Formal Verification of Cryptographic Protocols

Luca Durante

luca.durante@polito.it

Istituto di Elettronica e Ingegneria dell'Informazione e delle Telecomunicazioni - CNR
c/o Politecnico di Torino, corso Duca degli Abruzzi 24, I-10129 Torino (Italy)
Outline

formal specification ............................................................................................................. 3
formal verification .................................................................................................................. 16
spi calculus ............................................................................................................................. 21
testing equivalence ................................................................................................................ 24
environment knowledge representation .................................................................................. 29
concrete ES-LTS ..................................................................................................................... 35
symbolic data representation ................................................................................................. 39
symbolic ES-LTS ..................................................................................................................... 47
traces ......................................................................................................................................... 51
tool support ............................................................................................................................. 55
example ..................................................................................................................................... 56
experimental results ............................................................................................................... 58
related work ............................................................................................................................. 67
our results ................................................................................................................................. 72
future work ............................................................................................................................... 73
appendices ............................................................................................................................... 75
formal specification

- Cryptographic protocols should guarantee safe communications through unsafe communication channels

- By means of formal methods we want to verify a protocol meets some requirements

- This needs a precise formalization of:
  - cryptographic primitives
  - insecure channels
  - protocol’s behavior
  - secure communications
formal specification

perfect encryption

- symmetric / public / private key encryption
- a secret encrypted by means of $k / k^+ / k^-$ can be decrypted only if $k / k^- / k^+$ is known
- the encryption key cannot be deduced from the encrypted message
- an encrypted message is sufficiently redundant so that the decryption algorithm can detect whether or not it has succeeded in its task
formal specification

perfect encryption

- one-way functions (or hash functions)
  - if $H(m)$ is the hashing of $m$, then $m$ cannot be deduced from $H(m)$, i.e. $H()$ is noninvertible

- nonces
  - values used no more than once (for the same purpose), usually randomly generated
  - fresh: different when generated at different times or by different agents
  - unpredictable: cannot be guessed
  - an agent cannot guess and/or forge any protocol data
formal specification

perfect encryption

- The hypotheses of perfect encryption are an abstraction of the negligible probability an attacker could
  - get the cleartext from the cipher without the proper decryption key
  - invert a hash function
  - guess and/or forge any protocol secret
To keep simpler the theoretical framework, many formal techniques do not assume any algebraic property of operators (encryption, hashing, and so on)

then the operators are term constructors which define a free term algebra

i.e. each value has just only one representation and vice-versa

the following works try to overcome this limit:

[7, 15, 16, 20, 34, 36]
The assumptions of perfect encryption prevent from:

- taking into account the actual implementation of cryptographic primitives, that can lead to errors and bugs (e.g. IEEE 802.11 WEP authentication)
- taking into account bitwise operations (e.g. XOR), difficult to be represented in equational algebraic models, and often leading to undecidable models

Computational models represent data as bitstrings, and use a probabilistic approach to allow some of the perfect encryption assumptions to be dropped [6]
In order to abstractly define an insecure communication channel we have to define the properties of a potential attacker able to act on such an insecure channel.

The attacker’s reference model has been defined in Dolev-Yao [22].

The Dolev-Yao’s attacker can:\n
- look at, delete, reorder and replay any message sent over a public communication channel.

\[a\] consistently with the hypotheses of perfect encryption
The Dolev-Yao’s attacker can:\n\- decrypt any encrypted message for which it has got the right key, invert invertible functions, and split pairs into pieces\n\- generate its own nonces\n\- forge new messages starting from (pieces of) messages it already knows and possibly coming from past sessions of the protocol; it can then inject the forged messages into public communication channels

\(^a\)consistently with the hypotheses of perfect encryption
formal specification

attacker’s model

Example

\[ \Sigma = \{ \{m\}_k, k \} \implies m \in \hat{\Sigma} \]

\[ \overline{\Sigma} = \{ m, k \} \]
formal specification

attacker’s model

Example

\[ \Sigma = \{ \{m\}_k, k \} \implies m \in \hat{\Sigma} \]

\[ \bar{\Sigma} = \{ m, k \} \]

The Dolev-Yao’s attacker cannot take into account

- timing
- side-channel information, e.g. power consumption and resource usage
formal specification

protocol’s behavior

Wide Mouthed Frog Protocol

A

S

B

1) \{k_{AB}\}_{k_A^S}

2) \{k_{AB}\}_{k_B^S}

3) \{M\}_{k_{AB}}
formal specification

protocol’s behavior

Wide Mouthed Frog Protocol

1) A → S : \{k_{AB}\}k_{AS} on c_{AS}
2) S → B : \{k_{AB}\}k_{SB} on c_{SB}
3) A → B : \{M\}k_{AB} on c_{AB}
The previous specification does not capture a lot of aspects, e.g.:

- how does react an agent when it receives a message which differs from the expected one?
- is $k_{AB}$ “a priori” shared between $A$ and $B$ or is it freshly generated by $A$?
- in case we want to model more instances of each role, which of the data are instance-dependent and which of them are invariant?

Formal description techniques are able to describe a protocol in a complete and non-ambiguous way.
formal specification

formal description techniques

- Logics:
  - Multiset Rewriting [25]
  - BAN Logic [13]

- Process algebras:
  - CSP [33, 39]
  - CCS [27]
  - Spi Calculus [4]

- Ad-hoc formalisms:
  - BRUTUS [19]
  - Strand Spaces [41]
There are many security properties we could be interested in, mainly dependent on the protocol purpose. We will deal with:

- **secrecy** [2]
  the secrecy of a piece of information $d$ holds if an attacker, interacting with protocol parties, cannot learn $d$ (and cannot infer anything about it)

- **authentication** [32]
  an event $e$ authenticates an agent $A$ (with an agent $B$) if $e$ happens only after a message has been previously sent by $A$. This means that $A$ can be identified with certainty by $B$
formal verification

What do we verify?

- some kind of equivalence between two formal specifications, i.e. two specifications are indistinguishable by some kind of observer
- some kind of relationship between two formal specifications, e.g. refinement, implementation
- a formal specification satisfies some property
- the internal consistency of a formal specification, i.e. well formedness, satisfiability
formal verification

- How can we carry out such a verification?
  - Theorem proving [3, 12]
    - powerful technique: usually does not require simplifying hypotheses
    - requires skilled users
    - difficult to be automated
  - State-exploration [5, 7, 11, 24, 30, 31]
    - can be easily automated
    - needs models with finite behavior to some extent
    - affected by the state explosion problem
formal verification

- How can we carry out such a verification?
  - Control-flow analysis [8, 28]
    - does not need explicit enumeration of the state space
    - can be used on models with infinite behavior
    - limits on the kind of systems we can verify
    - limits on the kind of properties we can verify
  - Type-based techniques [1, 28, 29]
    - the Cryptic tool implements [1, 28]: types must be found and specified manually
    - the type checking implemented by Cryptic is sound, but incomplete
    - [29] tries to overcome the above limits, but it is a very preliminary work
Our choice:

- Formal description technique
  - Spi Calculus has been developed by M. Abadi and A. Gordon [4], and derives from Milner’s $\pi$-calculus [37].
  - Flexible and intuitive syntax and semantics to describe cryptographic protocols.

What do we verify? Verification of security properties relies on verification of may-testing equivalence.

Which technique do we select? state-exploration

formal verification

Our state exploration technique [24] checks for may-testing equivalence between spi processes.

The model describes all allowed interactions between the spi process and its environment.

The modelled environment is the most powerful one [22].

The model must be finite:
- bounded number of process instantiations;
- minimized environment knowledge;
- abstract model for environment’s messages\(^a\) without any restriction on the allowed messages.

\(^a\) also called symbolic model checking or lazy intruder
### spi calculus

<table>
<thead>
<tr>
<th>Terms</th>
<th>Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma, \rho, \theta$ ::= terms</td>
<td>$P, Q, R ::= \ldots$ processes</td>
</tr>
<tr>
<td>$m$ name</td>
<td>$\sigma\langle\rho\rangle.P$ output</td>
</tr>
<tr>
<td>$0$ name (the zero constant)</td>
<td>$\sigma(x).P$ input</td>
</tr>
<tr>
<td>$x$ name (a variable)</td>
<td>$P \mid Q$ composition</td>
</tr>
<tr>
<td>$y$ name (a variable)</td>
<td>$(\nu m)\ P$ restriction</td>
</tr>
<tr>
<td>$(\sigma, \rho)$ pair</td>
<td>!$P$ replication</td>
</tr>
<tr>
<td>$suc(\sigma)$ successor</td>
<td>$0$ nil</td>
</tr>
<tr>
<td>$H(\sigma)$ hashing</td>
<td>$[\sigma\text{ is } \rho]\ P$ match</td>
</tr>
<tr>
<td>${\sigma}_\rho$ shared-key encryption</td>
<td>$\text{let } (x, y) = \sigma \text{ in } P$ pair splitting</td>
</tr>
<tr>
<td>$\sigma^+$ public part</td>
<td>$\text{case } \sigma \text{ of } 0 : P \text{ suc}(x) : Q$ integer case</td>
</tr>
<tr>
<td>$\sigma^-$ private part</td>
<td>$\text{case } \sigma \text{ of } {x}_\rho \text{ in } P$ shared-key decr.</td>
</tr>
<tr>
<td>${[\sigma]}_\rho$ public-key encryption</td>
<td>$\text{case } \sigma \text{ of } [[x]]_\rho \text{ in } P$ decryption</td>
</tr>
<tr>
<td>$[[\sigma]]_\rho$ private-key signature</td>
<td>$\text{case } \sigma \text{ of } [[[x]]]_\rho \text{ in } P$ signature-check</td>
</tr>
</tbody>
</table>
Wide Mouthed Frog Protocol [4]

1) $A \rightarrow S : \{k_{AB}\}k_{AS}$ on $c_{AS}$
2) $S \rightarrow B : \{k_{AB}\}k_{SB}$ on $c_{SB}$
3) $A \rightarrow B : \{M\}k_{AB}$ on $c_{AB}$
Wide Mouthed Frog Protocol [4]

1) $A \rightarrow S : \{k_{AB}\}k_{AS}$ on $c_{AS}$

2) $S \rightarrow B : \{k_{AB}\}k_{SB}$ on $c_{SB}$

3) $A \rightarrow B : \{M\}k_{AB}$ on $c_{AB}$

$$P_A(M) \triangleq (\nu k_{AB})(c_{AS}\langle\{k_{AB}\}k_{AS}\rangle).c_{AB}\langle\{M\}k_{AB}\rangle.0)$$

$$P_S \triangleq c_{AS}(x_1). case \ x_1 \ of \ \{x_2\}k_{AS} \ in \ c_{SB}\langle\{x_2\}k_{SB}\rangle.0$$

$$P_B \triangleq c_{SB}(y_1). case \ y_1 \ of \ \{y_2\}k_{SB} \ in \ c_{AB}(y_3). case \ y_3 \ of \ \{y_4\}y_2 \ in \ F(y_4)$$

$$P_{wmf}(M) \triangleq (\nu k_{AS})(\nu k_{SB})(P_A(M) \mid P_S \mid P_B)$$
spi calculus

Wide Mouthed Frog Protocol [4]

1) $A \rightarrow S : \{k_{AB}\} k_{AS}$ on $c_{AS}$

2) $S \rightarrow B : \{k_{AB}\} k_{SB}$ on $c_{SB}$

3) $A \rightarrow B : \{M\} k_{AB}$ on $c_{AB}$

$P_A(M) \triangleq (\nu k_{AB})(c_{AS}\langle\{k_{AB}\}k_{AS}\rangle \cdot c_{AB}\langle\{M\}k_{AB}\rangle .0)$

$P_S \triangleq c_{AS}(x_1). case x_1 of \{x_2\}k_{AS} in c_{SB}\langle\{x_2\}k_{SB}\rangle .0$

$P_B \triangleq c_{SB}(y_1). case y_1 of \{y_2\}k_{SB} in c_{AB}(y_3). case y_3 of \{y_4\}y_2 in F(y_4)$

$P_{wmf}(M) \triangleq (\nu k_{AS})(\nu k_{SB})(P_A(M) \mid P_S \mid P_B)$
Spi Calculus

Wide Mouthed Frog Protocol [4]

