DECISION PROCEDURE FOR TRACE EQUIVALENCE

V. Cheval, H. Comon-Lundh, S. Delaune
LSV, ENS Cachan, CNRS, INRIA Saclay

15 November 2011
Cryptographic protocols

Most communications take place over a public network

Cryptographic protocols
• small programs designed to secure communication (e.g. secrecy)
• use cryptographic primitives (e.g. encryption, signature)

It important to verify their security
CONTEXT

• Reliable cryptography
  • Correct specification
• Implementation satisfying the specification
CONTEXT

- Reliable cryptography
- Correct specification
- Implementation satisfying the specification

- Some security properties
CONTEXT

- Reliable cryptography
- **Correct specification**
- Implementation satisfying the specification

- Some security properties

  **Reachability properties**
  - Secrecy, Authentication, ...
CONTEXT

- Reliable cryptography
- **Correct specification**
- Implementation satisfying the specification

Some security properties

**Reachability properties**
- Secrecy, Authentication, ...

**Equivalence properties**
- Anonymity, Privacy, Receipt-Freeness, ...
CONTEXT

- Formalism

Alice

Bob
Formalism

Alice

Bob
The intruder can
- intercept all messages
- transmit or modify messages
- test equality between messages
- initiate several sessions
The intruder can
- intercept all messages
- transmit or modify messages
- test equality between messages
- initiate several sessions
CONTEXT

- Reachability properties: secrecy, authentication,...
Can the intruder deduce Alice’s secret?
Can the intruder deduce Alice’s secret?
CONTEXT

- Equivalence properties: strong secret, anonymity,...

Alice  Intruder  Unknown
Can the intruder deduce the unknown’s identity?
CONTEXT

- Equivalence properties: strong secret, anonymity,...
Equivalence properties: strong secret, anonymity,...
CONTEXT

- Equivalence properties: strong secret, anonymity, ...

Alice  Intruder  Unknown  Charlene

Alice  Intruder  Unknown  Bob
Can the intruder distinguish the two situations?
CONTEXT

- Equivalence properties: strong secret, anonymity,...
Knowledge indistinguishability : static equivalence

Alice  Intruder  Bob
Knowledge indistinguishability: static equivalence

Alice  Intruder  Bob
Knowledge indistinguishability: static equivalence
Knowledge indistinguishability: static equivalence

Example with decryption: \( \text{dec}(\{x\}_y, y) = x \)
Knowledge indistinguishability: static equivalence

Example with decryption: $dec(\{x\}_y, y) = x$

$\Phi_1 : a, \{b\}_a, b$

$\Phi_2 : c, \{b\}_a, b$
Knowledge indistinguishability: static equivalence

Example with decryption:

\[
dec(\{x\}_y, y) = x \\
dec(M_2, M_1) = M_3
\]

\[
\Phi_1 : a, \{b\}_a, b \\
\Phi_2 : c, \{b\}_a, b
\]
Knowledge indistinguishability: static equivalence

Example with decryption:

\[ \text{dec}(\{x\}_y, y) = x \]
\[ \text{dec}(M_2, M_1) = M_3 \]
\[ \Phi_1 : a, \{b\}_a, b \quad \text{dec}(\{b\}_a, a) = b \]
\[ \Phi_2 : c, \{b\}_a, b \quad \text{dec}(\{b\}_a, c) \neq b \]
Knowledge indistinguishability: static equivalence

Example with decryption:

\[ \text{dec}(\{x\}_y, y) = x \]

\[ \text{dec}(M_2, M_1) = M_3 \]

\[ \Phi_1 : a, \{b\}_a, b \quad \text{dec}(\{b\}_a, a) = b \]

\[ \Phi_2 : c, \{b\}_a, b \quad \text{dec}(\{b\}_a, c) \neq b \]

Not equivalent
Knowledge indistinguishability: static equivalence

Example with decryption:
\[ \text{dec}(\{x\}_y, y) = x \]
\[ \text{dec}(M_2, M_1) = M_3 \]
\[ \Phi_1 : a, \{b\}_a, b \quad \text{dec}(\{b\}_a, a) = b \]
\[ \Phi_2 : c, \{b\}_a, b \quad \text{dec}(\{b\}_a, c) \neq b \]

