : (\mathrm{Enc}^{\mathrm{s}}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B)) \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, k_{AB})) \;; \\
A \rightsquigarrow B : (\mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, k_{AB}) \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{AB}} n_B)
\end{array}
$$

State explicitly the assumptions

A protocol narration does not explicitly state the initial knowledge and what is to be generated freshly during a protocol run.

Principals act concurrently

A protocol narration describes an idealised sequential trace of execution whereas the principals act concurrently.

$A \rightarrow B : M$ actually means

(i) A *asynchronously* sends M towards B,

(ii) B receives some message

State explicitly the assumptions

A protocol narration does not explicitly state the initial knowledge and what is to be generated freshly during a protocol run.

Principals act concurrently

A protocol narration describes an idealised sequential trace of execution whereas the principals act concurrently.

$A \rightarrow B : M$ actually means

(i) A *asynchronously* sends M towards B,

(ii) B receives some message (intended to be M)

Principals perform on-reception checks

(iii) B checks that the message it just received has the expected properties.

State explicitly the assumptions

A protocol narration does not explicitly state the initial knowledge and what is to be generated freshly during a protocol run.

A, S **share** k_{AS}
B, S **share** k_{BS}
A **generates** n_A ; B **generates** n_B ; S **generates** k_{AB} ;
$A \rightsquigarrow B : (A \,.\, n_A)$;
$B \rightsquigarrow S : (B \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, (n_A \,.\, n_B)))$;
$S \rightsquigarrow A : (\mathrm{Enc}^{\mathrm{s}}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B)) \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, k_{AB}))$;
$A \rightsquigarrow B : (\mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, k_{AB}) \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{AB}} n_B)$

Generating the checks

Current knowledge

$$\{A, B, S, k_{AS}, n_A\}$$

expected	
$(\mathrm{Enc}^{\mathrm{s}}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B)) \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, k_{AB}))$	

Generating the checks

Current knowledge

$$\{A, B, S, k_{AS}, n_A\}$$

expected	
$(\mathrm{Enc}^{s}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B)) \,.\, \mathrm{Enc}^{s}_{k_{BS}}(A \,.\, k_{AB}))$	

Generating the checks

Current knowledge

$$\{A, B, S, k_{AS}, n_A\}$$

expected	*actual*
$(\mathrm{Enc}^{s}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B)) \,.\, \mathrm{Enc}^{s}_{k_{BS}}(A \,.\, k_{AB}))$	x

Generating the checks

Current knowledge

$$\{A, B, S, k_{AS}, n_A\}$$

expected	*actual*
$(\mathrm{Enc}^{s}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B)) \,.\, \mathrm{Enc}^{s}_{k_{BS}}(A \,.\, k_{AB}))$	x
$\mathrm{Enc}^{s}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))$	$\pi_1\,(x)$
$\mathrm{Enc}^{s}_{k_{BS}}(A \,.\, k_{AB})$	$\pi_2\,(x)$

Generating the checks

Current knowledge

$$\{A, B, S, k_{AS}, n_A\}$$

expected	*actual*
$(\mathrm{Enc}^{s}_{k_{AS}}((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\,.\,\mathrm{Enc}^{s}_{k_{BS}}(A\,.\,k_{AB}))$	x
$\mathrm{Enc}^{s}_{k_{AS}}((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\pi_1\,(x)$
$\mathrm{Enc}^{s}_{k_{BS}}(A\,.\,k_{AB})$	$\pi_2\,(x)$
$((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x)$

Generating the checks

Current knowledge

$$\{A, B, S, k_{AS}, n_A\}$$

expected	*actual*
$(\mathrm{Enc}^{s}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B)) \,.\, \mathrm{Enc}^{s}_{k_{BS}}(A \,.\, k_{AB}))$	x
$\mathrm{Enc}^{s}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))$	$\pi_1\,(x)$
$\mathrm{Enc}^{s}_{k_{BS}}(A \,.\, k_{AB})$	$\pi_2\,(x)$
$((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))$	$\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x)$
$(B \,.\, k_{AB})$	$\pi_1\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x)\right)$
$(n_A \,.\, n_B)$	$\pi_2\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x)\right)$
B	$\pi_1\left(\pi_1\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x)\right)\right)$
k_{AB}	$\pi_2\left(\pi_1\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x)\right)\right)$
n_A	$\pi_1\left(\pi_2\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x)\right)\right)$
n_B	$\pi_2\left(\pi_2\left(\mathrm{Dec}^{s}_{k_{AS}}\pi_1\,(x)\right)\right)$

Generating the checks

Current knowledge

$$\{A, B, S, k_{AS}, n_A\}$$

expected	*actual*
$(\mathrm{Enc}^{\mathrm{s}}_{k_{AS}}((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\,.\,\mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A\,.\,k_{AB}))$	x
$\mathrm{Enc}^{\mathrm{s}}_{k_{AS}}((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\pi_1\,(x)$
$\mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A\,.\,k_{AB})$	$\pi_2\,(x)$
$((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1\,(x)$
$(B\,.\,k_{AB})$	$\pi_1\left(\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1\,(x)\right)$
$(n_A\,.\,n_B)$	$\pi_2\left(\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1\,(x)\right)$
B	$\pi_1\left(\pi_1\left(\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1\,(x)\right)\right)$
k_{AB}	$\pi_2\left(\pi_1\left(\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1\,(x)\right)\right)$
n_A	$\pi_1\left(\pi_2\left(\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1\,(x)\right)\right)$
n_B	$\pi_2\left(\pi_2\left(\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1\,(x)\right)\right)$

The Yahalom protocol in spi-calculus

$$
\begin{array}{l}
(\nu k_{AS}, k_{BS}) \\
\quad (\nu n_A)\, \overline{B}\langle (A \,.\, n_A)\rangle . A(x_2).\phi_2 \overline{B}\langle (\pi_2\,(x_2) \,.\, \mathrm{Enc}^{\mathrm{s}}_{\pi_2\left(\pi_1\left(\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1(x_2)\right)\right)} \pi_2\left(\pi_2\left(\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1\,(x_2)\right)\right))\rangle . \mathbf{0} \\
\;|\, (\nu n_B)\, B(x_0).\phi_0 \overline{S}\langle (B \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, (\pi_2\,(x_0) \,.\, n_B)))\rangle . B(x_3).\phi_3\, \mathbf{0} \\
\;|\, (\nu k_{AB}) \\
\qquad S(x_1).\phi_1 \\
\qquad \overline{A}\langle (\mathrm{Enc}^{\mathrm{s}}_{k_{AS}}((B \,.\, k_{AB}) \,.\, (\pi_1\left(\pi_2\left(\mathrm{Dec}^{\mathrm{s}}_{k_{BS}}\pi_2\,(x_1)\right)\right) \,.\, \pi_2\left(\pi_2\left(\mathrm{Dec}^{\mathrm{s}}_{k_{BS}}\pi_2\,(x_1)\right)\right))) \,.\, \mathrm{Enc}^{\mathrm{s}}_{k_{BS}}(A \,.\, k_{AB}))\rangle . \mathbf{0}
\end{array}
$$

$$
\begin{array}{lcl}
\phi_0 & = & [A = \pi_1\,(x_0)] \\
\phi_1 & = & [\pi_1\left(\pi_2\left(\mathrm{Dec}^{\mathrm{s}}_{k_{BS}}\pi_2\,(x_1)\right)\right) : \boldsymbol{M}] \wedge [B = \pi_1\,(x_1)] \wedge [A = \pi_1\left(\mathrm{Dec}^{\mathrm{s}}_{k_{BS}}\pi_2\,(x_1)\right)] \\
\phi_2 & = & [B = \pi_1\left(\pi_1\left(\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1\,(x_2)\right)\right)] \wedge [n_A = \pi_1\left(\pi_2\left(\mathrm{Dec}^{\mathrm{s}}_{k_{AS}}\pi_1\,(x_2)\right)\right)] \\
\phi_3 & = & [A = \pi_1\left(\mathrm{Dec}^{\mathrm{s}}_{k_{BS}}\pi_1\,(x_3)\right)] \wedge [n_B = \mathrm{Dec}^{\mathrm{s}}_{\pi_2\left(\mathrm{Dec}^{\mathrm{s}}_{k_{BS}}\pi_1(x_3)\right)}\pi_2\,(x_3)]
\end{array}
$$