2) $S \rightarrow B : \{k_{AB}\}angle_{k_{SB}}$ on $c_{SB}$

3) $A \rightarrow B : \{M\}angle_{k_{AB}}$ on $c_{AB}$

$P_A(M) \triangleq (\nu k_{AB})(\overline{c_{AB}}\langle\{M\}\rangle_{k_{AB}}.0)$

$P_S \triangleq \overline{c_{SB}}\langle\{k_{AB}\}\rangle_{k_{SB}}.0$

$P_{B} \triangleq c_{SB}(y_1). \mathrm{case} \ y_1 \ \mathrm{of} \ \{y_2\} \ \mathrm{in} \ c_{AB}(y_3). \ \mathrm{case} \ y_3 \ \mathrm{of} \ \{y_4\} \ y_2 \ \mathrm{in} \ F(y_4)$

$P_{wmf}(M) \triangleq (\nu k_{AS})(\nu k_{SB})(P_A(M) | P_S | P_B)$
spi calculus

Wide Mouthed Frog Protocol [4]

3) \( A \rightarrow B : \{M\}_{K_{AB}} \) on \( C_{AB} \)

\[
P_A(M) \triangleq (\nu K_{AB})(c_{AB}\langle\{M\}_{K_{AB}}\rangle.0)
\]

\[
P_B \triangleq c_{AB}(y_3). \text{ case } y_3 \text{ of } \{y_4\}_{K_{AB}} \text{ in } F(y_4)
\]

\[
P_{wmf}(M) \triangleq (\nu K_{AS})(\nu K_{SB})(P_A(M) | P_B)
\]
spi calculus

Wide Mouthed Frog Protocol [4]

\[ P_B \triangleq \]

\[ P_{wmf}(M) \triangleq (\nu k_{AS})(\nu k_{SB})(P_B) \]

\[ F(M) \]
spi calculus

Reaction operational semantics defined in [4]:

- **Reduction relation** $\geq$

  $P \geq Q$: $P$ reduces to $Q$ by performing an internal action.

  $[\sigma \text{ is } \sigma] \ P \geq P$ case $\{\sigma\}_\rho \text{ of } \{x\}_\rho$ in $P \geq P[\sigma/x]$

- **Structural equivalence** $\equiv$

  equivalence on processes based on simple and intuitive operator properties.

  $P | Q \equiv Q | P \quad \frac{m \not\in fn(P)}{(vm)(P|Q)\equiv P|(vm)Q} \quad \frac{P \geq Q}{P \equiv Q} \quad \frac{P \equiv Q \quad Q \equiv R}{P \equiv R}$

- **Reaction relation** $\rightarrow$

  $P \rightarrow P': P$ can evolve into $P'$ by performing an internal synchronization (between any two sub-processes).

  $\overline{\sigma}\{\rho\}.P|\sigma(x).Q \rightarrow P|Q[\rho/x]$
testing equivalence

- May-testing introduced by De Nicola and Hennessy on CCS [21].

- Spi calculus is a CCS extension.

- Definitions (from [4]):
  - when a test process $R$ runs in parallel with a process under test $P$, it may signal that $P$ has passed a test by means of a distinguished success action $\omega$;
  - $P, Q, R := \Omega$: $\Omega$ is a distinguished process that can perform only $\omega$. 
testing equivalence

Definitions (cont’d):

- \( P \downarrow \omega \) (\( P \) exhibits \( \omega \)) defined by:
  \[
  \Omega \downarrow \omega \quad \frac{P \downarrow \omega}{(P|Q) \downarrow \omega} \quad \frac{P \downarrow \omega}{(\nu m)P \downarrow \omega} \quad \frac{P \equiv Q}{Q \downarrow \omega}
  \]

- convergence: \( P \downarrow \omega \triangleq \exists Q \mid (P(\rightarrow) \ast Q) \land (Q \downarrow \omega) \)

- \( P \) may pass a test \( R \) \iff \( (P|R) \downarrow \omega \)

- testing preorder:
  \[
  P \sqsubseteq Q \triangleq \forall R \ ((P|R) \downarrow \omega \implies (Q|R) \downarrow \omega)
  \]

- may-testing equivalence:
  \[
  P \simeq Q \triangleq (P \sqsubseteq Q) \land (Q \sqsubseteq P)
  \]
testing equivalence

- Advantages:
  - may-testing corresponds to safety, as most typical security properties [4];
  - no explicit attacker’s specification is needed;
  - formulation of authenticity properties easy and compact.

\[
\begin{align*}
PA(M) & \triangleq (\nu \, k_{AB})(c_{AS} \langle \{k_{AB}\}k_{AS} \rangle \cdot c_{AB} \langle \{M\}k_{AB} \rangle .0) \\
P_S & \triangleq c_{AS}(x_1). \text{ case } x_1 \text{ of } \{x_2\}k_{AS} \text{ in } c_{SB} \langle \{x_2\}k_{SB} \rangle .0 \\
PB_{Spec}(M) & \triangleq c_{SB}(y_1). \text{ case } y_1 \text{ of } \{y_2\}k_{SB} \text{ in } c_{AB}(y_3). \text{ case } y_3 \text{ of } \{y_4\}y_2 \text{ in } F(M) \\
P_{wmf_{Spec}}(M) & \triangleq (\nu \, k_{AS})(\nu \, k_{SB})(PA(M) \mid PS \mid PB_{Spec}(M)) \\
\forall \, M \quad P_{wmf}(M) \simeq P_{wmf_{Spec}}(M)
\end{align*}
\]
testing equivalence

Advantages (cont’d):

- formulation of secrecy properties more accurate than the one based on environment knowledge:
Advantages (cont’d):

- formulation of secrecy properties more accurate than the one based on environment knowledge:

Initial environment knowledge: \( \{c, \ k^+\} \)

\[
P(M) \triangleq c\langle H(M) \rangle. \ 0 \quad P(M) \triangleq \overline{c}\langle \{[M]\}_{k^+} \rangle. \ 0 \quad P(M) \triangleq (\nu n)\overline{c}\langle \{[M, n]\}_{k^+} \rangle. \ 0
\]

Secrecy based on environment knowledge: satisfied
testing equivalence

Advantages (cont’d):

- formulation of secrecy properties more accurate than the one based on environment knowledge:

Initial environment knowledge: \{c, k^+, M, M'\}

\[
P(M) \overset{\Delta}{=} \overline{c}(H(M)). 0 \quad P(M) \overset{\Delta}{=} \overline{c}([M]_{k^+}). 0 \quad P(M) \overset{\Delta}{=} (\nu n)\overline{c}([M, n]_{k^+}). 0
\]

\[
P(M') \overset{\Delta}{=} \overline{c}(H(M')). 0 \quad P(M') \overset{\Delta}{=} \overline{c}([M']_{k^+}). 0 \quad P(M') \overset{\Delta}{=} (\nu n)\overline{c}([M', n]_{k^+}). 0
\]

The secret must not be recognized

\[
\forall M, M' \quad P(M) \simeq P(M')
\]
testing equivalence

How we check testing equivalence:

- the **quantification over contexts** problem is solved by means of an **ES-LTS** describing all the interactions of a spi process with the environment (all testers);
- may-testing equivalence is checked as **trace equivalence** on the **ES-LTS**;
- trace equivalence on the **ES-LTSs** is **necessary and sufficient** condition for may-testing equivalence (soundness and completeness).
environment knowledge representation

- \( A \): set of names \((m \in A)\).

- \( M(A) \): set terms \((\sigma, \sigma_1, \sigma_2 \in M(A)\) are finite).

- \( \Sigma \subseteq M(A) \): finite set of finite terms.

- \( \widehat{\Sigma} \): closure of \( \Sigma \) (the set of all spi calculus terms that can be built by combining the elements of \( \Sigma \) by means of term operators and their inverses).

- \( \Sigma \): minimal closure seed of \( \Sigma \) (the minimum subset of \( \widehat{\Sigma} \) that is enough to regenerate \( \widehat{\Sigma} \)).

\[
\Sigma = \{\{[m]\}_{a^+}, a^-\} \quad \overline{\Sigma} = \{\{[m]\}_{a^+}, a^-, m\}
\]
Let $\Sigma \subseteq \mathcal{M}(\mathcal{A})$ be a finite set of terms:

- **Finiteness**: $\overline{\Sigma}$ is finite.

- **Minimality**: Let $\sigma \in \overline{\Sigma}$, then $(\overline{\Sigma} \setminus \{\sigma\}) \subset \hat{\Sigma}$.

- **Closure preservation**: $\hat{\Sigma} = \hat{\Sigma}$.

- **Computability**: $\overline{\Sigma}$ can be computed in a finite number of steps (normalized reduction).

- **Decidability**: Let $\sigma \in \mathcal{M}(\mathcal{A})$ be any finite term, then, determining if $\sigma \in \hat{\Sigma}$ is decidable.
environment knowledge representation

$$\Sigma = \left\{ c, \{ [k_1] k_2 \}_{k_3^+}, \{ [m] \}_{k_1^+}, k_1^+, k_2 \right\} \quad \rho = k_3^- \quad \Sigma_0 = \Sigma \cup \rho$$
environment knowledge representation

$$\Sigma = \left\{ c, \{\{k_1\}k_2\}\}_{k_3^+}, \{[m]\}_{k_1^+}, k_1^+, k_2 \right\} \quad \rho = k_3^- \quad \Sigma_0 = \Sigma \cup \rho$$

$$\Sigma_0 = \left\{ c, \{\{k_1\}k_2\}\}_{k_3^+}, \{[m]\}_{k_1^+}, k_1^+, k_2, k_3^- \right\}$$
environment knowledge representation

\[ \Sigma = \left\{ c, \left\{ \left\{ k_1 \right\}_k \right\}_k^+, \left\{ [m] \right\}_k^+, \left\{ k_2 \right\}_k^{k+} \right\} \quad \rho = k_3^- \quad \Sigma_0 = \Sigma \cup \rho \]

\[ \Sigma_0 = \left\{ c, \left\{ \left\{ k_1 \right\}_k \right\}_k^+, \left\{ [m] \right\}_k^+, \left\{ k_1 \right\}_k^{k+}, \left\{ k_2 \right\}_k^{k+}, \left\{ k_3 \right\}_k^{k+} \right\} \]
environment knowledge representation

\[
\Sigma = \left\{ c, \{\{k_1\} \}_k^+, \{m\}_k^+, k_1^+, k_2 \right\} \quad \rho = k_3^- \quad \Sigma_0 = \Sigma \cup \rho
\]

\[
\Sigma_0 = \left\{ c, \{\{k_1\} \}_k^+, \{m\}_k^+, k_1^+, k_2, k_3^- \right\}
\]

\[
\Sigma_1 = \left\{ c, \{\{k_1\} \}_k^+, \{m\}_k^+, k_1^+, k_2, k_3^-, k_1 \right\}
\]
environment knowledge representation

$$\Sigma = \left\{ c, \{\{ k_1 \} \}_{k_2} \}^{k_3^+}, \{ [m] \}_{k_1^+}, k_1^+, k_2 \right\} \quad \rho = k_3^- \quad \Sigma_0 = \Sigma \cup \rho$$

$$\Sigma_0 = \left\{ c, \{\{ k_1 \} \}_{k_2} \}^{k_3^+}, \{ [m] \}_{k_1^+}, k_1^+, k_2, k_3^- \right\}$$

$$\Sigma_1 = \left\{ c, \{\{ k_1 \} \}_{k_2} \}^{k_3^+}, \{ [m] \}_{k_1^+}, k_1^+, k_2, k_3^- , \{ k_1 \}_{k_2} \right\}$$
environment knowledge representation

\[ \Sigma = \left\{ c, \{\{k_1\}k_2]\}_{k_3}, \{[m]\}_{k_1}, k_1^+, k_2 \right\} \quad \rho = k_3^- \quad \Sigma_0 = \Sigma \cup \rho \]

\[ \Sigma_0 = \left\{ c, \{\{k_1\}k_2]\}_{k_3}, \{[m]\}_{k_1}, k_1^+, k_2, k_3^- \right\} \]

\[ \Sigma_1 = \left\{ c, \{\{k_1\}k_2]\}_{k_3}, \{[m]\}_{k_1}, k_1^+, k_2, k_3^-, \{k_1\}k_2 \right\} \]

\[ \Sigma_2 = \left\{ c, \{\{k_1\}k_2]\}_{k_3}, \{[m]\}_{k_1}, k_1^+, k_2, k_3^-, k_1 \right\} \]
environment knowledge representation

