A decision procedure for proving symbolic equivalence

V. Cheval, H. Comon-Lundh, S. Delaune

LSV, Project SECSI

19 June 2010

Context

Automatic procedure for proving security properties on protocol

Context

Automatic procedure for proving security properties on protocol

Trace properties

- Examples : simple secret, authentication, . . .
- All traces of a protocol has to satisfy a certain property.
- Lot of previous works on those security properties.
- Tools already exists (example : ProVerif, Maude-NPA,. . .)

Context

Automatic procedure for proving security properties on protocol

Trace properties

- Examples : simple secret, authentication, ...
- All traces of a protocol has to satisfy a certain property.
- Lot of previous works on those security properties.
- Tools already exists (example : ProVerif, Maude-NPA,...)

Equivalence properties

- Examples : strong secret, dictionnary attacks, anonymity, ...
- Express the indistinguishability of two protocols
- Theoretical results (Baudet, Chevalier, Rusinowitch, ...)
- No general tool implemented

Security properties example : Anonymity

$$
\begin{array}{lll}
0. & A \longrightarrow B : & \mathrm{aenc}(\langle N_a, \mathrm{p}(A)\rangle, \mathrm{p}(B)) \\
1. & B \longrightarrow A : & \mathrm{aenc}(\langle N_a, \langle N_b, \mathrm{p}(B)\rangle\rangle, \mathrm{p}(A))
\end{array}
$$

Security properties example : Anonymity

$$
\begin{array}{lll}
0. & A \longrightarrow B : & \mathrm{aenc}(\langle N_a, \mathrm{p}(A)\rangle, \mathrm{p}(B)) \\
1. & B \longrightarrow A : & \mathrm{aenc}(\langle N_a, \langle N_b, \mathrm{p}(B)\rangle\rangle, \mathrm{p}(A))
\end{array}
$$

Security property : Anonymity

The identity of the principal A cannot be revealed to the attacker.

Security properties example : Anonymity

$$\begin{array}{lll} 0. & A \longrightarrow B : & \mathrm{aenc}(\langle N_a, \mathrm{p}(A)\rangle, \mathrm{p}(B)) \\ 1. & B \longrightarrow A : & \mathrm{aenc}(\langle N_a, \langle N_b, \mathrm{p}(B)\rangle\rangle, \mathrm{p}(A)) \end{array}$$

Security property : Anonymity

The identity of the principal A cannot be revealed to the attacker.

Formally

$$c(\mathrm{p}(a)).c(\mathrm{p}(a')).c(\mathrm{p}(b)) \mid P_A(a, b) \mid P_B(b, a)$$
$$\approx$$
$$c(\mathrm{p}(a)).c(\mathrm{p}(a')).c(\mathrm{p}(b)) \mid P_A(a', b) \mid P_B(b, a')$$

Previous works

Huttel (2002)

- Only spi-calculus (fixed primitives)
- Untractable implementation (multi-exponential complexity)
- Doesn't handle trace properties.

Blanchet, Abadi, Fournet (2008) : ProVerif

- Unbounded number of sessions
- Diff-equivalence : Observational equivalence between two process with the same structure but different messages.
- Very efficient
- Possibility of false attacks. Doesn't always terminate

Previous works

Cortier, Delaune (2009) + Baudet (2005) or Chevalier,Rusinowitch (2009)

- Bounded number of sessions
- Infinitely many traces are represented by constraint systems
- Observational equivalence of processes ⇔ symbolic equivalence of constraint systems
- Algorithm for the symbolic equivalence of positive constraint systems when the equational theory is given by a subterm convergent rewriting system.

Outline

1. Formalism
   - Constraint Systems
   - Equivalence
2. The rules
   - Definition and example
   - How to use them ?
3. Termination, Completeness and Soundness

Dolev-Yao

Rewrite rules

- $\mathrm{dec}(\mathrm{enc}(x, y), y) \to x$
- $\mathrm{adec}(\mathrm{aenc}(x, \mathrm{p}(y)), y) \to x$
- $\mathrm{check}(\mathrm{sign}(x, y), \mathrm{p}(y)) \to x$
- $\pi_1(\langle x, y \rangle) \to x$ and $\pi_2(\langle x, y \rangle) \to y$

Constraint system

$$\begin{array}{llr} 0. & A \longrightarrow B : & \mathrm{aenc}(\langle N_a, \mathrm{p}(A)\rangle, \mathrm{p}(B)) \\ 1. & B \longrightarrow A : & \mathrm{aenc}(\langle N_a, N_b, \mathrm{p}(B)\rangle, \mathrm{p}(A)) \end{array}$$

