

A Formal Semantics For Protocol Narrations

S. Briaïs 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

- $A \rightarrow B : M$ read “From A to B send M ”

- Messages

$$M, N ::= A | n | (M.N) | \{M\}_N | \dots \in \mathbf{M}$$

What is a protocol narration?

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

- Messages

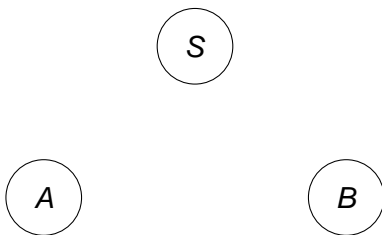
$$M, N ::= A | n | (M.N) | \{M\}_N | \dots \in \mathcal{M}$$

What is a protocol narration?

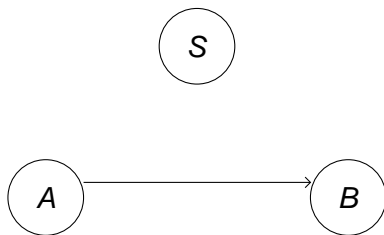
- Sequence of message exchanges
- $A \rightarrow B : M$ read “From A to B send M ”
- Messages

$$M, N ::= A | n | (M . N) | \{M\}_N | \dots \in \mathbf{M}$$

The Yahalom protocol

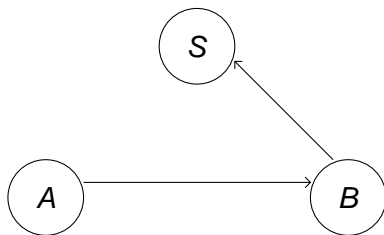


The Yahalom protocol



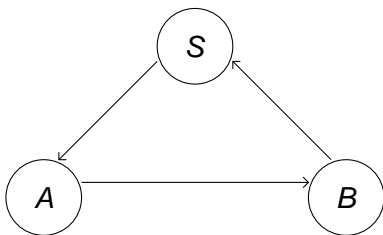
1 $A \rightarrow B : (A.n_A)$

The Yahalom protocol



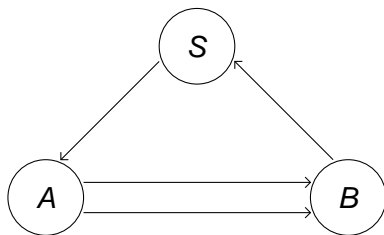
- 1 $A \rightarrow B : (A.n_A)$
- 2 $B \rightarrow S : (B.\{(A.(n_A.n_B))\}_{k_{BS}})$

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

$$\begin{aligned}
 & (\nu k_{AS}, k_{BS}) \\
 & (\nu n_A) \bar{B} \langle (A . n_A) \rangle . A(x_2) . \phi_2 \bar{B} \langle (\pi_2(x_2) . \{\pi_2(\pi_2(D_{k_{AS}}(\pi_1(x_2))))\}_{\pi_2(\pi_1(D_{k_{AS}}(\pi_1(x_2))))}) \rangle . \mathbf{0} \\
 & \| (\nu n_B) B(x_0) . \phi_0 \bar{S} \langle (B . \{(A . (\pi_2(x_0) . n_B))\}_{k_{BS}}) \rangle . B(x_3) . \phi_3 \mathbf{0} \\
 & \| (\nu k_{AB}) \\
 & S(x_1) . \phi_1 \\
 & \bar{A} \langle (\{(B . k_{AB}) . (\pi_1(\pi_2(D_{k_{BS}}(\pi_2(x_1)))) . \pi_2(\pi_2(D_{k_{BS}}(\pi_2(x_1)))))\}_{k_{AS}} . \{(A . k_{AB})\}_{k_{BS}}) \rangle . \mathbf{0}
 \end{aligned}$$

