Security protocols: formal models and verification

Sergiu Bursuc

School of Computer Science,
University of Bristol

Finse Winter School, 7 May 2015

Security protocols: roles and goals

Roles: $P_1, \ldots, P_n$
(e.g. clients, servers, devices, *things*, ...)

Goals:
- Secrecy
- Privacy
- Authentication
- Integrity
- Unlinkability
- ...

Security protocols: building blocks

1. Cryptographic primitives: encryption, signatures, commitments, hash functions, ...
2. Network communication

The attacker

- intrusion: network, computers, servers, etc
- dishonest execution of the protocol
- cryptanalysis

Formal attacks in practice

G. Lowe. **Breaking and fixing the Needham-Schroeder public-key protocol using FDR.** [TACAS 1996]

A. Armando, R. Carbone, L. Compagna, J. Cuellar, and L. Tobarra Abad. **Formal analysis of SAML 2.0 web browser single sign-on: Breaking the SAML-based single sign-on for google apps.** [FMSE 2008]

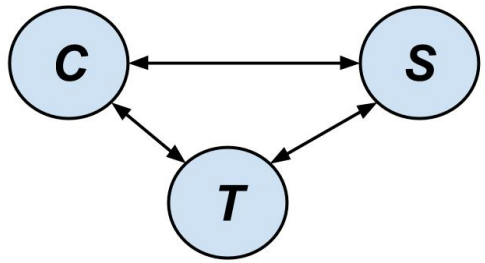
M. Bortolozzo, M. Centenaro, R. Focardi, and G. Steel. **Attacking and fixing PKCS11 security tokens.** [ACM CCS 2010]

D. Basin, C. Cremers, and S. Meier. **Provably repairing the ISO/IEC 9798 standard for entity authentication.** [POST 2012]

Plan

1. Protocols and attacks
2. Formal specification language
3. Case studies and verification

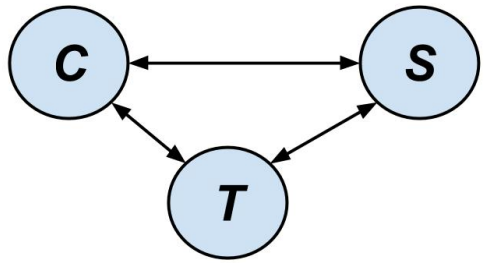
Needham-Schroeder symmetric key



Roles: C - client; S - server; T - third party

Goals: establish secret Kcs, authorise C, and authenticate S

Needham-Schroeder symmetric key



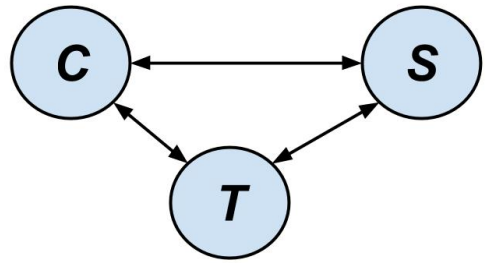
Roles: C - client; S - server; T - third party

Goals: establish secret Kcs, authorise C, and authenticate S

Keys: Kct (shared by C and T)
Kst (shared by S and T)

Nonces: Nc, Ns

Needham-Schroeder symmetric key



Roles: C - client; S - server; T - third party

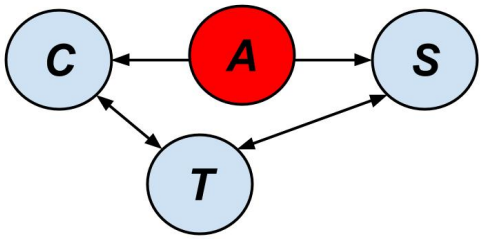
Goals: establish secret Kcs, authorise C, and authenticate S

Keys: Kct (shared by C and T)
Kst (shared by S and T)

Nonces: Nc, Ns

1. $C \rightarrow T \quad : \quad C, S, Nc$
2. $T \rightarrow C \quad : \quad \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S \quad : \quad \{Kcs, C\}_{Kst}$
4. $S \rightarrow C \quad : \quad \{Ns\}_{Kcs}$
5. $C \rightarrow S \quad : \quad \{inc(Ns)\}_{Kcs}$

Needham-Schroeder symmetric key



Roles: C - client; S - server; T - third party

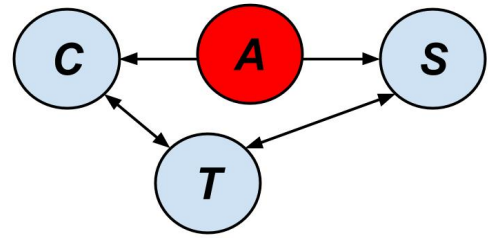
Goals: establish secret Kcs, authorise C, and authenticate S

Keys: Kct (shared by C and T)
Kst (shared by S and T)

Nonces: Nc, Ns

1. $C \rightarrow T \quad : \quad C, S, Nc$
2. $T \rightarrow C \quad : \quad \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S \quad : \quad \{Kcs, C\}_{Kst}$
4. $S \rightarrow C \quad : \quad \{Ns\}_{Kcs}$
5. $C \rightarrow S \quad : \quad \{inc(Ns)\}_{Kcs}$

Attack



Roles: C - client; S - server; T - third party

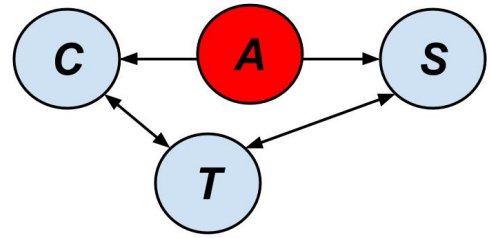
Goals: establish secret Kcs, authorise C, and authenticate S

Keys: Kct (shared by C and T)
Kst (shared by S and T)

Nonces: Nc, Ns

$$
\begin{array}{lllll}
1. & C \rightarrow A & : & C, S, Nc \\
1'. & A \rightarrow T & : & C, A, Nc \\
2'. & T \rightarrow A & : & \{Nc, Kca, \{Kca, C\}_{Kat}\}_{Kct} \\
2. & A \rightarrow C & : & \{Nc, Kca, \{Kca, C\}_{Kat}\}_{Kct} \\
3. & C \rightarrow A & : & \{Kca, C\}_{Kat} \\
4. & A \rightarrow C & : & \{Ns\}_{Kca} \\
5. & C \rightarrow A & : & \{inc(Ns)\}_{Kca}
\end{array}
$$

Needham-Schroeder symmetric key (v1)



Roles: C - client; S - server; T - third party

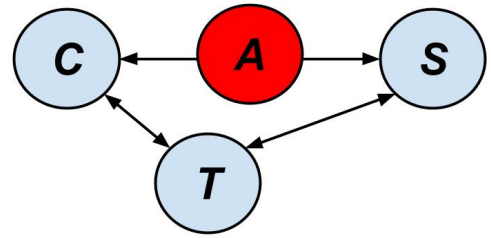
Goals: establish secret Kcs, authorise C, and authenticate S

Keys: Kct (shared by C and T)
Kst (shared by S and T)

Nonces: Nc, Ns

1. $C \rightarrow T$: C, S, Nc
2. $T \rightarrow C$: $\{Nc, S, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S$: $\{Kcs, C\}_{Kst}$
4. $S \rightarrow C$: $\{Ns\}_{Kcs}$
5. $C \rightarrow S$: $\{inc(Ns)\}_{Kcs}$

Needham-Schroeder symmetric key (v1)



Roles: C - client; S - server; T - third party

Goals: establish secret Kcs, authorise C, and authenticate S

Keys: Kct (shared by C and T)
Kst (shared by S and T)

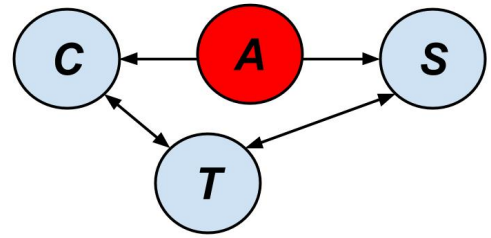
Nonces: Nc, Ns

1. $C \rightarrow T$: C, S, Nc
2. $T \rightarrow C$: $\{Nc, S, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S$: $\{Kcs, C\}_{Kst}$
4. $S \rightarrow C$: $\{Ns\}_{Kcs}$
5. $C \rightarrow S$: $\{inc(Ns)\}_{Kcs}$

4′. $S \rightarrow C$: $\{Ns'\}_{Kcs}$

5′. $C \rightarrow S$: $\{inc(Ns')\}_{Kcs}$

Attack 2



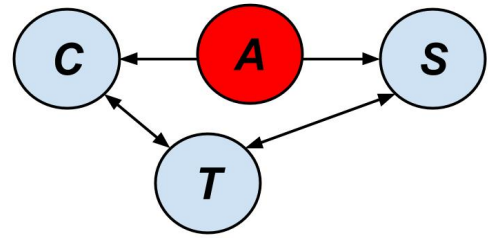
Roles: C - client; S - server; T - third party
Goals: establish secret Kcs, authorise C, and authenticate S
Keys: Kct (shared by C and T)
Kst (shared by S and T)
Nonces: Nc, Ns

1. $C \rightarrow T : C, S, Nc$
2. $T \rightarrow C : \{Nc, S, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S : \{Kcs, C\}_{Kst}$
4. $S \rightarrow C : \{Ns\}_{Kcs}$
5. $C \rightarrow S : \{inc(Ns)\}_{Kcs}$

$4'$. $S \rightarrow C : \{Ns'\}_{Kcs}$

$5'$. $C \rightarrow S : \{inc(Ns')\}_{Kcs}$

Attack 2

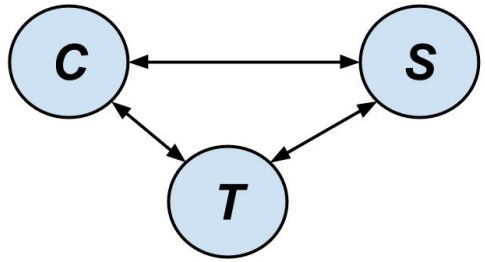


Roles: C - client; S - server; T - third party
Goals: establish secret Kcs, authorise C, and authenticate S
Keys: Kct (shared by C and T)
Kst (shared by S and T)
Nonces: Nc, Ns

1. $C \rightarrow T$: C, S, Nc
2. $T \rightarrow C$: $\{Nc, S, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S$: $\{Kcs, C\}_{Kst}$
4. $S \rightarrow C$: $\{Ns\}_{Kcs}$
5. $C \rightarrow S$: $\{inc(Ns)\}_{Kcs}$

4′. $A \rightarrow C$: $\{inc(Ns)\}_{Kcs}$
5′. $C \rightarrow A$: $\{inc(inc(Ns))\}_{Kcs}$

Needham-Schroeder symmetric key (v2)



Roles: C - client; S - server; T - third party

Goals: establish secret Kcs, authorise C, and authenticate S

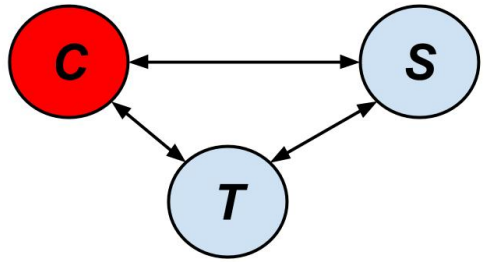
Keys: Kct (shared by C and T)

Kst (shared by S and T)

Nonces: Nc, Ns

1. $C \rightarrow T$: C, S, Nc
2. $T \rightarrow C$: $\{Nc, S, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S$: $\{Kcs, C\}_{Kst}$
4. $S \rightarrow C$: $\{S, Ns\}_{Kcs}$
5. $C \rightarrow S$: $\{C, inc(Ns)\}_{Kcs}$