\[ \Sigma = \left\{ c, \left\{ \left\{ k_1 \right\} k_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2 \right\} \quad \rho = k_3^- \quad \Sigma_0 = \Sigma \cup \rho \]

\[ \Sigma_0 = \left\{ c, \left\{ \left\{ k_1 \right\} k_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- \right\} \]

\[ \Sigma_1 = \left\{ c, \left\{ \left\{ k_1 \right\} k_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^-, \left\{ k_1 \right\} k_2 \right\} \]

\[ \Sigma_2 = \left\{ c, \left\{ \left\{ k_1 \right\} k_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^-, k_1 \right\} \]
environment knowledge representation

\[ \Sigma = \left\{ c, \left\{ \left\{ k_1 \right\} k_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^-, k_2 \right\} \quad \rho = k_3^- \quad \Sigma_0 = \Sigma \cup \rho \]

\[ \Sigma_0 = \left\{ c, \left\{ \left\{ k_1 \right\} k_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- \right\} \]

\[ \Sigma_1 = \left\{ c, \left\{ \left\{ k_1 \right\} k_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- , \left\{ k_1 \right\} k_2 \right\} \]

\[ \Sigma_2 = \left\{ c, \left\{ \left\{ k_1 \right\} k_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+ , k_2, k_3^-, k_1 \right\} \]

\[ \Sigma_3 = \left\{ c, \left\{ \left\{ k_1 \right\} k_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^-, k_1, m \right\} \]
environment knowledge representation

\[ \Sigma = \left\{ c, \left\{ \left\{ k_1 \right\}_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2 \right\} \]

\[ \rho = k_3^- \]

\[ \Sigma_0 = \Sigma \cup \rho \]

\[ \Sigma_0 = \left\{ c, \left\{ \left\{ k_1 \right\}_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- \right\} \]

\[ \Sigma_1 = \left\{ c, \left\{ \left\{ k_1 \right\}_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- , \left\{ k_1 \right\}_2 \right\} \]

\[ \Sigma_2 = \left\{ c, \left\{ \left\{ k_1 \right\}_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- , k_1 \right\} \]

\[ \Sigma_3 = \left\{ c, \left\{ \left\{ k_1 \right\}_2 \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- , k_1, m \right\} \]
environment knowledge representation

$$\Sigma = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right\} \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2 \right\} \quad \rho = k_3^- \quad \Sigma_0 = \Sigma \cup \rho$$

$$\Sigma_0 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right\} \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- \right\}$$

$$\Sigma_1 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right\} \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- \right\}$$

$$\Sigma_2 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right\} \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- \right\}$$

$$\Sigma_3 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right\} \right\}_{k_3^+}, \left\{ [m] \right\}_{k_1^+}, k_1^+, k_2, k_3^- \right\}$$

$$\Sigma_4 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right\} \right\}_{k_3^+}, \left\{ k_1^+ \right\}_{k_1^+}, k_2, k_3^- \right\}$$
environment knowledge representation

$$\Sigma = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right] \right\}_{k_3^+}, \left\{ m \right\}_{k_1^+}, k_1^+, k_2 \right\} \quad \rho = k_3^- \quad \Sigma_0 = \Sigma \cup \rho$$

$$\Sigma_0 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right] \right\}_{k_3^+}, \left\{ m \right\}_{k_1^+}, k_1^+, k_2, k_3^- \right\}$$

$$\Sigma_1 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right] \right\}_{k_3^+}, \left\{ m \right\}_{k_1^+}, k_1^+, k_2, k_3^-, \left\{ k_1 \right\} k_2 \right\}$$

$$\Sigma_2 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right] \right\}_{k_3^+}, \left\{ m \right\}_{k_1^+}, k_1^+, k_2, k_3^-, k_1 \right\}$$

$$\Sigma_3 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right] \right\}_{k_3^+}, \left\{ m \right\}_{k_1^+}, k_1^+, k_2, k_3^-, k_1, m \right\}$$

$$\Sigma_4 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right] \right\}_{k_3^+}, k_1^+, k_2, k_3^-, k_1, m \right\}$$

$$\Sigma_5 = \left\{ c, \left\{ \left[ \left\{ k_1 \right\} k_2 \right] \right\}_{k_3^+}, k_2, k_3^-, k_1, m \right\}$$
environment knowledge representation

\[
\Sigma = \left\{ \omega \left( \left[ \left\{ \{k_1\}k_2 \} \right\} \right]_{k_3^+}, \left[ m \right]_{k_1^+}, k_1^+, k_2 \right\} \quad \rho = k_3^- \\
\Sigma_0 = \Sigma \cup \rho \\
\Sigma_1 = \left\{ \omega \left( \left[ \left\{ \{k_1\}k_2 \} \right\} \right]_{k_3^+}, \left[ m \right]_{k_1^+}, k_1^+, k_2, k_3^- \right\} \\
\Sigma_2 = \left\{ \omega \left( \left[ \left\{ \{k_1\}k_2 \} \right\} \right]_{k_3^+}, \left[ m \right]_{k_1^+}, k_1^+, k_2, k_3^- \right\} \\
\Sigma_3 = \left\{ \omega \left( \left[ \left\{ \{k_1\}k_2 \} \right\} \right]_{k_3^+}, \left[ m \right]_{k_1^+}, k_1^- \right\} \\
\Sigma_4 = \left\{ \omega \left( \left[ \left\{ \{k_1\}k_2 \} \right\} \right]_{k_3^+}, k_1^+, k_2, k_3^- \right\} \\
\Sigma_5 = \left\{ \omega \left( \left[ \left\{ \{k_1\}k_2 \} \right\} \right]_{k_3^+}, k_2, k_3^- \right\} \\
\bar{\Sigma} \cup \left\{ \rho \right\} = \Sigma_5 \\
\delta_{\Sigma}^- (\rho) = \left\{ \left[ m \right]_{k_1^+}, k_1^+ \right\} \\
\delta_{\Sigma}^+ (\rho) = \left\{ k_3^-, k_1, m \right\}
\]
environment knowledge representation

- \( I \): countable set of names s.t. \( I \cap \mathcal{A} = \emptyset \) (indexes).

- The environment knowledge is a bijective function
  \[ K : \bar{\Sigma} \rightarrow L \quad (L \subset I). \]

- \( K \vdash \sigma \iff \sigma \in \hat{\Sigma}. \)

- When the environment receives a term \( \rho \), \( K \) is updated accordingly and becomes \( f(\rho, K) \):
  - the elements of \( \delta^\rightarrow(\rho) \) are removed from \( \text{dom}(K) \);
  - the elements of \( \delta^\leftarrow(\rho) \) are added to \( \text{dom}(K) \);
  - new indexes are assigned to terms of \( \delta^\leftarrow(\rho) \), as they are computed during the normalized reduction.
environment knowledge representation

Checking testing equivalence means to abstract away from the exact value of the exchanged data.

Indexes take into account how data is perceived by the environment.

Initial environment knowledge: \{\langle c, \ l_0 \rangle \}

\[
P \triangleq (\nu M, N)(\bar{c}\langle M \rangle. \bar{c}\langle N \rangle. \bar{c}\langle M \rangle. \ 0) \not\simeq Q \triangleq (\nu M, N)(\bar{c}\langle M \rangle. \bar{c}\langle N \rangle. \bar{c}\langle N \rangle. \ 0)
\]
\[
\{\langle c, \ l_0 \rangle, \ \langle M, \ l_1 \rangle, \langle N, \ l_2 \rangle \}
\]

\[
R \triangleq c(l_1). \ c(l_2). \ c(x). \ [l_1 \ is \ x]. \ \Omega
\]

\[
P \triangleq (\nu M, k)(\bar{c}\langle \{M\}_k \rangle. \bar{c}\langle k \rangle. \ 0) \simeq Q \triangleq (\nu N, h)(\bar{c}\langle \{N\}_h \rangle. \bar{c}\langle h \rangle. \ 0)
\]
\[
\{(c, \ l_0), \ (\{M\}_k, \ l_1)\}
\]
\[
\{(c, \ l_0), \ (k, \ l_2), \ (M, \ l_3)\}
\]
\[
\{(c, \ l_0), \ (N, \ l_2), \ (N, \ l_3)\}
\]
Canonical representation of $\rho$ wrt $K$: $\rho[K]$.

$$[K] = \left\{ \begin{array}{c} c \\ l_0 \\ \{ \begin{array}{c} \{k_1\} \\ k_2 \end{array} \} \end{array} \right\} \left[ \begin{array}{c} k_3^+ \\ l_1 \\ \{m\} \end{array} \right] \left[ \begin{array}{c} k_4^+ \\ l_2 \\ k_2 \end{array} \right]$$

$$\{[m]\}_{k_1^+}[K] = l_2$$

$$f(k_3^-, K) = \left\{ \begin{array}{c} c \\ l_0 \\ \{ \begin{array}{c} \{k_1\} \\ k_2 \end{array} \} \end{array} \right\} \left[ \begin{array}{c} k_3^- \\ l_1 \\ k_2 \\ l_4 \\ l_5 \\ l_6 \\ l_7 \end{array} \right]$$

$$\{[m]\}_{k_1^+}[f(k_3^-, K)] = \{[l_7]\}_{l_6^+}$$

$K$ is a substitution that replaces each occurrence of $\rho \in \text{dom}(K)$ with its corresponding index $K(\rho)$. 
concrete ES-LTS

- States of the ES-LTS, denoted $K \triangleright P$, are made up of a process $P$ and a knowledge function $K$.

- Transitions take the form $K \triangleright P \xrightarrow{\mu, \phi} K' \triangleright P'$ with
  - process action label $\mu$
  - complementary environment action label $\phi$

- Reaction transition $K \triangleright P \xrightarrow{\tau} K \triangleright P'$

- Output transition $K \triangleright P \xrightarrow{\sigma[K'], \delta_K} K' \triangleright P'$

- Input transition $K \triangleright P \xrightarrow{\sigma[K], \rho[K]} K \triangleright P'$
concrete ES-LTS

**Derivation rules**

\[ K \vdash \sigma \quad K' = f(\rho, K) \]

\[ \frac{K \triangleright \sigma \langle \rho \rangle . P}{\sigma[K']} \]

\[ K' \triangleright P \]

\[ \langle \delta_K^-(\rho), \delta_K^=(\rho), \rho \rangle[K'] \]

\[ K' \triangleright P \]

\[ K \vdash \rho \]

\[ K \triangleright \sigma(x) . P \]

\[ \sigma[K] \]

\[ \rho[K] \]

\[ K \triangleright P[\rho/x] \]
concrete ES-LTS
symbolic data representation

When a spi process is ready on an input event, the environment can interact with it by sending any term that can be built from its current knowledge.

Such a set of terms is infinite, then the concrete ES-LTS model contains an infinite number of transitions.

In the symbolic ES-LTS such a set of transitions is represented as a single transition, where the exchanged data is an unspecialized generic term $\gamma$, i.e. a term not constrained to assume any syntactical form.

$\gamma$ represents all the concrete terms the environment was able to generate when $\gamma$ was issued.
symbolic data representation

- $\gamma \in \Gamma$, where $\Gamma$ is an infinite, countable set of names, such that $\Gamma \cap I = \emptyset$, $\Gamma \cap A = \emptyset$.

- The use of generic terms leads to symbolic processes and symbolic knowledge functions.

- Each symbolic state is characterized by a finite set of unspecialized generic terms $G \subset \Gamma$.

- Function $\Upsilon$ belongs to the symbolic ES-LTS state and gives the current interpretation of each $\gamma \in G$.