The Yahalom protocol in spi-calculus

$$\begin{aligned}
 & (\nu k_{AS}, k_{BS}) \\
 & (\nu n_A) \bar{B} \langle (A . n_A) \rangle . A(x_2) . \phi_2 \bar{B} \langle (\pi_2(x_2) . \{\pi_2(\pi_2(D_{k_{AS}}(\pi_1(x_2))))\}_{\pi_2(\pi_1(D_{k_{AS}}(\pi_1(x_2))))}) \rangle . \mathbf{0} \\
 & \| (\nu n_B) B(x_0) . \phi_0 \bar{S} \langle (B . \{(A . (\pi_2(x_0) . n_B))\}_{k_{BS}}) \rangle . B(x_3) . \phi_3 \mathbf{0} \\
 & \| (\nu k_{AB}) \\
 & S(x_1) . \phi_1 \\
 & \bar{A} \langle (\{(B . k_{AB}) . (\pi_1(\pi_2(D_{k_{BS}}(\pi_2(x_1)))) . \pi_2(\pi_2(D_{k_{BS}}(\pi_2(x_1))))\})\}_{k_{AS}} . \{(A . k_{AB})\}_{k_{BS}}) \rangle . \mathbf{0}
 \end{aligned}$$

The Yahalom protocol in spi-calculus

$$\begin{aligned}
 & (\nu k_{AS}, k_{BS}) \\
 & (\nu n_A) \bar{B} \langle (A. n_A) \rangle . A(x_2) . \phi_2 \bar{B} \langle (\pi_2(x_2) . \{\pi_2(\pi_2(D_{k_{AS}}(\pi_1(x_2))))\})_{\pi_2(\pi_1(D_{k_{AS}}(\pi_1(x_2))))} \rangle . \mathbf{0} \\
 & \| (\nu n_B) B(x_0) . \phi_0 \bar{S} \langle (B. \{(A. (\pi_2(x_0) . n_B))\}_{k_{BS}}) \rangle . B(x_3) . \phi_3 \mathbf{0} \\
 & \| (\nu k_{AB}) \\
 & S(x_1) . \phi_1 \\
 & \bar{A} \langle (\{(B. k_{AB}) . (\pi_1(\pi_2(D_{k_{BS}}(\pi_2(x_1)))) . \pi_2(\pi_2(D_{k_{BS}}(\pi_2(x_1))))\})_{k_{AS}} . \{(A. k_{AB})\}_{k_{BS}}) \rangle . \mathbf{0}
 \end{aligned}$$

$$\phi_0 = [\pi_1(x_0) : \mathbf{M}] \wedge [\pi_2(x_0) : \mathbf{M}] \wedge [\pi_1(x_0) = A]$$

$$\begin{aligned}
 \phi_1 = & [\pi_1(x_1) : \mathbf{M}] \wedge [\pi_2(x_1) : \mathbf{M}] \wedge [D_{k_{BS}}(\pi_2(x_1)) : \mathbf{M}] \wedge [\pi_1(D_{k_{BS}}(\pi_2(x_1))) : \mathbf{M}] \wedge \\
 & [\pi_2(D_{k_{BS}}(\pi_2(x_1))) : \mathbf{M}] \wedge [\pi_1(\pi_2(D_{k_{BS}}(\pi_2(x_1)))) : \mathbf{M}] \wedge [\pi_2(\pi_2(D_{k_{BS}}(\pi_2(x_1)))) : \mathbf{M}] \wedge \\
 & [\pi_1(x_1) = B] \wedge [\pi_1(D_{k_{BS}}(\pi_2(x_1))) = A]
 \end{aligned}$$

$$\begin{aligned}
 \phi_2 = & [\pi_1(x_2) : \mathbf{M}] \wedge [\pi_2(x_2) : \mathbf{M}] \wedge [D_{k_{AS}}(\pi_1(x_2)) : \mathbf{M}] \wedge [\pi_1(D_{k_{AS}}(\pi_1(x_2))) : \mathbf{M}] \wedge \\
 & [\pi_2(D_{k_{AS}}(\pi_1(x_2))) : \mathbf{M}] \wedge [\pi_1(\pi_1(D_{k_{AS}}(\pi_1(x_2)))) : \mathbf{M}] \wedge [\pi_2(\pi_1(D_{k_{AS}}(\pi_1(x_2)))) : \mathbf{M}] \wedge \\
 & [\pi_1(\pi_2(D_{k_{AS}}(\pi_1(x_2)))) : \mathbf{M}] \wedge [\pi_2(\pi_2(D_{k_{AS}}(\pi_1(x_2)))) : \mathbf{M}] \wedge \\
 & [\pi_1(\pi_2(D_{k_{AS}}(\pi_1(x_2)))) = n_A] \wedge [\pi_1(\pi_1(D_{k_{AS}}(\pi_1(x_2)))) = B]
 \end{aligned}$$