Needham-Schroeder symmetric key (v2)



Roles: C - client; S - server; T - third party

Goals: establish secret Kcs, authorise C, and authenticate S

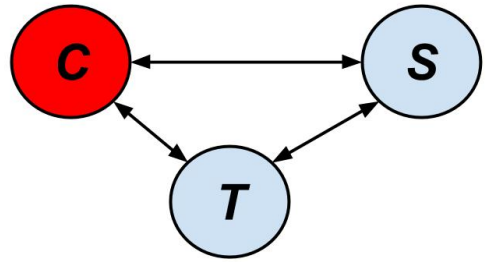
Keys: Kct (shared by C and T)

Kst (shared by S and T)

Nonces: Nc, Ns

1. $C \rightarrow T$: C, S, Nc
2. $T \rightarrow C$: $\{Nc, S, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S$: $\{Kcs, C\}_{Kst}$
4. $S \rightarrow C$: $\{S, Ns\}_{Kcs}$
5. $C \rightarrow S$: $\{C, inc(Ns)\}_{Kcs}$

Attack 3



Roles: C - client; S - server; T - third party

Goals: establish secret Kcs, authorise C, and authenticate S

Keys: Kct (shared by C and T)

Kst (shared by S and T)

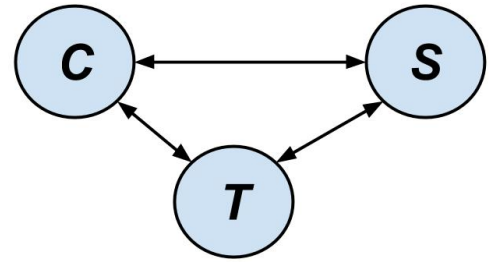
Nonces: Nc, Ns

1. $C \rightarrow T$: C, S, Nc
2. $T \rightarrow C$: $\{Nc, S, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S$: $\{Kcs, C\}_{Kst}$
4. $S \rightarrow C$: $\{S, Ns\}_{Kcs}$
5. $C \rightarrow S$: $\{C, inc(Ns)\}_{Kcs}$

...

3. $C \rightarrow S$: $\{Kcs, C\}_{Kst}$
4. $S \rightarrow C$: $\{S, Ns'\}_{Kcs}$
5. $C \rightarrow S$: $\{C, inc(Ns')\}_{Kcs}$

Needham-Schroeder symmetric key (v3)



Roles: C - client; S - server; T - third party

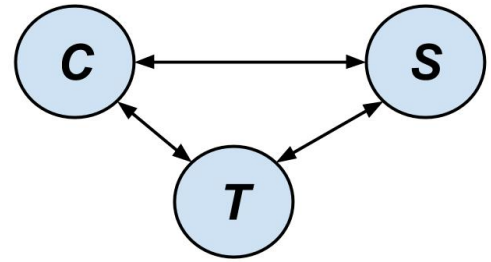
Goals: establish secret Kcs, authorise C, and authenticate S

Keys: Kct (shared by C and T)
Kst (shared by S and T)

Nonces: Nc, Ns

1. $C \rightarrow S \quad : \quad C$
2. $S \rightarrow C \quad : \quad \{C, Ns\}_{Kbs}$
3. $C \rightarrow T \quad : \quad C, S, Nc, \{C, Ns\}_{Kbs}$
4. $T \rightarrow C \quad : \quad \{Nc, S, Kcs, \{Kcs, Ns, C\}_{Kst}\}_{Kct}$
5. $C \rightarrow S \quad : \quad \{Kcs, Ns, C\}_{Kst}$
6. $S \rightarrow C \quad : \quad \{S, Ns\}_{Kcs}$
7. $C \rightarrow S \quad : \quad \{C, inc(Ns)\}_{Kcs}$

Needham-Schroeder symmetric key (v3)



Roles: C - client; S - server; T - third party

Goals: establish secret Kcs, authorise C, and authenticate S

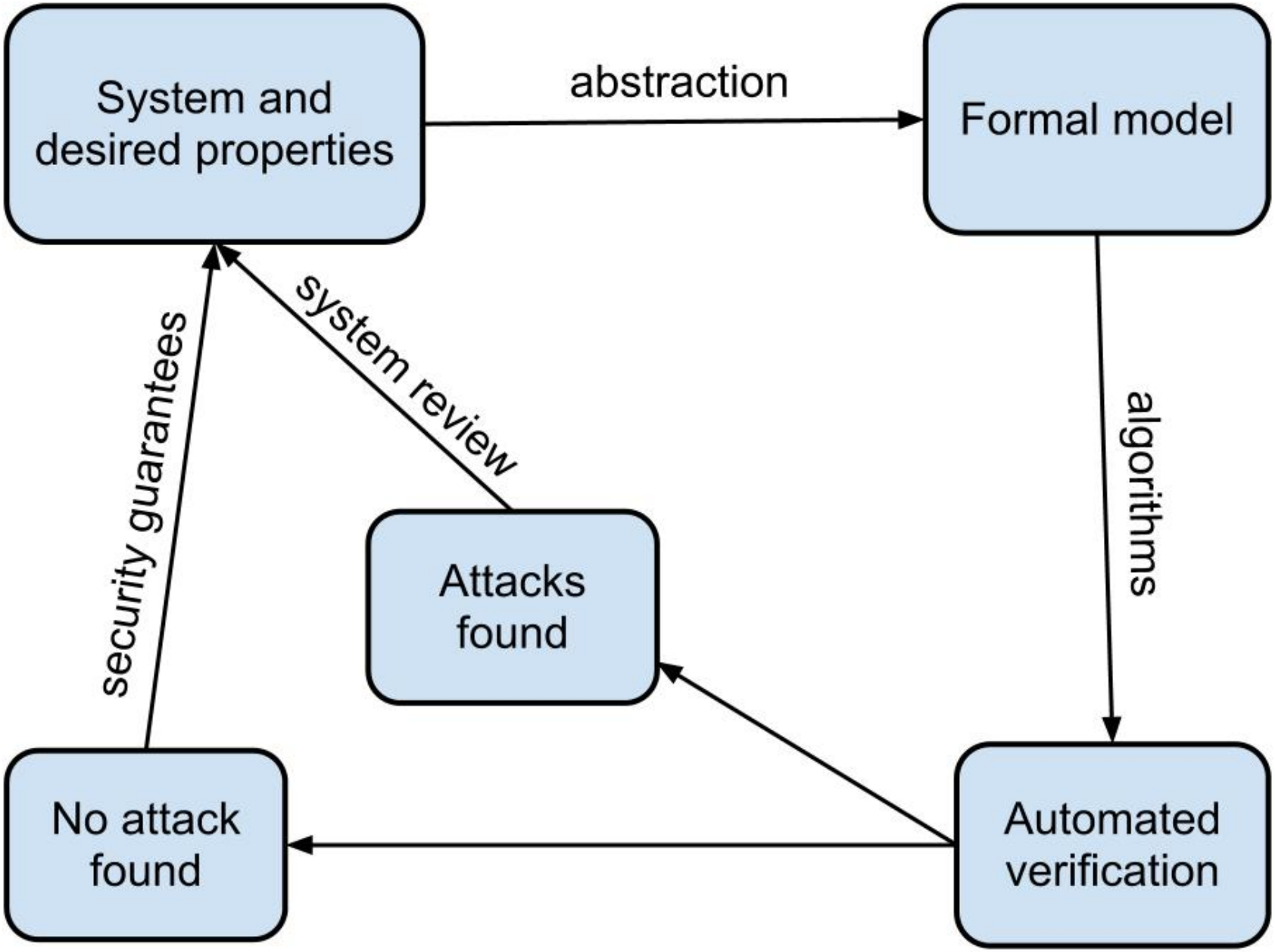
Keys: Kct (shared by C and T)
Kst (shared by S and T)

Nonces: Nc, Ns

1. $C \rightarrow S \quad : \quad C$
2. $S \rightarrow C \quad : \quad \{C, Ns\}_{Kbs}$
3. $C \rightarrow T \quad : \quad C, S, Nc, \{C, Ns\}_{Kbs}$
4. $T \rightarrow C \quad : \quad \{Nc, S, Kcs, \{Kcs, Ns, C\}_{Kst}\}_{Kct}$
5. $C \rightarrow S \quad : \quad \{Kcs, Ns, C\}_{Kst}$
6. $S \rightarrow C \quad : \quad \{S, Ns\}_{Kcs}$
7. $C \rightarrow S \quad : \quad \{C, inc(Ns)\}_{Kcs}$

Notes: i) compromised T; ii) Kerberos

Formal verification approach



Formal verification

---

system $\mathcal{S}$
environment $\mathcal{E}$
properties $\mathcal{P}$

does $\mathcal{S}$ satisfy $\mathcal{P}$ in $\mathcal{E}$?

Formal verification

Formalization

system $\mathcal{S}$	$\Rightarrow$	$\mathcal{M}(\mathcal{S})$
environment $\mathcal{E}$	$\Rightarrow$	$\mathcal{M}(\mathcal{E})$
properties $\mathcal{P}$	$\Rightarrow$	$\mathcal{M}(\mathcal{P})$
does $\mathcal{S}$ satisfy $\mathcal{P}$ in $\mathcal{E}$?	$\Rightarrow$	$\mathcal{M}(\mathcal{S}) \models_{\mathcal{M}(\mathcal{E})} \mathcal{M}(\mathcal{P})$?

Formal verification

Formalization		
system $\mathcal{S}$	$\Rightarrow$	$\mathcal{M}(\mathcal{S})$
environment $\mathcal{E}$	$\Rightarrow$	$\mathcal{M}(\mathcal{E})$
properties $\mathcal{P}$	$\Rightarrow$	$\mathcal{M}(\mathcal{P})$
does $\mathcal{S}$ satisfy $\mathcal{P}$ in $\mathcal{E}$?	$\Rightarrow$	$\mathcal{M}(\mathcal{S}) \models_{\mathcal{M}(\mathcal{E})} \mathcal{M}(\mathcal{P})$?

Verification

Formal model

- Messages as *terms*
- Roles as *processes*
- Security properties as *logical formulas*

Messages as terms

Term algebra $\mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

$$\begin{aligned} \mathcal{N} &= a, b, c, k_1, k_2, \ldots \\ \mathcal{X} &= x, y, z, \ldots \\ \mathcal{F} &= f_1, \ldots, f_k \end{aligned}$$

Messages as terms

Term algebra $\mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

$$\begin{aligned} \mathcal{N} &= a, b, c, k_1, k_2, \ldots \\ \mathcal{X} &= x, y, z, \ldots \\ \mathcal{F} &= f_1, \ldots, f_k \end{aligned}$$

- $\mathcal{N} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $\mathcal{X} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$ and $f \in \mathcal{F}$
  $\Longrightarrow f(t_1, \ldots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

Examples: $\mathsf{enc}(a, k)$,

Messages as terms

Term algebra $\mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

$$\begin{aligned} \mathcal{N} &= a, b, c, k_1, k_2, \ldots \\ \mathcal{X} &= x, y, z, \ldots \\ \mathcal{F} &= f_1, \ldots, f_k \end{aligned}$$

- $\mathcal{N} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $\mathcal{X} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$ and $f \in \mathcal{F}$
  $\Longrightarrow f(t_1, \ldots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

Examples: $\mathsf{enc}(a, k)$, $\mathsf{enc}(x, k)$,

Messages as terms

Term algebra $\mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

$$\begin{aligned} \mathcal{N} &= a, b, c, k_1, k_2, \ldots \\ \mathcal{X} &= x, y, z, \ldots \\ \mathcal{F} &= f_1, \ldots, f_k \end{aligned}$$

- $\mathcal{N} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $\mathcal{X} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$ and $f \in \mathcal{F}$
  $\Longrightarrow f(t_1, \ldots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