Constraint system

$$\begin{array}{ll} \mathrm{p}(A), \mathrm{p}(B), \{\langle N_a, \mathrm{p}(A)\rangle\}_{\mathrm{p}(B)} & \vdash \{\langle x, y\rangle\}_{\mathrm{p}(B)} \\ \mathrm{p}(B), \mathrm{p}(B), \{\langle N_a, \mathrm{p}(A)\rangle\}_{\mathrm{p}(B)}, \{\langle x, N_b, \mathrm{p}(B)\rangle\}_y & \vdash \{\langle N_a, z, \mathrm{p}(B)\rangle\}_{\mathrm{p}(A)} \end{array}$$

Solution of a constraint system

$$\mathsf{p}(A), \mathsf{p}(B), \{\langle N_a, \mathsf{p}(A)\rangle\}_{\mathsf{p}(B)} \vdash \{\langle x, y\rangle\}_{\mathsf{p}(B)}$$
$$\mathsf{p}(A), \mathsf{p}(B), \{\langle N_a, \mathsf{p}(A)\rangle\}_{\mathsf{p}(B)}, \{\langle x, N_b, \mathsf{p}(B)\rangle\}_y \vdash \{\langle N_a, z, \mathsf{p}(B)\rangle\}_{\mathsf{p}(A)}$$

Solution of a constraint system

$$\begin{array}{ll} & \quad ax_1 \qquad ax_2 \qquad\qquad ax_3 \qquad\qquad\qquad ax_4 \\ X_1 & \mathsf{p}(A), \mathsf{p}(B), \{\langle N_a, \mathsf{p}(A)\rangle\}_{\mathsf{p}(B)} \vdash \{\langle x, y\rangle\}_{\mathsf{p}(B)} \\ X_2 & \mathsf{p}(A), \mathsf{p}(B), \{\langle N_a, \mathsf{p}(A)\rangle\}_{\mathsf{p}(B)}, \{\langle x, N_b, \mathsf{p}(B)\rangle\}_y \vdash \{\langle N_a, z, \mathsf{p}(B)\rangle\}_{\mathsf{p}(A)} \end{array}$$

A solution

- $\sigma = \{x \mapsto N_a \; ; \; y \mapsto \mathsf{p}(a) \; ; \; z \mapsto N_b\}$, and
- $\theta = \{X_1 \mapsto ax_3 \; ; \; X_2 \mapsto ax_4\}$.

Outline

1. Formalism
   - Constraint Systems
   - Equivalence
2. The rules
   - Definition and example
   - How to use them ?
3. Termination, Completeness and Soundness

Static equivalence

Static equivalence : $\phi \sim \phi'$

Given two sequences of terms ϕ, ϕ', the intruder cannot distinguish them.

- $\forall (\xi, \xi') \in \Pi^2, \xi\phi\downarrow = \xi'\phi\downarrow \Leftrightarrow \xi\phi'\downarrow = \xi'\phi'\downarrow$
- $\forall \xi \in \Pi, \xi\phi\downarrow$ is a message $\Leftrightarrow \xi\phi'\downarrow$ is a message

Static equivalence

Static equivalence : $\phi \sim \phi'$

Given two sequences of terms ϕ, ϕ', the intruder cannot distinguish them.

- $\forall(\xi,\xi') \in \Pi^2, \xi\phi\downarrow = \xi'\phi\downarrow \Leftrightarrow \xi\phi'\downarrow = \xi'\phi'\downarrow$
- $\forall\xi \in \Pi, \xi\phi\downarrow$ is a message $\Leftrightarrow \xi\phi'\downarrow$ is a message

Example 1

- $\phi_1 = a,\ \mathsf{enc}(a,b),\ b$
- $\phi_2 = a,\ \mathsf{enc}(c,b),\ b$

Static equivalence

Static equivalence : $\phi \sim \phi'$

Given two sequences of terms ϕ, ϕ', the intruder cannot distinguish them.

- $\forall(\xi, \xi') \in \Pi^2, \xi\phi\downarrow = \xi'\phi\downarrow \Leftrightarrow \xi\phi'\downarrow = \xi'\phi'\downarrow$
- $\forall\xi \in \Pi, \xi\phi\downarrow$ is a message $\Leftrightarrow \xi\phi'\downarrow$ is a message

Example 1

- $\phi_1 = a,\ \mathsf{enc}(a, b),\ b$
- $\phi_2 = a,\ \mathsf{enc}(c, b),\ b$

Example 2

- $\phi_1 = a,\ \mathsf{enc}(a, b)$
- $\phi_2 = a,\ \mathsf{enc}(c, b)$

Symbolic equivalence

$C \approx_s C'$

Given two constraint systems, any two associated traces are staticly equivalent.