Outline

Situation in the pi calculus

- Spi calculus is an extension of the pi calculus that incorporates cryptographic primitives .

$$\begin{array}{rcl} P, Q & ::= & \mathbf{0} \mid a(x).P \mid \overline{a}\langle u\rangle.P \\ & \mid & [a=b]P \mid (\nu x)\,P \\ & \mid & P \mid Q \mid P+Q \mid\, !\,P \end{array}$$

- Open bisimulation (Sangiorgi) is at the basis of several tools that automatically checks equivalence of pi terms
  e.g. the Mobility Workbench (Victor)
- Can we extend this notion to the spi calculus?

Situation in the pi calculus

- Spi calculus is an extension of the pi calculus that incorporates cryptographic primitives more.

$$
\begin{array}{rcl}
P, Q & ::= & \mathbf{0} \mid E(x).P \mid \overline{E}\langle F\rangle.P \\
 & \mid & \phi P \mid (\nu x)\, P \\
 & \mid & P \,|\, Q \mid P + Q \mid\, !\, P
\end{array}
$$

$$
\begin{array}{rcl}
M, N & ::= & x \mid (M\,.\,N) \mid \mathrm{Enc}^{\mathrm{s}}_{N} M \\
E, F & ::= & \ldots \mid \pi_1\,(E) \mid \pi_2\,(E) \mid \mathrm{Dec}^{\mathrm{s}}_{F} E \\
\phi & ::= & [\,E = F\,] \mid [\,E : \mathcal{N}\,]
\end{array}
$$

- Open bisimulation (Sangiorgi) is at the basis of several tools that automatically checks equivalence of pi terms
  e.g. the Mobility Workbench (Victor)
- Can we extend this notion to the spi calculus?

Situation in the pi calculus

- Spi calculus is an extension of the pi calculus that incorporates cryptographic primitives .

$$\begin{array}{rcl} P, Q & ::= & \mathbf{0} \mid a(x).P \mid \overline{a}\langle u\rangle.P \\ & \mid & [a{=}b]P \mid (\nu x)\, P \\ & \mid & P \mid Q \mid P + Q \mid \,!\,P \end{array}$$

- Open bisimulation (Sangiorgi) is at the basis of several tools that automatically checks equivalence of pi terms
  e.g. the Mobility Workbench (Victor)
- Can we extend this notion to the spi calculus?

Situation in the pi calculus

- Spi calculus is an extension of the pi calculus that incorporates cryptographic primitives .

$$\begin{array}{rcl} P, Q & ::= & \mathbf{0} \mid a(x).P \mid \overline{a}\langle u \rangle.P \\ & \mid & [a=b]P \mid (\nu x)\, P \\ & \mid & P \,|\, Q \mid P + Q \mid \,!\, P \end{array}$$

- Open bisimulation (Sangiorgi) is at the basis of several tools that automatically checks equivalence of pi terms
  e.g. the Mobility Workbench (Victor)
- Can we extend this notion to the spi calculus?

Bisimulations in the pi calculus

The main differences is the way they handle substitutions

$$\begin{tikzcd} P \arrow[r, "\mu"] \arrow[d, dash] & P' \arrow[d, dash, dashed] \\ Q \arrow[r, dashed, "\mu"] & Q' \end{tikzcd}$$

Bisimulations in the pi calculus

The main differences is the way they handle substitutions

$$
\begin{tikzcd}
P \arrow[r, "a(x)"] \arrow[d, dash] & P' \arrow[d, dash, dashed] \\
Q \arrow[r, dashed, "a(x)"] & Q'
\end{tikzcd}
$$

Ground

Bisimulations in the pi calculus

The main differences is the way they handle substitutions

$$
\begin{tikzcd}
P \arrow[r, "a(x)"] \arrow[d, no head] & P' & P'\{^u/_x\} \arrow[d, dash, dashed] \\
Q \arrow[r, dashed, "a(x)"] & Q' & Q'\{^u/_x\}
\end{tikzcd}
$$

for any name u

Early/Late

Bisimulations in the pi calculus

The main differences is the way they handle substitutions

$$\begin{tikzcd} P \arrow[d, no head] & P\sigma \arrow[r, "\mu"] & P' \arrow[d, dashed, no head] \\ Q & Q\sigma \arrow[r, dashed, "\mu"] & Q' \end{tikzcd}$$

for any σ

Open

Bisimulations in the pi calculus

The main differences is the way they handle substitutions

$$
\begin{tikzcd}
P \arrow[d, dash, "D"] & P\sigma \arrow[r, "\mu"] & P' \arrow[d, dash, dashed, "D'"] \\
Q & Q\sigma \arrow[r, dashed, "\mu"] & Q'
\end{tikzcd}
$$

for any σ that respects D

Open

Distinctions D to prevent from fusing previously extruded names with free names.

Bisimulations in the pi calculus

The main differences is the way they handle substitutions

$$\begin{tikzcd} P \arrow[d, dash, "D"] & P\sigma \arrow[r, "\mu"] & P' \arrow[d, dashed, dash, "D'"] \\ Q & Q\sigma \arrow[r, dashed, "\mu"] & Q' \end{tikzcd}$$

for any σ that respects D

Open

The quantification over all substitutions gives a *call-by-need* flavor to the bisimulation. This idea is exploited by the tools which needs to inspect only *most general unifiers*.

$$\text{O-COMM-L}\ \frac{P \xrightarrow[M]{a(x)} P' \qquad Q \xrightarrow[N]{\overline{b}\,u} Q'}{P \mid Q \xrightarrow[MN[a=b]]{\tau} P'\{^{u}/_{x}\} \mid Q'}$$

Bisimulations in spi calculus

- Consider $P(M) := (\nu k)\, \overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_{k} M \rangle . \mathbf{0}$.
  We want $P(M) \approx P(N)$ since k is private and never revealed.

Bisimulations in spi calculus

- Consider $P(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_{k} M\rangle.\,\mathbf{0}$.
  We want $P(M) \approx P(N)$ since k is private and never revealed.

$$\begin{array}{ccc} P(M) & \text{———} & P(N) \\ \Big\downarrow{\scriptstyle \overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_{k} M} & & \\ \mathbf{0} & & \mathbf{0} \end{array}$$

Bisimulations in spi calculus

- Consider $P(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_k M\rangle.\,\mathbf{0}$.
  We want $P(M) \approx P(N)$ since k is private and never revealed.

$$
\begin{tikzcd}
P(M) \arrow[r, dash] \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_k M"'] & P(N) \arrow[d, dashed, "\overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_k N"] \\
\mathbf{0} & \mathbf{0}
\end{tikzcd}
$$

Bisimulations in spi calculus

- Consider $P(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_{k} M\rangle.\,\mathbf{0}$.
  We want $P(M) \approx P(N)$ since k is private and never revealed.

$$\begin{tikzcd}
P(M) \arrow[r, dash] \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_{k} M"'] & P(N) \arrow[d, dashed, "\overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_{k} N"] \\
\mathbf{0} & \mathbf{0}
\end{tikzcd}$$

- Bisimulations of the pi calculus are too fine-grained.