Examples: $\mathsf{enc}(a, k)$, $\mathsf{enc}(x, k)$, $\mathsf{enc}(\mathsf{enc}(x, k_1), k_2)$,

Messages as terms

Term algebra $\mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

$$\begin{aligned} \mathcal{N} &= a, b, c, k_1, k_2, \ldots \\ \mathcal{X} &= x, y, z, \ldots \\ \mathcal{F} &= f_1, \ldots, f_k \end{aligned}$$

- $\mathcal{N} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $\mathcal{X} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$ and $f \in \mathcal{F}$
  $\Longrightarrow f(t_1, \ldots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

Examples: $\mathrm{enc}(a, k)$, $\mathrm{enc}(x, k)$, $\mathrm{enc}(\mathrm{enc}(x, k_1), k_2)$, $\mathrm{dec}(x, k)$,

Messages as terms

Term algebra $\mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

$$\begin{aligned} \mathcal{N} &= a, b, c, k_1, k_2, \ldots \\ \mathcal{X} &= x, y, z, \ldots \\ \mathcal{F} &= f_1, \ldots, f_k \end{aligned}$$

- $\mathcal{N} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $\mathcal{X} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$ and $f \in \mathcal{F}$
  $\Longrightarrow f(t_1, \ldots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

Examples: $\mathsf{enc}(a, k)$, $\mathsf{enc}(x, k)$, $\mathsf{enc}(\mathsf{enc}(x, k_1), k_2)$, $\mathsf{dec}(x, k)$,
Equational theory: $u_1 = v_1, \ldots, u_n = v_n$ Example:
$\mathsf{dec}(\mathsf{enc}(x, y), y) = x$

Messages as terms

Term algebra $\mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

$$\begin{aligned} \mathcal{N} &= a, b, c, k_1, k_2, \ldots \\ \mathcal{X} &= x, y, z, \ldots \\ \mathcal{F} &= f_1, \ldots, f_k \end{aligned}$$

- $\mathcal{N} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $\mathcal{X} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$
- $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$ and $f \in \mathcal{F}$
  $\Longrightarrow f(t_1, \ldots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$

Examples: $\mathrm{enc}(a, k)$, $\mathrm{enc}(x, k)$, $\mathrm{enc}(\mathrm{enc}(x, k_1), k_2)$, $\mathrm{dec}(x, k)$,
Equational theory: $u_1 = v_1, \ldots, u_n = v_n$ Example:
$\mathrm{dec}(\mathrm{enc}(x, y), y) = x$ Note: both augments and restricts attacker's power

Equational theories

Symmetric key encryption:

$$\mathsf{dec}(\mathsf{enc}(x, y), y) = x$$

Equational theories

Symmetric key encryption:

$$\mathsf{dec}(\mathsf{enc}(x, y), y) = x$$

Public key encryption:

$$\mathsf{dec}(\mathsf{enc}(x, \mathsf{pub}(y)), y) = x$$

Equational theories

Symmetric key encryption:

$$\mathsf{dec}(\mathsf{enc}(x, y), y) = x$$

Public key encryption:

$$\mathsf{dec}(\mathsf{enc}(x, \mathsf{pub}(y)), y) = x$$

Signatures:

$$\begin{aligned} \mathsf{check}(\mathsf{sign}(x, y), \mathsf{pub}(y)) &= \mathsf{ok} \\ \mathsf{get}(\mathsf{sign}(x, y)) &= x \end{aligned}$$

Equational theories

Symmetric key encryption:

$$\mathsf{dec}(\mathsf{enc}(x, y), y) = x$$

Public key encryption:

$$\mathsf{dec}(\mathsf{enc}(x, \mathsf{pub}(y)), y) = x$$

Signatures:

$$\begin{aligned} \mathsf{check}(\mathsf{sign}(x, y), \mathsf{pub}(y)) &= \mathsf{ok} \\ \mathsf{get}(\mathsf{sign}(x, y)) &= x \end{aligned}$$

Blind signatures:

$$\begin{aligned} \mathsf{check}(\mathsf{sign}(x, y), \mathsf{pub}(y)) &= \mathsf{ok} \\ \mathsf{get}(\mathsf{sign}(x, y)) &= x \\ \mathsf{unblind}(\mathsf{sign}(\mathsf{blind}(x, y), z), y) &= \mathsf{sign}(x, z) \\ \mathsf{unblind}(\mathsf{blind}(x, y), y) &= x \end{aligned}$$

Equational theories

Modular exponentiation:

$$\exp(\exp(x, y), z) = \exp(\exp(x, z), y)$$

Equational theories

Modular exponentiation:

$$\exp(\exp(x, y), z) = \exp(\exp(x, z), y)$$

Re-randomizable encryption:

$$\begin{aligned} \mathrm{dec}(\mathrm{enc}(x, \mathrm{pub}(y), z), y) &= x \\ \mathrm{renc}(\mathrm{enc}(x, y, z), z') &= \mathrm{enc}(x, y, f(z, z')) \end{aligned}$$

Equational theories

Modular exponentiation:

$$\exp(\exp(x, y), z) = \exp(\exp(x, z), y)$$

Re-randomizable encryption:

$$\begin{aligned} \mathrm{dec}(\mathrm{enc}(x, \mathrm{pub}(y), z), y) &= x \\ \mathrm{renc}(\mathrm{enc}(x, y, z), z') &= \mathrm{enc}(x, y, f(z, z')) \end{aligned}$$

Homomorphic encryption:

$$\begin{aligned} \mathrm{dec}(\mathrm{enc}(x, \mathrm{pub}(y), z), y) &= x \\ \mathrm{enc}(x_1, y, z_1) \star \mathrm{enc}(x_2, y, z_2) &= \mathrm{enc}(x_1 + x_2, y, z_1 \star z_2) \end{aligned}$$

Intruder deduction: $T \vdash t$

$$\frac{T \vdash t_1 \quad \ldots \quad T \vdash t_k}{T \vdash f(t_1, \ldots, t_k)} \qquad \frac{T \vdash u}{T \vdash v} \quad \text{if} \quad u =_{\mathcal{E}} v$$

Intruder deduction: $T \vdash t$

$$\frac{T \vdash t_1 \quad \ldots \quad T \vdash t_k}{T \vdash f(t_1, \ldots, t_k)} \qquad \frac{T \vdash u}{T \vdash v} \quad \text{if} \quad u =_{\mathcal{E}} v$$

$\mathsf{enc}(s, k_1), \mathsf{enc}(k_1, k_2), \mathsf{sign}(k_2, k_3) \vdash s?$

Intruder deduction: $T \vdash t$

$$\frac{T \vdash t_1 \quad \ldots \quad T \vdash t_k}{T \vdash f(t_1, \ldots, t_k)} \qquad \frac{T \vdash u}{T \vdash v} \quad \text{if} \quad u =_{\mathcal{E}} v$$

$\mathsf{enc}(s, k_1), \mathsf{enc}(k_1, k_2), \mathsf{sign}(k_2, k_3) \vdash s?$
$\mathsf{enc}(s, \mathsf{enc}(s, k_1)), \mathsf{enc}(\mathsf{enc}(s, k_1), \mathsf{sign}(k_1, k_2)), k_1, k_2 \vdash s?$

Intruder deduction: $T \vdash t$

$$\frac{T \vdash t_1 \quad \ldots \quad T \vdash t_k}{T \vdash f(t_1, \ldots, t_k)} \qquad \frac{T \vdash u}{T \vdash v} \quad \text{if} \quad u =_{\mathcal{E}} v$$

$\mathsf{enc}(s, k_1), \mathsf{enc}(k_1, k_2), \mathsf{sign}(k_2, k_3) \vdash s?$
$\mathsf{enc}(s, \mathsf{enc}(s, k_1)), \mathsf{enc}(\mathsf{enc}(s, k_1), \mathsf{sign}(k_1, k_2)), k_1, k_2 \vdash s?$
$\mathsf{enc}(s, \mathsf{enc}(s, k_1)), \mathsf{enc}(\mathsf{enc}(s, k_1), \mathsf{sign}(k_1, k_2)), k_1, k_2' \vdash s?$

Intruder deduction and passive security

Intruder knowledge: $t_1, \ldots, t_n$

Intruder power: $\mathcal{E}$

Security question: $t_1, \ldots, t_n \vdash_{\mathcal{E}} s$?

Intruder deduction and passive security

Intruder knowledge: $t_1, \ldots, t_n$

Intruder power: $\mathcal{E}$

Security question: $t_1, \ldots, t_n \vdash_{\mathcal{E}} s$?

1. $C \to T \quad : \quad C, S, Nc$
2. $T \to C \quad : \quad \mathrm{enc}(\langle Nc, S, Kcs, \mathrm{enc}(\langle Kcs, C\rangle, Kst)\rangle, Kct)$
3. $C \to S \quad : \quad \mathrm{enc}(\langle Kcs, C\rangle, Kst)$
4. $S \to C \quad : \quad \mathrm{enc}(Nb, Kcs)$
5. $C \to S \quad : \quad \mathrm{enc}(inc(Nb), Kcs)$

Intruder knowledge (after 2 sessions): $C_1, C_2, S, Nc_1, Nc_2,$
$\mathrm{enc}(\langle Nc_1, S, Kc_1s, \mathrm{enc}(\langle Kc_1s, C_1\rangle, Kst)\rangle, Kc_1t),$
$\mathrm{enc}(\langle Nc_2, S, Kc_2s, \mathrm{enc}(\langle Kc_2s, C_2\rangle, Kst)\rangle, Kc_2t),$
$\mathrm{enc}(\langle Kc_1s, C_1\rangle, Kst), \mathrm{enc}(\langle Kc_2s, C_2\rangle, Kst),$
$\mathrm{enc}(Nb_1, Kc_1s), \mathrm{enc}(Nb_2, Kc_2s),$
$\mathrm{enc}(inc(Nb_1), Kc_1s), \mathrm{enc}(inc(Nb_2), Kc_2s)$
Security question: does the intruder know Kc_1s or Kc_2s ?

Formal verification

Formalization		
system $\mathcal{S}$	$\Rightarrow$	$\mathcal{M}(\mathcal{S})$
environment $\mathcal{E}$	$\Rightarrow$	$\mathcal{M}(\mathcal{E})$
properties $\mathcal{P}$	$\Rightarrow$	$\mathcal{M}(\mathcal{P})$
does $\mathcal{S}$ satisfy $\mathcal{P}$ in $\mathcal{E}$?	$\Rightarrow$	$\mathcal{M}(\mathcal{S}) \models_{\mathcal{M}(\mathcal{E})} \mathcal{M}(\mathcal{P})$?