- for all $(\theta, \sigma) \in \mathsf{Sol}(C)$,there exists σ' such that $(\theta, \sigma') \in \mathsf{Sol}(C')$ and $\phi\sigma \sim \phi'\sigma'$
- for all $(\theta, \sigma') \in \mathsf{Sol}(C')$, there exists σ such that $(\theta, \sigma) \in \mathsf{Sol}(C)$, and $\phi\sigma \sim \phi'\sigma'$

Example 1

$$\begin{array}{ll} A, B & \vdash x \\ A, B, \mathsf{enc}(x, K) & \vdash \mathsf{enc}(A, K) \end{array}$$

$$\begin{array}{ll} A, B & \vdash x \\ A, B, \mathsf{enc}(A, K) & \vdash \mathsf{enc}(A, K) \end{array}$$

Example 1

$$\begin{array}{ll} A, B & \vdash x \\ A, B, \mathsf{enc}(x, K) & \vdash \mathsf{enc}(A, K) \end{array}$$

$$\begin{array}{ll} A, B & \vdash x \\ A, B, \mathsf{enc}(A, K) & \vdash \mathsf{enc}(A, K) \end{array}$$

Non-equivalent

The substitution of recipe $\theta = \{X_1 \mapsto ax_2,\ X_2 \mapsto ax_3\}$ is only a solution for the first constraint system with $\sigma = \{x \mapsto B\}$, and

Example 2

$$\begin{array}{ll} a, b, \mathsf{enc}(n_a, k), & \vdash \mathsf{enc}(x, k) \\ a, b, \mathsf{enc}(n_a, k),\ \mathsf{enc}(\langle x, x\rangle, k),\ k & \vdash \mathsf{enc}(\langle n_a, n_a\rangle, k) \end{array}$$

$$\begin{array}{ll} a, b, \mathsf{enc}(n_a, k), & \vdash \mathsf{enc}(x, k) \\ a, b, \mathsf{enc}(n_a, k),\ \mathsf{enc}(\langle x, x\rangle, k),\ k' & \vdash \mathsf{enc}(\langle n_a, n_a\rangle, k) \end{array}$$

Example 2

$$\begin{array}{ll} a, b, \mathrm{enc}(n_a, k), & \vdash \mathrm{enc}(x, k) \\ a, b, \mathrm{enc}(n_a, k),\ \mathrm{enc}(\langle x, x\rangle, k),\ k & \vdash \mathrm{enc}(\langle n_a, n_a\rangle, k) \end{array}$$

$$\begin{array}{ll} a, b, \mathrm{enc}(n_a, k), & \vdash \mathrm{enc}(x, k) \\ a, b, \mathrm{enc}(n_a, k),\ \mathrm{enc}(\langle x, x\rangle, k),\ k' & \vdash \mathrm{enc}(\langle n_a, n_a\rangle, k) \end{array}$$

Non-equivalent

- A solution : $\sigma = \sigma' = \{x \mapsto n_a\}$, and $\theta = \{X_1 \mapsto ax_3,\ X_2 \mapsto ax_4\}$
- $\phi\sigma \not\sim \phi'\sigma'$: $\xi = f(\mathrm{dec}(ax_3, ax_5)), \xi' = \mathrm{dec}(ax_4, ax_5)$

Outline

1. Formalism
   - Constraint Systems
   - Equivalence
2. The rules
   - Definition and example
   - How to use them ?
3. Termination, Completeness and Soundness

General idea

- Input : two constraint systems : C and C'
- Problem : is $C \approx_s C'$?
- Reduce the problem to a finite conjunction of constraint systems equivalence :

$$C_1 \approx_s C'_1 \wedge \ldots \wedge C_n \approx_s C'_n$$

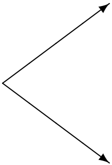
- Decidability of each $C_i \approx_s C'_i$ has to be trivial

Guessing from the top

Constructor rule

Partition of the solution which ends or not by the application of a public constructor

