Symbolic Verification of Cryptographic Protocols

Deducibility Constraints

David Baelde

LSV, ENS Paris-Saclay & Prosecco, Inria Paris

2017
Messages as terms

Terms
Assume a set of variables $\mathcal{X}$, and a set of names $\mathcal{N}$.
Assume a signature $\Sigma = \Sigma_c \uplus \Sigma_d$: constructor and destructor symbols.
Terms $t$, $u$, $v$, etc. are elements of $T(\Sigma, \mathcal{X} \cup \mathcal{N})$.
Constructor terms (messages) are elements of $T(\Sigma_c, \mathcal{N}) = \mathcal{M}$.

Equational theory
An equational theory is given by means of a finite set of equations. It represents (some) possible computations on terms.
Example: rewrite rules for standard primitives

Standard equational theory

The equational theory $\text{Estd}$ is given by:

\[
\text{sdec}(\text{senc}(x, y), y) =_{\text{Estd}} x
\]
\[
\text{adec}(\text{aenc}(x, \text{pub}(y)), y) =_{\text{Estd}} x
\]
\[
\text{proj}_i(\langle x_1, x_2 \rangle) =_{\text{Estd}} x_i
\]

Proposition

There exists a subterm-convergent rewrite system $\rightarrow$ such that the following conditions are equivalent:

- $u =_{\text{Estd}} v$;
- $u \leftrightarrow^* v$;
Example: rewrite rules for standard primitives

Standard equational theory

The equational theory $\text{Estd}$ is given by:

$$\text{sdec(\text{senc}(x, y), y)} = \text{Estd } x$$

$$\text{adec(\text{aenc}(x, \text{pub}(y)), y)} = \text{Estd } x$$

$$\text{proj}_i(\langle x_1, x_2 \rangle) = \text{Estd } x_i$$

Proposition

*There exists a subterm-convergent rewrite system $\rightarrow$ such that the following conditions are equivalent:*

- $u = \text{Estd } v$;
- $u \leftrightarrow^* v$;
- $u \rightarrow^* w \leftarrow^* v$ for some $w$;
Example: rewrite rules for standard primitives

Standard equational theory

The equational theory $E_{\text{std}}$ is given by:

\[
\text{sdec}(\text{senc}(x, y), y) =_{E_{\text{std}}} x \quad \text{adec}(\text{aenc}(x, \text{pub}(y)), y) =_{E_{\text{std}}} x
\]

\[
\text{proj}_i(\langle x_1, x_2 \rangle) =_{E_{\text{std}}} x_i
\]

Proposition

*There exists a subterm-convergent rewrite system $\rightarrow$ such that the following conditions are equivalent:*

- $u =_{E_{\text{std}}} v$;
- $u \leftrightarrow^* v$;
- $u \rightarrow^* w \leftarrow^* v$ for some $w$;
- $u \rightarrow^* w \leftarrow^* v$ for some constructor term $w$.  

David Baelde (ENS Saclay & Inria Paris)
Processes

Syntax

\[ P, Q, R ::= \text{in}(c, x).P \mid \text{out}(c, u).P \]
\[ \mid \text{if } u = v \text{ then } P \text{ else } Q \]
\[ \mid 0 \mid (P \mid Q) \mid \text{new } x.P \mid !P \]

Structural congruence

Let \( \equiv \) be the least congruence such that:

\[ 0 \mid P \equiv P \]
\[ P \mid Q \equiv Q \mid P \]
\[ P \mid (Q \mid R) \equiv (P \mid Q) \mid R \]
Process semantics

Reduction semantics

Rules can be applied modulo $\equiv$:

\[
\begin{align*}
\text{in}(c, x).P \mid \text{out}(c, u).Q \mid R & \rightsquigarrow P[x := u] \mid Q \mid R \\
& \quad \text{when } u =_{E} m \in \mathcal{M} \\
\text{if } u = v \text{ then } P \text{ else } Q \mid R & \rightsquigarrow P \mid R \\
& \quad \text{when } u =_{E} v \\
\text{if } u = v \text{ then } P \text{ else } Q \mid R & \rightsquigarrow Q \mid R \\
& \quad \text{when } u \neq_{E} v \\
(\text{new } x.P) \mid Q & \rightsquigarrow P[x := n] \mid Q \\
& \quad \text{when } n \text{ if fresh} \\
!P \mid Q & \rightsquigarrow P \mid !P \mid Q
\end{align*}
\]
Example: Needham-Schroeder