Bisimulations in spi calculus

- Consider $P(M) := (\nu k)\,\overline{c}\langle \mathsf{Enc}^{\mathrm{s}}_{k} M\rangle.\,\mathbf{0}$.
  We want $P(M) \approx P(N)$ since k is private and never revealed.

$$\begin{tikzcd}
P(M) \arrow[r, dash] \arrow[d, "\overline{c}\,(\nu k)\mathsf{Enc}^{\mathrm{s}}_{k} M"'] & P(N) \arrow[d, dashed, "\overline{c}\,(\nu k)\mathsf{Enc}^{\mathrm{s}}_{k} N"] \\
\mathbf{0} & \mathbf{0}
\end{tikzcd}$$

- Bisimulations of the pi calculus are too fine-grained.
- Some pair of messages should be indistinguishable.
- Bisimulations are extended with a data structure that represents the observer knowledge. This has led to various notions of *environment-sensitive* bisimulations (framed, alley, hedged, ...)

Hedged bisimulation def.

Borgström and Nestmann.

Hedge

A hedge $h \in \boldsymbol{H}$ is a finite set of pairs of messages.
Intuitively $(M, N) \in h$ means that M and N are indistinguishable.

A hedged bisimulation relates triples (h, P, Q).

Hedged bisimulation def.

Borgström and Nestmann.

$$P(M) := (\nu k)\, \overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_{k} M \rangle . \mathbf{0}$$

$$P(M) \qquad (c, c) \qquad P(N)$$

Hedged bisimulation def.

Borgström and Nestmann.

$$P(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{s}_{k} M\rangle.\,\mathbf{0}$$

$$\begin{array}{ccc} P(M) & (c, c) & P(N) \\ \big\downarrow{\scriptstyle \overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} M} & & \\ \mathbf{0} & & \end{array}$$

Hedged bisimulation def.

Borgström and Nestmann.

$$P(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{s}_{k} M\rangle.\,\mathbf{0}$$

$$\begin{tikzcd}
P(M) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} M"'] & (c, c) & P(N) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} N"] \\
\mathbf{0} & & \mathbf{0}
\end{tikzcd}$$

Hedged bisimulation def.

Borgström and Nestmann.

$$P(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_{k} M\rangle.\,\mathbf{0}$$

$$\begin{tikzcd}
P(M) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_{k} M"'] & (c, c) & P(N) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_{k} N"] \\
\mathbf{0} & (\mathrm{Enc}^{\mathrm{s}}_{k} M, \mathrm{Enc}^{\mathrm{s}}_{k} N) & \mathbf{0}
\end{tikzcd}$$

Hedged bisimulation (def.)

Borgström and Nestmann.

$$Q(M,N) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{s}_{k} M\rangle.\overline{c}\langle \mathrm{Enc}^{s}_{k} N\rangle.\,\mathbf{0}$$

$$\begin{array}{ccc}
Q(M,M) & (c,c) & Q(M,N) \\
\Big\downarrow{\scriptstyle \overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} M} & & \Big\downarrow{\scriptstyle \overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} M} \\
 & (\mathrm{Enc}^{s}_{k} M, \mathrm{Enc}^{s}_{k} M) & \\
\Big\downarrow{\scriptstyle \overline{c}\,\mathrm{Enc}^{s}_{k} M} & & \Big\downarrow{\scriptstyle \overline{c}\,\mathrm{Enc}^{s}_{k} N} \\
\mathbf{0} & (\mathrm{Enc}^{s}_{k} M, \mathrm{Enc}^{s}_{k} N) & \mathbf{0}
\end{array}$$

Hedged bisimulation def.

Borgström and Nestmann.

$$Q(M,N) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_k M\rangle.\overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_k N\rangle.\,\mathbf{0}$$

$$\begin{tikzcd}
Q(M,M) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_k M"'] & (c,c) & Q(M,N) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_k M"] \\
{} \arrow[d, "\overline{c}\,\mathrm{Enc}^{\mathrm{s}}_k M"'] & ({\color{blue}\mathrm{Enc}^{\mathrm{s}}_k M}, {\color{red}\mathrm{Enc}^{\mathrm{s}}_k M}) & {} \arrow[d, "\overline{c}\,\mathrm{Enc}^{\mathrm{s}}_k N"] \\
\mathbf{0} & ({\color{blue}\mathrm{Enc}^{\mathrm{s}}_k M}, {\color{green}\mathrm{Enc}^{\mathrm{s}}_k N}) & \mathbf{0}
\end{tikzcd}$$

Hedged bisimulation def.

Borgström and Nestmann.

$$Q(M,N) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_k M\rangle.\overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_k N\rangle.\,\mathbf{0}$$

$$\begin{array}{ccc}
Q(M,M) & (c,c) & Q(M,N) \\
\downarrow{\scriptstyle \overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_k M} & & \downarrow{\scriptstyle \overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_k M} \\
 & ({\color{blue}\mathrm{Enc}^{\mathrm{s}}_k M}, {\color{red}\mathrm{Enc}^{\mathrm{s}}_k M}) & \\
\downarrow{\scriptstyle \overline{c}\,\mathrm{Enc}^{\mathrm{s}}_k M} & & \downarrow{\scriptstyle \overline{c}\,\mathrm{Enc}^{\mathrm{s}}_k N} \\
\mathbf{0} & ({\color{blue}\mathrm{Enc}^{\mathrm{s}}_k M}, {\color{green}\mathrm{Enc}^{\mathrm{s}}_k N}) & \mathbf{0}
\end{array}$$

The hedge must be consistent def..

$$O := c(x).c(y).[x = y]\overline{c}\langle \mathrm{fail}\rangle.\,\mathbf{0}$$

Hedged bisimulation def.

Borgström and Nestmann.

$$R(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{s}_{k} M\rangle.\overline{c}\langle k\rangle.\,\mathbf{0}$$

$$\begin{array}{ccc}
R(M) & (c, c) & R(N) \\
\big\downarrow \overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} M & & \big\downarrow \overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} N \\
 & (\mathrm{Enc}^{s}_{k} M, \mathrm{Enc}^{s}_{k} N) & \\
\big\downarrow \overline{c}\,k & & \big\downarrow \overline{c}\,k \\
\mathbf{0} & (k, k) & \mathbf{0}
\end{array}$$

Hedged bisimulation def.

Borgström and Nestmann.

$$R(M) := (\nu k)\, \overline{c}\langle \mathrm{Enc}^{\mathrm{s}}_{k} M\rangle . \overline{c}\langle k\rangle .\, \mathbf{0}$$

$$\begin{tikzcd}
R(M) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_{k} M"'] & (c, c) & R(N) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{\mathrm{s}}_{k} N"] \\
{} \arrow[d, "\overline{c}\, k"'] & (\mathrm{Enc}^{\mathrm{s}}_{k} M, \mathrm{Enc}^{\mathrm{s}}_{k} N) & {} \arrow[d, "\overline{c}\, k"] \\
\mathbf{0} & \begin{array}{c}(k, k)\\ (M, N)\end{array} & \mathbf{0}
\end{tikzcd}$$

The hedge is analysed after outputs def..

