A Formal Semantics For Protocol Narrations

S. Briais U. Nestmann

School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne

Symposium on Trustworthy Global Computing
Edinburgh, UK, April 7-9, 2005

What is a protocol narration?

- Sequence of message exchanges
- $A \longrightarrow B : M$ read "From A to B send M"
- Messages

$$M, N ::= A \mid n \mid (M \,.\, N) \mid \{M\}_N \mid \cdots \quad \in \boldsymbol{M}$$

What is a protocol narration?

- Sequence of message exchanges
- $A \longrightarrow B : M$ read "From A to B send M"
- Messages

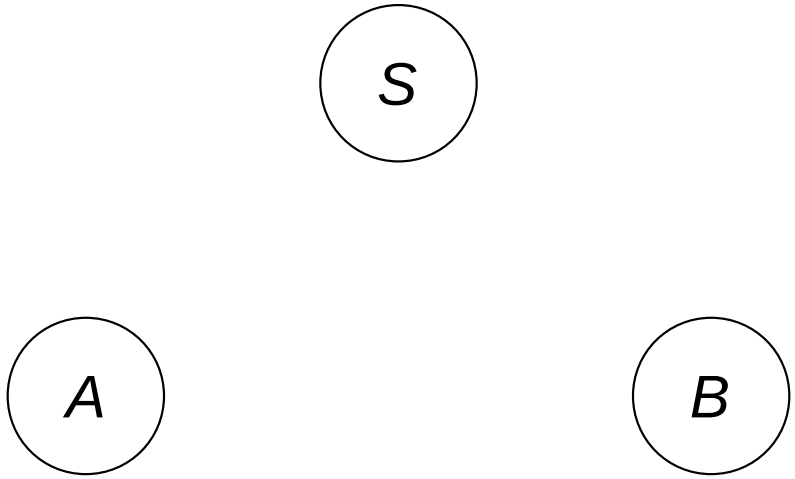
$$M, N ::= A \mid n \mid (M \,.\, N) \mid \{M\}_N \mid \cdots \quad \in \boldsymbol{M}$$

What is a protocol narration?

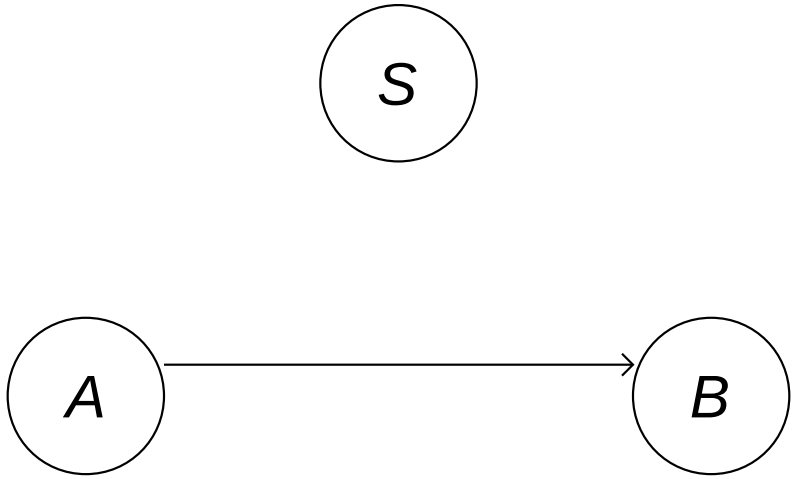
- Sequence of message exchanges
- $A \rightarrow B : M$ read "From A to B send M"
- Messages

$$M, N ::= A \mid n \mid (M \,.\, N) \mid \{M\}_N \mid \cdots \quad \in \boldsymbol{M}$$