- $\Upsilon(\gamma)$ is the domain of the knowlegde function of the state where $\gamma$ was generated ($\gamma$ represents the whole set of values $\hat{\Upsilon}(\gamma)$). $\Upsilon : G \rightarrow \text{dom}(K)$
symbolic data representation

- The infinite behaviors represented as a single symbolic one are indistinguishable until some filtering condition occurs:

\[ c(x) \cdots \text{case } x \text{ of } \{ \sigma \}_\rho \text{ in } Q \]

- When this happens, only the behaviors corresponding to the satisfying values are allowed to proceed: the set of terms represented by each unspecialized generic term \( \gamma \) is narrowed to the largest subset of \( \Upsilon(\gamma) \) compatible with the operation performed.
symbolic data representation

Such narrowings are equivalent to the substitution of one or more unspecialized generic terms with specialized generic terms or concrete terms.

\[ c(x). \cdots [x \text{ is } m]Q \]

if \( m \in \Upsilon(\gamma) \) then \( \langle m/\gamma \rangle \) narrows \( \Upsilon(\gamma) \) down to \( m \)

\[ c(x). \cdots \text{let}(x_1, x_2) = x \text{ in } Q \]

\( \langle (\gamma_1, \gamma_2)/\gamma \rangle \) narrows \( \Upsilon(\gamma) \) down to pairs

When a specialization \( \xi \) is applied, \( \Upsilon \) is updated accordingly, the new generic term interpretation function being denoted as \( \Upsilon\{\xi\} \).
symbolic data representation

Narrowing specializations are not adequate when knowledge functions are involved:

\[ K = \{(c, l_0), (d, l_1), \{(c, d)\}_{k_1}, l_2), \{\{(\gamma_0, \gamma_1)\}_{k_1}\}_{k_2}, l_3}\right\} \]

\[ \Upsilon = \{ (\gamma_0, \{c, d\}), (\gamma_1, \{c, d\}) \right\} \]

\[ P = (\nu k_2)\overline{c}\langle k_2 \rangle . P' \]
Narrowing specializations are not adequate when knowledge functions are involved:

\[ K = \{(c, l_0), (d, l_1), \{(c, d)\}_{k_1}, l_2), \{\{(\gamma_0, \gamma_1)\}_{k_1} \}_{k_2}, l_3) \} \]

\[ \Upsilon = \{(\gamma_0, \{c, d\}), (\gamma_1, \{c, d\})\} \]

\[ P = (\nu k_2)\overline{c}^{\langle k_2 \rangle}.P' \]
symbolic data representation

Narrowing specializations are not adequate when knowledge functions are involved:

\[ K = \{(c, l_0), (d, l_1), \{(c, d\}k_1, l_2), (\{(\gamma_0, \gamma_1\}k_1\}k_2, l_3)\} \]

\[ \Upsilon = \{ (\gamma_0, \{c, d\}), (\gamma_1, \{c, d\}) \} \]

\[ P = (\nu k_2)\overline{c}(k_2).P' \]
symbolic data representation

Narrowing specializations are not adequate when knowledge functions are involved:

\[ K = \{(c, l_0), (d, l_1), ((c, d)_{k_1}, l_2), \{(\gamma_0, \gamma_1)_{k_1}\}_{k_2}, l_3)\} \]

\[ \Upsilon = \{\gamma_0, \{c, d\}, \gamma_1, \{c, d\}\} \]

\[ P = (\nu k_2)\overline{c}\langle k_2 \rangle . P' \]
Narrowing specializations are not adequate when knowledge functions are involved:

\[ K = \{ (c, l_0), (d, l_1), \{(c, d)\}_{k_1}, l_2), (\{(\gamma_0, \gamma_1)\}_{k_1})_{k_2}, l_3) \} \]

\[ \Upsilon = \{ (\gamma_0, \{c, d\}), (\gamma_1, \{c, d\}) \} \]

\[ P = (\nu_{k_2})c\langle k_2 \rangle . P' \]
symbolic data representation

Narrowing specializations are not adequate when knowledge functions are involved:

\[ K = \{(c, l_0), (d, l_1), ((c, d)_{k_1}, l_2), ((\gamma_0, \gamma_1)_{k_1})_{k_2}, l_3)\}\]

\[ \Upsilon = \{ (\gamma_0, \{c, d\}), (\gamma_1, \{c, d\}) \} \]

\[ P = (\nu k_2)\overline{c}\langle k_2 \rangle.P' \]
symbolic data representation

Narrowing specializations are not adequate when knowledge functions are involved:

\[ K = \{(c, l_0), (d, l_1), \{(c, d)\}_{k_1}, l_2), (\{(\gamma_0, \gamma_1)\}_{k_1})_{k_2}, l_3)\} \]

\[ \Upsilon = \{ (\gamma_0, \{c, d\}), (\gamma_1, \{c, d\}) \} \]

\[ P = (\nu k_2)\bar{c}\langle k_2 \rangle . P' \]
symbolic data representation

Narrowing specializations are not adequate when knowledge functions are involved:

\[ K = \{(c, l_0), (d, l_1), \{\{(c, d)\}_{k_1}, l_2\}, \{\{\{c, d\}_{k_1}\}_{k_2}, l_3\}\} \]

\[ \Upsilon = \{(\gamma_0, \{c, d\}), (\gamma_1, \{c, d\})\} \]

\[ P = (\nu k_2)\overline{c}\langle k_2\rangle.P' \]

\[ K'_1 = \{(c, l_0), (d, l_1), \{\{(c, d)\}_{k_1}, l_2\}, (k_2, l_4)\} \]

by applying the specialization \( \langle c/\gamma_0, d/\gamma_1\rangle \)
Narrowing specializations are not adequate when knowledge functions are involved:

\[ K = \{(c, l_0), (d, l_1), \{(c, d)\}_{k_1}, l_2\}, \{(\{\gamma_0, \gamma_1\}\}_{k_1})_{k_2}, l_3\}\]

\[ \Upsilon = \{(\gamma_0, \{c, d\}), (\gamma_1, \{c, d\})\} \]

\[ P = (\nu k_2)\overline{c}\langle k_2\rangle.P' \]

\[ K_1' = \{(c, l_0), (d, l_1), \{(c, d)\}_{k_1}, l_2\}, (k_2, l_4)\} \]

by applying the specialization \(\langle c/\gamma_0, d/\gamma_1\rangle\)

\[ K_2' = \{(c, l_0), (d, l_1), \{(c, d)\}_{k_1}, l_2\}, (k_2, l_4), (\{(\gamma_0, \gamma_1)\}_{k_1}, l_5)\} \]

if specialization \(\langle c/\gamma_0, d/\gamma_1\rangle\) is never applied
**symbolic data representation**

- **Extended** narrowing specifications must be introduced: \( \langle \xi, \delta_\Lambda \rangle \), where \( \delta_\Lambda = \{\xi_1, \ldots, \xi_k\} \) is a set of forbidden further specializations.

- \( \Lambda \) is the set of specializations obtained as the union of all forbidden specializations accumulated up to the current state.

- \( \Lambda \) belongs to the symbolic ES-LTS state.

- \( (K \triangleright P)_{\Upsilon,\Lambda} \) is a symbolic state.
symbolic data representation

- A specialization $\xi$ is **compatible** with the prohibitions imposed by $\Lambda$ if it can be applied without violating the prohibitions imposed by $\Lambda$.

  $S_{\gamma, \Lambda}$ is the set of **allowed** specializations

Given $\Lambda = \{\langle c/\gamma_0, d/\gamma_1 \rangle\}$, $\langle c/\gamma_0 \rangle \in S_{\gamma, \Lambda}$ because any further specialization of $\gamma_1$ different from $\langle d/\gamma_1 \rangle$, such as for example $H(d)/\gamma_1$, can be applied after $\langle c/\gamma_0 \rangle$ without violating the prohibitions imposed by $\Lambda$.

- When a specialization $\xi \in S_{\gamma, \Lambda}$, is applied to a symbolic state, also $\Lambda$ is updated: $\Lambda\{\xi\}$.

$\{\langle c/\gamma_0, d/\gamma_1 \rangle\}\{c/\gamma_0\} = \{\langle d/\gamma_1 \rangle\}$

$\{\langle c/\gamma_0, d/\gamma_1 \rangle\}\{d/\gamma_0\} = \emptyset$
symbolic data representation

Unification: \( \sigma \cdot \rho = \{ \xi \in S_{\mathcal{Y}, \Lambda} \mid \sigma[\xi] = \rho[\xi] \} \)

\( \gamma_0 \cdot \{ \gamma_1 \} \gamma_2 \) useful to compute which behaviors can successfully execute the match operation \( \gamma_0 \) is \( \{ \gamma_1 \} \gamma_2 \).

\( \mathcal{Y}(\gamma_0) = \{ \{ m \}_d, \{ m \}_e \} \quad \mathcal{Y}(\gamma_1) = \mathcal{Y}(\gamma_2) = \{ m, d, e \} \)

\( \xi_1 = \langle \{ m \}_d / \gamma_0, m / \gamma_1, d / \gamma_2 \rangle : \)
\[ [\gamma_0[\xi_1] \text{ is } \{ \gamma_1 \} \gamma_2[\xi_1]] = \{ \{ m \}_d \text{ is } \{ m \}_d \} \]

\( \xi_2 = \langle \{ m \}_e / \gamma_0, m / \gamma_1, e / \gamma_2 \rangle : \)
\[ [\gamma_0[\xi_2] \text{ is } \{ \gamma_1 \} \gamma_2[\xi_2]] = \{ \{ m \}_e \text{ is } \{ m \}_e \} \]

\( \xi_3 = \langle \{ \gamma'_0 \} \gamma''_0 / \gamma_0, \gamma'_0 / \gamma_1, \gamma''_0 / \gamma_2 \rangle, \) with \( \mathcal{Y}(\gamma'_0) = \mathcal{Y}(\gamma''_0) = \mathcal{Y}(\gamma_0) : \)
\[ [\gamma_0[\xi_3] \text{ is } \{ \gamma_1 \} \gamma_2[\xi_3]] = \{ \{ \gamma'_0 \} \gamma''_0 \text{ is } \{ \gamma'_0 \} \gamma''_0 \} \]
symbolic ES-LTS

- **Specialization transition** \((K \triangleright P)_{\Upsilon,\Lambda} \xrightarrow{\tau_{[K']}} (K' \triangleright P')_{\Upsilon',\Lambda'}\)

- **Output transition** \((K \triangleright P)_{\Upsilon,\Lambda} \xrightarrow{\sigma_{[\xi][K']}} (K' \triangleright P')_{\Upsilon',\Lambda'}\)

- **Input transition** \((K \triangleright P)_{\Upsilon,\Lambda} \xrightarrow{\gamma[K]} (K \triangleright P')_{\Upsilon',\Lambda}\)
symbolic ES-LTS

Symbolic derivation rules

\[
\begin{align*}
K \vdash \sigma & \quad \langle \xi, \delta_\Lambda \rangle \in \Theta(\rho, K_{\Upsilon, \Lambda}) & \quad K'_{\Upsilon', \Lambda'} = f_{\langle \xi, \delta_\Lambda \rangle}(\rho, K_{\Upsilon, \Lambda}) \\
(K \triangleright \sigma(\rho).P)_{\Upsilon, \Lambda} & \quad \frac{\sigma[\xi][K']}{\langle \xi, \delta_\Lambda \rangle[K'], \langle \delta_{\Upsilon'}(\rho), \delta_{\Upsilon'}(\rho), \rho \rangle[K'] \rightarrow (K' \triangleright P[\xi])_{\Upsilon', \Lambda'}
\end{align*}
\]

\[
\begin{align*}
K \vdash \sigma & \quad \gamma \notin \text{dom}(\Upsilon) \\
(K \triangleright \sigma(x).P)_{\Upsilon, \Lambda} & \quad \frac{\sigma[K]}{\gamma \rightarrow (K \triangleright P[\gamma/x])_{\Upsilon \cup \{\gamma, \text{dom}(K)\}}, \Lambda}
\end{align*}
\]

\[
\begin{align*}
\xi \in \sigma & \quad \rho & \quad \xi \not\in \top \\
(K \triangleright ([\sigma \text{ is } \rho]P))_{\Upsilon, \Lambda} & \quad \frac{\tau}{\xi[K[\xi]] \rightarrow (K \triangleright [\sigma \text{ is } \rho]P)_{\Upsilon, \Lambda}\{\xi\}}
\end{align*}
\]
symbolic ES-LTS

\[
\begin{array}{c}
0 \\
\tau\{\{M]\}_k, c(x). [x \text{ is } H(M)] F(M) \\
\tau\{\{M]\}_k, c(y_1). \text{ case } y_1 \text{ of } \{y_2\}_k \text{ in } \tau(H(y_2)) \cdot 0
\end{array}
\]

\[
\begin{array}{c}
1.1 \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k
\end{array}
\]

\[
\begin{array}{c}
1.2 \\
\tau\{\{M]\}_k, c(x). [x \text{ is } H(M)] F(M) \\
\tau\{\{M]\}_k, c(y_1). \text{ case } y_1 \text{ of } \{y_2\}_k \text{ in } \tau(H(y_2)) \cdot 0
\end{array}
\]

\[
\begin{array}{c}
2.1 \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k
\end{array}
\]

\[
\begin{array}{c}
2.2 \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k
\end{array}
\]

\[
\begin{array}{c}
2.3 \\
\tau\{\{M]\}_k, c(x). [x \text{ is } H(M)] F(M) \\
\tau\{\{M]\}_k, c(y_1). \text{ case } y_1 \text{ of } \{y_2\}_k \text{ in } \tau(H(y_2)) \cdot 0
\end{array}
\]

\[
\begin{array}{c}
3.1 \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k
\end{array}
\]

\[
\begin{array}{c}
3.2 \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k
\end{array}
\]

\[
\begin{array}{c}
3.3 \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k
\end{array}
\]

\[
\begin{array}{c}
4.1 \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k
\end{array}
\]

\[
\begin{array}{c}
4.2 \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k
\end{array}
\]

\[
\begin{array}{c}
4.3 \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k \\
\tau\{\{M]\}_k
\end{array}
\]
symbolic ES-LTS