Hedged bisimulation def.

Borgström and Nestmann.

$$S_1(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{s}_{k} M\rangle.c(x).[x=k]\overline{c}\langle k\rangle.\mathbf{0}$$
$$S_2(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{s}_{k} M\rangle.c(x).\mathbf{0}$$

$$\begin{tikzcd}
S_1(M) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} M"'] & (c,c) & S_2(M) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} M"] \\
{} \arrow[d, "c(x)"'] & (\mathrm{Enc}^{s}_{k} M, \mathrm{Enc}^{s}_{k} M) & {} \arrow[d, "c(x)"] \\
{[x=k]\mathbf{0}} & & \mathbf{0}
\end{tikzcd}$$

Hedged bisimulation def.

Borgström and Nestmann.

$$S_1(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{s}_{k} M\rangle.c(x).[x=k]\overline{c}\langle k\rangle.\,\mathbf{0}$$
$$S_2(M) := (\nu k)\,\overline{c}\langle \mathrm{Enc}^{s}_{k} M\rangle.c(x).\,\mathbf{0}$$

$$\begin{tikzcd}
S_1(M) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} M"'] & (c, c) & S_2(M) \arrow[d, "\overline{c}\,(\nu k)\mathrm{Enc}^{s}_{k} M"] \\
\cdot \arrow[d, "c(x)"'] & (\mathrm{Enc}^{s}_{k} M, \mathrm{Enc}^{s}_{k} M) & \cdot \arrow[d, "c(x)"] \\
{[x=k]\,\mathbf{0}} & & \mathbf{0}
\end{tikzcd}$$

The possible pairs of input messages are constructed using the current knowledge and possibly some *fresh names* def.

Open hedged bisimulation def.

Delaying instantiation of input variables

- Which names are subjects to substitutions?
  - Input variables.
- What are the possible objects of substitutions?
  - Messages constructed using the knowledge available at the moment of the input and possibly some fresh names.
- A variable dynamically typed as a name is not replaced by a compound message LTS.

Open hedged bisimulation def.

Delaying instantiation of input variables

- Which names are subjects to substitutions?
  - Input variables.
- What are the possible objects of substitutions?
  - Messages constructed using the knowledge available at the moment of the input and possibly some fresh names.
- A variable dynamically typed as a name is not replaced by a compound message LTS.

Open hedged bisimulation def.

Delaying instantiation of input variables

- Which names are subjects to substitutions?
  - Input variables.
- What are the possible objects of substitutions?
  - Messages constructed using the knowledge available at the moment of the input and possibly some fresh names.
- A variable dynamically typed as a name is not replaced by a compound message LTS.

Open hedged bisimulation def.

Delaying instantiation of input variables

- Which names are subjects to substitutions?
  - Input variables.
- What are the possible objects of substitutions?
  - Messages constructed using the knowledge available at the moment of the input and possibly some fresh names.
- A variable dynamically typed as a name is not replaced by a compound message LTS.

Hence the form of S-environments $se = (h, v, \prec, (\gamma_l, \gamma_r))$.

Open hedged bisimulation def.

Delaying instantiation of input variables

- Which names are subjects to substitutions?
  - Input variables.
- What are the possible objects of substitutions?
  - Messages constructed using the knowledge available at the moment of the input and possibly some fresh names.
- A variable dynamically typed as a name is not replaced by a compound message LTS.

Hence the form of S-environments $se = (h, v, \prec, (\gamma_l, \gamma_r))$.

consistency of S-environments

A S-environment is consistent if for any instantiation of input variables, the resulting hedge is consistent.

Symbolic characterisation

- Relies on the definition of a *symbolic LTS* def.
- The idea is to record —without checking— the conditions needed to enable a transition.

$$P \overset{\mu}{\underset{\Phi}{\longmapsto}} P'$$

- The symbolic LTS helps to characterise precisely the set of substitutions σ such that $P\sigma \xrightarrow{\mu} P'$.
- Given a symbolic transition $P \overset{\mu}{\underset{\Phi}{\longmapsto}} P'$, there is a finite complete set of solutions of Φ.

Outline

1. From protocol narrations to spi calculus
2. An open variant of bisimulation for the spi calculus
3. A formalization in Coq

Representation of binders

de Bruijn indices

Representation of $a(x).[\mathrm{Dec}^{s}_{k}\, x : \boldsymbol{M}](\nu l)\, \overline{b}\langle l\rangle.\,\mathbf{0}$

$$z\ y\ x\ \cdots\ l\ k\ j\ \cdots\ c\ b\ a$$

$$0\ \lambda.\ [\mathrm{Dec}^{s}_{11}\ 0 : \boldsymbol{M}]\ \nu\ \overline{3}\langle 0\rangle.\mathbf{0}$$

Representation of binders

de Bruijn indices

Several operations have to be defined to handle de Bruijn indices. more

Example: $\mathrm{lift}_d(k, t)$ makes room for k new binders in t

$$z\ y\ x\ \cdots\ c\ b\ a$$

$$\lambda.\overline{1}\langle 0\rangle.\mathbf{0}$$

$$z\ y\ x\ \cdots\ c\ b\ a$$

$$0\ \lambda.\overline{0}\langle 24\rangle.\mathbf{0}$$

$$z\ y\ x\ \cdots\ c\ b\ a$$

$$\lambda.(\overline{1}\langle 0\rangle.\mathbf{0} \mid 1\ \lambda.\overline{0}\langle 25\rangle.\mathbf{0})$$

Representation of binders

de Bruijn indices

In practise:

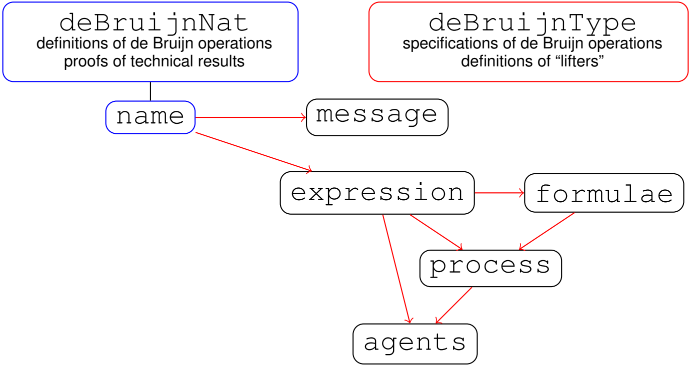
- 5 operations on indices, 6 types (names, messages, ...)
- about 60 useful facts relate these operations
- not scalable and tedious to define and prove several times the same operations/facts

Representation of binders

de Bruijn indices

Instead

1. define on names
2. lift to other types



Abstracting the labelled transition system

- There are several LTS to define.
- Some properties are shared
(e.g. structural congruence preserves the transitions)
- These LTS all follow the same pattern.
- Instead of defining each LTS separately, we make a functor and thus defer the definition of the semantics to the definitions of the semantics of actions.