The Yahalom protocol

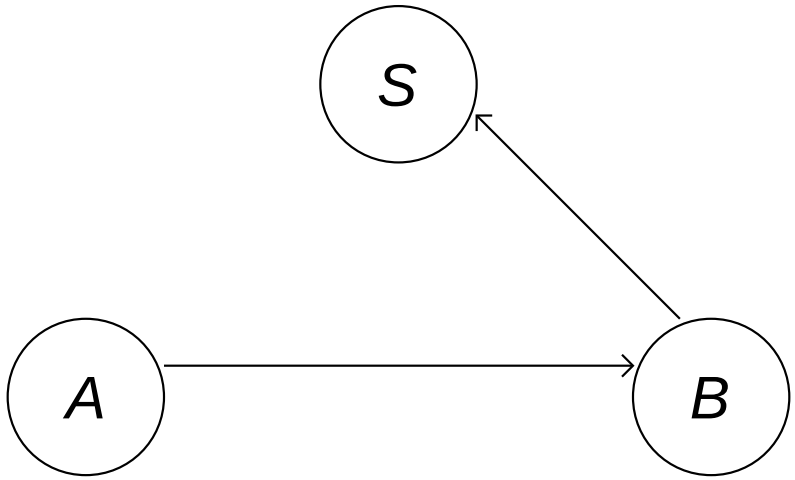


The Yahalom protocol



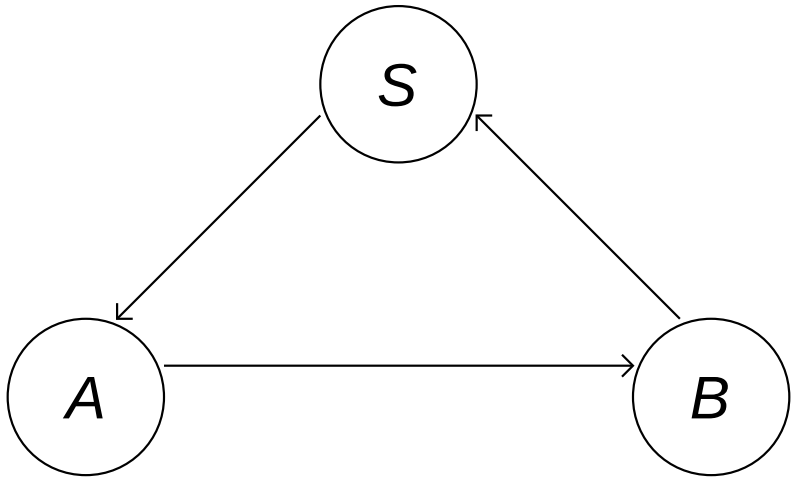
1. $A \rightarrow B : (A \,.\, n_A)$

The Yahalom protocol



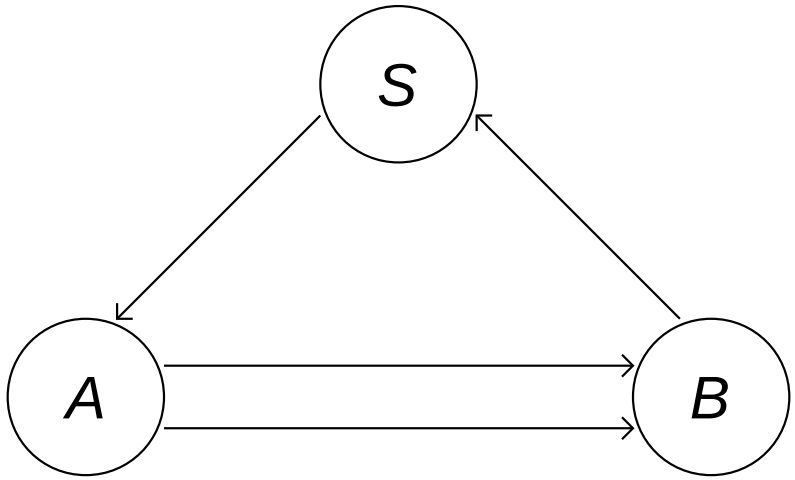
1. $A \rightarrow B : (A \,.\, n_A)$
2. $B \rightarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}})$

The Yahalom protocol



1. $A \rightarrow B : (A \,.\, n_A)$
2. $B \rightarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}})$
3. $S \rightarrow A : (\{((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))\}_{k_{AS}} \,.\, \{(A \,.\, k_{AB})\}_{k_{BS}})$

The Yahalom protocol



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

The Yahalom protocol in spi-calculus

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

The Yahalom protocol in spi-calculus

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

The Yahalom protocol in spi-calculus

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

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

Related work

- Spyer (with Gensoul)
- Casper (Lowe)
- CAPSL (Millen)
- LySa (Bodei, Buchholtz, Degano, Nielson, Nielson)

Related work

- Spyer (with Gensoul)
- Casper (Lowe)
- CAPSL (Millen)
- LySa (Bodei, Buchholtz, Degano, Nielson, Nielson)
- セキュリティプロトコルの略式記法から spi 計算への変換 (Sumii, Tatsuzawa, Yonezawa)

Outline

Outline

The Yahalom protocol

$$
\begin{aligned}
&A \rightsquigarrow B : (A \,.\, n_A);\\
&B \rightsquigarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}});\\
&S \rightsquigarrow A : (\{((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))\}_{k_{AS}} \,.\, \{(A \,.\, k_{AB})\}_{k_{BS}});\\
&A \rightsquigarrow B : (\{(A \,.\, k_{AB})\}_{k_{BS}} \,.\, \{n_B\}_{k_{AB}})
\end{aligned}
$$

The Yahalom protocol

private k_{AS}; A **knows** k_{AS}; S **knows** k_{AS};

$$
\begin{aligned}
&A \rightsquigarrow B : (A \,.\, n_A);\\
&B \rightsquigarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}});\\
&S \rightsquigarrow A : (\{((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))\}_{k_{AS}} \,.\, \{(A \,.\, k_{AB})\}_{k_{BS}});\\
&A \rightsquigarrow B : (\{(A \,.\, k_{AB})\}_{k_{BS}} \,.\, \{n_B\}_{k_{AB}})
\end{aligned}
$$

- A secret key k_{AS} is assumed to be shared between A and S

The Yahalom protocol

private k_{AS}; A **knows** k_{AS}; S **knows** k_{AS};
private k_{BS}; B **knows** k_{BS}; S **knows** k_{BS};

$$A \rightsquigarrow B : (A \,.\, n_A);$$
$$B \rightsquigarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}});$$
$$S \rightsquigarrow A : (\{((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))\}_{k_{AS}} \,.\, \{(A \,.\, k_{AB})\}_{k_{BS}});$$
$$A \rightsquigarrow B : (\{(A \,.\, k_{AB})\}_{k_{BS}} \,.\, \{n_B\}_{k_{AB}})$$

- A secret key k_{AS} is assumed to be shared between A and S
- A secret key k_{BS} is assumed to be shared between B and S

The Yahalom protocol

private k_{AS}; A **knows** k_{AS}; S **knows** k_{AS};
private k_{BS}; B **knows** k_{BS}; S **knows** k_{BS};
A **generates** n_A;

$A \rightsquigarrow B : (A \,.\, n_A)$;
$B \rightsquigarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}})$;
$S \rightsquigarrow A : (\{((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))\}_{k_{AS}} \,.\, \{(A \,.\, k_{AB})\}_{k_{BS}})$;
$A \rightsquigarrow B : (\{(A \,.\, k_{AB})\}_{k_{BS}} \,.\, \{n_B\}_{k_{AB}})$

- A secret key k_{AS} is assumed to be shared between A and S
- A secret key k_{BS} is assumed to be shared between B and S
- A generates a fresh nonce n_A

The Yahalom protocol

private k_{AS}; A **knows** k_{AS}; S **knows** k_{AS};
private k_{BS}; B **knows** k_{BS}; S **knows** k_{BS};
A **generates** n_A; B **generates** n_B;

$$A \rightsquigarrow B : (A \,.\, n_A);$$
$$B \rightsquigarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}});$$
$$S \rightsquigarrow A : (\{((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))\}_{k_{AS}} \,.\, \{(A \,.\, k_{AB})\}_{k_{BS}});$$
$$A \rightsquigarrow B : (\{(A \,.\, k_{AB})\}_{k_{BS}} \,.\, \{n_B\}_{k_{AB}})$$

- A secret key k_{AS} is assumed to be shared between A and S
- A secret key k_{BS} is assumed to be shared between B and S
- A generates a fresh nonce n_A
- B generates a fresh nonce n_B