} Verification

- Messages as *terms*
- Roles as *processes*
- Security properties as *logical formulas*

Process algebra: [Abadi, Fournet 2001] and [Blanchet 2001]

$$\begin{array}{ll} \text{new } n; P & \text{let } x = u \text{ in } P \\ \text{in}(c, u); P & \text{out}(c, u); P \\ P \mid Q & !P \\ \text{if } u = v \text{ then } P \text{ else } Q & \end{array}$$

Process algebra: [Abadi, Fournet 2001] and [Blanchet 2001]

$\text{new } n; P$ $\quad$ $\text{let } x = u \text{ in } P$

$\text{in}(c, u); P$ $\quad$ $\text{out}(c, u); P$

$P \mid Q$ $\quad$ $!P$

$\text{if } u = v \text{ then } P \text{ else } Q$

$\text{new } k;$

$\text{out}(c, \mathsf{pub}(k)); \text{in}(c, x);$

$\text{let } y = \mathsf{dec}(x, k) \text{ in}$

$\text{out}(c, y)$

Process algebra: [Abadi, Fournet 2001] and [Blanchet 2001]

$$\begin{array}{lll} \text{new } n; P & \text{let } x = u \text{ in } P & \text{new } k; \\ \text{in}(c, u); P & \text{out}(c, u); P & \text{out}(c, \mathsf{pub}(k)); \text{in}(c, x); \\ P \mid Q & !P & \text{let } y = \mathsf{dec}(x, k) \text{ in} \\ \text{if } u = v \text{ then } P \text{ else } Q & & \text{out}(c, y) \end{array}$$

Security : $P \models \texttt{att}:k?$

Process algebra: [Abadi, Fournet 2001] and [Blanchet 2001]

$\text{new } n; P$ $\quad$ $\text{let } x = u \text{ in } P$

$\text{in}(c, u); P$ $\quad$ $\text{out}(c, u); P$

$P \mid Q$ $\quad$ $!P$

$\text{if } u = v \text{ then } P \text{ else } Q$

$\text{new } k; \text{new } s; \text{out}(c, \text{enc}(s, \text{pub}(k)))$

$\text{out}(c, \text{pub}(k)); \text{in}(c, x);$

$\text{let } y = \text{dec}(x, k) \text{ in}$

$\text{out}(c, y)$

Security : $\quad P \models \texttt{att:}k?$

$P \models \texttt{att:}s?$

Process algebra: [Abadi, Fournet 2001] and [Blanchet 2001]

$\text{new } n; P$	$\text{let } x = u \text{ in } P$	$\text{new } k; \text{new } s; \text{out}(c, \text{enc}(s, \text{pub}(k)))$
$\text{in}(c, u); P$	$\text{out}(c, u); P$	$\text{out}(c, \text{pub}(k)); \text{in}(c, x);$
$P \mid Q$	$!P$	$\text{let } y = \text{dec}(x, k) \text{ in}$
$\text{if } u = v \text{ then } P \text{ else } Q$		$\texttt{event DEC}(y); \text{out}(c, y)$

Security : $P \models \texttt{att:}k?$

$P \models \texttt{att:}s \rightsquigarrow \texttt{event:DEC}(s)$

Process algebra: [Abadi, Fournet 2001] and [Blanchet 2001]

$\text{new } n; P$	$\text{let } x = u \text{ in } P$	$\text{new } k; \text{new } s; \text{out}(c, \text{enc}(s, \text{pub}(k)))$
$\text{in}(c, u); P$	$\text{out}(c, u); P$	$\text{out}(c, \text{pub}(k)); \text{in}(c, x);$
$P \mid Q$	$!P$	$\text{let } y = \text{dec}(x, k) \text{ in}$
$\text{if } u = v \text{ then } P \text{ else } Q$		$\texttt{event DEC}(y); \text{out}(c, y)$

Security : $P \models \texttt{att:}k?$

$P \models \texttt{att:}s \rightsquigarrow \texttt{event:DEC}(s)$

Tools: ProVerif, Avispa, Scyther, Tamarin, etc

Configurations $(\mathcal{N}, \mathcal{M}, \mathcal{P})$

- $\mathcal{N}$ - names representing fresh data in an execution
- $\mathcal{M}$ - terms representing messages sent over the network
- $\mathcal{P}$ - set of processes that are being executed in parallel

Configurations $(\mathcal{N}, \mathcal{M}, \mathcal{P})$

- $\mathcal{N}$ - names representing fresh data in an execution
- $\mathcal{M}$ - terms representing messages sent over the network
- $\mathcal{P}$ - set of processes that are being executed in parallel

$$\begin{aligned}&\text{new } k; \text{new } s; \text{out}(c, \text{enc}(s, \text{pub}(k)))\\ &\text{out}(c, \text{pub}(k)); \text{in}(c, x);\\ &\text{let } y = \text{dec}(x, k) \text{ in out}(c, y)\end{aligned}$$

- $\mathcal{N} = \{k, s\}$
- $\mathcal{M} = \{\text{enc}(s, \text{pub}(k)), \text{pub}(k)\}$
- $\mathcal{P} = \{\text{in}(c, x); \text{let } y = \text{dec}(x, k) \text{ in out}(c, y)\}$

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

(NIL) $\quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{0\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P})$

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

(NIL) $\quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{0\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P})$

(BANG) $\quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{!P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P, !P\})$

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

$$\texttt{(NIL)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathtt{0}\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P})$$

$$\texttt{(BANG)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{!P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P, !P\})$$

$$\texttt{(PAR)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P \mid Q\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P, Q\})$$

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

(NIL) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{0\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P})$

(BANG) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{!P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P, !P\})$

(PAR) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P \mid Q\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P, Q\})$

(NEW) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\text{new } n; P\}) \rightsquigarrow (\mathcal{N} \cup \{n'\}, \mathcal{M}, \mathcal{P} \cup \{P\})$
where $n' \notin \mathcal{N}$

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

(COMM) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{out}(c, t); P\ ,\ \mathrm{in}(c, x); Q\})$
$\rightsquigarrow (\mathcal{N}, \mathcal{M}', \mathcal{P} \cup \{P, Q[x \mapsto t]\})$

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

(COMM) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{out}(c, t); P\ ,\ \mathrm{in}(c, x); Q\})$
$\rightsquigarrow (\mathcal{N}, \mathcal{M}', \mathcal{P} \cup \{P, Q[x \mapsto t]\})$
where $\mathcal{M}' = \mathcal{M} \cup \{t\}$, if $\mathcal{M} \vdash c$,
and $\mathcal{M}' = \mathcal{M}$, otherwise

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

(COMM) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{out}(c, t); P\ ,\ \mathrm{in}(c, x); Q\})$
$\rightsquigarrow (\mathcal{N}, \mathcal{M}', \mathcal{P} \cup \{P, Q[x \mapsto t]\})$
where $\mathcal{M}' = \mathcal{M} \cup \{t\}$, if $\mathcal{M} \vdash c$,
and $\mathcal{M}' = \mathcal{M}$, otherwise

(OUT) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{out}(c, t); P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}', \mathcal{P} \cup \{P\})$
where $\mathcal{M}' = \mathcal{M} \cup \{t\}$, if $\mathcal{M} \vdash c$

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

(COMM) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{out}(c, t); P\ ,\ \mathrm{in}(c, x); Q\})$
$\rightsquigarrow (\mathcal{N}, \mathcal{M}', \mathcal{P} \cup \{P, Q[x \mapsto t]\})$
where $\mathcal{M}' = \mathcal{M} \cup \{t\}$, if $\mathcal{M} \vdash c$,
and $\mathcal{M}' = \mathcal{M}$, otherwise

(OUT) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{out}(c, t); P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}', \mathcal{P} \cup \{P\})$
where $\mathcal{M}' = \mathcal{M} \cup \{t\}$, if $\mathcal{M} \vdash c$

(IN) $(\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{in}(c, x); Q\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{Q[x \mapsto t]\})$
if $\mathcal{M} \vdash c$ and $\mathcal{M} \vdash t$

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

$$(\mathtt{IF_T}) \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\text{if } U = V \text{ then } P \text{ else } Q\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P\}) \quad \text{if } U =_{\mathcal{E}} V$$

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

$$(\mathtt{IF_T}) \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\text{if } U = V \text{ then } P \text{ else } Q\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P\}) \quad \text{if } U =_{\mathcal{E}} V$$

$$(\mathtt{IF_F}) \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\text{if } U = V \text{ then } P \text{ else } Q\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{Q\}) \quad \text{if } U \neq_{\mathcal{E}} V$$

Operational semantics: $(\mathcal{N}, \mathcal{M}, \mathcal{P}) \rightsquigarrow (\mathcal{N}', \mathcal{M}', \mathcal{P}')$

$$(\mathtt{IF_T}) \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\text{if } U = V \text{ then } P \text{ else } Q\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P\}) \quad \text{if } U =_{\mathcal{E}} V$$

$$(\mathtt{IF_F}) \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\text{if } U = V \text{ then } P \text{ else } Q\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{Q\}) \quad \text{if } U \neq_{\mathcal{E}} V$$

$$(\mathtt{LET}) \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\text{let } x = T \text{ in } P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{P[x \mapsto T]\})$$

Needham-Schroeder in applied pi-calculus

1. $C \to T \quad : \quad C, S, Nc$
2. $T \to C \quad : \quad \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \to S \quad : \quad \{Kcs, C\}_{Kst}$
4. $S \to C \quad : \quad \{Ns\}_{Kcs}$
5. $C \to S \quad : \quad \{inc(Ns)\}_{Kcs}$

Client(C, S)

$\text{new } Nc;$
$\text{out}(net, \langle C, S, Nc \rangle);$
$\text{in}(net, x_T);$
$\text{let } \langle = Nc, x_{\text{kcs}}, x_{\text{ciph}} \rangle = \text{dec}(x_T, k(C)) \text{ in}$
$\text{out}(net, x_{ciph});$
$\text{in}(net, x_S);$
$\text{let } x_{Ns} = \text{dec}(x_S, x_{\text{kcs}}) \text{ in}$
$\text{out}(net, \text{enc}(\text{inc}(x_{Ns}), x_{\text{kcs}}))$

Needham-Schroeder in applied pi-calculus

1. $C \to T \quad : \quad C, S, Nc$
2. $T \to C \quad : \quad \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \to S \quad : \quad \{Kcs, C\}_{Kst}$
4. $S \to C \quad : \quad \{Ns\}_{Kcs}$
5. $C \to S \quad : \quad \{inc(Ns)\}_{Kcs}$

Third Party

$\text{in}(net, \langle x_C, x_S, x_{Nc} \rangle);$
$\text{new } k_{CS};$
$\text{let } y_S = \text{enc}(\langle k_{CS}, x_C \rangle, k(x_S)) \text{ in}$
$\text{let } y_C = \text{enc}(\langle x_{Nc}, c, y_S \rangle, k(x_C)) \text{ in}$
$\text{out}(net, y_C)$

Needham-Schroeder in applied pi-calculus

$$
\begin{array}{lrcl}
1. & C \to T & : & C, S, Nc \\
2. & T \to C & : & \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct} \\
3. & C \to S & : & \{Kcs, C\}_{Kst} \\
4. & S \to C & : & \{Ns\}_{Kcs} \\
5. & C \to S & : & \{inc(Ns)\}_{Kcs}
\end{array}
$$

Server(S)

$\mathrm{in}(net, x_{req})$;
$\mathrm{let}\ \langle x_{Kcs}, x_C \rangle = \mathsf{dec}(x_{req}, k(S))\ \mathrm{in}$
$\mathrm{new}\ Ns$;
$\mathrm{out}(net, \mathsf{enc}(Ns, x_{Kcs}))$;
$\mathrm{in}(net, x_{resp})$;
$\mathrm{if}\ \mathsf{inc}(Ns) = \mathsf{dec}(x_{resp}, x_{Kcs})\ \mathrm{then}\ OK$

Formal verification

Formalization		
system $\mathcal{S}$	$\Rightarrow$	$\mathcal{M}(\mathcal{S})$
environment $\mathcal{E}$	$\Rightarrow$	$\mathcal{M}(\mathcal{E})$
properties $\mathcal{P}$	$\Rightarrow$	$\mathcal{M}(\mathcal{P})$
does $\mathcal{S}$ satisfy $\mathcal{P}$ in $\mathcal{E}$?	$\Rightarrow$	$\mathcal{M}(\mathcal{S}) \models_{\mathcal{M}(\mathcal{E})} \mathcal{M}(\mathcal{P})$?