$$\begin{aligned}
 \phi_3 = & [\pi_1(x_3) : \mathbf{M}] \wedge [\pi_2(x_3) : \mathbf{M}] \wedge [D_{k_{BS}}(\pi_1(x_3)) : \mathbf{M}] \wedge \\
 & [D_{\pi_2(D_{k_{BS}}(\pi_1(x_3)))}(\pi_2(x_3)) : \mathbf{M}] \wedge [\pi_1(D_{k_{BS}}(\pi_1(x_3))) : \mathbf{M}] \wedge [\pi_2(D_{k_{BS}}(\pi_1(x_3))) : \mathbf{M}] \wedge \\
 & [\pi_1(D_{k_{BS}}(\pi_1(x_3))) = A] \wedge [D_{\pi_2(D_{k_{BS}}(\pi_1(x_3)))}(\pi_2(x_3)) = n_B]
 \end{aligned}$$

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

- 1 Extending protocol narrations
- 2 Compiling protocol narrations
- 3 Executing protocol narrations
- 4 Rewriting protocol narrations... in spi-calculus

Outline

- 1 Extending protocol narrations
- 2 Compiling protocol narrations
- 3 Executing protocol narrations
- 4 Rewriting protocol narrations... in spi-calculus

The Yahalom protocol

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

The Yahalom protocol

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

$A \rightsquigarrow B: (A . n_A);$

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

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

$A \rightsquigarrow B: (\{(A . k_{AB})\}_{k_{BS}} . \{n_B\}_{k_{AB}})$

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

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}

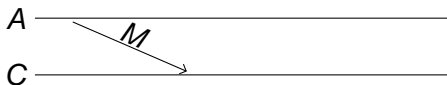
Outline

- 1 Extending protocol narrations
- 2 Compiling protocol narrations**
- 3 Executing protocol narrations
- 4 Rewriting protocol narrations... in spi-calculus

Concurrency

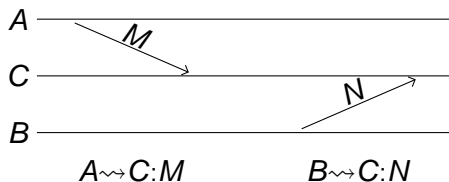


Concurrency

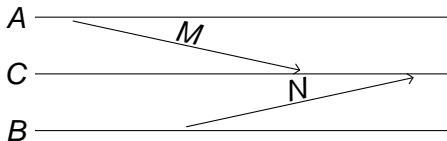


$$A \rightsquigarrow C : M$$

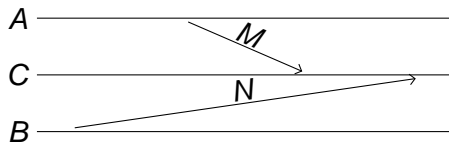
Concurrency



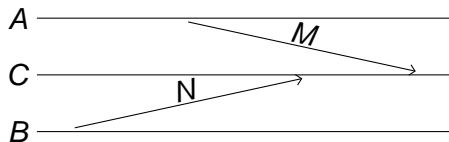
Concurrency



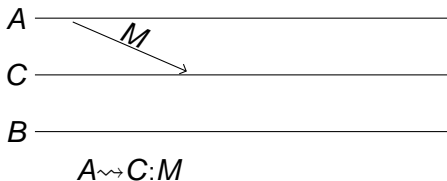
Concurrency



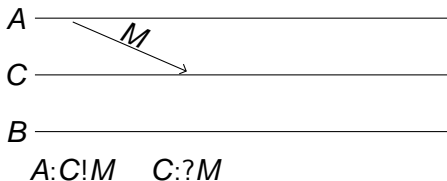
Concurrency



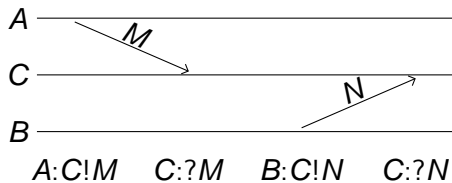
Concurrency



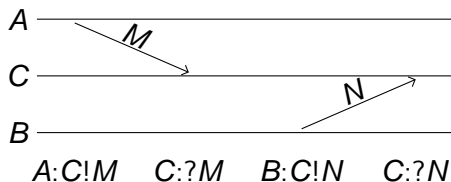
Concurrency



Concurrency

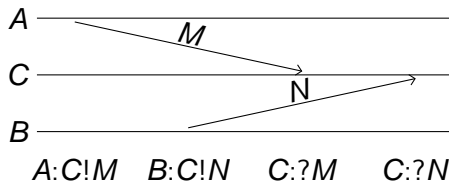


Concurrency



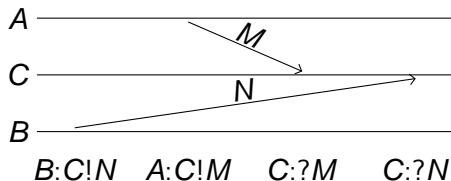
$$A : \dots ; B : \dots \cong B : \dots ; A : \dots \quad \text{if } A \neq B$$