Abstracting the labelled transition system

We rely on a set of actions $\mathcal{A}$ and several functions to manipulate them:

- mkSil : $\mathcal{A}$ (silent)
- mkInp : $\boldsymbol{E} \rightarrow \mathcal{A} \cup \{\bot\}$ (input)
- mkOutp : $\boldsymbol{E} \times \boldsymbol{E} \rightarrow (\mathcal{A} \times \boldsymbol{E}) \cup \{\bot\}$ (output)
- mkRes : $\mathcal{A} \rightarrow \mathcal{A} \cup \{\bot\}$ (restriction)
- mkIf : $\boldsymbol{F} \times \mathcal{A} \rightarrow \mathcal{A} \cup \{\bot\}$ (guard)
- mkInt : $\mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A} \cup \{\bot\}$ (interact)

Abstracting the labelled transition system

We then define a parametrised LTS.

$$\textsc{Input}\ \frac{\mathsf{mkInp}(E) = \alpha \in \mathcal{A}}{E\lambda.P \xrightarrow{\alpha} \lambda.P} \qquad \textsc{Output}\ \frac{\mathsf{mkOutp}(E, F) = (\alpha, M) \in \mathcal{A} \times \boldsymbol{E}}{\overline{E}\langle F\rangle.P \xrightarrow{\alpha} \langle M\rangle P}$$

$$\textsc{Close-L}\ \frac{P \xrightarrow{\alpha} F \qquad Q \xrightarrow{\beta} C \qquad \mathsf{mkInt}(\alpha, \beta) = \gamma \in \mathcal{A}}{P \mid Q \xrightarrow{\gamma} F \bullet C}$$

Overview of the formalization

- Monadic pi calculus
- Pi LTS
- Spi calculus
- Hedges and their properties
- Spi LTS: standard, with type constraints, symbolic and their properties
- Crash test: result about structural congruence
- Late hedged bisimulation, correctness of up-to techniques
- Small examples of bisimulations

Conclusion

- A formal semantics for protocol narrations.
  - A rigorous translation into spi calculus.
- An open style definition of bisimulation for the spi calculus.
  - It is a sound proof technique.
  - It is an extension of open bisimulation of the pi calculus.
  - Its projection down to the pi calculus has enabled us to better understand the original notion of open bisimulation.
  - A symbolic characterisation as a promising first step towards mechanisation.
- A formalization in a proof assistant.
  - Very useful while elaborating the theory.
  - Already a framework to reason formally about cryptographic protocols in Coq.

Future work

- Study furthermore open hedged bisimilarity.
  - Congruence properties.
  - Mechanisation.
- Complete the formalization in Coq.
  - Realise the dream of having a correct-by-construction equivalence checker for the spi calculus.
  - Define smart tactics for reasoning directly in Coq (e.g. interface with the tool that handles the decidable fragment)

Future work

- Study furthermore open hedged bisimilarity.
  - Congruence properties.
  - Mechanisation.
- Complete the formalization in Coq.
  - Realise the dream of having a correct-by-construction equivalence checker for the spi calculus.
  - Define smart tactics for reasoning directly in Coq (e.g. interface with the tool that handles the decidable fragment)
- Demos?

The end.

The spi calculus back

Syntax

- Countably infinite set of *names*.
  Communication channels, nonces, atomic data, ...
- Messages

$$M, N \quad ::= \quad x \mid (M\,.\,N) \mid \mathrm{Enc}^{\mathrm{s}}_{N} M$$

- Expressions

$$\begin{array}{rcl} E, F & ::= & x \mid (E\,.\,F) \mid \mathrm{Enc}^{\mathrm{s}}_{F} E \\ & \mid & \pi_1\,(E) \mid \pi_2\,(E) \mid \mathrm{Dec}^{\mathrm{s}}_{F} E \end{array}$$

- Guards

$$\phi \quad ::= \quad [\,E = F\,] \mid [\,E : \mathcal{N}\,]$$

Syntax back

continued

- Processes

$$\begin{array}{rcl} P, Q & ::= & \mathbf{0} \mid E(x).P \mid \overline{E}\langle F\rangle.P \\ & \mid & \phi P \mid (\nu x)\, P \\ & \mid & P \,|\, Q \mid P + Q \mid\; !\, P \end{array}$$

- Agents

$$\begin{array}{rcl} A & ::= & P \\ & \mid & (x)P \\ & \mid & (\nu \tilde{z})\,\langle M\rangle P \quad \text{where } \{\tilde{z}\} \subseteq \mathrm{n}(M) \end{array}$$

Labelled transitions system back

Late semantics

$$\textsc{Input}\ \frac{\mathbf{e}_{\mathrm{c}}(E) = a \in \mathcal{N}}{E(x).P \xrightarrow{a} (x)P} \qquad \textsc{Output}\ \frac{\mathbf{e}_{\mathrm{c}}(E) = a \in \mathcal{N} \qquad \mathbf{e}_{\mathrm{c}}(F) = M \in \boldsymbol{M}}{\overline{E}\langle F\rangle.P \xrightarrow{\overline{a}} \langle M\rangle P}$$

$$\textsc{Close-L}\ \frac{P \xrightarrow{a} F \qquad Q \xrightarrow{\overline{a}} C}{P \mid Q \xrightarrow{\tau} F \bullet C} \qquad \textsc{IfThen}\ \frac{P \xrightarrow{\mu} P'}{\phi P \xrightarrow{\mu} P'}\ \mathbf{e}(\phi) = \mathbf{true}$$

$$\textsc{Res}\ \frac{P \xrightarrow{\mu} A}{(\nu z)\, P \xrightarrow{\mu} (\boldsymbol{\nu} z)\, A}\ z \notin \mathrm{n}(\mu) \qquad \textsc{Par-L}\ \frac{P \xrightarrow{\mu} A}{P \mid Q \xrightarrow{\mu} A \mid Q}$$

+ SUM, REP- et ALPHA.

Evaluation of expressions and guards back

- Expressions:

$$
\begin{array}{rcll}
\mathbf{e}_{\mathrm{c}}(a) & := & a & \\
\mathbf{e}_{\mathrm{c}}(\mathrm{Enc}^{\mathrm{s}}_{F}E) & := & \mathrm{Enc}^{\mathrm{s}}_{N}M & \text{if } \mathbf{e}_{\mathrm{c}}(E) = M \in \boldsymbol{M} \\
 & & & \text{and } \mathbf{e}_{\mathrm{c}}(F) = N \in \boldsymbol{M} \\
\mathbf{e}_{\mathrm{c}}((E_1 \,.\, E_2)) & := & (M_1 \,.\, M_2) & \text{if } \mathbf{e}_{\mathrm{c}}(E_1) = M_1 \in \boldsymbol{M} \\
 & & & \text{and } \mathbf{e}_{\mathrm{c}}(E_2) = M_2 \in \boldsymbol{M} \\
\mathbf{e}_{\mathrm{c}}(\mathrm{Dec}^{\mathrm{s}}_{F}E) & := & M & \text{if } \mathbf{e}_{\mathrm{c}}(E) = \mathrm{Enc}^{\mathrm{s}}_{N}M \in \boldsymbol{M} \\
 & & & \text{and } \mathbf{e}_{\mathrm{c}}(F) = N \in \boldsymbol{M} \\
\mathbf{e}_{\mathrm{c}}(\pi_1(E)) & := & M_1 & \text{if } \mathbf{e}_{\mathrm{c}}(E) = (M_1 \,.\, M_2) \in \boldsymbol{M} \\
\mathbf{e}_{\mathrm{c}}(\pi_2(E)) & := & M_2 & \text{if } \mathbf{e}_{\mathrm{c}}(E) = (M_1 \,.\, M_2) \in \boldsymbol{M} \\
\mathbf{e}_{\mathrm{c}}(E) & := & \perp & \text{otherwise}
\end{array}
$$