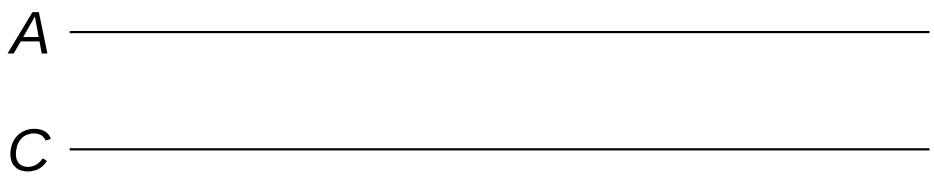
The Yahalom protocol

private k_{AS}; A **knows** k_{AS}; S **knows** k_{AS};
private k_{BS}; B **knows** k_{BS}; S **knows** k_{BS};
A **generates** n_A; B **generates** n_B;
S **generates** k_{AB};
$A \rightsquigarrow B : (A \,.\, n_A)$;
$B \rightsquigarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}})$;
$S \rightsquigarrow A : (\{((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))\}_{k_{AS}} \,.\, \{(A \,.\, k_{AB})\}_{k_{BS}})$;
$A \rightsquigarrow B : (\{(A \,.\, k_{AB})\}_{k_{BS}} \,.\, \{n_B\}_{k_{AB}})$

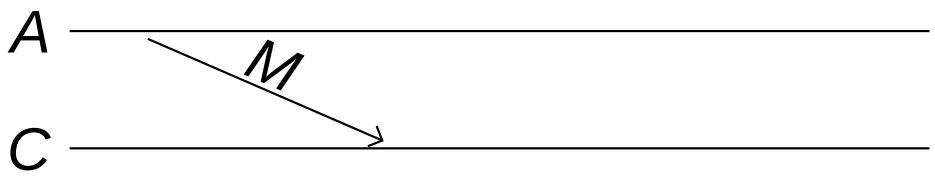
- A secret key k_{AS} is assumed to be shared between A and S
- A secret key k_{BS} is assumed to be shared between B and S
- A generates a fresh nonce n_A
- B generates a fresh nonce n_B
- S creates a fresh key k_{AB}