Concurrency



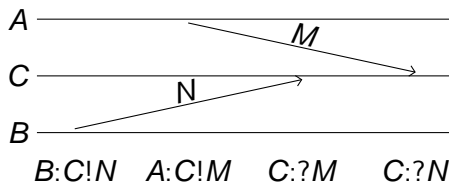
$$A : \dots ; B : \dots \cong B : \dots ; A : \dots \quad \text{if } A \neq B$$

Concurrency



$$A : \dots ; B : \dots \cong B : \dots ; A : \dots \quad \text{if } A \neq B$$

Concurrency



$$A : \dots ; B : \dots \cong B : \dots ; A : \dots \quad \text{if } A \neq B$$

How to generate the checks?

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

How to generate the checks?

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

How to generate the checks?

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

How to generate the checks?

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

How to generate the checks?

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

- B expects x to have the same type as M

How to generate the checks?

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

- 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

$$(\{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \cdot \{(A \cdot k_{AB})\}_{k_{BS}})$$

Analysing a knowledge set

$$\begin{array}{c}
 A \\
 B \\
 S \\
 k_{AS} \\
 n_A \\
 (\{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \cdot \{(A \cdot k_{AB})\}_{k_{BS}}) \quad X
 \end{array}$$

Analysing a knowledge set

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

Analysing a knowledge set

A	A
B	B
S	S
k_{AS}	k_{AS}
n_A	n_A
$(\{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \cdot \{(A \cdot k_{AB})\}_{k_{BS}})$	X
$\{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}}$	$\pi_1(X)$
$\{(A \cdot k_{AB})\}_{k_{BS}}$	$\pi_2(X)$

Analysing a knowledge set

$$\begin{array}{c}
 A \\
 B \\
 S \\
 k_{AS} \\
 n_A \\
 (\{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \cdot \{(A \cdot k_{AB})\}_{k_{BS}}) \\
 \quad \{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \\
 \quad \quad \{(A \cdot k_{AB})\}_{k_{BS}} \\
 \quad ((B \cdot k_{AB}) \cdot (n_A \cdot n_B))
 \end{array}$$

$$\begin{array}{c}
 A \\
 B \\
 S \\
 k_{AS} \\
 n_A \\
 x \\
 \pi_1(x) \\
 \pi_2(x) \\
 D_{k_{AS}}(\pi_1(x))
 \end{array}$$

Analysing a knowledge set

$$\begin{array}{c}
 A \\
 B \\
 S \\
 k_{AS} \\
 n_A \\
 (\{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \cdot \{(A \cdot k_{AB})\}_{k_{BS}}) \\
 \quad \{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \\
 \quad \quad \{(A \cdot k_{AB})\}_{k_{BS}} \\
 \quad ((B \cdot k_{AB}) \cdot (n_A \cdot n_B)) \\
 \quad (B \cdot k_{AB}) \\
 \quad (n_A \cdot n_B)
 \end{array}$$

$$\begin{array}{c}
 A \\
 B \\
 S \\
 k_{AS} \\
 n_A \\
 x \\
 \pi_1(x) \\
 \pi_2(x) \\
 D_{k_{AS}}(\pi_1(x)) \\
 \pi_1(D_{k_{AS}}(\pi_1(x))) \\
 \pi_2(D_{k_{AS}}(\pi_1(x)))
 \end{array}$$