<table>
<thead>
<tr>
<th>$I(s_{ka}, p_{kb})$</th>
<th>$R(s_{kb}, n_{b}, honest)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>new $n_{a}$.</td>
<td>in($c, y$).</td>
</tr>
<tr>
<td>out($c, a_{enc}(\langle pub(s_{ka}), n_{a}\rangle, p_{kb})$).</td>
<td>let $p_{ka} = proj_1(adec(y, s_{kb})$) in</td>
</tr>
<tr>
<td></td>
<td>let $n_{a} = proj_2(adec(y, s_{kb})$) in</td>
</tr>
<tr>
<td>in($c, x$).</td>
<td>out($c, a_{enc}(\langle n_{a}, n_{b}\rangle, p_{ka})$).</td>
</tr>
<tr>
<td>if $n_{a} = proj_1(adec(x, s_{ka})$ then</td>
<td>in($c, z$).</td>
</tr>
<tr>
<td>out($c, a_{enc}(proj_2(adec(x, s_{ka})), p_{kb})$)</td>
<td>if $n_{b} = adec(z, s_{kb})$ then</td>
</tr>
<tr>
<td></td>
<td>if $p_{ka} = honest$ then</td>
</tr>
<tr>
<td></td>
<td>out($c, s_{enc}(secret, n_{b})$)</td>
</tr>
</tbody>
</table>
Example: Needham-Schroeder

\[
\begin{align*}
I(ska, pk_b) & \quad R(skb, nb, honest) \\
\text{new } n_a. & \quad \text{in}(c, y). \\
\text{out}(c, \text{aenc}(\langle \text{pub}(sk_a), n_a \rangle, pk_b)). & \quad \text{let } pk_a = \text{proj}_1(\text{adec}(y, skb)) \text{ in} \\
\text{in}(c, x). & \quad \text{let } n_a = \text{proj}_2(\text{adec}(y, skb)) \text{ in} \\
\text{if } n_a = \text{proj}_1(\text{adec}(x, sk_a) \text{ then} & \quad \text{out}(c, \text{aenc}(\langle n_a, nb \rangle, pk_a)). \\
\text{out}(c, \text{aenc}(\text{proj}_2(\text{adec}(x, sk_a)), pk_b)) & \quad \text{in}(c, z). \\
\text{if } n_a = \text{proj}_1(\text{adec}(x, sk_a) \text{ then} & \quad \text{if } pk_a = \text{honest} \text{ then} \\
\text{out}(c, \text{aenc}(\text{proj}_2(\text{adec}(x, sk_a)), pk_b)) & \quad \text{out}(c, \text{senc}(\text{secret}, nb))
\end{align*}
\]

Scenario \((sk_a, skb, nb \in \mathcal{N})\)

\[
\text{out}(c, \langle \text{pub}(sk_a), \text{pub}(skb) \rangle). \quad (I(ska, \text{pub}(skb)) \mid R(skb, nb, \text{pub}(sk_a)))
\]
Example: Needham-Schroeder