Not equivalent
Knowledge indistinguishability: static equivalence

Example with decryption: \( \text{dec} (\{x\}_y, y) = x \)

\( \Phi_1 : a, \{b\}_a, b \)
\( \text{dec}(\{b\}_a, a) = b \)

\( \Phi_2 : c, \{b\}_a, b \)
\( \text{dec}(\{b\}_a, c) \neq b \)

Not equivalent

No test
Knowledge indistinguishability: static equivalence

Example with decryption: \( \text{dec}(\{x\}_y, y) = x \)

\[ \Phi_1 : a, \{b\}_a, b \]
\[ \Phi_2 : c, \{b\}_a, b \]
\[ \text{dec}(\{b\}_a, a) = b \]
\[ \text{dec}(\{b\}_a, c) \neq b \]

Not equivalent

\[ \Phi_1 : a, \{b\}_a \]
\[ \Phi_2 : c, \{b\}_a \]

Equivalent

No test
Most of the previous works focus on stronger equivalence


- B. Blanchet, M. Abadi, and C. Fournet. *Automated verification of selected equivalences for security protocols*.

  ➡ Tool: B. Blanchet, *ProVerif*

Trace equivalence for simple processes without else branches

Example

Two problematic examples:
Two problematic examples:

Example

Two problematic examples:

MOTIVATION

Example

Two problematic examples:


\[
\{\langle N_a, pk(k_A) \rangle \}_p k(k_B)
\]

\[
\{\langle N_a, N_b, pk(k_B) \rangle \}_p k(k_A)
\]
Example

Two problematic examples:
Example

Two problematic examples:
MOTIVATION

Example

Alice

Intruder

Charlene

Intruder

Bob
MOTIVATION

- Example
  
  Alice
  
  Charlene
  
  Bob
MOTIVATION

- Example

$\{\langle N_a, pk(k_A) \rangle \}_{pk(k_B)}$
MOTIVATION

Example

\[ \{\langle N_a, pk(k_A) \rangle \}_{pk(k_B)} \]

\[ \{\langle x, y \rangle \}_{pk(k_B)} \]
MOTIVATION

Example

\[
\{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)} \rightarrow \{ \langle x, y \rangle \}_{pk(k_B)} \rightarrow pk(k_A) = y
\]
Example

\[\{\langle N_a, pk(k_A) \rangle \}_p k(k_B)\]

\[\{\langle x, y \rangle \}_p k(k_B)\]

\[\{\langle x, N_b, pk(k_B) \rangle \}_y\]

\[pk(k_A) = y\]
Example

\[
\{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)} \rightarrow \{ \langle x, y \rangle \}_{pk(k_B)} \rightarrow \{ \langle x, N_b, pk(k_B) \rangle \}_y \rightarrow \{ N \}_{pk(k_A)} \rightarrow pk(k_A) = y
\]
Example

\[
\{(N_a, pk(k_A))\}_{pk(k_B)}
\]

\[
\{(x, y)\}_{pk(k_B)}
\]

\[
\{N\}_{pk(k_A)}
\]

\[
\{N\}_{pk(k_C)}
\]

\[
\{N, pk(k_B)\}
\]

\[
\{x, N_b, pk(k_B)\}
\]
Example

\[
\begin{align*}
\{\langle N_a, pk(k_A) \rangle \}_p(k_B) & \quad \rightarrow \quad \{\langle x, y \rangle \}_p(k_B) \\
\{\langle x, N_b, pk(k_B) \rangle \}_y & \quad \rightarrow \quad pk(k_A) = y \\
\{\langle x, N_b, pk(k_B) \rangle \}_y & \quad \rightarrow \quad \{N \}_p(k_A) \\
\{\langle x, N_b, pk(k_B) \rangle \}_y & \quad \rightarrow \quad \{N \}_p(k_C) \\
\{\langle N_c, pk(k_C) \rangle \}_p(k_B) & \quad \rightarrow \quad \{\langle x, y \rangle \}_p(k_B) \\
\{\langle x, N_b, pk(k_B) \rangle \}_y & \quad \rightarrow \quad pk(k_C) = y \\
\{\langle x, N_b, pk(k_B) \rangle \}_y & \quad \rightarrow \quad \{N \}_p(k_C) \\
\{\langle x, N_b, pk(k_B) \rangle \}_y & \quad \rightarrow \quad \{N \}_p(k_A) \\
\end{align*}
\]
MOTIVATION