} Verification

- Messages as *terms*
- Roles as *processes*
- Security properties as *logical formulas*

Security properties: secrecy as reachability

$$\underbrace{(\mathcal{N}_0, \mathcal{M}_0, \{P_0\}) \rightsquigarrow^* (\mathcal{N}, \mathcal{M}, \mathcal{P}) \text{ and } \mathcal{M} \vdash t?}$$

Security properties: secrecy as reachability

$$\underbrace{(\mathcal{N}_0, \mathcal{M}_0, \{P_0\}) \rightsquigarrow^* (\mathcal{N}, \mathcal{M}, \mathcal{P}) \text{ and } \mathcal{M} \vdash t?}_{P_0 \models \text{att} : t}$$

Security properties: secrecy as reachability

$$\underbrace{(\mathcal{N}_0, \mathcal{M}_0, \{P_0\}) \rightsquigarrow^* (\mathcal{N}, \mathcal{M}, \mathcal{P}) \text{ and } \mathcal{M} \vdash t?}_{P_0 \models \mathsf{att} : t}$$

$$P_0 = \begin{cases} \mathrm{new}\ k; \mathrm{new}\ s; \mathrm{out}(c, \mathsf{enc}(s, \mathsf{pub}(k))) \\ \mathrm{out}(c, \mathsf{pub}(k)); \mathrm{in}(c, x); \\ \mathrm{let}\ y = \mathsf{dec}(x, k)\ \mathrm{in}\ \mathrm{out}(c, y) \end{cases}$$

$$P_0 \not\models \mathsf{att} : k$$
$$P_0 \models \mathsf{att} : s$$

Security properties: secrecy as reachability

$$\underbrace{(\mathcal{N}_0, \mathcal{M}_0, \{P_0\}) \rightsquigarrow^* (\mathcal{N}, \mathcal{M}, \mathcal{P}) \text{ and } \mathcal{M} \vdash t?}_{P_0 \models \mathsf{att} : t}$$

$$P_0 = \begin{cases} \mathrm{new}\ k; \mathrm{new}\ s; \mathrm{out}(c, \mathsf{enc}(s, \mathsf{pub}(k))) \\ \mathrm{out}(c, \mathsf{pub}(k)); \mathrm{in}(c, x); \\ \mathrm{let}\ y = \mathsf{dec}(x, k)\ \mathrm{in}\ \mathrm{out}(c, y) \end{cases}$$

$$P_0 \not\models \mathsf{att} : k$$
$$P_0 \models \mathsf{att} : s$$

$$(\emptyset, \emptyset, \{P_0\}) \rightsquigarrow^* (\mathcal{N}, \mathcal{M}, \mathcal{P}) \text{ and } \mathcal{M} \vdash s$$

- $\mathcal{N} = \{k, s\}$
- $\mathcal{M} = \{\mathsf{enc}(s, \mathsf{pub}(k)), \mathsf{pub}(k), s\}$
- $\mathcal{P} = \emptyset$

Key secrecy in Needham-Schroeder

1. $C \rightarrow T \quad : \quad C, S, Nc$
2. $T \rightarrow C \quad : \quad \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S \quad : \quad \{Kcs, C\}_{Kst}$
4. $S \rightarrow C \quad : \quad \{Ns\}_{Kcs}$
5. $C \rightarrow S \quad : \quad \{inc(Ns)\}_{Kcs}$

Demo

Security properties: privacy as equivalence

$$Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2)$$

vs

$$Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2)$$

Security properties: privacy as equivalence

$$\text{new } r; !\text{out}(c, \text{enc}(d, \text{pub}(k_A), r)) \mid A \mid S_1 \mid S_2$$

Security properties: privacy as equivalence

$$\mathrm{new}\, r; !\mathrm{out}(c, \mathsf{enc}(d, \mathsf{pub}(k_A), r)) \mid A \mid S_1 \mid S_2$$

$$!\mathrm{new}\, r; \mathrm{out}(c, \mathsf{enc}(d, \mathsf{pub}(k_A), r))) \mid A \mid S_1 \mid S_2$$

Security properties: privacy as equivalence

$$\mathrm{new}\, r; !\mathrm{out}(c, \mathsf{enc}(d, \mathsf{pub}(k_A), r)) \mid A \mid S_1 \mid S_2$$

$$!\mathrm{new}\, r; \mathrm{out}(c, \mathsf{enc}(d, \mathsf{pub}(k_A), r))) \mid A \mid S_1 \mid S_2$$

$$P[d] \sim P[d']$$
$$P[d] \sim \mathcal{I}[d]$$

Examples: electronic voting, weak secrets, bids, reviews, like buttons, etc

Security properties: unlinkability as equivalence

$$\begin{array}{l}\text{new } r_1; \text{new } r_2; \\ \text{out}(c, \text{enc}(s_1, \text{pub}(k_A), r_1)) \mid \\ \text{out}(c, \text{enc}(s_2, \text{pub}(k_A), r_2)) \mid \\ A \mid S_1 \mid S_2\end{array} \quad \text{vs} \quad \begin{array}{l}\text{new } r_1; \text{new } r_2; \\ \text{out}(c, \text{enc}(s_1, \text{pub}(k_A), r_1)) \mid \\ \text{out}(c, \text{enc}(s_1, \text{pub}(k_A), r_2)) \mid \\ \mid A \mid S_1 \mid S_2\end{array}$$

Security properties: unlinkability as equivalence

$$\begin{array}{l}\text{new } r_1; \text{new } r_2; \\ \text{out}(c, \text{enc}(s_1, \text{pub}(k_A), r_1)) \mid \\ \text{out}(c, \text{enc}(s_2, \text{pub}(k_A), r_2)) \mid \\ A \mid S_1 \mid S_2\end{array} \quad \text{vs} \quad \begin{array}{l}\text{new } r_1; \text{new } r_2; \\ \text{out}(c, \text{enc}(s_1, \text{pub}(k_A), r_1)) \mid \\ \text{out}(c, \text{enc}(s_1, \text{pub}(k_A), r_2)) \mid \\ \mid A \mid S_1 \mid S_2\end{array}$$

$$P[s_1] \mid P[s_2] \sim P[s_1] \mid P[s_1]$$

Examples: RFID tags, location, healthcare, etc

Security properties: unlinkability as equivalence

$$\begin{array}{l}\text{new } r_1; \text{new } r_2; \\ \text{out}(c, \text{enc}(s_1, \text{pub}(k_A), r_1)) \mid \\ \text{out}(c, \text{enc}(s_2, \text{pub}(k_A), r_2)) \mid \\ A \mid S_1 \mid S_2\end{array} \quad \text{vs} \quad \begin{array}{l}\text{new } r_1; \text{new } r_2; \\ \text{out}(c, \text{enc}(s_1, \text{pub}(k_A), r_1)) \mid \\ \text{out}(c, \text{enc}(s_1, \text{pub}(k_A), r_2)) \mid \\ \mid A \mid S_1 \mid S_2\end{array}$$

$$P[s_1] \mid P[s_2] \sim P[s_1] \mid P[s_1]$$

Examples: RFID tags, location, healthcare, etc

$$\mathit{Client}(C, S_1) \mid \mathit{Client}(C, S_1) \mid T \mid \mathit{Server}(S_1) \mid \mathit{Server}(S_2)$$

vs

$$\mathit{Client}(C, S_1) \mid \mathit{Client}(C, S_2) \mid T \mid \mathit{Server}(S_1) \mid \mathit{Server}(S_2)$$

Static equivalence

Term context: $C[\epsilon_1, \ldots, \epsilon_n]$ applied to $t_1, \ldots, t_n$ gives $C[t_1, \ldots, t_n]$

Static equivalence

Term context: $C[\epsilon_1, \ldots, \epsilon_n]$ applied to $t_1, \ldots, t_n$ gives $C[t_1, \ldots, t_n]$
Observations:

$$\mathcal{O}(\mathcal{N}, \mathcal{M}) = \{(C_1, C_2) \mid (C_1, C_2) \cap \mathcal{N} = \emptyset \text{ and } C_1[\mathcal{M}] =_{\mathcal{E}} C_2[\mathcal{M}]\}$$

Static equivalence

Term context: $C[\epsilon_1, \ldots, \epsilon_n]$ applied to $t_1, \ldots, t_n$ gives $C[t_1, \ldots, t_n]$
Observations:

$$\mathcal{O}(\mathcal{N}, \mathcal{M}) = \{(C_1, C_2) \mid (C_1, C_2) \cap \mathcal{N} = \emptyset \text{ and } C_1[\mathcal{M}] =_{\mathcal{E}} C_2[\mathcal{M}]\}$$

Static equivalence: $\mathcal{O}(\mathcal{N}_1, \mathcal{M}_1) = \mathcal{O}(\mathcal{N}_2, \mathcal{M}_2)$?

Static equivalence

Term context: $C[\epsilon_1, \ldots, \epsilon_n]$ applied to $t_1, \ldots, t_n$ gives $C[t_1, \ldots, t_n]$
Observations:

$$\mathcal{O}(\mathcal{N}, \mathcal{M}) = \{(C_1, C_2) \mid (C_1, C_2) \cap \mathcal{N} = \emptyset \text{ and } C_1[\mathcal{M}] =_{\mathcal{E}} C_2[\mathcal{M}]\}$$

Static equivalence: $\mathcal{O}(\mathcal{N}_1, \mathcal{M}_1) = \mathcal{O}(\mathcal{N}_2, \mathcal{M}_2)$?

$$\mathcal{M}_1 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_1, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$
$$\mathcal{M}_2 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_2, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$

- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_1, r_2\}$?

Static equivalence

Term context: $C[\epsilon_1, \ldots, \epsilon_n]$ applied to $t_1, \ldots, t_n$ gives $C[t_1, \ldots, t_n]$
Observations:

$$\mathcal{O}(\mathcal{N}, \mathcal{M}) = \{(C_1, C_2) \mid (C_1, C_2) \cap \mathcal{N} = \emptyset \text{ and } C_1[\mathcal{M}] =_\mathcal{E} C_2[\mathcal{M}]\}$$

Static equivalence: $\mathcal{O}(\mathcal{N}_1, \mathcal{M}_1) = \mathcal{O}(\mathcal{N}_2, \mathcal{M}_2)$?

$$\mathcal{M}_1 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_1, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$
$$\mathcal{M}_2 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_2, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$

- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_1, r_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{s_1, s_2\}$?

Static equivalence

Term context: $C[\epsilon_1, \ldots, \epsilon_n]$ applied to $t_1, \ldots, t_n$ gives $C[t_1, \ldots, t_n]$
Observations:

$$\mathcal{O}(\mathcal{N}, \mathcal{M}) = \{(C_1, C_2) \mid (C_1, C_2) \cap \mathcal{N} = \emptyset \text{ and } C_1[\mathcal{M}] =_\mathcal{E} C_2[\mathcal{M}]\}$$

Static equivalence: $\mathcal{O}(\mathcal{N}_1, \mathcal{M}_1) = \mathcal{O}(\mathcal{N}_2, \mathcal{M}_2)$?

$$\mathcal{M}_1 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_1, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$
$$\mathcal{M}_2 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_2, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$

- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_1, r_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{s_1, s_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_2\}$?

Static equivalence

Term context: $C[\epsilon_1, \ldots, \epsilon_n]$ applied to $t_1, \ldots, t_n$ gives $C[t_1, \ldots, t_n]$
Observations:

$$\mathcal{O}(\mathcal{N}, \mathcal{M}) = \{(C_1, C_2) \mid (C_1, C_2) \cap \mathcal{N} = \emptyset \text{ and } C_1[\mathcal{M}] =_\mathcal{E} C_2[\mathcal{M}]\}$$

Static equivalence: $\mathcal{O}(\mathcal{N}_1, \mathcal{M}_1) = \mathcal{O}(\mathcal{N}_2, \mathcal{M}_2)$?

$$\mathcal{M}_1 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_1, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$
$$\mathcal{M}_2 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_2, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$

- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_1, r_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{s_1, s_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_1\}$?