- Guards:

$$
\begin{array}{rcll}
\mathbf{e}([E = F]) & := & \mathbf{true} & \text{si } \mathbf{e}_{\mathrm{c}}(E) = \mathbf{e}_{\mathrm{c}}(F) = M \in \boldsymbol{M} \\
\mathbf{e}([E : \mathcal{N}]) & := & \mathbf{true} & \text{si } \mathbf{e}_{\mathrm{c}}(E) = a \in \mathcal{N} \\
\mathbf{e}(\phi) & := & \mathbf{false} & \text{otherwise}
\end{array}
$$

Late hedged bisimulation back

A symmetric consistent hedged relation $\mathcal{R}$ is a *(strong) late hedged bisimulation* if whenever $(h, P, Q) \in \mathcal{R}$, we have that

1. if $P \xrightarrow{\tau} P'$ then
   there exists Q' such that $Q \xrightarrow{\tau} Q'$ and $(h, P', Q') \in \mathcal{R}$
2. if $P \xrightarrow{a} (x)P'$ (with $x \notin \mathrm{n}(\pi_1(h))$)
   and $(a, b) \in h$ then
   there exist y and Q' such that $Q \xrightarrow{b} (y)Q'$ (with $y \notin \mathrm{n}(\pi_2(h))$)
   and for all B and (M, N) such that $h \vdash_B (M, N)$
   we have $(h \cup B, P'\{^M/_x\}, Q'\{^N/_y\}) \in \mathcal{R}$.
3. if $P \xrightarrow{\overline{a}} (\nu\tilde{c})\,\langle M\rangle P'$ (with $\{\tilde{c}\} \cap \mathrm{n}(\pi_1(h)) = \emptyset$)
   and $(a, b) \in h$ then
   there exist $\tilde{d}$, Q' and N such that $Q \xrightarrow{\overline{b}} (\nu\tilde{d})\,\langle N\rangle Q'$
   (with $\left\{\tilde{d}\right\} \cap \mathrm{n}(\pi_2(h)) = \emptyset$)
   and $(\mathcal{I}(h \cup \{(M, N)\}), P', Q') \in \mathcal{R}$.

Synthesis of a hedge and possible inputs back

Synthesis of a hedge

The synthesis $\mathcal{S}(h)$ is the smallest set that satisfies

$$\textsc{Syn-Inc}\ \frac{(M, N) \in h}{(M, N) \in \mathcal{S}(h)}$$

$$\textsc{Syn-Enc-s}\ \frac{(M_1, N_1) \in \mathcal{S}(h) \qquad (M_2, N_2) \in \mathcal{S}(h)}{(\mathrm{Enc}^{\mathrm{s}}_{M_2} M_1, \mathrm{Enc}^{\mathrm{s}}_{N_2} N_1) \in \mathcal{S}(h)}$$

$$\textsc{Syn-Pair}\ \frac{(M_1, N_1) \in \mathcal{S}(h) \qquad (M_2, N_2) \in \mathcal{S}(h)}{((M_1 \,.\, M_2), (N_1 \,.\, N_2)) \in \mathcal{S}(h)}$$

Synthesis of a hedge and possible inputs back

Possible inputs

Let $h \in \boldsymbol{H}$, $(M, N) \in \boldsymbol{M} \times \boldsymbol{M}$
Let $B \subseteq \mathcal{N} \times \mathcal{N}$ a consistent hedge such that

- $\pi_1(B) \cap \mathrm{n}(\pi_1(h)) = \emptyset$
- $\pi_2(B) \cap \mathrm{n}(\pi_2(h)) = \emptyset$

i.e. the names of B are fresh component-wise w.r.t. those of h.
We write $h \vdash_B (M, N)$ if

- $\forall (b_1, b_2) \in B : b_1 \in \mathrm{n}(M) \vee b_2 \in \mathrm{n}(N)$
- $(M, N) \in \mathcal{S}(h \cup B)$

Analysis of a hedge and irreducibles back

Analysis

The analysis $\mathcal{A}(h)$ is the smallest hedge that is closed by $\mathrm{analz}(\cdot)$.

$$\textsc{Ana-Inc}\ \frac{(M, N) \in h}{(M, N) \in \mathrm{analz}(h)}$$

$$\textsc{Ana-Dec-s}\ \frac{(\mathsf{Enc}^{\mathrm{s}}_{M_2} M_1, \mathsf{Enc}^{\mathrm{s}}_{N_2} N_1) \in \mathrm{analz}(h) \qquad (M_2, N_2) \in \mathcal{S}(h)}{(M_1, N_1) \in \mathrm{analz}(h)}$$

$$\textsc{Ana-Fst}\ \frac{((M_1 \,.\, M_2), (N_1 \,.\, N_2)) \in \mathrm{analz}(h)}{(M_1, N_1) \in \mathrm{analz}(h)}$$

$$\textsc{Ana-Snd}\ \frac{((M_1 \,.\, M_2), (N_1 \,.\, N_2)) \in \mathrm{analz}(h)}{(M_2, N_2) \in \mathrm{analz}(h)}$$

Analysis of a hedge and irreducibles back

Irreducibles

$\mathcal{I}(h)$ is the smallest hedge such that $\mathcal{S}(\mathcal{I}(h)) = \mathcal{S}(\mathcal{A}(h))$.

Definition

A hedge h is irreducible iff $\mathcal{I}(h) = h$.

Consistency of a hedge back

Consistency

A hedge h is consistent iff:
Whenever $(M, N) \in h$

- $M \in \mathcal{N} \iff N \in \mathcal{N}$
- whenever $(M', N') \in h$: $M = M' \iff N = N'$
- $M \neq (M_1 \,.\, M_2)$ and $N \neq (N_1 \,.\, N_2)$
- if $M = \mathrm{Enc}^{\mathrm{s}}_{M_2} M_1$ then $(M_2, N_2) \notin \mathcal{S}(h)$
- if $N = \mathrm{Enc}^{\mathrm{s}}_{N_2} N_1$ then $(M_2, N_2) \notin \mathcal{S}(h)$

Lemma

A consistent hedge is irreducible.

S-environments back

Definition (S-environment)

A S-environment is a quadruple $se = (h, v, \prec, (\gamma_l, \gamma_r))$ where $h \in \boldsymbol{H}$, $v \subseteq \mathcal{N} \times \mathcal{N}$ is a consistent hedge, $\prec \subseteq h \times v$, $\gamma_l \subseteq \pi_1(v)$ and $\gamma_r \subseteq \pi_2(v)$.

Hedge available

The *hedge available* to $(x, y) \in v$ according to $\prec$ is defined by $se|_{(x,y)} := \{(M, N) \in h \mid (M, N) \prec (x, y)\}$.

Concrete hedge

The *concrete hedge* of *se* is $\mathfrak{H}(se) := h \cup v$.