$a,\ b \vdash \mathrm{aenc}(x, \mathrm{p}(a))$



Guessing from the top

Constructor rule

Partition of the solution which ends or not by the application of a public constructor

$$a,\ b \vdash \mathrm{aenc}(x, \mathrm{p}(a))$$

$$a,\ b \vdash x$$
$$a,\ b \vdash \mathrm{p}(a)$$

Guessing from the top

Constructor rule

Partition of the solution which ends or not by the application of a public constructor

$$a,\ b \vdash \mathrm{aenc}(x, \mathrm{p}(a))$$

→

$$a,\ b \vdash x$$
$$a,\ b \vdash \mathrm{p}(a)$$

→

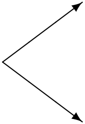
$$a,\ b \vdash_{\mathtt{NoCons}} \mathrm{aenc}(x, \mathrm{p}(a))$$

Guessing from the bottom

Destructor rule

Partition of the solution where a cypher can be decrypt or not.

$\mathrm{aenc}(b, \mathrm{p}(a)),\ a \vdash x$



Guessing from the bottom

Destructor rule

Partition of the solution where a cypher can be decrypt or not.

$$\mathrm{aenc}(b, \mathrm{p}(a)),\ a \vdash x$$

$$\overset{\texttt{IsDest}}{\mathrm{aenc}(b, \mathrm{p}(a))},\ a \vdash a$$

$$\overset{\texttt{IsDest}}{\mathrm{aenc}(b, \mathrm{p}(a))},\ a,\ b \vdash x$$

Guessing from the bottom

Destructor rule

Partition of the solution where a cypher can be decrypt or not.

$$\mathrm{aenc}(b, \mathrm{p}(a)),\ a \vdash x$$

→ IsDest:

$$\mathrm{aenc}(b, \mathrm{p}(a)),\ a \vdash a$$

$$\mathrm{aenc}(b, \mathrm{p}(a)),\ a,\ b \vdash x$$

→ NoDest:

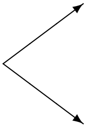
$$\mathrm{aenc}(b, \mathrm{p}(a)),\ a \vdash x$$

Outline

1. Formalism
   - Constraint Systems
   - Equivalence
2. The rules
   - Definition and example
   - How to use them ?
3. Termination, Completeness and Soundness

Application of the rules on a constraint systems couple

$$\begin{cases} A,\ B \vdash \mathsf{enc}(A, B) \\ C,\ D \vdash \mathsf{enc}(x, C) \end{cases}$$



Application of the rules on a constraint systems couple

$$\left\{ \begin{array}{l} A,\ B \vdash \mathrm{enc}(A, B) \\ \\ C,\ D \vdash \mathrm{enc}(x, C) \end{array} \right. \longrightarrow \left\{ \begin{array}{l} A,\ B \vdash A \\ A,\ B \vdash B \\ \\ C,\ D \vdash x \\ C,\ D \vdash C \end{array} \right.$$

Application of the rules on a constraint systems couple

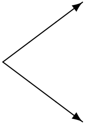
$$\begin{cases} A,\ B \vdash \mathrm{enc}(A, B) \\ \\ C,\ D \vdash \mathrm{enc}(x, C) \end{cases}$$

$$\begin{cases} A,\ B \vdash A \\ A,\ B \vdash B \\ \\ C,\ D \vdash x \\ C,\ D \vdash C \end{cases}$$

$$\begin{cases} A,\ B \vdash_{\mathtt{NoCons}} \mathrm{enc}(A, B) \\ \\ C,\ D \vdash_{\mathtt{NoCons}} \mathrm{enc}(x, C) \end{cases}$$

Application of the rules on a constraint systems couple

$$\left\{\begin{array}{l} A,\ B \vdash \mathrm{enc}(A, B) \\ \\ C,\ D \vdash C \end{array}\right.$$



Application of the rules on a constraint systems couple

$$\left\{\begin{array}{l} A,\ B \vdash A \\ A,\ B \vdash B \\ \\ \bot \end{array}\right.$$

$$\left\{\begin{array}{l} A,\ B \vdash \mathrm{enc}(A, B) \\ \\ C,\ D \vdash C \end{array}\right.$$

Application of the rules on a constraint systems couple

$$\left\{\begin{array}{l} A,\ B \vdash \mathsf{enc}(A, B) \\ \\ C,\ D \vdash C \end{array}\right.$$

$$\left\{\begin{array}{l} A,\ B \vdash A \\ A,\ B \vdash B \\ \\ \bot \end{array}\right.$$

$$\left\{\begin{array}{l} A,\ B \vdash_{\mathtt{NoCons}} \mathsf{enc}(A, B) \\ \\ C,\ D \vdash_{\mathtt{NoCons}} C \end{array}\right.$$