Example

\[\{\langle N_a, pk(k_A) \rangle \}_{pk(k_B)}\]

Unknown

\[\{\langle N_c, pk(k_C) \rangle \}_{pk(k_B)}\]

Unknown

\[\{\langle x, N_b, pk(k_B) \rangle \}_y\]

Intruder

Bob

\[\{\langle x, y \rangle \}_{pk(k_B)}\]

\[\{x, N_b, pk(k_B) \}_y\]

\[\{N\}_{pk(k_A)}\]

\[\{N\}_{pk(k_C)}\]

\[\{x, N_b, pk(k_B) \}_y\]

\[pk(k_A) = y\]

\[pk(k_C) = y\]
**MOTIVATION**

- **Example**

  - *Unknown* → *Intruder* → *Bob*
    - $\{\langle N_a, pk(k_A) \rangle \}_{pk(k_B)}$
    - $\{\langle N_I, pk(k_A) \rangle \}_{pk(k_B)}$
    - $pk(k_A) = y$
  
  - *Unknown* → *Intruder* → *Bob*
    - $\{\langle N_C, pk(k_C) \rangle \}_{pk(k_B)}$
    - $\{\langle N_I, pk(k_A) \rangle \}_{pk(k_B)}$
    - $pk(k_C) = y$
MOTIVATION

Example

Unknown \(\{\langle N_a, pk(k_A) \rangle \}_{pk(k_B)}\) Intruder \(\{\langle N_I, pk(k_A) \rangle \}_{pk(k_B)}\) Bob \(pk(k_A) = pk(k_A)\)

Unknown \(\{\langle N_c, pk(k_C) \rangle \}_{pk(k_B)}\) Intruder \(\{\langle x, N_b, pk(k_B) \rangle \}_y\) Bob \(pk(k_C) = y\)

Unknown \(\{\langle N_I, N_b, pk(k_B) \rangle \}_{pk(k_A)}\) Intruder \(\{N\}_{pk(k_C)}\) Bob \(\{\langle N_I, N_b, pk(k_B) \rangle \}_{pk(k_A)}\)
Example

\[
\begin{align*}
\langle N_a, pk(k_A) \rangle_{pk(k_B)} &\quad \xrightarrow{} \quad \langle N_I, pk(k_A) \rangle_{pk(k_B)} \\
\langle N_c, pk(k_C) \rangle_{pk(k_B)} &\quad \xrightarrow{} \quad \langle N_I, N_b, pk(k_B) \rangle_{pk(k_A)}
\end{align*}
\]
Example

\[
\{N_a, pk(k_A)\}_{pk(k_B)} \rightarrow \{N_I, pk(k_A)\}_{pk(k_B)}
\]

Unknown

Intruder

Bob

\[
\{N_c, pk(k_C)\}_{pk(k_B)} \rightarrow \{N_I, N_b, pk(k_B)\}_{pk(k_A)}
\]

Unknown

Intruder

Bob

\[
pk(k_A) = pk(k_C)
\]
CONTRIBUTION

Decision procedure for verification of trace equivalence

- Infinitely many traces are represented by symbolic constraint system
  + Protocol possibly non-determinist and with non trivial else branches
  + Private channels
    - Fixed set of cryptographic primitives: symmetric and asymmetric encryption, pairing and signature
    - Bounded number of sessions (no replication in the process algebra)
One constraint system = one interleaving = several traces
One constraint system = one interleaving = several traces

\[ pk(k_A), pk(k_B), pk(k_C), N_I \]
One constraint system = one interleaving = several traces

\[ \langle N_a, pk(k_A) \rangle_{pk(k_B)} \]