2.2  \( c \{M\}_k \)
\( l_0 \)  \( l_1 \)
\( c(x).[x \text{ is } H(M)] \; F(M) \)
\( | \; \text{case } \gamma_0 \text{ of } \{y_2\}_k \; \text{in } \overline{e}(H(y_2)). \; 0 \)

3.2  \( c \{M\}_k \)
\( l_0 \)  \( l_1 \)
\( [\gamma_1 \text{ is } H(M)] \; F(M) \)
\( | \; \text{case } \gamma_0 \text{ of } \{y_2\}_k \; \text{in } \overline{e}(H(y_2)). \; 0 \)

2.2.a  \( c \{M\}_k \)
\( l_0 \)  \( l_1 \)
\( c(x).[x \text{ is } H(M)] \; F(M) \)
\( | \; \overline{e}(H(M)). \; 0 \)

3.2.a  \( c \{M\}_k \)
\( l_0 \)  \( l_1 \)
\( [\gamma_1 \text{ is } H(M)] \; F(M) \)
\( | \; \overline{e}(H(M)). \; 0 \)

3.2.a'  \( c \{M\}_k \)
\( l_0 \)  \( l_1 \)
\( [\gamma_1 \text{ is } H(M)] \; F(M) \)
\( | \; \overline{e}(H(M)). \; 0 \)

3.3  \( c \{M\}_k \)
\( l_0 \)
\( [\gamma_1 \text{ is } \ldots] \; F(M) \)
\( | \; \overline{e}(H(M)). \; 0 \)

4.2  \( c \{M\}_k \)
\( l_0 \)  \( l_1 \)  \( l_2 \)
\( [\gamma_1 \text{ is } H(M)] \; F(M) \)

4.2'  \( c \{M\}_k \)
\( l_0 \)  \( l_1 \)  \( l_2 \)
\( [\gamma_1 \text{ is } H(M)] \; F(M) \)

4.3  \( c \)
\( l_0 \)
\( [\gamma_1 \text{ is } \ldots] \; F(M) \)
Both concrete and symbolic traces have been defined.

Traces take into account only observable actions.

In symbolic traces:
- A specialization action is observable only when it is followed by an input or an output action;
- Sequences of specialization actions are collapsed into a single one;
- Each specialization action is merged with the following input or output action.
The concrete traces of a spi process $P$ with an initial intruder knowledge $K$ are denoted as $\text{ctr}(P, K)$.

In a similar way we denote the symbolic traces of $P$ as $\text{str}(P, K)$.

On symbolic traces a partial order $\subseteq_K$ is defined.

$\subseteq_K$ can be extended to sets of symbolic traces:

$$T_1 \subseteq_K T_2 \iff \forall t_1 \in T_1 \ \exists t_2 \in T_2 \ | \ t_1 \subseteq_K t_2$$
Let $P$ and $Q$ be two spi processes, and $K$ be a knowledge function with $\text{dom}(K) = \text{fn}(P) \cup \text{fn}(Q)$. Then,

- **Theorem**
  $$\text{ctr}(P, K) \subseteq \text{ctr}(Q, K) \iff \text{str}(P, K) \subseteq_K \text{str}(Q, K)$$

- **Corollary**
  $$\text{ctr}(P, K) = \text{ctr}(Q, K) \iff \text{str}(P, K) \simeq_K \text{str}(Q, K)$$

where $$\simeq_K = \subseteq_K \cap \subseteq_K^{-1}$$
traces

Let $P$ and $Q$ be any two spi processes, and $K$ an initial knowledge function with $\text{dom}(K) = \text{fn}(P) \cup \text{fn}(Q)$. Then

- **Soundness** $\text{ctr}(P, K) \subseteq \text{ctr}(Q, K) \Rightarrow P \sqsubseteq Q$

- **Completeness** $P \sqsubseteq Q \Rightarrow \text{ctr}(P, K) \subseteq \text{ctr}(Q, K)$

If a difference in the traces of two spi descriptions (typically a specification and a corresponding implementation of a protocol) is found, it can be used to automatically build the spi calculus description of an intruder process that can exploit the difference.
tool support

A prototypal tool has been recently implemented with the following structure and it has been tested with some classical authentication protocols: it can analyze systems with a few protocol sessions.

Diagram:

- **Spi-calculus descriptions**
  - $P$
  - $P_{\text{spec}}$

- **Parser & ES-LTS generator**
  - $P$
  - $P_{\text{spec}}$

- **Symbolic ES-LTSs**
  - $P$
  - $P_{\text{spec}}$

- **Trace Inclusion checker**
  - Traces not included

- **Spi-calculus Intruder constructor**
  - Intruder(s) in spi-calculus

---

Politecnico di Torino, October 2009
Luca Durante
Formal Verification of Cryptographic Protocols – p. 55/86
example - wmf authentication flaw
example - wmf authentication flaw

\[ 2' \quad S_1 \rightarrow I(B_1): \{k_{AB_1}\}^{k_{SB}}_{c_{SB_1}} \]
example - wmf authentication flaw

\[2'_1 \quad S_1 \rightarrow I(B_1) : \{ k_{AB_1} \} k_{SB} \text{ on } c_{SB_1}\]

\[2''_1 \quad I(S_1) \rightarrow B_1 : \{ k_{AB_1} \} k_{SB} \text{ on } c_{SB_1}\]
example - wmf authentication flaw

2'\) \ S_1 \rightarrow I(B_1) : \ \{k_{AB_1}\}k_{SB} \ on \ c_{SB_1}

2''\) \ I(S_1) \rightarrow B_1 : \ \{k_{AB_1}\}k_{SB} \ on \ c_{SB_1}

2''\) \ I(S_2) \rightarrow B_2 : \ \{k_{AB_1}\}k_{SB} \ on \ c_{SB_2}
example - wmf authentication flaw

2'\) \(S_1 \rightarrow I(B_1) : \{k_{AB_1}\}k_{SB} \text{ on } c_{SB_1}\)

2''\) \(I(S_1) \rightarrow B_1 : \{k_{AB_1}\}k_{SB} \text{ on } c_{SB_1}\)

2''\) \(I(S_2) \rightarrow B_2 : \{k_{AB_1}\}k_{SB} \text{ on } c_{SB_2}\)

3'\) \(A_1 \rightarrow I(B_1) : \{M_1\}k_{AB_1} \text{ on } c_{AB_1}\)
example - wmf authentication flaw

2\text{'}_1 \quad S_1 \rightarrow I(B_1) : \{k_{AB_1}\}k_{SB} \text{ on } c_{SB_1}

2\text{''}_1 \quad I(S_1) \rightarrow B_1 \quad : \{k_{AB_1}\}k_{SB} \text{ on } c_{SB_1}

2\text{''}_2 \quad I(S_2) \rightarrow B_2 \quad : \{k_{AB_1}\}k_{SB} \text{ on } c_{SB_2}

3\text{'}_1 \quad A_1 \rightarrow I(B_1) : \{M_1\}k_{AB_1} \text{ on } c_{AB_1}

3\text{''}_1 \quad I(A_1) \rightarrow B_1 \quad : \{M_1\}k_{AB_1} \text{ on } c_{AB_1}
example - wmf authentication flaw

2'₁) \( S_1 \rightarrow I(B_1) : \{k_{AB_1}\}k_{SB} \) on \( c_{SB_1} \)

2''₁) \( I(S_1) \rightarrow B_1 : \{k_{AB_1}\}k_{SB} \) on \( c_{SB_1} \)

2''₂) \( I(S_2) \rightarrow B_2 : \{k_{AB_1}\}k_{SB} \) on \( c_{SB_2} \)

3'₁) \( A_1 \rightarrow I(B_1) : \{M_1\}k_{AB_1} \) on \( c_{AB_1} \)

3''₁) \( I(A_1) \rightarrow B_1 : \{M_1\}k_{AB_1} \) on \( c_{AB_1} \)

3''₂) \( I(A_2) \rightarrow B_2 : \{M_1\}k_{AB_1} \) on \( c_{AB_2} \)
**example - wmf authentication flaw**

\[
P_A(M, c_{AS}, c_{AB}) \triangleq (\nu \ k_{AB}) \ (\overline{c_{AS}} \ \langle \{k_{AB}\} \ k_{AS}\rangle \ \cdot \ \overline{c_{AB}} \ \langle \{M\} \ k_{AB}\rangle)
\]

\[
P_S(c_{AS}, c_{SB}) \triangleq c_{AS}(x_1) \ \cdot \ \text{case} \ \ x_1 \ \text{of} \ \ \{x_2\} \ k_{AS} \ \text{in} \ \overline{c_{SB}} \ \langle \{x_2\} \ k_{SB}\rangle
\]

\[
P_B(c_{SB}, c_{AB}, c_F) \triangleq c_{SB}(y_1) \ \cdot \ \text{case} \ \ y_1 \ \text{of} \ \ \{y_2\} \ k_{SB} \ \text{in} \ \overline{c_{AB}}(y_3) \ \cdot \ \text{case} \ \ y_3 \ \text{of} \ \ \{y_4\} \ y_2 \ \text{in} \ \overline{c_F}(y_4)
\]

\[
P_{\text{spec}}(M, c_{SB}, c_{AB}, c_F) \triangleq c_{SB}(y_1) \ \cdot \ \text{case} \ \ y_1 \ \text{of} \ \ \{y_2\} \ k_{SB} \ \text{in} \ \overline{c_{AB}}(y_3) \ \cdot \ \text{case} \ \ y_3 \ \text{of} \ \ \{y_4\} \ y_2 \ \text{in} \ \overline{c_F}(M)
\]

\[
P_{\text{wmf}}(M_1, M_2) \triangleq (\nu \ k_{AS})(\nu \ k_{SB})(
\begin{align*}
P_A(M_1, c_{AS_1}, c_{AB_1}) &| P_S(c_{AS_1}, c_{SB_1}) &| P_B(c_{SB_1}, c_{AB_1}, c_F_1) \\
P_A(M_2, c_{AS_2}, c_{AB_2}) &| P_S(c_{AS_2}, c_{SB_2}) &| P_B(c_{SB_2}, c_{AB_2}, c_F_2)
\end{align*}
\]

\[
P_{\text{wmf}_{\text{spec}}}(M_1, M_2) \triangleq (\nu \ k_{AS})(\nu \ k_{SB})(
\begin{align*}
P_A(M_1, c_{AS_1}, c_{AB_1}) &| P_S(c_{AS_1}, c_{SB_1}) &| P_{\text{spec}}(M_1, c_{SB_1}, c_{AB_1}, c_F_1) \\
P_A(M_2, c_{AS_2}, c_{AB_2}) &| P_S(c_{AS_2}, c_{SB_2}) &| P_{\text{spec}}(M_2, c_{SB_2}, c_{AB_2}, c_F_2)
\end{align*}
\]