Outline

Concurrency

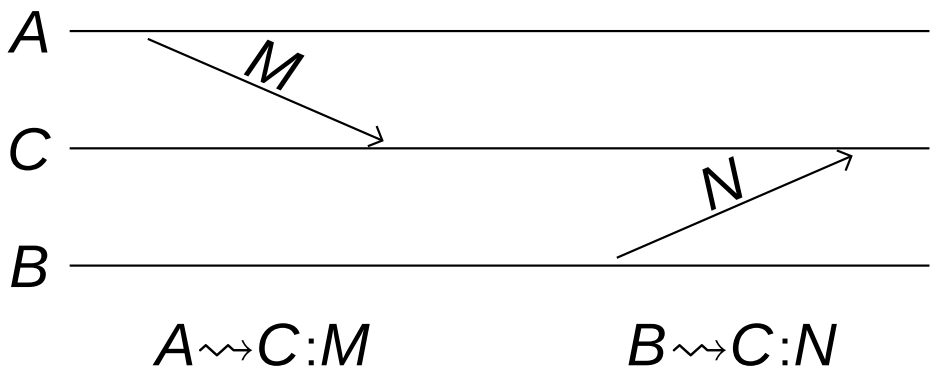


Concurrency

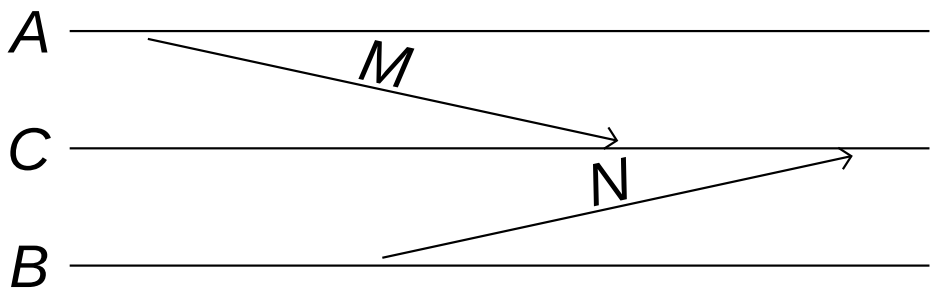


$$A \rightsquigarrow C : M$$

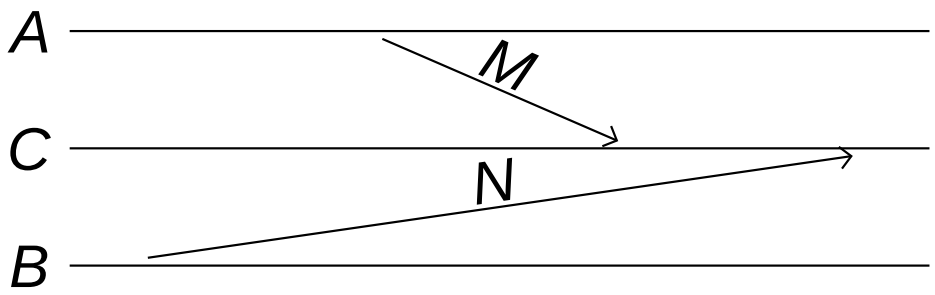
Concurrency



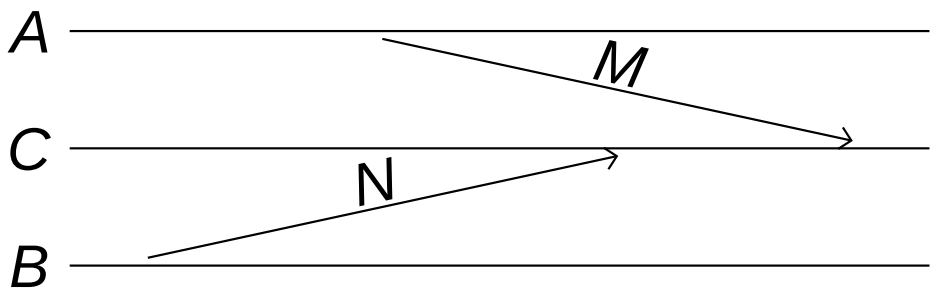
Concurrency



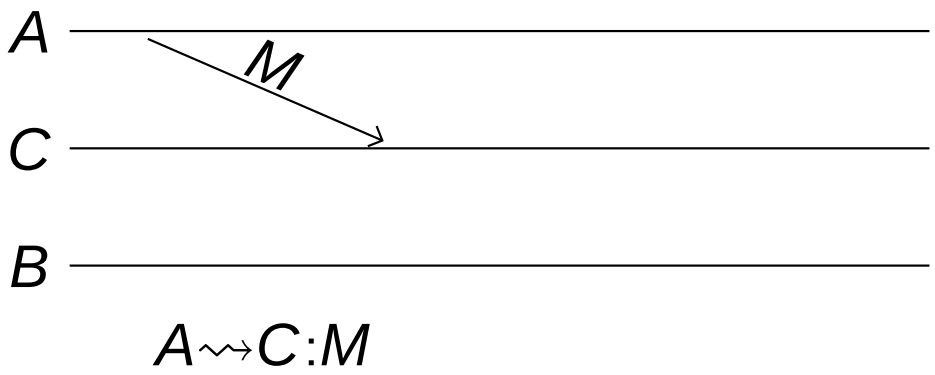
Concurrency



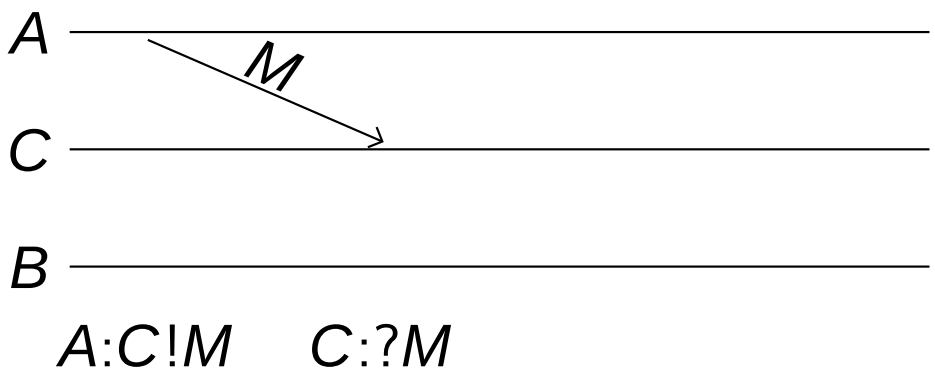
Concurrency



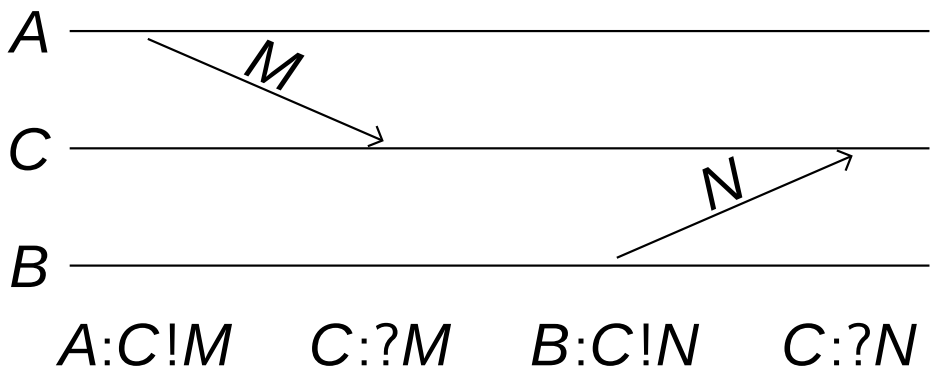
Concurrency



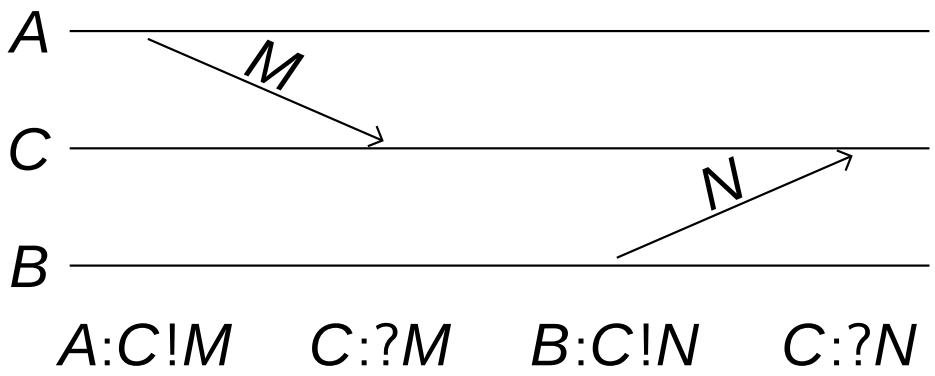
Concurrency



Concurrency

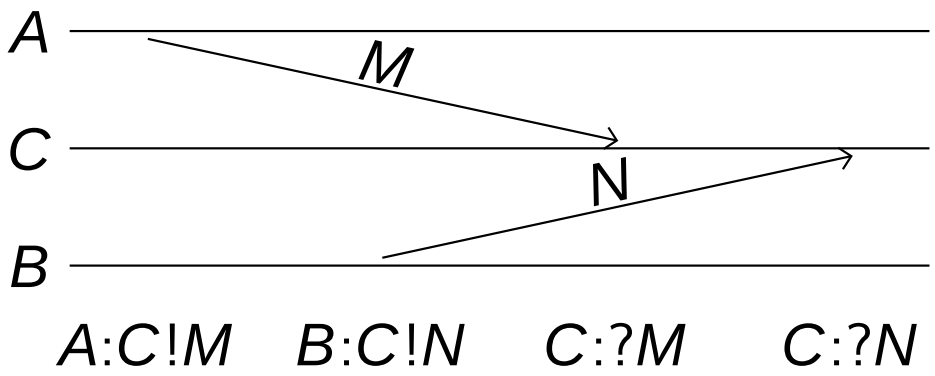


Concurrency



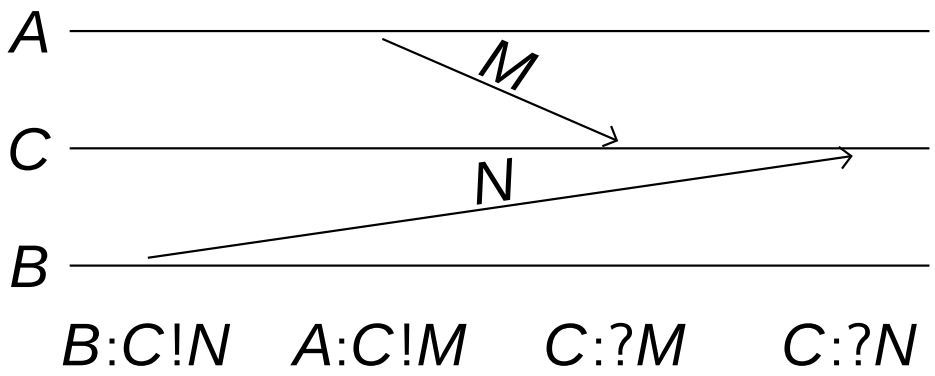
$$A : \cdots ; B : \cdots \cong B : \cdots ; A : \cdots \qquad \text{if } A \neq B$$