Alice \quad Intruder \quad Bob

\[ pk(k_A), pk(k_B), pk(k_C), N_I, \{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)} \]
One constraint system = one interleaving = several traces

\[ \{\langle N_a, pk(k_A) \rangle\}_{pk(k_B)} \rightarrow \{\langle x, y \rangle\}_{pk(k_B)} \rightarrow \]

\[ pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A) \rangle\}_{pk(k_B)} \vdash \{\langle x, y \rangle\}_{pk(k_B)} \]
One constraint system = one interleaving = several traces

\[ \{\langle N_a, pk(k_A) \rangle\}_{pk(k_B)} \quad \{\langle x, y \rangle\}_{pk(k_B)} \quad pk(k_A) = y \]

\[ pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A) \rangle\}_{pk(k_B)} \vdash \{\langle x, y \rangle\}_{pk(k_B)} \]

\[ y \overset{?}{=} pk(k_A) \]
One constraint system = one interleaving = several traces

\[\{\langle Na, pk(k_A)\rangle\}_{pk(k_B)} \rightarrow \{\langle x, y\rangle\}_{pk(k_B)} \rightarrow \{\langle x, Nb, pk(k_B)\rangle\}_y \rightarrow pk(k_A) = y\]
CONSTRRAINT SYSTEM

- One constraint system = one interleaving = several traces

\[ \{\langle N_a, pk(k_A) \rangle\}_{pk(k_B)} \quad \{\langle x, y \rangle\}_{pk(k_B)} \quad \{\langle x, N_b, pk(k_B) \rangle\}_y \quad pk(k_A) = y \]

\[ D : pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A) \rangle\}_{pk(k_B)} \quad ? \quad \{\langle x, y \rangle\}_{pk(k_B)} \]

\[ \Phi : pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A) \rangle\}_{pk(k_B)} \quad \{\langle x, N_b, pk(k_B) \rangle\}_y \]

\[ E : y \overset{?}{=} pk(k_A) \]
One constraint system = one interleaving = several traces

\[
\begin{align*}
D &: pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A) \rangle \}_pk(k_B) \vdash \{\langle x, y \rangle \}_pk(k_B) \\
\Phi &: pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A) \rangle \}_pk(k_B), \{\langle x, N_b, pk(k_B) \rangle \}_y \\
E &: y = pk(k_A)
\end{align*}
\]
One solution of a constraint system = one trace

\[ D : pk(k_A), pk(k_B), pk(k_C), N_I, \{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)} \vdash \{ \langle x, y \rangle \}_{pk(k_B)} \]

\[ \Phi : pk(k_A), pk(k_B), pk(k_C), N_I, \{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)}, \{ \langle x, N_b, pk(k_B) \rangle \} \}_{y} \]

\[ E : y = pk(k_A) \]
One solution of a constraint system = one trace

\[ D : \ pk(k_A), pk(k_B), pk(k_C), N, \{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)} \vdash \{ \langle x, y \rangle \}_{pk(k_B)} \]

\[ \Phi : \ pk(k_A), pk(k_B), pk(k_C), N, \{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)}, \{ \langle x, N_b, pk(k_B) \rangle \}_{y} \]

\[ E : \ y = pk(k_A) \]

A solution is a pair of substitution \((\sigma, \theta)\) here:

- \(\sigma\) describe the messages
- \(\theta\) describe how the messages are deduced
One solution of a constraint system = one trace

\[ D : \text{pk}(k_A), \text{pk}(k_B), \text{pk}(k_C), N_I, \{ \langle N_a, \text{pk}(k_A) \rangle \} \text{pk}(k_B) \vdash \{ \langle x, y \rangle \} \text{pk}(k_B) \]

\[ \Phi : \text{pk}(k_A), \text{pk}(k_B), \text{pk}(k_C), N_I, \{ \langle N_a, \text{pk}(k_A) \rangle \} \text{pk}(k_B), \{ \langle x, N_b, \text{pk}(k_B) \rangle \} \}

\[ E : y = \text{pk}(k_A) \]

A solution is a pair of substitution \((\sigma, \theta)\) where:

- \(\sigma\) describe the messages
- \(\theta\) describe how the messages are deduced

\[ \sigma = \{ x \rightarrow N_I; y \rightarrow \text{pk}(k_A) \} \]
One solution of a constraint system = one trace

\[ D : pk(k_A), pk(k_B), pk(k_C), N_I, \{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)} \vdash \{ \langle x, y \rangle \}_{pk(k_B)} \]

\[ \Phi : pk(k_A), pk(k_B), pk(k_C), N_I, \{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)}, \{ \langle x, N_b, pk(k_B) \rangle \}_{y} \]

\[ E : y = pk(k_A) \]