Static equivalence

Term context: $C[\epsilon_1, \ldots, \epsilon_n]$ applied to $t_1, \ldots, t_n$ gives $C[t_1, \ldots, t_n]$
Observations:

$$\mathcal{O}(\mathcal{N}, \mathcal{M}) = \{(C_1, C_2) \mid (C_1, C_2) \cap \mathcal{N} = \emptyset \text{ and } C_1[\mathcal{M}] =_{\mathcal{E}} C_2[\mathcal{M}]\}$$

Static equivalence: $\mathcal{O}(\mathcal{N}_1, \mathcal{M}_1) = \mathcal{O}(\mathcal{N}_2, \mathcal{M}_2)$?

$$\mathcal{M}_1 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_1, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$
$$\mathcal{M}_2 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_2, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$

- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_1, r_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{s_1, s_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_1\}$? $C_1 = enc(s_1, \epsilon_2, r_2)$ and $C_2 = \epsilon_3$
- $\mathcal{N}_1 = \mathcal{N}_2 = \{s_2\}$?

Static equivalence

Term context: $C[\epsilon_1, \ldots, \epsilon_n]$ applied to $t_1, \ldots, t_n$ gives $C[t_1, \ldots, t_n]$
Observations:

$$\mathcal{O}(\mathcal{N}, \mathcal{M}) = \{(C_1, C_2) \mid (C_1, C_2) \cap \mathcal{N} = \emptyset \text{ and } C_1[\mathcal{M}] =_{\mathcal{E}} C_2[\mathcal{M}]\}$$

Static equivalence: $\mathcal{O}(\mathcal{N}_1, \mathcal{M}_1) = \mathcal{O}(\mathcal{N}_2, \mathcal{M}_2)$?

$$\mathcal{M}_1 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_1, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$
$$\mathcal{M}_2 = \mathsf{enc}(s_1, \mathsf{pub}(k), r_1), \mathsf{enc}(s_2, \mathsf{pub}(k), r_2), \mathsf{pub}(k)$$

- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_1, r_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{s_1, s_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_2\}$?
- $\mathcal{N}_1 = \mathcal{N}_2 = \{r_1\}$? $C_1 = enc(s_1, \epsilon_2, r_2)$ and $C_2 = \epsilon_3$
- $\mathcal{N}_1 = \mathcal{N}_2 = \{s_2\}$? $C_1 = enc(s_1, \epsilon_2, r_2)$ and $C_2 = \epsilon_3$

Observational equivalence: $P_1 \sim P_2$

$$\texttt{(OUT)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{out}(c, t); P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}', \mathcal{P} \cup \{P\})$$
$$\text{where } \mathcal{M}' = \mathcal{M} \cup \{t\}, \text{ if } \mathcal{M} \vdash c$$

$$\texttt{(IN)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{in}(c, x); Q\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{Q[x \mapsto t]\})$$
$$\text{if } \mathcal{M} \vdash c \text{ and } \mathcal{M} \vdash t$$

Observational equivalence: $P_1 \sim P_2$

$$\texttt{(OUT)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{out}(c, t); P\}) \xrightarrow{\mathrm{out}(c,.)} (\mathcal{N}, \mathcal{M}', \mathcal{P} \cup \{P\})$$
$$\text{where } \mathcal{M}' = \mathcal{M} \cup \{t\}, \text{ if } \mathcal{M} \vdash c$$

$$\texttt{(IN)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{in}(c, x); Q\}) \xrightarrow{\mathrm{in}(c,\mathcal{C})} (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{Q[x \mapsto t]\})$$
$$\text{if } \mathcal{M} \vdash c \text{ and } \mathcal{C}[\mathcal{M}] =_{\mathcal{E}} t$$

Observational equivalence: $P_1 \sim P_2$

$$\texttt{(OUT)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{out}(c, t); P\}) \xrightarrow{\mathrm{out}(c,.)} (\mathcal{N}, \mathcal{M}', \mathcal{P} \cup \{P\})$$
$$\text{where } \mathcal{M}' = \mathcal{M} \cup \{t\}, \text{ if } \mathcal{M} \vdash c$$

$$\texttt{(IN)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{in}(c, x); Q\}) \xrightarrow{\mathrm{in}(c,\mathcal{C})} (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{Q[x \mapsto t]\})$$
$$\text{if } \mathcal{M} \vdash c \text{ and } \mathcal{C}[\mathcal{M}] =_{\mathcal{E}} t$$

Traces: $(\mathcal{N}_0, \mathcal{M}_0, \{P\}) \xrightarrow{\alpha_1 \ldots \alpha_k} (\mathcal{N}, \mathcal{M}, \mathcal{P})$

Observational equivalence: $P_1 \sim P_2$

$$\texttt{(OUT)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{out}(c, t); P\}) \xrightarrow{\mathrm{out}(c,.)} (\mathcal{N}, \mathcal{M}', \mathcal{P} \cup \{P\})$$
$$\text{where } \mathcal{M}' = \mathcal{M} \cup \{t\}, \text{ if } \mathcal{M} \vdash c$$

$$\texttt{(IN)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{\mathrm{in}(c, x); Q\}) \xrightarrow{\mathrm{in}(c,\mathcal{C})} (\mathcal{N}, \mathcal{M}, \mathcal{P} \cup \{Q[x \mapsto t]\})$$
$$\text{if } \mathcal{M} \vdash c \text{ and } \mathcal{C}[\mathcal{M}] =_{\mathcal{E}} t$$

Traces: $(\mathcal{N}_0, \mathcal{M}_0, \{P\}) \xrightarrow{\alpha_1 \ldots \alpha_k} (\mathcal{N}, \mathcal{M}, \mathcal{P})$
Observational equivalence: $P \sim Q$ iff for every trace

$$(\mathcal{N}_0, \mathcal{M}_0, \{P_1\}) \xrightarrow{\alpha_1 \ldots \alpha_k} (\mathcal{N}_1, \mathcal{M}_1, \mathcal{P}_1)$$

there is a trace

$$(\mathcal{N}_0, \mathcal{M}_0, \{P_2\}) \xrightarrow{\alpha_1 \ldots \alpha_k} (\mathcal{N}_2, \mathcal{M}_2, \mathcal{P}_2)$$

such that

$$\mathcal{O}(\mathcal{N}_1, \mathcal{M}_1) = \mathcal{O}(\mathcal{N}_2, \mathcal{M}_2)$$

Privacy and unlinkability with Needham-Schroeder

$$
\begin{array}{lcll}
1. & C \rightarrow T & : & C, S, Nc \\
2. & T \rightarrow C & : & \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct} \\
3. & C \rightarrow S & : & \{Kcs, C\}_{Kst} \\
4. & S \rightarrow C & : & \{Ns\}_{Kcs} \\
5. & C \rightarrow S & : & \{inc(Ns)\}_{Kcs}
\end{array}
$$

$$
\mathcal{P} \left\{ \begin{array}{c} Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\ \text{vs} \\ Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2) \end{array} \right.
$$

$$
\mathcal{U} \left\{ \begin{array}{c} Client(C, S_1) \mid Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\ \text{vs} \\ Client(C, S_1) \mid Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2) \end{array} \right.
$$

Demo

Privacy and unlinkability with Needham-Schroeder

$$
\begin{array}{lll}
1. & C \rightarrow T & : \quad C, \{S\}_{Kct}, Nc \\
2. & T \rightarrow C & : \quad \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct} \\
3. & C \rightarrow S & : \quad \{Kcs, C\}_{Kst} \\
4. & S \rightarrow C & : \quad \{Ns\}_{Kcs} \\
5. & C \rightarrow S & : \quad \{inc(Ns)\}_{Kcs}
\end{array}
$$

$$
\mathcal{P} \left\{ \begin{array}{c}
Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\
\text{vs} \\
Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2)
\end{array} \right.
$$

$$
\mathcal{U} \left\{ \begin{array}{c}
Client(C, S_1) \mid Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\
\text{vs} \\
Client(C, S_1) \mid Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2)
\end{array} \right.
$$

Demo

Privacy and unlinkability with Needham-Schroeder

$$
\begin{array}{lrcl}
1. & C \rightarrow T & : & C, \{S\}_{Kct}, Nc \\
2. & T \rightarrow C & : & \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct} \\
3. & C \rightarrow S & : & \{Kcs, C\}_{Kst} \\
4. & S \rightarrow C & : & \{Ns\}_{Kcs} \\
5. & C \rightarrow S & : & \{inc(Ns)\}_{Kcs}
\end{array}
$$

$$
\mathcal{P} \left\{ \begin{array}{c} Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\ \text{vs} \\ Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2) \end{array} \right.
$$

[which is stronger?]

$$
\mathcal{U} \left\{ \begin{array}{c} Client(C, S_1) \mid Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\ \text{vs} \\ Client(C, S_1) \mid Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2) \end{array} \right.
$$

Demo

Privacy and unlinkability with Needham-Schroeder

$$
\begin{array}{lrcl}
1. & C \rightarrow T & : & C, \{S, Nc\}_{Kct} \\
2. & T \rightarrow C & : & \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct} \\
3. & C \rightarrow S & : & \{Kcs, C\}_{Kst} \\
4. & S \rightarrow C & : & \{Ns\}_{Kcs} \\
5. & C \rightarrow S & : & \{inc(Ns)\}_{Kcs}
\end{array}
$$

$$
\mathcal{P} \left\{ \begin{array}{c} Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\ \text{vs} \\ Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2) \end{array} \right.
$$

[which is stronger?]

$$
\mathcal{U} \left\{ \begin{array}{c} Client(C, S_1) \mid Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\ \text{vs} \\ Client(C, S_1) \mid Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2) \end{array} \right.
$$

Demo

Privacy and unlinkability with Needham-Schroeder

$$
\begin{array}{lrcl}
1. & C \rightarrow T & : & C, \{C, S, Nc\}_{Kct} \\
2. & T \rightarrow C & : & \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct} \\
3. & C \rightarrow S & : & \{Kcs, C\}_{Kst} \\
4. & S \rightarrow C & : & \{Ns\}_{Kcs} \\
5. & C \rightarrow S & : & \{inc(Ns)\}_{Kcs}
\end{array}
$$

$$
\mathcal{P} \left\{ \begin{array}{c} Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\ \text{vs} \\ Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2) \end{array} \right.
$$

[which is stronger?]

$$
\mathcal{U} \left\{ \begin{array}{c} Client(C, S_1) \mid Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\ \text{vs} \\ Client(C, S_1) \mid Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2) \end{array} \right.
$$

Demo

Privacy and unlinkability with Needham-Schroeder

$$
\begin{array}{lrcl}
1. & C \rightarrow T & : & C??, \{C, S, Nc\}_{Kct} \\
2. & T \rightarrow C & : & \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct} \\
3. & C \rightarrow S & : & \{Kcs, C\}_{Kst} \\
4. & S \rightarrow C & : & \{Ns\}_{Kcs} \\
5. & C \rightarrow S & : & \{inc(Ns)\}_{Kcs}
\end{array}
$$

$$
\mathcal{P} \left\{ \begin{array}{c} Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\ \text{vs} \\ Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2) \end{array} \right.
$$

[which is stronger?]

$$
\mathcal{U} \left\{ \begin{array}{c} Client(C, S_1) \mid Client(C, S_1) \mid T \mid Server(S_1) \mid Server(S_2) \\ \text{vs} \\ Client(C, S_1) \mid Client(C, S_2) \mid T \mid Server(S_1) \mid Server(S_2) \end{array} \right.
$$

Demo

Correspondence assertions: informally

Integrity: does the result a party obtains correspond to reality ?
Authorisation: is a party allowed to access a resource ?
Authentication: am I really talking to the expected party ?
Agreement: did P_1 and P_2 agree on the same value ?