Concurrency



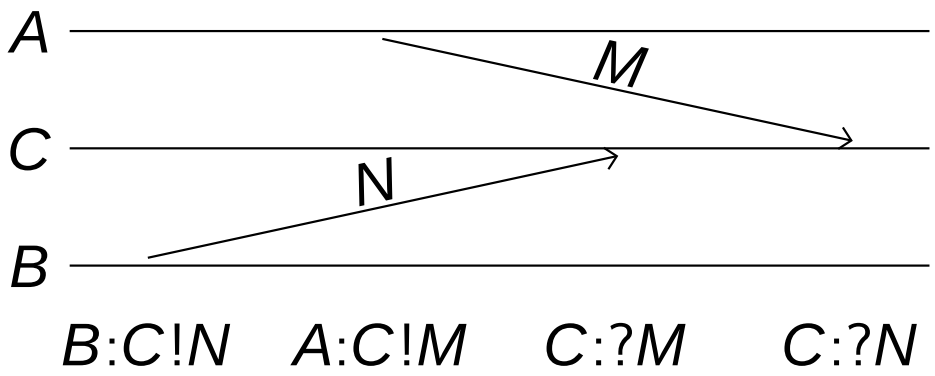
$$A : \cdots ; B : \cdots \cong B : \cdots ; A : \cdots \qquad \text{if } A \neq B$$

Concurrency



$$A : \cdots ; B : \cdots \cong B : \cdots ; A : \cdots \qquad \text{if } A \neq B$$

Concurrency



$$A : \cdots ; B : \cdots \cong B : \cdots ; A : \cdots \qquad \text{if } A \neq B$$

How to generate the checks?

$$A \rightsquigarrow B : M \quad \longrightarrow \quad A : B!M ; B : ?M$$

How to generate the checks?

$$A \rightsquigarrow B : M \quad \longrightarrow \quad A : B!M ; B : ?x$$

How to generate the checks?

$$A \rightsquigarrow B : M \quad \longrightarrow \quad A : B!M ; B : ?x ; B : [x = M]$$

How to generate the checks?

$$A \rightsquigarrow B : M \quad \longrightarrow \quad A : B!M ; B : ?x ; B : \phi_x$$

How to generate the checks?

$$A \rightsquigarrow B : M \quad \longrightarrow \quad A : B!M ; B : ?x ; B : \phi_x$$

- B expects x to have the same type as M

How to generate the checks?

$$A \rightsquigarrow B : M \quad \longrightarrow \quad A : B!M ; B : ?x ; B : \phi_x$$

- B expects x to have the same type as M
- B can use his acquired knowledge to check x consistency

The Yahalom protocol

private k_{AS}; A **knows** k_{AS}; S **knows** k_{AS};
private k_{BS}; B **knows** k_{BS}; S **knows** k_{BS};
A **generates** n_A; B **generates** n_B;
S **generates** k_{AB};
$A \rightsquigarrow B : (A \,.\, n_A)$;
$B \rightsquigarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}})$;
$S \rightsquigarrow A : (\{((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))\}_{k_{AS}} \,.\, \{(A \,.\, k_{AB})\}_{k_{BS}})$;
$A \rightsquigarrow B : (\{(A \,.\, k_{AB})\}_{k_{BS}} \,.\, \{n_B\}_{k_{AB}})$

- A secret key k_{AS} is assumed to be shared between A and S
- A secret key k_{BS} is assumed to be shared between B and S
- A generates a fresh nonce n_A
- B generates a fresh nonce n_B
- S creates a fresh key k_{AB}

The Yahalom protocol

private k_{AS}; A **knows** k_{AS}; S **knows** k_{AS};
private k_{BS}; B **knows** k_{BS}; S **knows** k_{BS};
A **generates** n_A; B **generates** n_B;
S **generates** k_{AB};
$A \rightsquigarrow B : (A \,.\, n_A)$;
$B \rightsquigarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}})$;
$S \rightsquigarrow A : (\{((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))\}_{k_{AS}} \,.\, \{(A \,.\, k_{AB})\}_{k_{BS}})$;
$A \rightsquigarrow B : (\{(A \,.\, k_{AB})\}_{k_{BS}} \,.\, \{n_B\}_{k_{AB}})$

- A secret key k_{AS} is assumed to be shared between A and S
- A secret key k_{BS} is assumed to be shared between B and S
- A generates a fresh nonce n_A
- B generates a fresh nonce n_B
- S creates a fresh key k_{AB}

Analysing a knowledge set

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

The Yahalom protocol

private k_{AS}; A **knows** k_{AS}; S **knows** k_{AS};
private k_{BS}; B **knows** k_{BS}; S **knows** k_{BS};
A **generates** n_A; B **generates** n_B;
S **generates** k_{AB};
$A \rightsquigarrow B : (A \,.\, n_A)$;
$B \rightsquigarrow S : (B \,.\, \{(A \,.\, (n_A \,.\, n_B))\}_{k_{BS}})$;
$S \rightsquigarrow A : (\{((B \,.\, k_{AB}) \,.\, (n_A \,.\, n_B))\}_{k_{AS}} \,.\, \{(A \,.\, k_{AB})\}_{k_{BS}})$;
$A \rightsquigarrow B : (\{(A \,.\, k_{AB})\}_{k_{BS}} \,.\, \{n_B\}_{k_{AB}})$

- A secret key k_{AS} is assumed to be shared between A and S
- A secret key k_{BS} is assumed to be shared between B and S
- A generates a fresh nonce n_A
- B generates a fresh nonce n_B
- S creates a fresh key k_{AB}

Analysing a knowledge set

$$A$$
$$B$$
$$S$$
$$k_{AS}$$
$$n_A$$
$$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$$

Analysing a knowledge set

$$A$$
$$B$$
$$S$$
$$k_{AS}$$
$$n_A$$
$$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}}) \qquad x$$

Analysing a knowledge set

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$	x

What is a knowledge set?

It is a finite subset of $M \times E$.
It connects reality to expectations.