A solution is a pair of substitution \((\sigma, \theta)\) where:
- \(\sigma\) describe the messages
- \(\theta\) describe how the messages are deduced

\[ \sigma = \{ x \rightarrow N_I; y \rightarrow pk(k_A) \} \]
One solution of a constraint system = one trace

\[
D : \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A) \rangle\} \leftarrow \{\langle x, y \rangle\} \; \Phi : \; pk(k_A), pk(k_B), pk(k_C), N_I, \{\langle N_a, pk(k_A) \rangle\} \leftarrow \{\langle x, N_b, pk(k_B) \rangle\} \; \]

\[
E : \; y = pk(k_A)
\]

A solution is a pair of substitution \((\sigma, \theta)\) where:
- \(\sigma\) describe the messages
- \(\theta\) describe how the messages are deduced

\[
\sigma = \{x \rightarrow N_I; \; y \rightarrow pk(k_A)\}
\]

\[
\theta = \{X_1 \rightarrow \{\langle ax_4, ax_1 \rangle\} \; ax_2\}
\]
One solution of a constraint system = one trace

\[ D : \text{pk}(k_A), \text{pk}(k_B), \text{pk}(k_C), N_I, \{ \langle N_a, \text{pk}(k_A) \rangle \}_{\text{pk}(k_B)} \vdash \{ \langle x, y \rangle \}_{\text{pk}(k_B)} \]

\[ \Phi : \text{pk}(k_A), \text{pk}(k_B), \text{pk}(k_C), N_I, \{ \langle N_a, \text{pk}(k_A) \rangle \}_{\text{pk}(k_B)}, \{ \langle x, N_b, \text{pk}(k_B) \rangle \}_{\text{pk}(k_B)} \]

\[ ax_1 \quad ax_2 \quad ax_3 \quad ax_4 \quad ax_5 \quad ax_6 \]

\[ E : \; y = \text{pk}(k_A) \]

A solution is a pair of substitution \((\sigma, \theta)\) where:

- \(\sigma\) describes the messages
- \(\theta\) describes how the messages are deduced

\[ \sigma = \{ x \rightarrow N_I; \; y \rightarrow \text{pk}(k_A) \} \]

\[ \sigma = \{ x \rightarrow N_a; \; y \rightarrow \text{pk}(k_A) \} \]

\[ \theta = \{ X_1 \rightarrow \{ \langle ax_4, ax_1 \rangle \}_{ax_2} \} \]

\[ \theta = \{ X_1 \rightarrow ax_5 \} \]
Set of constraint systems

- \{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)} = \{ \langle x, y \rangle \}_{pk(k_B)} = \{ \langle x, N_b, pk(k_B) \rangle \}_{y} = \{ \langle x, N_b, pk(k_B) \rangle \}_{y} = \{ N \}_{pk(k_A)} = \{ N \}_{pk(k_C)} = y
Set of constraint systems
CONSTRAINT SYSTEM

Set of constraint systems

\[ \{ \langle N_a, pk(k_A) \rangle \}_{pk(k_B)} \Rightarrow \{ \langle x, y \rangle \}_{pk(k_B)} \Rightarrow \{ \langle x, N_b, pk(k_B) \rangle \}_{y} \Rightarrow \{ N \}_{pk(k_A)} \]

\[ pk(k_A) = y \]

\[ \{ \langle x, N_b, pk(k_B) \rangle \}_{y} \Rightarrow \{ x, y \}_{pk(k_B)} \Rightarrow \{ x, N_b, pk(k_B) \}_{y} \Rightarrow \{ N \}_{pk(k_C)} \]

\[ pk(k_C) = y \]

\[ \{ C_1 ; C_2 \} \simeq \{ C'_1 ; C'_2 \} \]
Set of constraint systems

Symbolic equivalence between sets of constraint systems
Why sets of constraint systems are necessary?

\[
\begin{align*}
\{\langle N_A, pk(k_A) \rangle \}_{pk(k_B)} & \quad y = pk(K_A) \\
\{\langle N_c, pk(k_C) \rangle \}_{pk(k_B)} & \quad y = pk(K_C)
\end{align*}
\]
Why sets of constraint systems are necessary?
Why sets of constraint systems are necessary?

\[ S = \{ C_1 ; C_2 ; C_3 \} \]

\[ S' = \{ C_1' ; C_2' ; C_3' ; C_4' \} \]
Symbolic equivalence between sets of constraint systems