\[
\begin{array}{ll}
2'_1 \ \delta_1 & \rightarrow \ I(B_1) : \ \{k_{AB_1}\} \ k_{SB} \ \text{on} \ c_{SB_1} \\
2''_1 \ \delta_3 & \rightarrow \ I(S_1) \ \rightarrow \ B_1 : \ \{k_{AB_1}\} \ k_{SB} \ \text{on} \ c_{SB_1} \\
3'_1 & \delta_2 \ 
\rightarrow \ I(S_2) \ 
\rightarrow \ B_2 : \ \{k_{AB_1}\} \ k_{SB} \ \text{on} \ c_{SB_2} \\
3''_1 \ \delta_3 & \rightarrow \ I(A_1) \ \rightarrow \ B_1 : \ \{M_1\} \ k_{AB_1} \ \text{on} \ c_{AB_1} \\
3'''_1 & \delta_3 \ 
\rightarrow \ I(A_2) \ \rightarrow \ B_2 : \ \{M_1\} \ k_{AB_1} \ \text{on} \ c_{AB_2}
\end{array}
\]

\[
\begin{array}{ll}
2'_1 & \triangleq \ c_{SB_1}(l_{10}). \\
2''_1 & \triangleq \overline{c_{SB_1}}(l_{10}). \\
2''_2 & \triangleq \overline{c_{SB_2}}(l_{10}). \\
3'_1 & \triangleq \ c_{AB_1}(l_{11}). \\
3''_1 & \triangleq \overline{c_{AB_1}}(l_{11}). \\
3'''_1 & \triangleq \overline{c_{AB_2}}(l_{11}). \\
\end{array}
\]

\[
\begin{array}{ll}
f_{c_{F_1}}(x_1). & \text{[}x_1 \ \text{is} \ M_1] \\
f_{c_{F_2}}(x_2). & \text{[}x_2 \ \text{is} \ M_1]. \ \Omega
\end{array}
\]

\[
\begin{array}{ll}
2'_1 & \downarrow (\tau, T). (l_3, \langle \top, \emptyset \rangle, \langle \emptyset, \emptyset, l_{10} \rangle) \\
& \downarrow (l_3, \langle \emptyset, \emptyset, l_{10} \rangle) \\
2''_1 & \downarrow (\tau, T). (l_3, \gamma_0) \\
& \downarrow (l_3, l_{10}) \\
2''_2 & \downarrow (\tau, (l_{10}/\gamma_0)). (l_8, \gamma_1) \\
& \downarrow (l_8, l_{10}) \\
3'_1 & \downarrow (\tau, (l_{10}/\gamma_1)). (l_1, \langle \top, \emptyset \rangle, \langle \emptyset, \emptyset, l_{11} \rangle) \\
& \downarrow (l_1, \langle \emptyset, \emptyset, l_{11} \rangle) \\
3''_1 & \downarrow (\tau, T). (l_{11}, \gamma_2) \\
3'''_1 & \downarrow (l_{11}, l_{11}) \\
3''''_1 & \downarrow (\tau, (l_{11}/\gamma_2)). (l_6, \gamma_3) \\
& \downarrow (l_6, l_{11}) \\
\end{array}
\]

\[
\begin{array}{ll}
3'_1 & \downarrow (\tau, (l_{11}/\gamma_3)). (l_4, \langle \top, \emptyset \rangle, \langle \emptyset, \emptyset, l_{2} \rangle) \\
& \downarrow (l_4, \langle \emptyset, \emptyset, l_{2} \rangle) \\
3''_1 & \downarrow (l_{11}/\gamma_3)). (l_9, \langle \emptyset, \emptyset, l_{2} \rangle) \\
& \downarrow (l_9, \langle \emptyset, \emptyset, l_{2} \rangle)
\end{array}
\]

symbolic trace

concrete trace
S\textsuperscript{3}A [24], OFMC [7], STA [10] and Casper [31] have been tested [14] in their

- error-detection capabilities (w.r.t. known flaws)
- performances (analyzing protocols without known flaws)

on a Subset of the Security Protocols Open Repository [38]: only protocols that can be analyzed by all the tools have been taken into account.
<table>
<thead>
<tr>
<th>#</th>
<th>Protocol</th>
<th>Attack Type</th>
<th>S(^3)A</th>
<th>OFMC</th>
<th>STA</th>
<th>Casper</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Andrew Secure RPC</td>
<td>Freshness</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>BAN mod. Andrew Sec. RPC</td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>BAN concrete Andrew Sec. RPC</td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>CCITT x509 (3)</td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>Denning-Sacco shared key</td>
<td>Freshness</td>
<td>Y</td>
<td>Y</td>
<td>N(^h)</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>Kao Chow authentication 1</td>
<td>Freshness</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>KSL (rep. part)</td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>KSL</td>
<td>Parallel session</td>
<td>Y</td>
<td>N(^e)</td>
<td>Y</td>
<td>N(^r)</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Parallel session</td>
<td>Y</td>
<td>N(^e)</td>
<td>N(^f)</td>
<td>N(^r)</td>
</tr>
<tr>
<td>11</td>
<td>Neumann Stubblebine (rep. part)</td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>12</td>
<td>Neumann Stubblebine</td>
<td>Type-flaw</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N(^h)</td>
</tr>
</tbody>
</table>
# experimental results - error-detection

<table>
<thead>
<tr>
<th>#</th>
<th>Protocol</th>
<th>Attack Type</th>
<th>$S^3A$</th>
<th>OFMC</th>
<th>STA</th>
<th>Casper</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Needham-Schroeder Public Key</td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>14</td>
<td>Needham-Schroeder Symmetric Key</td>
<td>Freshness</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>15</td>
<td>Otway Rees</td>
<td>Type-flaw</td>
<td>$N^h$</td>
<td>Y</td>
<td>$N^h$</td>
<td>N</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>Type-flaw</td>
<td>$N^h$</td>
<td>Y</td>
<td>$N^h$</td>
<td>N</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>Type-flaw</td>
<td>$N^h$</td>
<td>$N^f$</td>
<td>$N^h$</td>
<td>N</td>
</tr>
<tr>
<td>18</td>
<td>SPLICE/AS</td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>Binding</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>Parallel session</td>
<td>$N^r$</td>
<td>Y</td>
<td>Y</td>
<td>$N^r$</td>
</tr>
<tr>
<td>21</td>
<td>Hwang/Chen mod. SPLICE/AS</td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>22</td>
<td>Clark/Jacob mod. Hwang/Chen</td>
<td>Freshness</td>
<td>$N^r$</td>
<td>Y</td>
<td>N</td>
<td>$N^r$</td>
</tr>
<tr>
<td>23</td>
<td>TMN</td>
<td>Other</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>Other</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>Parallel session</td>
<td>$N^r$</td>
<td>$N^f$</td>
<td>$N^f$</td>
<td>N</td>
</tr>
</tbody>
</table>
## experimental results - error-detection

<table>
<thead>
<tr>
<th>#</th>
<th>Protocol</th>
<th>Attack Type</th>
<th>$S^3A$</th>
<th>OFMC</th>
<th>STA</th>
<th>Casper</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Woo/Lam mutual authentication</td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>Type-flaw</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N&lt;sup&gt;h&lt;/sup&gt;</td>
</tr>
<tr>
<td>28</td>
<td>Woo/Lam $\Pi$</td>
<td>Parallel session</td>
<td>Y</td>
<td>N&lt;sup&gt;f&lt;/sup&gt;</td>
<td>N&lt;sup&gt;f&lt;/sup&gt;</td>
<td>N</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>31</td>
<td>Woo/Lam $\Pi^1$</td>
<td>Type-flaw</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N&lt;sup&gt;h&lt;/sup&gt;</td>
</tr>
<tr>
<td>32</td>
<td>Woo/Lam $\Pi^2$</td>
<td>Type-flaw</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N&lt;sup&gt;h&lt;/sup&gt;</td>
</tr>
<tr>
<td>33</td>
<td>Woo/Lam $\Pi^3$</td>
<td>Type-flaw</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N&lt;sup&gt;h&lt;/sup&gt;</td>
</tr>
<tr>
<td>34</td>
<td>Yahalom</td>
<td>Type-flaw</td>
<td>N&lt;sup&gt;h&lt;/sup&gt;</td>
<td>Y</td>
<td>N&lt;sup&gt;h&lt;/sup&gt;</td>
<td>N</td>
</tr>
<tr>
<td>35</td>
<td>BAN simplified Yahalom</td>
<td>Type-flaw</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>36</td>
<td></td>
<td>Parallel session</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>37</td>
<td></td>
<td>Type-flaw</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td><strong>30</strong></td>
<td><strong>32</strong></td>
<td><strong>28</strong></td>
<td><strong>18</strong></td>
</tr>
</tbody>
</table>
### Distribution of the Test Results by Type

<table>
<thead>
<tr>
<th>Key</th>
<th>Meaning of the Test Result</th>
<th>$^{3}S$</th>
<th>OFMC</th>
<th>STA</th>
<th>Casper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Flaw detected with a standard specification</td>
<td>30</td>
<td>32</td>
<td>28</td>
<td>18</td>
</tr>
<tr>
<td>N</td>
<td>Flaw undetected</td>
<td>—</td>
<td>—</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$N^h$</td>
<td>Flaw detected only with a custom specification</td>
<td>4</td>
<td>—</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$N^r$</td>
<td>Resources exhausted.</td>
<td>3</td>
<td>—</td>
<td>—</td>
<td>4</td>
</tr>
<tr>
<td>$N^f$</td>
<td>The tool stops when it detects the first flaw</td>
<td>—</td>
<td>3</td>
<td>3</td>
<td>—</td>
</tr>
<tr>
<td>$N^e$</td>
<td>Internal error or no result</td>
<td>—</td>
<td>2</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
## Experimental Results - Execution Times

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Scenario</th>
<th>S²A</th>
<th>OFMC</th>
<th>STA</th>
<th>Casper</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAN modified</td>
<td>1A 1B</td>
<td>0.04 s</td>
<td>0.01 s</td>
<td>0.15 s</td>
<td>N⁺r</td>
</tr>
<tr>
<td>CCITT X.509 (3)</td>
<td>2A 1B</td>
<td>2.11 s</td>
<td>0.02 s</td>
<td>1218.33 s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2A 2B</td>
<td>N⁺r</td>
<td>0.07 s</td>
<td>N⁺r</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3A 2B</td>
<td>0.24 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3A 3B</td>
<td>1.61 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4A 3B</td>
<td>5.62 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4A 4B</td>
<td>N⁺r</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lowe modified</td>
<td>1A 1B 1S</td>
<td>0.07 s</td>
<td>0.01 s</td>
<td>0.03 s</td>
<td>76.8 s</td>
</tr>
<tr>
<td>Denning-Sacco shared key</td>
<td>2A 1B 1S</td>
<td>1.50 s</td>
<td>0.03 s</td>
<td>1.48 s</td>
<td>154.26 s</td>
</tr>
<tr>
<td></td>
<td>2A 2B 1S</td>
<td>611.20 s</td>
<td>0.03 s</td>
<td>109.21 s</td>
<td>168.52 s</td>
</tr>
<tr>
<td></td>
<td>2A 2B 2S</td>
<td>N⁺r</td>
<td>0.11 s</td>
<td>2289.59 s</td>
<td>N⁺r</td>
</tr>
<tr>
<td></td>
<td>3A 3B 3S</td>
<td>3.93 s</td>
<td></td>
<td></td>
<td>N⁺r</td>
</tr>
<tr>
<td></td>
<td>4A 4B 4S</td>
<td>N⁺r</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**experimental results - execution times**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Scenario</th>
<th>$S^{3A}$</th>
<th>OFMC</th>
<th>STA</th>
<th>Casper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kao Chow Authentication v. 2</td>
<td>1A 1B 1S</td>
<td>0.15 s</td>
<td>0.02 s</td>
<td>0.12 s</td>
<td>15.20 s</td>
</tr>
<tr>
<td></td>
<td>2A 1B 1S</td>
<td>1.70 s</td>
<td>0.03 s</td>
<td>0.44 s</td>
<td>21.82 s</td>
</tr>
<tr>
<td></td>
<td>2A 2B 1S</td>
<td>21.20 s</td>
<td>0.03 s</td>
<td>7.71 s</td>
<td>29.35 s</td>
</tr>
<tr>
<td></td>
<td>2A 2B 2S</td>
<td>$N^r$</td>
<td>0.36 s</td>
<td>1162.87 s</td>
<td>$N^r$</td>
</tr>
<tr>
<td></td>
<td>3A 3B 3S</td>
<td>$N^r$</td>
<td>$N^r$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kao Chow Authentication v. 3</td>
<td>1A 1B 1S</td>
<td>0.04 s</td>
<td>0.03 s</td>
<td>0.40 s</td>
<td>216.10 s</td>
</tr>
<tr>
<td></td>
<td>2A 1B 1S</td>
<td>2.09 s</td>
<td>0.03 s</td>
<td>1.33 s</td>
<td>260.30 s</td>
</tr>
<tr>
<td></td>
<td>2A 2B 1S</td>
<td>26.46 s</td>
<td>0.04 s</td>
<td>14.11 s</td>
<td>416.49 s</td>
</tr>
<tr>
<td></td>
<td>2A 2B 2S</td>
<td>$N^r$</td>
<td>0.45 s</td>
<td>$N^r$</td>
<td>$N^r$</td>
</tr>
<tr>
<td></td>
<td>3A 2B 2S</td>
<td>1.42 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3A 3B 2S</td>
<td>12.70 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3A 3B 3S</td>
<td>$N^r$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Experimental Results - Execution Times