Analysing a knowledge set

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$	x

Analysing a knowledge set

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$	x
$\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}$	$\pi_1(x)$
$\{(A\,.\,k_{AB})\}_{k_{BS}}$	$\pi_2(x)$

Analysing a knowledge set

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$	x
$\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}$	$\pi_1(x)$
$\{(A\,.\,k_{AB})\}_{k_{BS}}$	$\pi_2(x)$
$((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\mathrm{D}_{k_{AS}}(\pi_1(x))$

Analysing a knowledge set

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$	x
$\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}$	$\pi_1(x)$
$\{(A\,.\,k_{AB})\}_{k_{BS}}$	$\pi_2(x)$
$((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\mathsf{D}_{k_{AS}}(\pi_1(x))$
$(B\,.\,k_{AB})$	$\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))$
$(n_A\,.\,n_B)$	$\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))$

Analysing a knowledge set

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$	x
$\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}$	$\pi_1(x)$
$\{(A\,.\,k_{AB})\}_{k_{BS}}$	$\pi_2(x)$
$((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\mathsf{D}_{k_{AS}}(\pi_1(x))$
$(B\,.\,k_{AB})$	$\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))$
$(n_A\,.\,n_B)$	$\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))$
B	$\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x))))$
k_{AB}	$\pi_2(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x))))$
n_A	$\pi_1(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x))))$
n_B	$\pi_2(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x))))$

Analysis of a knowledge set

$$\mathcal{A}(K) = \bigcup_{n \in \mathbb{N}} \mathcal{A}_n(K)$$

$$\text{ANA-INC-INIT}\ \frac{(M,E) \in K}{(M,E) \in \mathcal{A}_0(K)}$$

$$\text{ANA-INC-REC}\ \frac{(M,E) \in \mathcal{A}_n(K) \qquad M \in \boldsymbol{N} \cup \boldsymbol{A}}{(M,E) \in \mathcal{A}_{n+1}(K)}$$

$$\text{ANA-FST}\ \frac{((M\,.\,N),E) \in \mathcal{A}_n(K)}{(M,\pi_1(E)) \in \mathcal{A}_{n+1}(K)} \qquad \text{ANA-SND}\ \frac{((M\,.\,N),E) \in \mathcal{A}_n(K)}{(N,\pi_2(E)) \in \mathcal{A}_{n+1}(K)}$$

$$\text{ANA-DEC}\ \frac{(\{M\}_N,E) \in \mathcal{A}_n(K) \qquad (N,F) \in \mathcal{S}(\mathcal{A}_n(K))}{(M,\mathsf{D}_F(E)) \in \mathcal{A}_{n+1}(K)}$$

$$\text{ANA-DEC-REC}\ \frac{(\{M\}_N,E) \in \mathcal{A}_n(K) \qquad (N,F) \notin \mathcal{S}(\mathcal{A}_n(K))}{(\{M\}_N,E) \in \mathcal{A}_{n+1}(K)}$$

How to generate the checks?

$$A \rightsquigarrow B : M \quad \longrightarrow \quad A : B!M ; B : ?x ; B : \phi_x$$

- B expects x to have the same type as M
- B can use his acquired knowledge to check x consistency

Computing the formula

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$	x
$\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}$	$\pi_1(x)$
$\{(A\,.\,k_{AB})\}_{k_{BS}}$	$\pi_2(x)$
$((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\mathrm{D}_{k_{AS}}(\pi_1(x))$
$(B\,.\,k_{AB})$	$\pi_1(\mathrm{D}_{k_{AS}}(\pi_1(x)))$
$(n_A\,.\,n_B)$	$\pi_2(\mathrm{D}_{k_{AS}}(\pi_1(x)))$
B	$\pi_1(\pi_1(\mathrm{D}_{k_{AS}}(\pi_1(x))))$
k_{AB}	$\pi_2(\pi_1(\mathrm{D}_{k_{AS}}(\pi_1(x))))$
n_A	$\pi_1(\pi_2(\mathrm{D}_{k_{AS}}(\pi_1(x))))$
n_B	$\pi_2(\pi_2(\mathrm{D}_{k_{AS}}(\pi_1(x))))$

Computing the formula

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$	x
$\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}$	$\pi_1(x)$
$\{(A\,.\,k_{AB})\}_{k_{BS}}$	$\pi_2(x)$
$((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\mathsf{D}_{k_{AS}}(\pi_1(x))$
$(B\,.\,k_{AB})$	$\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))$
$(n_A\,.\,n_B)$	$\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))$
B	$\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x))))$
k_{AB}	$\pi_2(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x))))$
n_A	$\pi_1(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x))))$
n_B	$\pi_2(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x))))$

The result of the computation

$$
\begin{aligned}
&[A:\boldsymbol{M}] \wedge [B:\boldsymbol{M}] \wedge [S:\boldsymbol{M}] \wedge [k_{AS}:\boldsymbol{M}] \wedge [n_A:\boldsymbol{M}] \wedge [x:\boldsymbol{M}] \\
\wedge\, &[\pi_1(x):\boldsymbol{M}] \wedge [\pi_2(x):\boldsymbol{M}] \\
\wedge\, &[\mathsf{D}_{k_{AS}}(\pi_1(x)):\boldsymbol{M}] \\
\wedge\, &[\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x))):\boldsymbol{M}] \wedge [\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x))):\boldsymbol{M}] \\
\wedge\, &[\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \wedge [\pi_2(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \\
\wedge\, &[\pi_1(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \wedge [\pi_2(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}]
\end{aligned}
$$

Computing the formula

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$	x
$\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}$	$\pi_1(x)$
$\{(A\,.\,k_{AB})\}_{k_{BS}}$	$\pi_2(x)$
$((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\mathsf{D}_{k_{AS}}(\pi_1(x))$
$(B\,.\,k_{AB})$	$\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))$
$(n_A\,.\,n_B)$	$\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))$
B	$\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x))))$
k_{AB}	$\pi_2(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x))))$
n_A	$\pi_1(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x))))$
n_B	$\pi_2(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x))))$

The result of the computation

$$
\begin{aligned}
&[A:\boldsymbol{M}] \wedge [B:\boldsymbol{M}] \wedge [S:\boldsymbol{M}] \wedge [k_{AS}:\boldsymbol{M}] \wedge [n_A:\boldsymbol{M}] \wedge [x:\boldsymbol{M}] \\
&\wedge [\pi_1(x):\boldsymbol{M}] \wedge [\pi_2(x):\boldsymbol{M}] \\
&\wedge [\mathsf{D}_{k_{AS}}(\pi_1(x)):\boldsymbol{M}] \\
&\wedge [\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x))):\boldsymbol{M}] \wedge [\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x))):\boldsymbol{M}] \\
&\wedge [\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \wedge [\pi_2(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \\
&\wedge [\pi_1(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \wedge [\pi_2(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \\
&\wedge [\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))) = B]
\end{aligned}
$$