Correspondence assertions: events

$\mathrm{new}\ n; P \qquad \mathrm{let}\ x = u\ \mathrm{in}\ P$

$\mathrm{in}(c, u); P \qquad \mathrm{out}(c, u); P$

$P \mid Q \qquad !P$

$\mathrm{if}\ u = v\ \mathrm{then}\ P\ \mathrm{else}\ Q$

$\mathsf{event}\mathcal{E}(u_1, \ldots, u_n) \qquad ; \mathsf{P}$

Correspondence assertions: events

$\mathrm{new}\ n; P \qquad \mathrm{let}\ x = u\ \mathrm{in}\ P$

$\mathrm{in}(c, u); P \qquad \mathrm{out}(c, u); P$

$P \mid Q \qquad !P$

$\mathrm{if}\ u = v\ \mathrm{then}\ P\ \mathrm{else}\ Q$

$\mathsf{event}\mathcal{E}(u_1, \ldots, u_n) \qquad ; \mathrm{P}$

$\mathrm{new}\ k; \mathrm{new}\ s; \mathrm{out}(c, \mathsf{enc}(s, \mathsf{pub}(k)))$

$\mathrm{out}(c, \mathsf{pub}(k)); \mathrm{in}(c, x);$

$\mathrm{let}\ y = \mathsf{dec}(x, k)\ \mathrm{in}$

$\mathsf{eventDEC}(y); \mathrm{out}(c, y)$

Correspondence assertions: formally

$$\texttt{(EV)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P} \cup \{\mathsf{eventE}(t_1, \ldots, t_n); P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{L}', \mathcal{P} \cup \{P\}) \quad \text{where } \mathcal{L}' = \mathcal{L} \cup \mathsf{E}(t_1, \ldots, t_n)$$

Correspondence assertions: formally

$$\texttt{(EV)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P} \cup \{\mathsf{event}\mathsf{E}(t_1, \ldots, t_n); P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{L}', \mathcal{P} \cup \{P\}) \quad \text{where } \mathcal{L}' = \mathcal{L} \cup \mathsf{E}(t_1, \ldots, t_n)$$

Syntax

- Predicates $\rho := \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n) \mid u = v \mid \mathsf{att} : t$
- Formulas $\Phi := \rho \mid \Phi \wedge \Phi \mid \Phi \vee \Phi$
- Assertions: $\Phi_1 \implies \Phi_2$

Correspondence assertions: formally

$$\texttt{(EV)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P} \cup \{\mathsf{event}\mathsf{E}(t_1, \ldots, t_n); P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{L}', \mathcal{P} \cup \{P\}) \quad \text{where } \mathcal{L}' = \mathcal{L} \cup \mathsf{E}(t_1, \ldots, t_n)$$

Syntax

- Predicates $\rho := \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n) \mid u = v \mid \mathsf{att} : t$
- Formulas $\Phi := \rho \mid \Phi \wedge \Phi \mid \Phi \vee \Phi$
- Assertions: $\Phi_1 \implies \Phi_2$

Semantics

- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n)$ when $\mathsf{E}(t_1, \ldots, t_n) \in \mathcal{L}$

Correspondence assertions: formally

$$\texttt{(EV)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P} \cup \{\mathsf{event}\mathsf{E}(t_1, \ldots, t_n); P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{L}', \mathcal{P} \cup \{P\}) \quad \text{where } \mathcal{L}' = \mathcal{L} \cup \mathsf{E}(t_1, \ldots, t_n)$$

Syntax

- Predicates $\rho := \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n) \mid u = v \mid \mathsf{att} : t$
- Formulas $\Phi := \rho \mid \Phi \wedge \Phi \mid \Phi \vee \Phi$
- Assertions: $\Phi_1 \implies \Phi_2$

Semantics

- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n)$ when $\mathsf{E}(t_1, \ldots, t_n) \in \mathcal{L}$
- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models u = v$ when $u =_{\mathcal{E}} v$

Correspondence assertions: formally

$$\texttt{(EV)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P} \cup \{\mathsf{event}\mathsf{E}(t_1, \ldots, t_n); P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{L}', \mathcal{P} \cup \{P\}) \quad \text{where } \mathcal{L}' = \mathcal{L} \cup \mathsf{E}(t_1, \ldots, t_n)$$

Syntax

- Predicates $\rho := \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n) \mid u = v \mid \mathsf{att} : t$
- Formulas $\Phi := \rho \mid \Phi \wedge \Phi \mid \Phi \vee \Phi$
- Assertions: $\Phi_1 \implies \Phi_2$

Semantics

- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n)$ when $\mathsf{E}(t_1, \ldots, t_n) \in \mathcal{L}$
- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models u = v$ when $u =_{\mathcal{E}} v$
- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \mathsf{att} : t$ when $\mathcal{M} \vdash t$

Correspondence assertions: formally

$$\texttt{(EV)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P} \cup \{\mathsf{event}\mathsf{E}(t_1, \ldots, t_n); P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{L}', \mathcal{P} \cup \{P\}) \quad \text{where } \mathcal{L}' = \mathcal{L} \cup \mathsf{E}(t_1, \ldots, t_n)$$

Syntax

- Predicates $\rho := \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n) \mid u = v \mid \mathsf{att} : t$
- Formulas $\Phi := \rho \mid \Phi \wedge \Phi \mid \Phi \vee \Phi$
- Assertions: $\Phi_1 \implies \Phi_2$

Semantics

- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n)$ when $\mathsf{E}(t_1, \ldots, t_n) \in \mathcal{L}$
- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models u = v$ when $u =_{\mathcal{E}} v$
- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \mathsf{att} : t$ when $\mathcal{M} \vdash t$
- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \Phi_1 \wedge \Phi_2, \Phi_1 \vee \Phi_2$ when ...

Correspondence assertions: formally

$$\texttt{(EV)} \quad (\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P} \cup \{\mathsf{event}\mathsf{E}(t_1, \ldots, t_n); P\}) \rightsquigarrow (\mathcal{N}, \mathcal{M}, \mathcal{L}', \mathcal{P} \cup \{P\}) \quad \text{where } \mathcal{L}' = \mathcal{L} \cup \mathsf{E}(t_1, \ldots, t_n)$$

Syntax

- Predicates $\rho := \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n) \mid u = v \mid \mathsf{att} : t$
- Formulas $\Phi := \rho \mid \Phi \wedge \Phi \mid \Phi \vee \Phi$
- Assertions: $\Phi_1 \implies \Phi_2$

Semantics

- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \mathsf{ev} : \mathsf{E}(t_1, \ldots, t_n)$ when $\mathsf{E}(t_1, \ldots, t_n) \in \mathcal{L}$
- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models u = v$ when $u =_{\mathcal{E}} v$
- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \mathsf{att} : t$ when $\mathcal{M} \vdash t$
- $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \Phi_1 \wedge \Phi_2, \Phi_1 \vee \Phi_2$ when ...
- $(\mathcal{N}_0, \mathcal{M}_0, \mathcal{L}_0, \{P\}) \models \Phi_1 \implies \Phi_2$ when
  for every reachable configuration $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P})$
  with $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \Phi_1\sigma$
  we have $(\mathcal{N}, \mathcal{M}, \mathcal{L}, \mathcal{P}) \models \Phi_2\sigma$

Examples

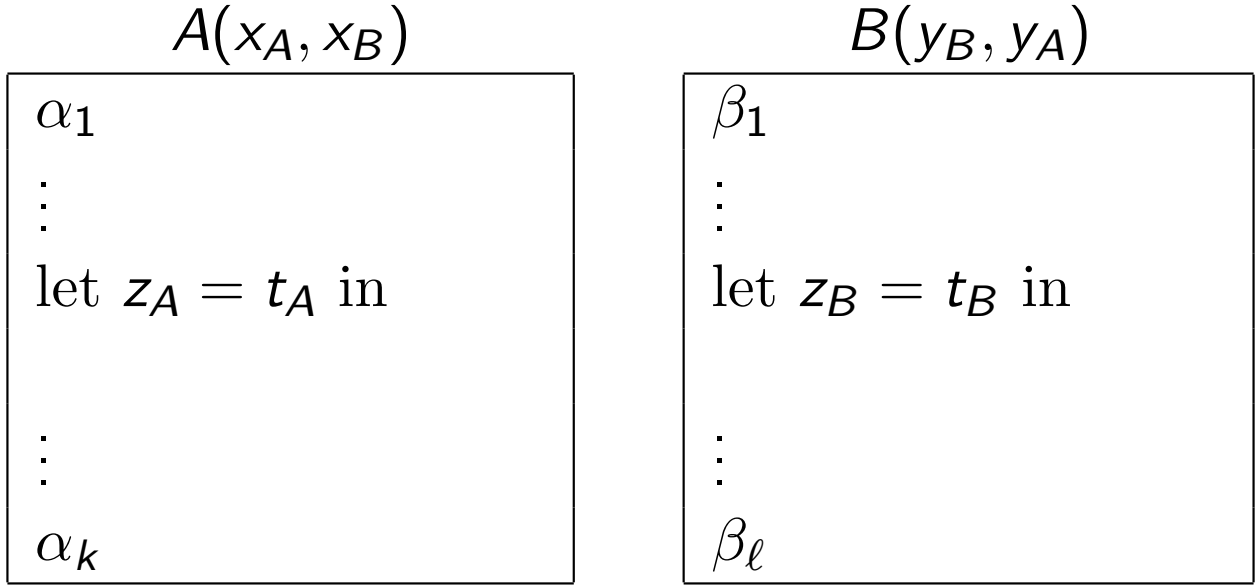
Data protection:

$$P_0 \begin{cases} \text{new } k; \text{new } s; \text{out}(c, \text{enc}(s, \text{pub}(k))) \\ \text{out}(c, \text{pub}(k)); \text{in}(c, x); \\ \text{let } y = \text{dec}(x, k) \text{ in} \\ \text{eventDEC}(y); \text{out}(c, y) \end{cases}$$

$$P_0 \models \text{att} : s \implies \text{ev} : \text{DEC}(s)$$

Examples

Agreement:



Examples

Agreement:

$A(x_A, x_B)$

α_1

$\vdots$

let $z_A = t_A$ in

event AS(x_A, x_B, z_A)

$\vdots$

α_k

$B(y_B, y_A)$

β_1

$\vdots$

let $z_B = t_B$ in

event BS(y_B, y_A, z_B)

$\vdots$

β_ℓ

Examples

Agreement:

$A(x_A, x_B)$

α_1

$\vdots$

$\text{let } z_A = t_A \text{ in}$

$\text{event } \mathsf{AS}(x_A, x_B, z_A)$

$\vdots$

α_k

$B(y_B, y_A)$

β_1

$\vdots$

$\text{let } z_B = t_B \text{ in}$

$\text{event } \mathsf{BS}(y_B, y_A, z_B)$

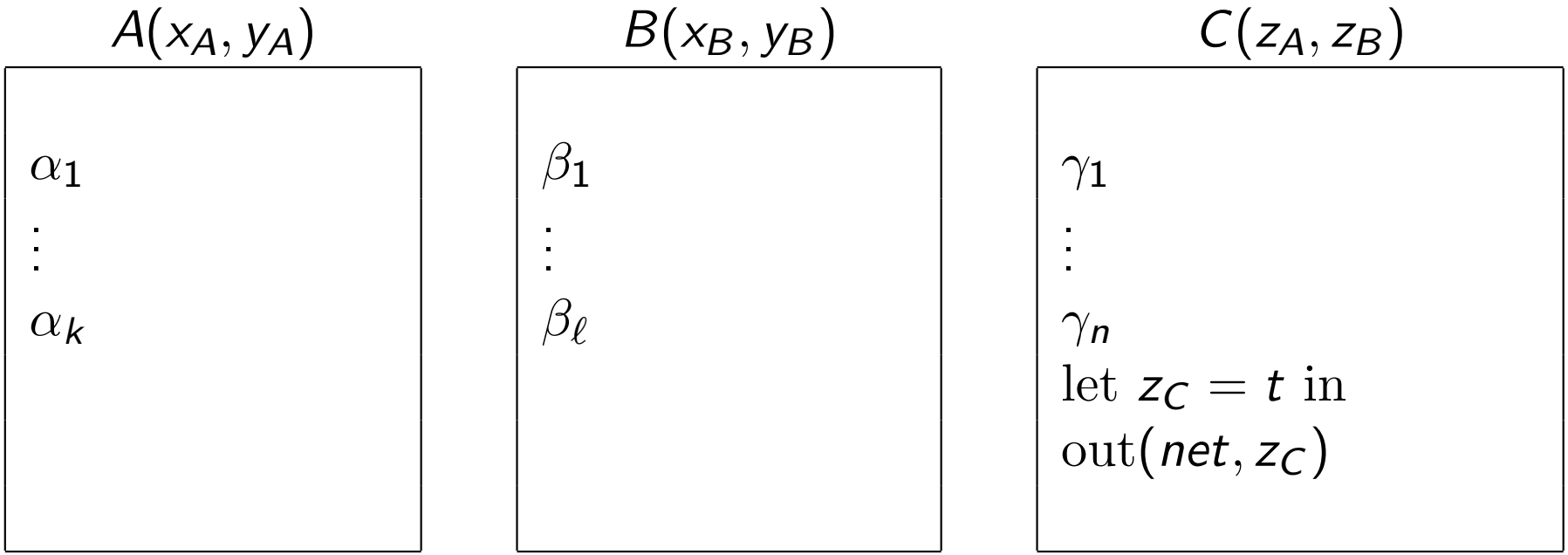
$\vdots$

β_ℓ

$$(!A \mid !B) \quad \models \quad \mathsf{ev} : \mathsf{BS}(x_1, x_2, x_3) \implies \mathsf{ev} : \mathsf{AS}(x_2, x_1, x_3)$$