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Scenario</th>
<th>$S^3A$</th>
<th>OFMC</th>
<th>STA</th>
<th>Casper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amended Needham Schroeder</td>
<td>1A 1B 1S</td>
<td>0.69 s</td>
<td>0.06 s</td>
<td>0.40 s</td>
<td>$N^r$</td>
</tr>
<tr>
<td>Symmetric Key</td>
<td>2A 1B 1S</td>
<td>307.10 s</td>
<td>0.16 s</td>
<td>97.20 s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2A 2B 1S</td>
<td>$N^r$</td>
<td>0.47 s</td>
<td>$N^r$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2A 2B 2S</td>
<td>3.58 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3A 2B 2S</td>
<td>$N^e$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lowe modified Yahalom</td>
<td>1A 1B 1S</td>
<td>0.07 s</td>
<td>0.06 s</td>
<td>0.03 s</td>
<td>11.46 s</td>
</tr>
<tr>
<td></td>
<td>2A 1B 1S</td>
<td>4.33 s</td>
<td>0.20 s</td>
<td>0.37 s</td>
<td>12.76 s</td>
</tr>
<tr>
<td></td>
<td>2A 2B 1S</td>
<td>3906.00 s</td>
<td>2.88 s</td>
<td>74.90 s</td>
<td>14.25 s</td>
</tr>
<tr>
<td></td>
<td>2A 2B 2S</td>
<td>$N^r$</td>
<td>$N^r$</td>
<td>1534.90 s</td>
<td>20.96 s</td>
</tr>
<tr>
<td></td>
<td>3A 2B 2S</td>
<td>$N^r$</td>
<td></td>
<td>385.60 s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3A 3B 2S</td>
<td></td>
<td></td>
<td></td>
<td>$N^r$</td>
</tr>
</tbody>
</table>
## Experimental Results - Execution Times

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Scenario</th>
<th>$S^3A$</th>
<th>OFMC</th>
<th>STA</th>
<th>Casper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paulson’s</td>
<td>1A 1B 1S</td>
<td>0.06 s</td>
<td>0.02 s</td>
<td>0.07 s</td>
<td>99.19 s</td>
</tr>
<tr>
<td>strengthened Yahalom</td>
<td>2A 1B 1S</td>
<td>2.95 s</td>
<td>0.03 s</td>
<td>0.62 s</td>
<td>124.25 s</td>
</tr>
<tr>
<td></td>
<td>2A 2B 1S</td>
<td>1858.10 s</td>
<td>0.21 s</td>
<td>73.20 s</td>
<td>154.28 s</td>
</tr>
<tr>
<td></td>
<td>2A 2B 2S</td>
<td>N$^r$</td>
<td>1.89 s</td>
<td>N$^r$</td>
<td>377.13 s</td>
</tr>
<tr>
<td></td>
<td>3A 2B 2S</td>
<td>N$^e$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
related work

Boreale, De Nicola, Pugliese [12]:

- Testing equivalence verification of a spi calculus dialect:
  - public key encryption, hashing and integers are missing;
  - keys are atomic.
- Theorem proving approach.
- Based on "trace equivalence" (sound and complete) defined on an ES-LTS similar to our one, but
  - environment knowledge is not minimized;
  - transition labels are redundant, but at the same time not enough to define traces: "trace equivalence" requires step-by-step equivalence of environment knowledge.
related work

Symbolic techniques (Amadio & Lugiez [5], Huima [30], Boreale [9, 11], Fiore & Abadi [26], Millen & Shmatikov [35]):

- Reachability analysis only (no testing equivalence checking).
- Public channels only.
- Knowledge minimization is not addressed.
- Only Boreale and Fiore & Abadi deal with spi calculus (but they use simplified language versions).
- Fiore & Abadi and Millen & Shmatikov deal with non-atomic keys, but with limitations.
Symbolic techniques (Basin & Mödersheim & Viganò [7]):

- Reachability analysis only (no testing equivalence checking).
- Public channels only.
- Knowledge minimization is not addressed.
- Protocols are described by means of HLPSL (High-Level Protocol Specification Language).
- Full support to non-atomic keys.
- Limited support for operators with algebraic properties, (i.e. exponentiation).
- Besides lazy intruder, also lazy demand-driven state exploration.
related work

Clarke, Jha & Marrero [18, 19]:

- Reachability analysis only.
- Protocol description language less expressive than spi calculus.
- State exploration.
- No symbolic technique is used, but the length of messages from the intruder is restricted.
- The environment knowledge is minimized, but knowledge management relies on atomic keys.
- The technique is implemented by a tool (Brutus), which however is not publicly distributed.
related work

Song [40]:
- Reachability analysis only.
- Protocol description language less expressive than spi calculus (atomic keys only, public channels only, tests allowed only while receiving).
- Uses a proof technique based on strand spaces.
- Handles infinite models (with unbounded number of sessions).
- Proofs are fully automated, but termination is not guaranteed.
- The technique is implemented by a tool (Athena), which is available in a prototypical version.
our results

- A new sound and complete technique for checking testing equivalence by state exploration.

- Deals with the whole spi calculus (with a bound only on the number of parallel processes).

- Addresses performance issues by:
  - minimized intruder knowledge representation;
  - symbolic data representation;
  - (irredundant) trace labels and simple trace inclusion checks.

- Can be fully automated and yields the spi calculus description(s) of possible intruders.

- Has tool support.
future work

- Theory can be further improved:
  - extending the language (recursion, etc.);
  - dealing with unbounded number of sessions;
  - dealing with algebraic operators (partially done [16])
  - introducing reduction techniques (based on symmetry, partial order, etc.) (partially done [17]).

- The tool can be further improved:
  - on-the-fly checking;
  - user interface/integrated environment;
  - automatic code generation (partially done [23]).
future work

A single technique and/or tool is not enough to solve all the problems, so new challenges are:

- integrated frameworks where more tools and techniques cooperate, e.g. model checking, theorem provers, and so on
- new techniques able to exploit the compositionality of the analysis
- new techniques able to deal with new cryptographic protocols, e.g. *open-ended* protocols
**App. A - reaction operational semantics**

- reduction relation $\succ$

\[
[\sigma \text{ is } \sigma] \quad P \succ P
\]

\[
let \ (x, y) = (\sigma, \rho) \text{ in } P \succ P[\sigma/x, \rho/y]
\]

\[
case \ 0 \ of \ 0 : P \ \text{suc}(x) : Q \succ P
\]

\[
case \ \text{suc}(\sigma) \ of \ 0 : P \ \text{suc}(x) : Q \succ Q[\sigma/x]
\]

\[
case \ \{\sigma\}\rho \ of \ \{x\}\rho \text{ in } P \succ P[\sigma/x]
\]

\[
case \ \{[\sigma]\}\rho^+ \ of \ \{[x]\}\rho^- \text{ in } P \succ P[\sigma/x]
\]

\[
case \ \{\{\sigma\}\}\rho^- \ of \ \{\{x\}\}\rho^+ \text{ in } P \succ P[\sigma/x]
\]
App. A - reaction operational semantics

- structural equivalence \( \equiv \)

\[
\begin{align*}
P & \equiv P \\
P \parallel 0 & \equiv P \\
P \parallel Q & \equiv Q \parallel P \\
(P \parallel Q) \parallel R & \equiv (P \parallel Q) \parallel R \\
(\nu m_1)(\nu m_2) P & \equiv (\nu m_2)(\nu m_1) P \\
(\nu b)0 & \equiv 0 \\
\end{align*}
\]

\[
\begin{align*}
(m \not\in \text{fn}(P)) & \Rightarrow (\nu m)(P \parallel Q) \equiv P \parallel (\nu m)Q \\
(P \parallel Q) \equiv P & \parallel Q \\

\end{align*}
\]

\[
\begin{align*}
P \equiv Q & \Rightarrow Q \equiv R \\
(P \equiv R) & \Rightarrow P \equiv P' \\
(P \equiv Q \parallel Q') & \Rightarrow Q \equiv P \\
(\nu m)P \equiv (\nu m)P' \\
\end{align*}
\]
reaction relation $\rightarrow$

\[ \overline{\sigma}(\rho).P | \sigma(x).Q \rightarrow P | Q[\rho/x] \]

\[
\begin{align*}
\frac{P \equiv P'}{P \rightarrow Q} & \quad \frac{P \rightarrow P'}{P | Q \rightarrow P'|Q} & \quad \frac{P \rightarrow P'}{(\nu m)P \rightarrow (\nu m)P'}
\end{align*}
\]

\[ P \equiv P' \quad P' \rightarrow Q' \quad Q' \equiv Q \]
App. B - closure rules

\[ \sigma \in \Sigma \Rightarrow \sigma \in \hat{\Sigma} \quad (1) \]
\[ \sigma \in \hat{\Sigma} \Rightarrow suc(\sigma) \in \hat{\Sigma} \quad (2) \text{ (successor)} \]
\[ \sigma_1 \in \hat{\Sigma} \land \sigma_2 \in \hat{\Sigma} \Rightarrow (\sigma_1, \sigma_2) \in \hat{\Sigma} \quad (3) \text{ (pairing)} \]
\[ \sigma_1 \in \hat{\Sigma} \land \sigma_2 \in \hat{\Sigma} \Rightarrow \{\sigma_1\}_{\sigma_2} \in \hat{\Sigma} \quad (4) \text{ (shared key encryption)} \]
\[ \sigma \in \hat{\Sigma} \Rightarrow H(\sigma) \in \hat{\Sigma} \quad (5) \text{ (hashing)} \]
\[ \sigma_1 \in \hat{\Sigma} \land \sigma_2^+ \in \hat{\Sigma} \Rightarrow \{[\sigma_1]\}_{\sigma_2^+} \in \hat{\Sigma} \quad (6) \text{ (public key encryption)} \]
\[ \sigma_1 \in \hat{\Sigma} \land \sigma_2^- \in \hat{\Sigma} \Rightarrow \{[\sigma_1]\}_{\sigma_2^-} \in \hat{\Sigma} \quad (7) \text{ (private key signature)} \]
\[ \sigma \in \hat{\Sigma} \Rightarrow \sigma^+ \in \hat{\Sigma} \land \sigma^- \in \hat{\Sigma} \quad (8) \text{ (key pair extraction)} \]
App. B - closure rules

\[ \text{suc}(\sigma) \in \hat{\Sigma} \Rightarrow \sigma \in \hat{\Sigma} \]  (9) (predecessor)

\[ (\sigma_1, \sigma_2) \in \hat{\Sigma} \Rightarrow \sigma_1 \in \hat{\Sigma} \land \sigma_2 \in \hat{\Sigma} \]  (10) (projection)

\[ \{\sigma_1\}_{\sigma_2} \in \hat{\Sigma} \land \sigma_2 \in \hat{\Sigma} \Rightarrow \sigma_1 \in \hat{\Sigma} \]  (11) (shared key decryption)

\[ \{[\sigma_1]\}_{\sigma_2^+} \in \hat{\Sigma} \land \sigma_2^- \in \hat{\Sigma} \Rightarrow \sigma_1 \in \hat{\Sigma} \]  (12) (public key decryption)

\[ \{[\sigma_1]\}_{\sigma_2^-} \in \hat{\Sigma} \land \sigma_2^+ \in \hat{\Sigma} \Rightarrow \sigma_1 \in \hat{\Sigma} \]  (13) (signature check)

\[ \sigma^+ \in \hat{\Sigma} \land \sigma^- \in \hat{\Sigma} \Rightarrow \sigma \in \hat{\Sigma} \]  (14) (key pair reconstruction)
App. B - closure seed rules