Computing the formula

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}\,.\,\{(A\,.\,k_{AB})\}_{k_{BS}})$	x
$\{((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))\}_{k_{AS}}$	$\pi_1(x)$
$\{(A\,.\,k_{AB})\}_{k_{BS}}$	$\pi_2(x)$
$((B\,.\,k_{AB})\,.\,(n_A\,.\,n_B))$	$\mathrm{D}_{k_{AS}}(\pi_1(x))$
$(B\,.\,k_{AB})$	$\pi_1(\mathrm{D}_{k_{AS}}(\pi_1(x)))$
$(n_A\,.\,n_B)$	$\pi_2(\mathrm{D}_{k_{AS}}(\pi_1(x)))$
B	$\pi_1(\pi_1(\mathrm{D}_{k_{AS}}(\pi_1(x))))$
k_{AB}	$\pi_2(\pi_1(\mathrm{D}_{k_{AS}}(\pi_1(x))))$
n_A	$\pi_1(\pi_2(\mathrm{D}_{k_{AS}}(\pi_1(x))))$
n_B	$\pi_2(\pi_2(\mathrm{D}_{k_{AS}}(\pi_1(x))))$

The result of the computation

$$
\begin{aligned}
&[A:\boldsymbol{M}] \wedge [B:\boldsymbol{M}] \wedge [S:\boldsymbol{M}] \wedge [k_{AS}:\boldsymbol{M}] \wedge [n_A:\boldsymbol{M}] \wedge [x:\boldsymbol{M}] \\
&\wedge [\pi_1(x):\boldsymbol{M}] \wedge [\pi_2(x):\boldsymbol{M}] \\
&\wedge [\mathsf{D}_{k_{AS}}(\pi_1(x)):\boldsymbol{M}] \\
&\wedge [\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x))):\boldsymbol{M}] \wedge [\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x))):\boldsymbol{M}] \\
&\wedge [\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \wedge [\pi_2(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \\
&\wedge [\pi_1(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \wedge [\pi_2(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \\
&\wedge [\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))) = B] \wedge [\pi_1(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))) = n_A]
\end{aligned}
$$

The formula of an analysed knowledge set

$$\Phi(K) \stackrel{\text{def}}{=} \bigwedge_{(M,E)\in K} [E : M] \;\wedge\; \bigwedge_{(M,E_i)\in K \wedge (M,E_j)\in K \wedge E_i \neq E_j} [E_i = E_j]$$

Removing redundancies

$$
\begin{aligned}
&[A:\boldsymbol{M}] \wedge [B:\boldsymbol{M}] \wedge [S:\boldsymbol{M}] \wedge [k_{AS}:\boldsymbol{M}] \wedge [n_A:\boldsymbol{M}] \wedge [x:\boldsymbol{M}] \\
&\wedge [\pi_1(x):\boldsymbol{M}] \wedge [\pi_2(x):\boldsymbol{M}] \\
&\wedge [\mathsf{D}_{k_{AS}}(\pi_1(x)):\boldsymbol{M}] \\
&\wedge [\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x))):\boldsymbol{M}] \wedge [\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x))):\boldsymbol{M}] \\
&\wedge [\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \wedge [\pi_2(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \\
&\wedge [\pi_1(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \wedge [\pi_2(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))):\boldsymbol{M}] \\
&\wedge [\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))) = B] \wedge [\pi_1(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))) = n_A]
\end{aligned}
$$

Removing redundancies

$$[\pi_2(x) : \boldsymbol{M}]$$

$$\begin{aligned} &\wedge [\pi_2(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))) : \boldsymbol{M}] \\ &\wedge [\pi_2(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))) : \boldsymbol{M}] \\ \wedge [\pi_1(\pi_1(\mathsf{D}_{k_{AS}}(\pi_1(x)))) = B] &\wedge [\pi_1(\pi_2(\mathsf{D}_{k_{AS}}(\pi_1(x)))) = n_A] \end{aligned}$$

Outline

Labelled transition system

$$\textsc{Send}\ \frac{[\![E]\!] = M \in \boldsymbol{M}}{A{:}B!E;X \xrightarrow{A:B!M} X} \qquad \textsc{Receive}\ \frac{}{A{:}?x;X \xrightarrow{A:?M} X\{^{M}/_{x}\}_{A}}\ M \in \boldsymbol{M}$$

$$\textsc{Check}\ \frac{X \xrightarrow{A:\beta} X'}{A{:}\phi;X \xrightarrow{A:\beta} X'}\ [\![\phi]\!] = \textbf{true}$$

$$\textsc{Open}\ \frac{X \xrightarrow{A:(\nu\tilde{n})\,B!M} X'}{\nu z;X \xrightarrow{A:(\nu z\tilde{n})\,B!M} X'}\ z \in \mathrm{n}(M) \setminus \{\tilde{n}\}$$

$$\textsc{Rearrange}\ \frac{X \cong_{\alpha} X' \qquad X' \xrightarrow{A:\beta} X''}{X \xrightarrow{A:\beta} X''}$$

Labelled transition system

$$\textsc{Send}\ \frac{[\![E]\!] = M \in \boldsymbol{M}}{A{:}B!E;X \xrightarrow{A:B!M} X} \qquad \textsc{Receive}\ \frac{}{A{:}?x;X \xrightarrow{A:?M} X\{^{M}/_{x}\}_{A}}\ M \in \boldsymbol{M}$$

$$\textsc{Check}\ \frac{X \xrightarrow{A:\beta} X'}{A{:}\phi;X \xrightarrow{A:\beta} X'}\ [\![\phi]\!] = \textbf{true}$$

$$\textsc{Open}\ \frac{X \xrightarrow{A:(\nu\tilde{n})\,B!M} X'}{\nu z;X \xrightarrow{A:(\nu z\tilde{n})\,B!M} X'}\ z \in \mathsf{n}(M) \setminus \{\tilde{n}\}$$

$$\textsc{Rearrange}\ \frac{X \cong_{\alpha} X' \qquad X' \xrightarrow{A:\beta} X''}{X \xrightarrow{A:\beta} X''}$$