<table>
<thead>
<tr>
<th>$I(\text{sk}_a, \text{pk}_b)$</th>
<th>$R(\text{sk}_b, n_b, \text{honest})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>new $n_a$.</td>
<td></td>
</tr>
<tr>
<td>out($c$, aenc($\langle\text{pub}(\text{sk}_a), n_a\rangle$, $\text{pk}_b$)).</td>
<td></td>
</tr>
<tr>
<td>in($c$, $x$).</td>
<td></td>
</tr>
<tr>
<td>if $n_a = \text{proj}_1(\text{adec}(x, \text{sk}_a))$ then</td>
<td></td>
</tr>
<tr>
<td>out($c$, aenc($\text{proj}_2(\text{adec}(x, \text{sk}_a))$, $\text{pk}_b$))</td>
<td></td>
</tr>
<tr>
<td>if $n_a = \text{proj}_1(\text{adec}(y, \text{sk}_b))$ in</td>
<td></td>
</tr>
<tr>
<td>let $\text{pk}_a = \text{proj}_1(\text{adec}(y, \text{sk}_b))$ in</td>
<td></td>
</tr>
<tr>
<td>let $n_a = \text{proj}_2(\text{adec}(y, \text{sk}_b))$ in</td>
<td></td>
</tr>
<tr>
<td>out($c$, aenc($\langle n_a, n_b \rangle$, $\text{pk}_a$))</td>
<td></td>
</tr>
<tr>
<td>in($c$, $z$).</td>
<td></td>
</tr>
<tr>
<td>if $n_b = \text{adec}(z, \text{sk}_b)$ then</td>
<td></td>
</tr>
<tr>
<td>if $\text{pk}_a = \text{honest}$ then</td>
<td></td>
</tr>
<tr>
<td>out($c$, senc(secret, $n_b$))</td>
<td></td>
</tr>
</tbody>
</table>

Scenario ($\text{sk}_a, \text{sk}_b, n_b, \text{sk}_i \in \mathcal{N}$)

$$\text{out}(c, \langle \text{sk}_i, \text{pub}(\text{sk}_a), \text{pub}(\text{sk}_b)\rangle). \ (I(\text{sk}_a, \text{pub}(\text{sk}_i)) \ | \ R(\text{sk}_b, n_b, \text{pub}(\text{sk}_a)))$$
Secrecy

Definition

*P does not ensure the secrecy of u if*, for some *A* in which no name occurs free, and some arbitrary *Q*,

\[ P \mid A \sim^* \text{out}(c, u).0 \mid Q \]
Secrecy

Definition

$P$ does not ensure the secrecy of $u$ if, for some $A$ in which no name occurs free, and some arbitrary $Q$, 

$$P \mid A \rightsquigarrow^* \text{out}(c, u).0 \mid Q$$

A lot of redundancy in that definition!
A **configuration** is a pair \((P, \Phi)\) where
- \(P\) is a ground process; (processes still identified up to \(\equiv\))
- \(\Phi \subseteq \mathcal{M}\) is called a frame. (attacker’s knowledge)

\[
\begin{align*}
\text{(out}(c, u).P \mid Q, \Phi) & \xrightarrow{\text{out}(c, u)} (P \mid Q, \Phi \cup \{u\}) & \text{where } u =_E v \in \mathcal{M} \\
\text{(in}(c, x).P \mid Q, \Phi) & \xrightarrow{\text{in}(c, u)} (P[x := u] \mid Q, \Phi) & \text{where } u \in \mathcal{M}, u =_E t \text{ for some } t \in T(\Sigma, \Phi) \\
\text{(if } u = v \text{ then } P \text{ else } Q \mid R, \Phi) & \xrightarrow{\tau} (P \mid R, \Phi) & \text{when } u =_E v \\
\text{(if } u = v \text{ then } P \text{ else } Q \mid R, \Phi) & \xrightarrow{\tau} (Q \mid R, \Phi) & \text{when } u \not= _E v \\
\text{(new } x.P) \mid Q, \Phi) & \xrightarrow{\tau} (P[x := n] \mid Q, \Phi) & \text{for some fresh } n \\
(!P \mid Q, \Phi) & \xrightarrow{\tau} (P \mid !P \mid Q, \Phi)
\end{align*}
\]
Theorem

\( P \) does not ensure the secrecy of \( u \) iff

\[ \exists \text{ tr, } P', \Phi', \; t \in T(\Sigma, \Phi') \text{ such that } (P, \emptyset) \xrightarrow{\text{tr}} (P', \Phi') \text{ and } u \equivE t. \]
Assume a slight simplification: attackers do not use ! and new.

**Theorem**

$P$ does not ensure the secrecy of $u$ iff

$$\exists \ tr, P', \Phi', t \in T(\Sigma, \Phi') \ such \ that \ (P, \emptyset) \xrightarrow{tr} (P', \Phi') \ and \ u =_{E} t.$$  

More generally, the following are equivalent:

- there is a trace $(P, \Phi) \xrightarrow{tr} (P', \Phi')$ such that $u =_{E} t \in T(\Sigma, \Phi')$;
- there is an attacker $A$ with terms in $T(\Sigma, \mathcal{X} \cup \Phi)$ such that $P \mid A \rightsquigarrow^{*} Q \mid \text{out}(c, u)$ for some $Q$. 