Analysing a knowledge set

$$\begin{array}{l}
 A \\
 B \\
 S \\
 k_{AS} \\
 n_A \\
 (\{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \cdot \{(A \cdot k_{AB})\}_{k_{BS}}) \\
 \quad \{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \\
 \quad \quad \{(A \cdot k_{AB})\}_{k_{BS}} \\
 \quad ((B \cdot k_{AB}) \cdot (n_A \cdot n_B)) \\
 \quad (B \cdot k_{AB}) \\
 \quad (n_A \cdot n_B) \\
 B \\
 k_{AB} \\
 n_A \\
 n_B
 \end{array}$$

$$\begin{array}{l}
 A \\
 B \\
 S \\
 k_{AS} \\
 n_A \\
 x \\
 \pi_1(x) \\
 \pi_2(x) \\
 D_{k_{AS}}(\pi_1(x)) \\
 \pi_1(D_{k_{AS}}(\pi_1(x))) \\
 \pi_2(D_{k_{AS}}(\pi_1(x))) \\
 \pi_1(\pi_1(D_{k_{AS}}(\pi_1(x)))) \\
 \pi_2(\pi_1(D_{k_{AS}}(\pi_1(x)))) \\
 \pi_1(\pi_2(D_{k_{AS}}(\pi_1(x)))) \\
 \pi_2(\pi_2(D_{k_{AS}}(\pi_1(x))))
 \end{array}$$

Analysis of a knowledge set

$$\mathcal{A}(K) = \bigcup_{n \in \mathbb{N}} \mathcal{A}_n(K) \quad \text{ANA-INC-INIT} \quad \frac{(M, E) \in K}{(M, E) \in \mathcal{A}_0(K)}$$

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

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

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

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

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

How to generate the checks?

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

- 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
$(\{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \cdot \{(A \cdot k_{AB})\}_{k_{BS}})$	x
$\{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}}$	$\pi_1(x)$
$\{(A \cdot k_{AB})\}_{k_{BS}}$	$\pi_2(x)$
$((B \cdot k_{AB}) \cdot (n_A \cdot n_B))$	$D_{k_{AS}}(\pi_1(x))$
$(B \cdot k_{AB})$	$\pi_1(D_{k_{AS}}(\pi_1(x)))$
$(n_A \cdot n_B)$	$\pi_2(D_{k_{AS}}(\pi_1(x)))$
B	$\pi_1(\pi_1(D_{k_{AS}}(\pi_1(x))))$
k_{AB}	$\pi_2(\pi_1(D_{k_{AS}}(\pi_1(x))))$
n_A	$\pi_1(\pi_2(D_{k_{AS}}(\pi_1(x))))$
n_B	$\pi_2(\pi_2(D_{k_{AS}}(\pi_1(x))))$

Computing the formula

$$\begin{array}{l}
 A \\
 B \\
 S \\
 k_{AS} \\
 n_A \\
 (\{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \cdot \{(A \cdot k_{AB})\}_{k_{BS}}) \\
 \quad \{((B \cdot k_{AB}) \cdot (n_A \cdot n_B))\}_{k_{AS}} \\
 \quad \quad \{(A \cdot k_{AB})\}_{k_{BS}} \\
 \quad ((B \cdot k_{AB}) \cdot (n_A \cdot n_B)) \\
 \quad (B \cdot k_{AB}) \\
 \quad (n_A \cdot n_B) \\
 B \\
 k_{AB} \\
 n_A \\
 n_B
 \end{array}$$

$$\begin{array}{l}
 A \\
 B \\
 S \\
 k_{AS} \\
 n_A \\
 x \\
 \pi_1(x) \\
 \pi_2(x) \\
 D_{k_{AS}}(\pi_1(x)) \\
 \pi_1(D_{k_{AS}}(\pi_1(x))) \\
 \pi_2(D_{k_{AS}}(\pi_1(x))) \\
 \pi_1(\pi_1(D_{k_{AS}}(\pi_1(x)))) \\
 \pi_2(\pi_1(D_{k_{AS}}(\pi_1(x)))) \\
 \pi_1(\pi_2(D_{k_{AS}}(\pi_1(x)))) \\
 \pi_2(\pi_2(D_{k_{AS}}(\pi_1(x))))
 \end{array}$$