Labelled transition system

$$\textsc{Send}\ \frac{[\![E]\!] = M \in \boldsymbol{M}}{A{:}B!E;X \xrightarrow{A:B!M} X} \qquad \textsc{Receive}\ \frac{}{A{:}?x;X \xrightarrow{A:?M} X\{^{M}/_{x}\}_{A}}\ M \in \boldsymbol{M}$$

$$\textsc{Check}\ \frac{X \xrightarrow{A:\beta} X'}{A{:}\phi;X \xrightarrow{A:\beta} X'}\ [\![\phi]\!] = \textbf{true}$$

$$\textsc{Open}\ \frac{X \xrightarrow{A:(\nu\tilde{n})\,B!M} X'}{\nu z;X \xrightarrow{A:(\nu z\tilde{n})\,B!M} X'}\ z \in \mathsf{n}(M) \setminus \{\tilde{n}\}$$

$$\textsc{Rearrange}\ \frac{X \cong_{\alpha} X' \qquad X' \xrightarrow{A:\beta} X''}{X \xrightarrow{A:\beta} X''}$$

Outline

Translation of executable narrations in spi

$$X_{/A} = \text{“projection of } X \text{ on } A\text{”}$$

$$\mathcal{T}[\![X]\!] \stackrel{\text{def}}{=} (\nu n)_{n \in R(X)} \prod_{A \in \mathcal{A}(X)} X_{/A}$$

Conclusion

Abadi, 2000.

1. One should make explicit what is known (public, private) before a protocol run, and what is to be generated freshly during a protocol run.
2. One should make explicit which checks the individual principals are expected to carry out on the reception of messages.
3. Principals act concurrently, in contrast to the apparently sequential idealized execution of a run according to a narration.
4. Concurrency occurs also at the level of different protocol sessions, which may happen to be executed simultaneously while sharing principals across.

Conclusion

Abadi, 2000.

1. One should make explicit what is known (public, private) before a protocol run, and what is to be generated freshly during a protocol run.
2. One should make explicit which checks the individual principals are expected to carry out on the reception of messages.
3. Principals act concurrently, in contrast to the apparently sequential idealized execution of a run according to a narration.
4. Concurrency occurs also at the level of different protocol sessions, which may happen to be executed simultaneously while sharing principals across.

Conclusion

Abadi, 2000.

1. One should make explicit what is known (public, private) before a protocol run, and what is to be generated freshly during a protocol run.
2. One should make explicit which checks the individual principals are expected to carry out on the reception of messages.
3. Principals act concurrently, in contrast to the apparently sequential idealized execution of a run according to a narration.
4. Concurrency occurs also at the level of different protocol sessions, which may happen to be executed simultaneously while sharing principals across.

Conclusion

Abadi, 2000.

1. One should make explicit what is known (public, private) before a protocol run, and what is to be generated freshly during a protocol run.
2. One should make explicit which checks the individual principals are expected to carry out on the reception of messages.
3. Principals act concurrently, in contrast to the apparently sequential idealized execution of a run according to a narration.
4. Concurrency occurs also at the level of different protocol sessions, which may happen to be executed simultaneously while sharing principals across.

Conclusion

Abadi, 2000.

1. One should make explicit what is known (public, private) before a protocol run, and what is to be generated freshly during a protocol run.
2. One should make explicit which checks the individual principals are expected to carry out on the reception of messages.
3. Principals act concurrently, in contrast to the apparently sequential idealized execution of a run according to a narration.
4. Concurrency occurs also at the level of different protocol sessions, which may happen to be executed simultaneously while sharing principals across.

Future work

- Concurrency at the level of protocol sessions.

Future work

- Concurrency at the level of protocol sessions.

$$A \rightsquigarrow S : \{M\}_{k_{AS}}$$

Future work

- Concurrency at the level of protocol sessions.

$$A \rightsquigarrow S : \{M\}_{k_{AS}}$$

Future work

- Concurrency at the level of protocol sessions.

$$A \rightsquigarrow S : \{M\}_{k_{AS}}$$

- Formal reasoning using executable narrations.

Questions?

Bibliography

J.A. Clark and J.L. Jacob
A Survey Of Authentication Protocol Literature.
Technical Report 1.0, University of York, 1997.

M. Abadi
Security Protocols And Their Properties.
Foundations of Secure Computation, 2000.

Synthesis of a knowledge set

$$K \subset \mathcal{S}(K)$$

$$\text{SYN-PAIR}\ \frac{(M,E) \in \mathcal{S}(K) \qquad (N,F) \in \mathcal{S}(K)}{((M\,.\,N),(E\,.\,F)) \in \mathcal{S}(K)}$$

$$\text{SYN-ENC}\ \frac{(M,E) \in \mathcal{S}(K) \qquad (N,F) \in \mathcal{S}(K)}{(\{M\}_N,\{E\}_F) \in \mathcal{S}(K)}$$

Consistency formula of a knowledge set

$$\begin{aligned}\Phi(K) \stackrel{\text{def}}{=} \quad & \bigwedge_{((M\,.\,N),E)\in K} \left([\pi_1(E):\boldsymbol{M}] \wedge [\pi_2(E):\boldsymbol{M}]\right) \\ \wedge \quad & \bigwedge_{(\{M\}_N,E)\in K \,\wedge\, (N,F)\in \mathcal{S}(K)} [\mathsf{D}_F(E):\boldsymbol{M}] \\ \wedge \quad & \bigwedge_{(M,E_i)\in K \,\wedge\, (M,E_j)\in K \,\wedge\, E_i\neq E_j} [E_i = E_j]\end{aligned}$$

Adding hashing

$$\text{SYN-HASH}\ \frac{(M,E) \in \mathcal{S}(K)}{(\mathsf{H}(M),\mathsf{H}(E)) \in \mathcal{S}(K)}$$

$$\Phi(K) \stackrel{\text{def}}{=} \cdots \wedge \bigwedge_{(\mathsf{H}(M),E)\in K \,\wedge\, (M,F)\in\mathcal{S}(K)} [E = \mathsf{H}(F)]$$

Consistency formula of a knowledge set

$$\Phi(K) \stackrel{\text{def}}{=} \bigwedge_{(M,E)\in K} [E : \boldsymbol{M}] \wedge \bigwedge_{(M,E_i)\in K \wedge (M,E_j)\in \mathcal{S}(K) \wedge E_i \neq E_j} [E_i = E_j]$$