Respectful substitutions back

Definition (Respectful substitutions)

Let (σ, ρ) be a pair of substitutions, $B \subseteq \mathcal{N} \times \mathcal{N}$ a consistent hedge and $se = (h, v, \prec, (\gamma_l, \gamma_r))$ a S-environment. We say that (σ, ρ) *respects se with B* — written $(\sigma, \rho) \rhd_B se$ — if

1. $\mathrm{supp}(\sigma) \subseteq \pi_1(v)$
2. $\mathrm{supp}(\rho) \subseteq \pi_2(v)$
3. $\forall (b_1, b_2) \in B : b_1 \in \mathrm{n}(\sigma(\pi_1(v))) \vee b_2 \in \mathrm{n}(\rho(\pi_2(v)))$
4. $\pi_1(B) \cap (\mathrm{n}(\pi_1(h)) \setminus \pi_1(v)) = \emptyset$
5. $\pi_2(B) \cap (\mathrm{n}(\pi_2(h)) \setminus \pi_2(v)) = \emptyset$
6. $\forall (x, y) \in v : (x\sigma, y\rho) \in \mathcal{S}(\mathcal{I}(se|_{(x,y)}(\sigma, \rho) \cup B))$
7. $\forall x \in \gamma_l : x\sigma \in \mathcal{N}$
8. $\forall y \in \gamma_r : y\rho \in \mathcal{N}$

Open hedged bisimulation back

A symmetric consistent open hedged relation $\mathcal{R}$ is an *open hedged bisimulation* if for all $(se, P, Q) \in \mathcal{R}$, for all σ, ρ and B such that $(\sigma, \rho) \triangleright_B se$,

internal communications

if $P\sigma \xhookrightarrow[S_1]{\tau} P'$ then

there exist Q' and S_2 such that $Q\rho \xhookrightarrow[S_2]{\tau} Q'$

and $(se_B^{(\sigma,\rho)} +_c (S_1, S_2), P', Q') \in \mathcal{R}$

Open hedged bisimulation back

A symmetric consistent open hedged relation $\mathcal{R}$ is an *open hedged bisimulation* if for all $(se, P, Q) \in \mathcal{R}$, for all σ, ρ and B such that $(\sigma, \rho) \triangleright_B se$,

inputs

if $P\sigma \xhookrightarrow[S_1]{a} (x)P'$ (with $x \notin \mathrm{n}(\pi_1(\mathfrak{H}(se_B^{(\sigma,\rho)}))))$
and $(a, b) \in \mathcal{S}(\mathcal{I}(\mathfrak{H}(se_B^{(\sigma,\rho)})))$ then
there exist y, Q' and S_2 such that $Q\rho \xhookrightarrow[S_2]{b} (y)Q'$ (with $y \notin \mathrm{n}(\pi_2(\mathfrak{H}(se_B^{(\sigma,\rho)}))))$
and $(se_B^{(\sigma,\rho)} +_{\mathrm{i}} (x, y) +_{\mathrm{c}} (S_1, S_2), P', Q') \in \mathcal{R}$

Open hedged bisimulation back

A symmetric consistent open hedged relation $\mathcal{R}$ is an *open hedged bisimulation* if for all $(se, P, Q) \in \mathcal{R}$, for all σ, ρ and B such that $(\sigma, \rho) \triangleright_B se$,

outputs

if $P\sigma \xrightarrow[S_1]{\overline{a}} (\nu\tilde{c})\,\langle M\rangle P'$ (with $\{\tilde{c}\} \cap \mathrm{n}(\pi_1(\mathfrak{H}(se_B^{(\sigma,\rho)}))) = \emptyset$)
and $(a, b) \in \mathcal{S}(\mathcal{I}(\mathfrak{H}(se_B^{(\sigma,\rho)})))$ then
there exist $\tilde{d}$, N, Q' and S_2 such that $Q\rho \xrightarrow[S_2]{\overline{b}} (\nu\tilde{d})\,\langle N\rangle Q'$
(with $\left\{\tilde{d}\right\} \cap \mathrm{n}(\pi_2(\mathfrak{H}(se_B^{(\sigma,\rho)}))) = \emptyset$)
and $(se_B^{(\sigma,\rho)} +_{\mathrm{o}} (M, N) +_{\mathrm{c}} (S_1, S_2), P', Q') \in \mathcal{R}$

A LTS that collects type constraints back

$$\text{NC-SILENT}\ \frac{}{\tau.P \xhookrightarrow[\emptyset]{\tau} P}$$

$$\text{NC-INPUT}\ \frac{\mathbf{e}_c(E) = a \in \mathcal{N}}{E(x).P \xhookrightarrow[\{a\}]{a} (x)P}$$

$$\text{NC-OUTPUT}\ \frac{\mathbf{e}_c(E) = a \in \mathcal{N} \qquad \mathbf{e}_c(F) = M \in \boldsymbol{M}}{\overline{E}\langle F\rangle.P \xhookrightarrow[\{a\}]{\overline{a}} \langle M\rangle P}$$

$$\text{NC-IFTHEN}\ \frac{P \xhookrightarrow[S]{\mu} A}{\phi P \xhookrightarrow[S \cup \mathbf{nc}(\phi)]{\mu} A}\ \mathbf{e}(\phi) = \mathbf{true}$$

where $\mathbf{nc}([E:\mathcal{N}]) := \{\mathbf{e}_c(E)\}$ and $\mathbf{nc}([E=F]) := \emptyset$.

Properties back

Theorem

The two semantics are equivalent:

1. *If* $P \xrightarrow{\mu} A$ *there exists* $S \subseteq \mathcal{N}$ *such that* $P \underset{S}{\overset{\mu}{\hookrightarrow}} A$.
2. *If* $P \underset{S}{\overset{\mu}{\hookrightarrow}} A$ *then* $P \xrightarrow{\mu} A$.

Lemma

If $P \underset{S}{\overset{\mu}{\hookrightarrow}} A$ *and* $\sigma : \mathcal{N} \to \mathbf{M}$ *is a substitution such that* $S\sigma \subseteq \mathcal{N}$ *then* $P\sigma \underset{S\sigma}{\overset{\mu\sigma}{\hookrightarrow}} A\sigma$.

A symbolic LTS back

$$\text{S-GUARD}\ \frac{P \overset{\mu}{\underset{c}{\longmapsto}} A}{\phi P \overset{\mu}{\underset{c\&\{\phi\}}{\longmapsto}} A} \qquad \text{S-INPUT}\ \frac{}{E(x).P \overset{\mathbf{e}_{\mathrm{a}}(E)}{\underset{\{[E:\mathcal{N}]\}}{\longmapsto}} (x)P}$$

$$\text{S-OUTPUT}\ \frac{}{\overline{E}\langle F\rangle.P \overset{\overline{\mathbf{e}_{\mathrm{a}}(E)}}{\underset{\{[E:\mathcal{N}],[F:\boldsymbol{M}]\}}{\longmapsto}} \langle \mathbf{e}_{\mathrm{a}}(F)\rangle P}$$

$$\text{S-CLOSE-L}\ \frac{P \overset{E}{\underset{c_1}{\longmapsto}} F \qquad Q \overset{\overline{E'}}{\underset{c_2}{\longmapsto}} C}{P \mid Q \overset{\tau}{\underset{\{[E=E']\}\&c_1\&c_2}{\longmapsto}} F \bullet C}$$

$$\text{S-RES}\ \frac{P \overset{\mu}{\underset{c}{\longmapsto}} A}{(\nu z)\, P \overset{\mu}{\underset{\nu_+(z,c)}{\longmapsto}} (\boldsymbol{\nu} z)\, A}\ z \notin \mathrm{n}(\mu)$$