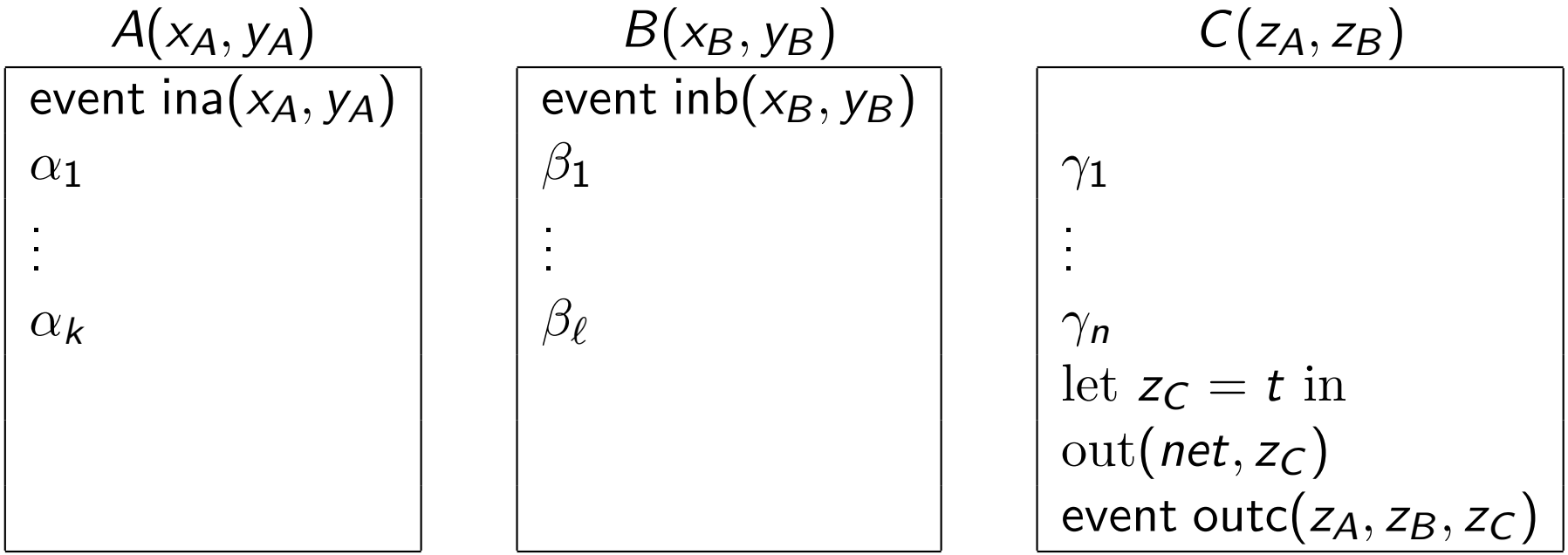
Examples

Integrity:



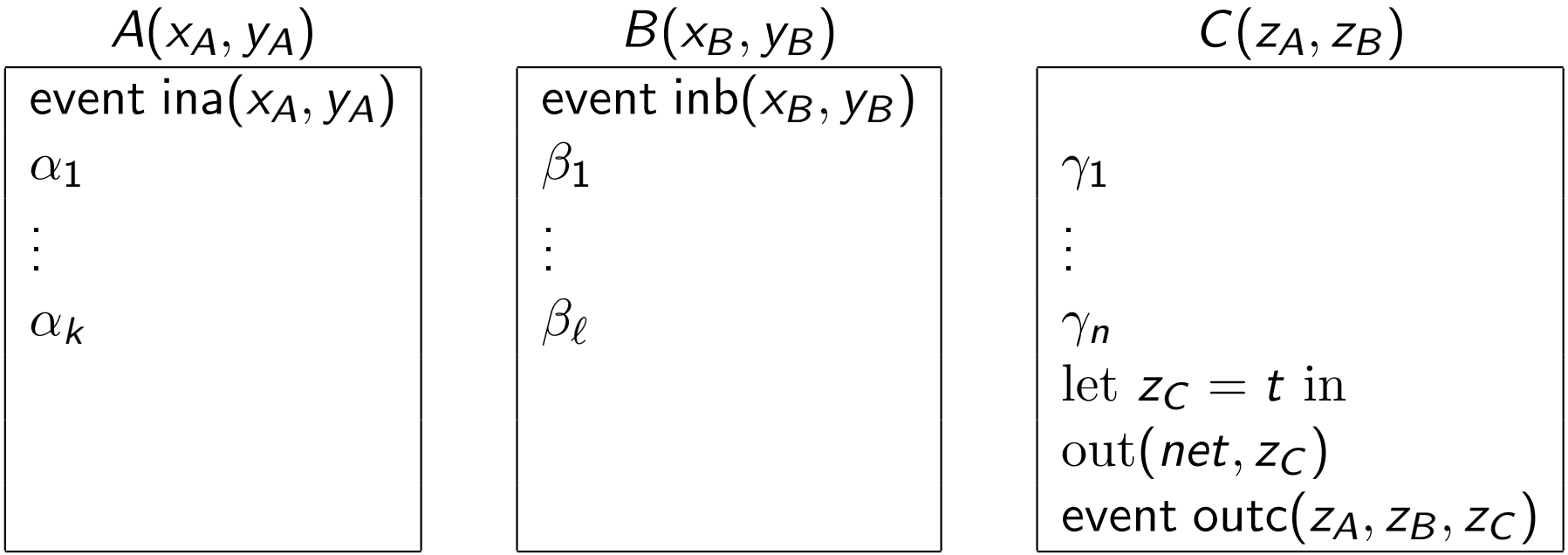
Examples

Integrity:



Examples

Integrity:



$$(\,!A \mid !B \mid !C \,) \quad \models \quad \mathsf{ev} : \mathsf{outc}(x_1, x_2, x_3) \implies \mathsf{ev} : \mathsf{ina}(x_1, y_1) \ \wedge \ \mathsf{ev} : \mathsf{inb}(x_2, y_2) \wedge x_3 = y_1 + y_2$$

Examples

Authorisation and Authentication for Needham-Schroeder.

Case studies and verification

Formal authentication in Needham-Schroeder

$$
\begin{array}{lrcl}
1. & C \to T & : & C, S, Nc \\
2. & T \to C & : & \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct} \\
3. & C \to S & : & \{Kcs, C\}_{Kst} \\
4. & S \to C & : & \{Ns\}_{Kcs} \\
5. & C \to S & : & \{inc(Ns)\}_{Kcs}
\end{array}
$$

Client(C, S)

$\mathrm{new}\ Nc; \mathrm{out}(net, \langle C, S, Nc\rangle);$
$\mathrm{in}(net, x_T);$
$\mathrm{let}\ \langle = Nc, x_{Kcs}, x_{\mathrm{ciph}}\rangle = \mathrm{dec}(x_T, k(C))\ \mathrm{in}$
$\mathrm{out}(net, x_{ciph}); \mathrm{in}(net, x_S);$
$\mathrm{let}\ x_{Ns} = \mathrm{dec}(x_S, x_{Kcs})\ \mathrm{in}$
event GoodResponse$(C, S, Nc, x_{Ns}, x_{Kcs})$
$\mathrm{out}(net, \mathrm{enc}(\mathrm{inc}(x_{Ns}), x_{Kcs}))$

Formal authentication in Needham-Schroeder

1. $C \to T \quad : \quad C, S, Nc$
2. $T \to C \quad : \quad \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \to S \quad : \quad \{Kcs, C\}_{Kst}$
4. $S \to C \quad : \quad \{Ns\}_{Kcs}$
5. $C \to S \quad : \quad \{inc(Ns)\}_{Kcs}$

Third Party

$\text{in}(net, \langle x_C, x_S, x_{Nc} \rangle)$;
event Authorised(x_C, x_S, x_{Nc});
new k_{CS};
let $y_S = \text{enc}(\langle k_{CS}, x_C \rangle, k(x_S))$ in
let $y_C = \text{enc}(\langle x_{Nc}, c, y_S \rangle, k(x_C))$ in
$\text{out}(net, y_C)$

Formal authentication in Needham-Schroeder

1. $C \rightarrow T \quad : \quad C, S, Nc$
2. $T \rightarrow C \quad : \quad \{Nc, Kcs, \{Kcs, C\}_{Kst}\}_{Kct}$
3. $C \rightarrow S \quad : \quad \{Kcs, C\}_{Kst}$
4. $S \rightarrow C \quad : \quad \{Ns\}_{Kcs}$
5. $C \rightarrow S \quad : \quad \{inc(Ns)\}_{Kcs}$

Server(S)

$\text{in}(net, x_{req})$;
$\text{let } \langle x_{Kcs}, x_C \rangle = \text{dec}(x_{req}, k(S)) \text{ in}$
$\text{new } Ns$;
event GrantingAccess(x_C, S, Ns, x_{Kcs});
$\text{out}(net, \text{enc}(Ns, x_{Kcs}))$;
$\text{in}(net, x_{resp})$;
$\text{if inc}(Ns) = \text{dec}(x_{resp}, x_{Kcs}) \text{ then}$
event AccessGranted(x_C, S, Ns, x_{Kcs})

Formal authentication in Needham-Schroeder

$$\text{ev} : \text{GoodResponse}(C, S, x_{Nc}, x_{Ns}, x_{Kcs}) \implies$$
$$\text{ev} : \text{GrantingAccess}(C, S, x_{Ns}, x_{Kcs})$$

$$\text{ev} : \text{AccessGranted}(C, S, x_{Ns}, x_{Kcs}) \implies$$
$$\text{ev} : \text{Authorised}(C, S, x_{Nc})$$

Secure multi-party computation

Privacy-supporting cloud computing

Resources

Laboratoire Spécification et Vérification
Security Protocols Open Repository
www.lsv.ens-cachan.fr/spore/

Bruno Blanchet
ProVerif: Cryptographic protocol verifier in the formal model
http://prosecco.gforge.inria.fr/personal/bblanche/proverif/

Hubert Comon-Lundh and Stéphanie Delaune
Formal Security Proofs.
Software Safety and Security, 2012

Véronique Cortier and Steve Kremer
Formal Models and Techniques for Analyzing Security Protocols: A Tutorial.
Foundations and Trends in Programming Languages, 2014.

Research challenges

Protocols

Verification procedures

Relation to implementations