\[ m \in \overline{\Sigma} \iff m \in \hat{\Sigma} \]

\[ \text{suc}(\sigma) \notin \overline{\Sigma} \]

\[ (\sigma_1, \sigma_2) \notin \overline{\Sigma} \]

\[ \{\sigma_1\}\sigma_2 \in \overline{\Sigma} \iff \sigma_2 \notin \hat{\Sigma} \]

\[ H(\sigma) \in \overline{\Sigma} \iff \sigma \notin \hat{\Sigma} \]

\[ \{[\sigma_1]\}\sigma_2^+ \in \overline{\Sigma} \iff \sigma_2^+ \notin \hat{\Sigma} \lor \sigma_1 \notin \hat{\Sigma} \]

\[ \{[\sigma_1]\}\sigma_2^- \in \overline{\Sigma} \iff \sigma_2^- \notin \hat{\Sigma} \lor \sigma_1 \notin \hat{\Sigma} \]

\[ \sigma^+ \in \overline{\Sigma} \iff \sigma \notin \hat{\Sigma} \]

\[ \sigma^- \in \overline{\Sigma} \iff \sigma \notin \hat{\Sigma} \]
App. C - reduction rules

\[ H(\sigma) \in \Sigma_i \land r(\sigma, \Sigma_1) \leadsto \langle \{ H(\sigma) \}, (5), \emptyset \rangle \in R(\Sigma_i) \]

\[ \{\sigma_1\}\sigma_2^+ \in \Sigma_i \land r(\sigma_1, \Sigma_1) \land r(\sigma_2^+, \Sigma_1) \leadsto \langle \{\{\sigma_1\}\sigma_2^+\}, (6), \emptyset \rangle \in R(\Sigma_i) \]

\[ \{\sigma_1\}\sigma_2^- \in \Sigma_i \land r(\sigma_1, \Sigma_1) \land r(\sigma_2^-, \Sigma_1) \leadsto \langle \{\{\sigma_1\}\sigma_2^-\}, (7), \emptyset \rangle \in R(\Sigma_i) \]

\[ \sigma^+ \in \Sigma_i \land r(\sigma, \Sigma_1) \leadsto \langle \{\sigma^+\}, (8), \emptyset \rangle \in R(\Sigma_i) \]

\[ \sigma^- \in \Sigma_i \land r(\sigma, \Sigma_1) \leadsto \langle \{\sigma^-\}, (8), \emptyset \rangle \in R(\Sigma_i) \]

\[ \text{suc}(\sigma) \in \Sigma_i \leadsto \langle \{\text{suc}(\sigma)\}, (9), \{\sigma\} \rangle \in R(\Sigma_i) \]

\[ (\sigma_1, \sigma_2) \in \Sigma_i \leadsto \langle \{(\sigma_1, \sigma_2)\}, (10), \{\sigma_1, \sigma_2\} \rangle \in R(\Sigma_i) \]

\[ \{\sigma_1\}\sigma_2 \in \Sigma_i \land r(\sigma_2, \Sigma_1) \leadsto \langle \{\{\sigma_1\}\sigma_2\}, (11), \{\sigma_1\} \rangle \in R(\Sigma_i) \]

\[ \{\sigma_1\}\sigma_2^+ \in \Sigma_i \land r(\sigma_2^-, \Sigma_1) \land \neg r(\sigma_1, \Sigma_1) \leadsto \langle \{\{\sigma_1\}\sigma_2^+\}, (12), \{\sigma_1, \{\sigma_1\}\sigma_2^+\} \rangle \in R(\Sigma_i) \]

\[ \{\sigma_1\}\sigma_2^- \in \Sigma_i \land r(\sigma_2^+, \Sigma_1) \land \neg r(\sigma_1, \Sigma_1) \leadsto \langle \{\{\sigma_1\}\sigma_2^-\}, (13), \{\sigma_1, \{\sigma_1\}\sigma_2^-\} \rangle \in R(\Sigma_i) \]

\[ \sigma^+ \in \Sigma_i \land \sigma^- \in \Sigma_i \leadsto \langle \{\sigma^+, \sigma^-\}, (14), \{\sigma\} \rangle \in R(\Sigma_i) \]
boolean \( r(\sigma, \Sigma) \) {
    if \( \sigma \in \Sigma \) then return TRUE;
    else if \( \sigma = \text{suc}(\sigma_1) \) then return \( r(\sigma_1, \Sigma) \);
    else if \( \sigma = (\sigma_1, \sigma_2) \) then return \( r(\sigma_1, \Sigma) \land r(\sigma_2, \Sigma) \);
    else if \( \sigma = \{\sigma_1\}_{\sigma_2} \) then return \( r(\sigma_1, \Sigma) \land r(\sigma_2, \Sigma) \);
    else if \( \sigma = H(\sigma_1) \) then return \( r(\sigma_1, \Sigma) \);
    else if \( \sigma = \{[\sigma_1]\}_{\sigma_2^+} \) then return \( r(\sigma_1, \Sigma) \land r(\sigma_2^+, \Sigma) \);
    else if \( \sigma = [\{\sigma_1\}]_{\sigma_2^-} \) then return \( r(\sigma_1, \Sigma) \land r(\sigma_2^-, \Sigma) \);
    else if \( \sigma = \sigma_1^+ \) then return \( r(\sigma_1, \Sigma) \);
    else if \( \sigma = \sigma_1^- \) then return \( r(\sigma_1, \Sigma) \);
    else (\( \sigma \in \mathcal{A} \setminus \Sigma \)) return FALSE;
}
App. D - concrete ES-LTS derivation rules

\[
P \equiv P' \quad K_1 \triangleright P' \quad \frac{\mu}{\phi} \quad K_2 \triangleright Q' \quad Q' \equiv Q
\]

\[
K_1 \triangleright P \quad \frac{\mu}{\phi} \quad K_2 \triangleright Q
\]

\[
P \rightarrow P' \quad K \triangleright P \quad \frac{\tau}{\phi} \quad K \triangleright P'
\]

\[
K \triangleright P \quad \frac{\mu}{\phi} \quad K' \triangleright P'
\]

\[
K \triangleright (\nu b) P \quad \frac{\mu}{\phi} \quad K' \triangleright (\nu b) P'
\]

\[
K \vdash \sigma \quad K' = f(\rho, K)
\]

\[
K \triangleright \bar{\sigma}(\rho).P \quad \frac{\sigma[K']}{\bar{\sigma}[K]} \quad K' \triangleright P
\]

\[
K \vdash \sigma \quad K \vdash \rho
\]

\[
K \triangleright \sigma(x).P \quad \frac{\sigma[K]}{\rho[K]} \quad K \triangleright P[\rho/x]
\]
\[ \delta_K^-(\rho) = \left\{ \langle \theta, \theta[K] \rangle \mid \theta \in \delta^-_\Sigma(\rho) \right\} \quad \text{with} \quad \overline{\Sigma} = \text{dom}(K) \]

\[ \delta_K^+(\rho) = \left\{ \left\langle \{\{\sigma_1\}\}_{\sigma_2^-}, \{\{\sigma_1\}\}_{\sigma_2^+} \right\rangle \mid \{\{\sigma_1\}\}_{\sigma_2^+} \in \delta^+_{\overline{\Sigma}}(\rho) \right\} \cup \left\{ \left\langle \{\{\sigma_1\}\}_{\sigma_2^+}, \{\{\sigma_1\}\}_{\sigma_2^-} \right\rangle \mid \{\{\sigma_1\}\}_{\sigma_2^-} \in \delta^+_{\overline{\Sigma}}(\rho) \right\} \]

\[ \delta_K = \langle \delta_K^-(\rho), \delta_K^+(\rho), \rho \rangle[K'] = \langle \delta_K^-(\rho)[K'], \delta_K^+(\rho)[K'], \rho[K'] \rangle \]

i.e. \( \delta_K^-(\rho)[K'] \) takes the form \( (\theta[K'], \theta[K][K']) = (\theta[K'], \theta[K]) \)
\[(K \triangleright P)_{\Upsilon,\Lambda}\{\xi\} = (K[\xi] \triangleright P[\xi])_{\Upsilon[\xi],\Lambda[\xi]} = (K' \triangleright P')_{\Upsilon',\Lambda'}\]

\[f_{\langle\xi,\delta\Lambda\rangle}(\rho, K_{\Upsilon,\Lambda}) = f(\rho[\xi], K[\xi])_{\Upsilon[\xi],(\Lambda \cup \delta\Lambda)[\xi]}\]

\[\Theta(\rho, K_{\Upsilon,\Lambda}) = \{\langle\xi, \delta\Lambda\rangle \mid \xi \in S_{\Upsilon,\Lambda}, \delta\Lambda \in S_{\Upsilon}, f_{\langle\xi,\delta\Lambda\rangle}(\rho, K_{\Upsilon,\Lambda}) \in K_{\Upsilon,\Lambda}\}\]

\[
\begin{align*}
(K \triangleright P)_{\Upsilon,\Lambda} \xrightarrow{\xi[K[\xi]]} & (K \triangleright P')_{\Upsilon,\Lambda}\{\xi\} \\
(K \triangleright (P|Q))_{\Upsilon,\Lambda} \xrightarrow{\xi[K[\xi]]} & (K \triangleright (P'|Q))_{\Upsilon,\Lambda}\{\xi\}
\end{align*}
\]

\[
\begin{align*}
\xi \in & \sigma \bullet \rho \quad \xi \notin \top \\
(K \triangleright (\sigma(\theta).P \mid \rho(x).Q))_{\Upsilon,\Lambda} \xrightarrow{\xi[K[\xi]]} & (K \triangleright (\sigma(\theta).P \mid \rho(x).Q))_{\Upsilon,\Lambda}\{\xi\}
\end{align*}
\]

\[
\begin{align*}
\gamma',\gamma'' \notin & \text{dom}(\Upsilon) \\
(K \triangleright (\text{let } (x,y) = \gamma \text{ in } P))_{\Upsilon,\Lambda} \xrightarrow{\gamma'\gamma''/\gamma} & (K \triangleright (\text{let } (x,y) = \gamma \text{ in } P))_{\Upsilon,\Lambda}\{(\gamma',\gamma'')/\gamma\}
\end{align*}
\]
App. E - symbolic ES-LTS derivation rules

\[ (K \triangleright (\text{case } \gamma \text{ of } 0:P \text{ suc}(x):Q))_{\gamma',\Lambda} \xrightarrow{\tau} (K \triangleright (\text{case } \gamma \text{ of } 0:P \text{ suc}(x):Q))_{\gamma',\Lambda\{0/\gamma\}} \]

\[ \gamma' \notin \text{dom}(\Upsilon) \]

\[ (K \triangleright (\text{case } \gamma \text{ of } 0:P \text{ suc}(x):Q))_{\gamma',\Lambda} \xrightarrow{\tau} (K \triangleright (\text{case } \gamma \text{ of } 0:P \text{ suc}(x):Q))_{\gamma',\Lambda\{suc(\gamma')/\gamma\}} \]

\[ \xi \in \eta \circ \rho \quad \xi \neq \top \]

\[ (K \triangleright (\text{case } \eta \text{ of } \{x\}_\rho \text{ in } P))_{\gamma',\Lambda} \xrightarrow{\tau} (K \triangleright (\text{case } \eta \text{ of } \{x\}_\rho \text{ in } P))_{\gamma',\Lambda\{\xi\}} \]

\[ \xi \in \eta \oplus \rho \quad \xi \neq \top \]

\[ (K \triangleright (\text{case } \eta \text{ of } \{x\}_\rho \text{ in } P))_{\gamma',\Lambda} \xrightarrow{\tau} (K \triangleright (\text{case } \eta \text{ of } \{x\}_\rho \text{ in } P))_{\gamma',\Lambda\{\xi\}} \]

\[ \xi \in \eta \ominus \rho \quad \xi \neq \top \]

\[ (K \triangleright (\text{case } \eta \text{ of } \{x\}_\rho \text{ in } P))_{\gamma',\Lambda} \xrightarrow{\tau} (K \triangleright (\text{case } \eta \text{ of } \{x\}_\rho \text{ in } P))_{\gamma',\Lambda\{\xi\}} \]
References


[27] R. Focardi and R. Gorrieri. The compositional security checker: A tool for the verification of information flow se-


[38] Projet EVA (Explication et Vérification Automatique de protocoles cryptographiques). Security protocols open repository. Available online,