To check whether $P$ and $P'$ are trace equivalent, we have to check that:

$S \approx S'$, for all symbolic interleaving

Symbolic equivalence $S \approx S$

- For all $C \in S$, for all $(\theta, \sigma) \in \text{Sol}(C)$, there exists $C' \in S'$ and $\sigma'$ such that $(\theta, \sigma') \in \text{Sol}(C')$ and $\Phi\sigma \sim \Phi'\sigma'$

- and conversely
Previous works on constraint system


2. Y. Chevalier and M. Rusinowitch. Decidability of equivalence of symbolic derivations.


Focus on:
- symbolic equivalence between two constraint systems (All)
- positive constraint system (no disequations) (All)
- subterm convergent equational theory (1,2 & 3)
- more restricted equational theory (4 & 5)
THE ALGORITHM

- Set of rules

```
Test

\[ C \quad \text{T} \quad \neg \text{T} \quad C_1 \quad C_2 \]
```
THE ALGORITHM

- Set of rules

![Diagram of algorithm with rules]

- How to apply the rules:
  \[
  \{C^1; C^2; \ldots\} \approx \{C^n; \ldots\}
  \]
  \[
  \{C^1_1; C^2_1; \ldots\} \approx \{C^n_1; \ldots\} \quad \{C^1_2; C^2_2; \ldots\} \approx \{C^n_2; \ldots\}
  \]
THE ALGORITHM

- A complete execution
A complete execution

\[ S \approx S' \]

\[ \mathcal{T}, \neg\mathcal{T} \]

\[ \mathcal{T}_1, \neg\mathcal{T}_1 \]
THE ALGORITHM

- A complete execution

\[
S \approx S' \quad \text{with} \quad T \quad \text{and} \quad \neg T
\]

\[
T_1 \quad \neg T_1 \quad T_2 \quad \neg T_2
\]
THE ALGORITHM

- A complete execution

The application of the rules creates a binary tree where each node is a pair of sets of constraint systems.
The symbolic equivalence is syntactically decided on each leaf.
Example of rule: Cons

Test $\mathcal{T} = \exists X_1, X_2 \text{ s.t. } X = \text{enc}(X_1, X_2)$

\[
\begin{cases}
\ldots \\
T \vdash_X \text{enc}(u_1, u_2) \\
\ldots 
\end{cases}
\]

\[
\begin{array}{c}
\mathcal{T} \\
\mathcal{T} \neg
\end{array}
\]

\[
\begin{cases}
T \vdash_{X_1} u_1 \\
T \vdash_{X_2} u_2 \\
X = \text{enc}(X_1, X_2) \\
\ldots
\end{cases}
\]

\[
\begin{cases}
\ldots \\
T \vdash_X \text{enc}(u_1, u_2) \\
\text{Top}(X) \neq \text{enc} \\
\ldots
\end{cases}
\]
THE ALGORITHM

- The solved form of a constraint system
  - Existence of solutions (Reachability)
    \[ m_1, \ldots, m_n \vdash x \]
    \[ m_1, \ldots, m_n, \ldots, m_n' \vdash y \]
  - Matching solutions (including disequations)
    \[
    \begin{align*}
    a, b & \vdash x \\
    a, b, c & \vdash y \\
    x & \neq y
    \end{align*}
    \quad
    \begin{align*}
    a, b & \vdash x \\
    a, b, c & \vdash y \\
    x & \neq f(y)
    \end{align*}
    \]
  - Static equivalence
    \[
    \begin{align*}
    a, \{b\}_c & \vdash x \\
    a, \{b\}_c, c & \vdash y
    \end{align*}
    \quad
    \begin{align*}
    a, b & \vdash x \\
    a, b, c & \vdash y
    \end{align*}
    \]
Let \((S_0, S'_0)\) be an initial pair of set of constraint systems, we have:

\((S, S')\)
RESULT

Let \((S_0, S'_0)\) be an initial pair of set of constraint systems, we have:

If all leaves \((S, S')\) on the tree satisfy the testing condition then \(S_0 \approx S'_0\).
RESULT

Let \((S_0, S'_0)\) be an initial pair of set of constraint systems, we have:

If all leaves \((S, S')\) on the tree satisfy the testing condition then \(S_0 \approx S'_0\).