Transition constraints back

- A transition constraint has the form $(\nu\tilde{z})\,\Phi$ where Φ is a finite set of guards and $\tilde{z}$ is a finite set of names that occur in Φ, i.e. $\{\tilde{z}\} \subseteq \mathrm{n}(\Phi)$
- Composition of constraints:
  - Conjunction of $c_1 = (\nu\tilde{z}_1)\,\Phi_1$ and $c_2 = (\nu\tilde{z}_2)\,\Phi_2$ where $\{\tilde{z}_1\} \cap \{\tilde{z}_2\} = \emptyset$, $\{\tilde{z}_1\} \cap \mathrm{fn}(c_2) = \{\tilde{z}_2\} \cap \mathrm{fn}(c_1) = \emptyset$

$$c_1 \,\&\, c_2 := (\nu\tilde{z}_1\tilde{z}_2)\,(\Phi_1 \cup \Phi_2)$$

  - Restriction of name x.
    If $c = (\nu\tilde{z})\,\Phi$ and $x \notin \{\tilde{z}\}$:

$$\begin{array}{rcll} \nu_+(x, c) & := & (\nu x\tilde{z})\,\Phi & \text{if } x \in \mathrm{fn}(c) \\ & := & c & \text{otherwise} \end{array}$$

Abstract evaluation back

Abstract evaluation of expressions:

$$
\begin{array}{rcll}
\mathbf{e}_{\mathrm{a}}(a) & := & a & \text{if } a \in \mathcal{N} \\
\mathbf{e}_{\mathrm{a}}(\mathrm{Enc}^{\mathrm{s}}_{F} E) & := & \mathrm{Enc}^{\mathrm{s}}_{\mathbf{e}_{\mathrm{a}}(F)} \mathbf{e}_{\mathrm{a}}(E) & \\
\mathbf{e}_{\mathrm{a}}((E\,.\,F)) & := & (\mathbf{e}_{\mathrm{a}}(E)\,.\,\mathbf{e}_{\mathrm{a}}(F)) & \\
\mathbf{e}_{\mathrm{a}}(\mathrm{Dec}^{\mathrm{s}}_{F} E) & := & E_1 & \text{if } \mathbf{e}_{\mathrm{a}}(E) = \mathrm{Enc}^{\mathrm{s}}_{E_2} E_1 \\
 & & \mathrm{Dec}^{\mathrm{s}}_{\mathbf{e}_{\mathrm{a}}(F)} \mathbf{e}_{\mathrm{a}}(E) & \text{otherwise} \\
\mathbf{e}_{\mathrm{a}}(\pi_1\,(E)) & := & E_1 & \text{if } \mathbf{e}_{\mathrm{a}}(E) = (E_1\,.\,E_2) \\
 & & \pi_1\,(\mathbf{e}_{\mathrm{a}}(E)) & \text{otherwise} \\
\mathbf{e}_{\mathrm{a}}(\pi_2\,(E)) & := & E_2 & \text{if } \mathbf{e}_{\mathrm{a}}(E) = (E_1\,.\,E_2) \\
 & & \pi_2\,(\mathbf{e}_{\mathrm{a}}(E)) & \text{otherwise}
\end{array}
$$

Properties back

Define $>_{\mathrm{o}}$ as being the smallest precongruence on expressions that satisfies:

- $\pi_1\,((E_1\,.\,E_2)) >_{\mathrm{o}} E_1$ if $\mathbf{e}_{\mathrm{c}}(E_2) \neq \bot$
- $\pi_2\,((E_1\,.\,E_2)) >_{\mathrm{o}} E_2$ if $\mathbf{e}_{\mathrm{c}}(E_1) \neq \bot$
- $\mathrm{Dec}^{\mathrm{s}}_{E_2}\mathrm{Enc}^{\mathrm{s}}_{E_2}E_1 >_{\mathrm{o}} E_1$ if $\mathbf{e}_{\mathrm{c}}(E_2) \neq \bot$

Extend this relation to agents in:

- $A >^{=}_{\mathrm{o}} B$ (A, B are concrete agents)
- $A >^{\mathbf{e}}_{\mathrm{o}} B$ (A is symbolic, B is concrete)

(two ways to handle concretions)

Properties back

continued

Theorem

Let $P, Q \in \mathbf{P}$ and assume that $P >_{\mathrm{o}} Q$.

1. *If $P \overset{\mu}{\underset{S}{\hookrightarrow}} A$ then $Q \overset{\mu}{\underset{S}{\hookrightarrow}} B$ and $A >_{\mathrm{o}}^{=} B$*
2. *If $Q \overset{\mu}{\underset{S}{\hookrightarrow}} B$ then $P \overset{\mu}{\underset{S}{\hookrightarrow}} A$ and $A >_{\mathrm{o}}^{=} B$*

Theorem

Let $P, Q \in \mathbf{P}$ and $\sigma : \mathcal{N} \to \mathbf{M}$ a substitution.

1. *If $P \overset{\mu_s}{\underset{c}{\longmapsto}} A$ and $\mathbf{e}(c\sigma) = \mathbf{true}$ then $P\sigma \xhookrightarrow[\mathbf{nc}(c\sigma)]{\mathbf{e}_{\mathrm{c}}(\mu_s\sigma)} B$ with $A\sigma >_{\mathrm{o}}^{\mathbf{e}} B$*
2. *If $P\sigma \overset{\mu}{\underset{S}{\hookrightarrow}} B$ then $P \overset{\mu_s}{\underset{c}{\longmapsto}} A$ with $\mathbf{e}(c\sigma) = \mathbf{true}$, $\mathbf{nc}(c\sigma) = S$, $\mathbf{e}_{\mathrm{c}}(\mu_s\sigma) = \mu$ and $A\sigma >_{\mathrm{o}}^{\mathbf{e}} B$*

Operations on de Bruijn indices back

- Parametrised by the binding depth d
- $\mathrm{mem}_d(i, t)$ returns **true** iff i is free in t
- $\mathrm{lift}_d(k, t)$ makes room for k new binders in t
  Used in parallel composition of an agent and a process:

$$
\begin{array}{rcl}
(\lambda.P) \mid Q & := & \lambda.(P \mid \mathrm{lift}_0(1, Q)) \\
(\nu^k\langle F\rangle P) \mid Q & := & \nu^k\langle F\rangle(P \mid \mathrm{lift}_0(k, Q))
\end{array}
$$

For instance:

$z\; y\; x\; \cdots c\; b\; a$

$\lambda.\overline{1}\langle 0\rangle.\mathbf{0}$

$z\; y\; x\; \cdots c\; b\; a$

$0\; \lambda.\overline{0}\langle 24\rangle.\mathbf{0}$

$z\; y\; x\; \cdots c\; b\; a$

$\lambda.(\overline{1}\langle 0\rangle.\mathbf{0} \mid 1\; \lambda.\overline{0}\langle 25\rangle.\mathbf{0})$

Operations on de Bruijn indices back

continued

- $\mathsf{swap}_d(k, t)$ makes a circular permutation of the k first indices in t
- $\mathsf{low}_d(t)$ removes the first index
- Used in restriction of an agent:

$$\begin{array}{rcll} \boldsymbol{\nu}(\lambda.P) & := & \lambda.\nu\,\mathsf{swap}_0(1, P) & \\ \boldsymbol{\nu}(\nu^k\langle F\rangle P) & := & \nu^{k+1}\langle F\rangle P & \text{if } \mathsf{mem}_k(0, F) = \textbf{true} \\ & := & \nu^k\langle \mathsf{low}_k(F)\rangle\nu\,\mathsf{swap}_0(k, P) & \text{otherwise} \end{array}$$

- $\mathsf{lsubst}_d(k, \overline{E}, t)$ substitutes the $|\overline{E}|$ first indices with the corresponding expression of $\overline{E}$ in t. The k first indices are bound in $\overline{E}$.

$$(\lambda.P) \bullet (\nu^k\langle F\rangle Q) \quad := \quad \nu^k(\mathsf{lsubst}_0(k, F, P) \mid Q)$$