Application of the rules on a constraint systems couple

$$\begin{cases} A,\ B \vdash \mathsf{enc}(A, B) \\ C,\ D \vdash x \end{cases}$$

Application of the rules on a constraint systems couple

$$\left\{ \begin{array}{l} A,\ B \vdash \mathrm{enc}(A, B) \\ \\ C,\ D \vdash x \end{array} \right.$$

$$\left\{ \begin{array}{l} A,\ B \vdash A \\ A,\ B \vdash B \\ \\ C,\ D \vdash x_1 \\ C,\ D \vdash x_2 \\ x = \mathrm{enc}(x_1, x_2) \end{array} \right.$$

Application of the rules on a constraint systems couple

$$\begin{cases} A,\ B \vdash \mathrm{enc}(A,B) \\ \\ C,\ D \vdash x \end{cases}$$

$$\longrightarrow \begin{cases} A,\ B \vdash A \\ A,\ B \vdash B \\ \\ C,\ D \vdash x_1 \\ C,\ D \vdash x_2 \\ x = \mathrm{enc}(x_1, x_2) \end{cases}$$

$$\longrightarrow \begin{cases} A,\ B \vdash_{\mathtt{NoCons}} \mathrm{enc}(A,B) \\ \\ C,\ D \vdash_{\mathtt{NoCons}} x \end{cases}$$

Soundness and completeness

Theorem (Soundness)

If all leaves of a tree, whose root is labeled with (C_0, C'_0) *(a pair of initial constraints), are labeled either with* $(\perp, \perp)$ *or with some* (C, C') *with* $C \neq \perp, C' \neq \perp$*, then* $C_0 \approx_s C'_0$.

Soundness and completeness

Theorem (Soundness)

If all leaves of a tree, whose root is labeled with (C_0, C_0') *(a pair of initial constraints), are labeled either with* $(\bot, \bot)$ *or with some* (C, C') *with* $C \neq \bot, C' \neq \bot$*, then* $C_0 \approx_s C_0'$.

Theorem (Completeness)

If (C_0, C_0') *is a pair of initial constraints such that* $C_0 \approx_s C_0'$*, then all leaves of a tree, whose root is labeled with* (C_0, C_0')*, are labeled either with* $(\bot, \bot)$ *or with some* (C, C') *with* $C \neq \bot$ *and* $C' \neq \bot$.

Termination problem

Consider the initial pair of contraints (C, C') given below:

$$C = \begin{cases} a & \vdash & \mathrm{enc}(x_1, x_2) \\ a, b & \vdash & x_1 \end{cases} \qquad C' = \begin{cases} a & \vdash & y_1 \\ a, b & \vdash & \mathrm{enc}(y_1, y_2) \end{cases}$$

Termination problem

$$C_1 = \begin{cases} a & \vdash & x_1 \\ a & \vdash & x_2 \\ a, b & \vdash & x_1 \end{cases} \qquad C'_1 = \begin{cases} a & \vdash & z_1 \\ a & \vdash & z_2 \\ a, b & \vdash & \mathsf{enc}(\mathsf{enc}(z_1, z_2), y_2) \end{cases}$$

with $y_1 \stackrel{?}{=} \mathsf{enc}(z_1, z_2)$

Termination problem

$$C_1 = \begin{cases} a & \vdash & \mathsf{enc}(t_1, t_2) \\ a & \vdash & x_2 \\ a, b & \vdash & t_1 \\ a, b & \vdash & t_2 \end{cases}$$

with $x_1 \stackrel{?}{=} \mathsf{enc}(t_1, t_2)$

$$C'_1 = \begin{cases} a & \vdash & z_1 \\ a & \vdash & z_2 \\ a, b & \vdash & \mathsf{enc}(z_1, z_2) \\ a, b & \vdash & y_2 \end{cases}$$

with $y_1 \stackrel{?}{=} \mathsf{enc}(z_1, z_2)$

Termination theorem

Theorem

There exists a strategy on the rules which terminates.

Demo

Demo

Future Works

Theory

1. Extension to non positive constraint systems (Ongoing work)
2. Extension to symbolic equivalence of constraint system set (Ongoing work)
3. Extension to trace equivalence of non deterministic protocol (Ongoing work)
4. Other cryptographic primitives

Implementation

1. Symbolic equivalence of positive constraint systems (Done)
2. Trace equivalence of positive protocol (Done but not efficient)