If \(S_0 \approx S'_0\) then all leaves \((S, S')\) on the tree satisfy the testing condition.
RESULT

Let \((S_0, S'_0)\) be an initial pair of set of constraint systems, we have:

If all leaves \((S, S')\) on the tree satisfy the testing condition then \(S_0 \approx S'_0\).

If \(S_0 \approx S'_0\) then all leaves \((S, S')\) on the tree satisfy the testing condition.

The strategy terminates
FUTURE WORK

- Contribution
  - Decision procedure for trace equivalence
    - Infinitely many traces are represented by symbolic constraint system
    - Protocol possibly non-determinist and with non trivial else branches
    - Private channels
    - Fixed set of cryptographic primitives: symmetric and asymmetric encryption, pairing and signature
    - Bounded number of sessions (no replication in the process algebra)

- Future work
  - Experiment shows that the implementation is not efficient enough
  - More cryptographic primitives
  - Link with ProVerif
The disequations problem

\[ a, b \vdash x_1 \]
\[ D : a, b \vdash x_2 \]
\[ a, b \vdash y \]
\[ E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle \]
The disequations problem

\[ E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle \]
The disequations problem

\[ E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle \]

\[ x_2 = a \]
The disequations problem

\[ E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle \]

\[ x_2 = a \]

\[ E : [x_1 \neq y] \land y \neq \langle x_1, a, b \rangle \]
The disequations problem

\[ E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle \]

\[ E : [x_1 \neq y] \land y \neq \langle x_1, a, b \rangle \]

\[ x_2 = a \]

\[ y = \langle y_1, y_2, y_3 \rangle \]
TERMINATION

- The disequations problem

\[ E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle \]

\[ \downarrow \]

\[ x_2 = a \]

\[ E : [x_1 \neq y] \land y \neq \langle x_1, a, b \rangle \]

\[ \downarrow \]

\[ y = \langle y_1, y_2, y_3 \rangle \]

\[ E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \land \langle y_1, y_2, y_3 \rangle \neq \langle x_1, a, b \rangle \]
TERMINATION

The disequations problem

\[ E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle \]

\[ x_2 = a \]

\[ E : [x_1 \neq y] \land y \neq \langle x_1, a, b \rangle \]

\[ y = \langle y_1, y_2, y_3 \rangle \]

\[ E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \land \langle y_1, y_2, y_3 \rangle \neq \langle x_1, a, b \rangle \]
The disequations problem

\[ E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle \]

\[ x_2 = a \]

\[ E : [x_1 \neq y] \land y \neq \langle x_1, a, b \rangle \]

\[ y = \langle y_1, y_2, y_3 \rangle \]

\[ E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \land \langle y_1, y_2, y_3 \rangle \neq \langle x_1, a, b \rangle \]

\[ E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \land [y_1 \neq x_1 \lor y_2 \neq a \lor y_3 \neq b] \]
TERMINATION

- The disequations problem

\[ E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle \]

\[ x_2 = a \]

\[ E : [x_1 \neq y] \land y \neq \langle x_1, a, b \rangle \]

\[ y = \langle y_1, y_2, y_3 \rangle \]

\[ E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \land \langle y_1, y_2, y_3 \rangle \neq \langle x_1, a, b \rangle \]

\[ E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \land [y_1 \neq x_1 \lor y_2 \neq a \lor y_3 \neq b] \]

\[ y_3 = b \]
TERMINATION

- The disequations problem

\[ E : [x_1 \neq y \lor x_2 \neq a] \land y \neq \langle x_1, x_2, b \rangle \]

\[ x_2 = a \]

\[ E : [x_1 \neq y] \land y \neq \langle x_1, a, b \rangle \]

\[ y = \langle y_1, y_2, y_3 \rangle \]

\[ E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \land \langle y_1, y_2, y_3 \rangle \neq \langle x_1, a, b \rangle \]

\[ E : [x_1 \neq \langle y_1, y_2, y_3 \rangle] \land [y_1 \neq x_1 \lor y_1 \neq a \lor y_3 \neq b] \]

\[ y_3 = b \]

\[ E : [x_1 \neq \langle y_1, y_2, b \rangle] \land [y_1 \neq x_1 \lor y_2 \neq a] \]