The result of the computation

$$\begin{aligned}
 & [A:\mathbf{M}] \wedge [B:\mathbf{M}] \wedge [S:\mathbf{M}] \wedge [k_{AS}:\mathbf{M}] \wedge [n_A:\mathbf{M}] \wedge [x:\mathbf{M}] \\
 \wedge & [\pi_1(x):\mathbf{M}] \wedge [\pi_2(x):\mathbf{M}] \\
 \wedge & [D_{k_{AS}}(\pi_1(x)):\mathbf{M}] \\
 \wedge & [\pi_1(D_{k_{AS}}(\pi_1(x))):\mathbf{M}] \wedge [\pi_2(D_{k_{AS}}(\pi_1(x))):\mathbf{M}] \\
 \wedge & [\pi_1(\pi_1(D_{k_{AS}}(\pi_1(x)))):\mathbf{M}] \wedge [\pi_2(\pi_1(D_{k_{AS}}(\pi_1(x)))):\mathbf{M}] \\
 \wedge & [\pi_1(\pi_2(D_{k_{AS}}(\pi_1(x)))):\mathbf{M}] \wedge [\pi_2(\pi_2(D_{k_{AS}}(\pi_1(x)))):\mathbf{M}]
 \end{aligned}$$

Computing the formula

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

The result of the computation

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

Computing the formula

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

The result of the computation

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

The formula of an analysed knowledge set

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

Removing redundancies

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

Removing redundancies

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

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

Outline

- 1 Extending protocol narrations
- 2 Compiling protocol narrations
- 3 Executing protocol narrations**
- 4 Rewriting protocol narrations... in spi-calculus

Labelled transition system

$$\text{SEND} \frac{\llbracket E \rrbracket = M \in \mathbf{M}}{A: B!E; X \xrightarrow{A: B!M} X}$$

$$\text{RECEIVE} \frac{}{A: ?x; X \xrightarrow{A: ?M} X\{M/x\}_A} M \in \mathbf{M}$$

$$\text{CHECK} \frac{X \xrightarrow{A: \beta} X'}{A: \phi; X \xrightarrow{A: \beta} X'} \llbracket \phi \rrbracket = \mathbf{true}$$

$$\text{OPEN} \frac{X \xrightarrow{A: (\nu \tilde{n}) B!M} X'}{\nu z; X \xrightarrow{A: (\nu z \tilde{n}) B!M} X'} z \in n(M) \setminus \{\tilde{n}\}$$

$$\text{REARRANGE} \frac{X \cong_{\alpha} X' \quad X' \xrightarrow{A: \beta} X''}{X \xrightarrow{A: \beta} X''}$$

Labelled transition system

$$\text{SEND} \frac{\llbracket E \rrbracket = M \in \mathbf{M}}{A: B!E; X \xrightarrow{A: B!M} X} \quad \text{RECEIVE} \frac{}{A: ?x; X \xrightarrow{A: ?M} X\{M/x\}_A} M \in \mathbf{M}$$

$$\text{CHECK} \frac{X \xrightarrow{A: \beta} X'}{A: \phi; X \xrightarrow{A: \beta} X'} \llbracket \phi \rrbracket = \mathbf{true}$$

$$\text{OPEN} \frac{X \xrightarrow{A: (\nu \tilde{n}) B!M} X'}{\nu z; X \xrightarrow{A: (\nu z \tilde{n}) B!M} X'} z \in n(M) \setminus \{\tilde{n}\}$$

$$\text{REARRANGE} \frac{X \cong_{\alpha} X' \quad X' \xrightarrow{A: \beta} X''}{X \xrightarrow{A: \beta} X''}$$

Labelled transition system

$$\text{SEND} \frac{\llbracket E \rrbracket = M \in \mathbf{M}}{A: B!E; X \xrightarrow{A: B!M} X} \quad \text{RECEIVE} \frac{}{A: ?x; X \xrightarrow{A: ?M} X\{M/x\}_A} M \in \mathbf{M}$$

$$\text{CHECK} \frac{X \xrightarrow{A: \beta} X'}{A: \phi; X \xrightarrow{A: \beta} X'} \llbracket \phi \rrbracket = \mathbf{true}$$

$$\text{OPEN} \frac{X \xrightarrow{A: (\nu \tilde{n}) B!M} X'}{\nu z; X \xrightarrow{A: (\nu z \tilde{n}) B!M} X'} z \in n(M) \setminus \{\tilde{n}\}$$

$$\text{REARRANGE} \frac{X \cong_{\alpha} X' \quad X' \xrightarrow{A: \beta} X''}{X \xrightarrow{A: \beta} X''}$$

Outline

- 1 Extending protocol narrations
- 2 Compiling protocol narrations
- 3 Executing protocol narrations
- 4 Rewriting protocol narrations... in spi-calculus**

Translation of executable narrations in spi

X/A = “projection of X on A ”

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

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

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

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

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

Consistency formula of a knowledge set

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

Adding hashing

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

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

Consistency formula of a knowledge set

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