Assume a slight simplification: attackers do not use ! and new.

**Theorem**

*P does not ensure the secrecy of u iff*

\[ \exists \operatorname{tr}, P', \Phi', t \in T(\Sigma, \Phi') \text{ such that } (P, \emptyset) \xrightarrow{\operatorname{tr}} (P', \Phi') \text{ and } u =_E t. \]

More generally, the following are equivalent:

- there is a trace \((P, \Phi) \xrightarrow{\operatorname{tr}} (P', \Phi')\) such that \(u =_E t \in T(\Sigma, \Phi')\);
- there is an attacker \(A\) with terms in \(T(\Sigma, \mathcal{X} \cup \Phi)\) such that \(P \mid A \rightsquigarrow^* Q \mid \operatorname{out}(c, u)\) for some \(Q\).

**Note:** adding a communication rule to the LTS would not change anything.
A trivial modification

We don’t care **how** a term can be derived, but only **if** it can be.

**Deduction**

Assume a relation $S \vdash u$ such that

$S \vdash u \iff u \in \mathcal{M}$ and there exists $t \in \mathcal{T}(\Sigma, S)$ such that $t \equiv_E u$.

**Modified LTS**

$$(\text{in}(c, x).P \mid Q, \Phi) \xrightarrow{\text{in}(c,u)} (P[x := u] \mid Q, \Phi) \text{ when } \Phi \vdash u$$
Example: Deduction system for standard primitives

\[
\frac{u \quad v}{\langle u, v \rangle} \quad \frac{\langle u, v \rangle}{u} \quad \frac{\langle u, v \rangle}{v} \quad \frac{u}{\text{pub}(u)}
\]
Example: Deduction system for standard primitives

\[
\begin{array}{cccc}
  \frac{u \quad v}{\langle u, v \rangle} & \frac{\langle u, v \rangle}{u} & \frac{\langle u, v \rangle}{v} & \frac{u}{\text{pub}(u)} \\
  \frac{u \quad v}{\text{senc}(u, v)} & \frac{\text{senc}(u, v) \quad v}{u} & \frac{u \quad v}{\text{aenc}(u, v)} & \frac{\text{aenc}(u, \text{pub}(v)) \quad v}{u}
\end{array}
\]

Example: Deduction system for standard primitives

\[
\begin{align*}
\frac{u \quad v}{\langle u, v \rangle} & \quad \frac{\langle u, v \rangle}{u} & \quad \frac{\langle u, v \rangle}{v} & \quad \frac{u}{\text{pub}(u)} \\
\frac{u \quad v}{\text{senc}(u, v)} & \quad \frac{\text{senc}(u, v) \quad v}{u} & \quad \frac{u \quad v}{\text{aenc}(u, v)} & \quad \frac{\text{aenc}(u, \text{pub}(v)) \quad v}{u}
\end{align*}
\]


**Lemma**

For all \( S \subseteq \mathcal{M} \),

\[ S \vdash_{\text{std}} u \iff u \in \mathcal{M} \text{ and } \exists t \in T(\Sigma_{\text{std}}, S) \text{ such that } t =_{\text{Estd}} u. \]
The insecurity problem

From now on, restrict to the standard primitives: $\text{senc}(\cdot, \cdot), \text{aenc}(\cdot, \cdot), \langle \cdot, \cdot \rangle$.

The insecurity problem

Given some $(P, \Phi)$ and $u \in M$, does there exist $(P, \Phi) \xrightarrow{\text{tr}} (P', \Phi')$ such that $\Phi' \vdash u$?

Remarks:

- Undecidable for unbounded number of sessions.
- NP-hard for bounded number of sessions.

Next:
- Symbolic verification and constraint solving yields NP procedure.
Intruder detection

Problem
Given $S \subseteq M$ and $u \in M$, does $S \vdash u$?

Theorem
For the standard primitives, the intruder detection problem is in PTIME.

Proof sketch.
Say that a derivation is non-repeating when its branches never contain a repetition of a term.
In such derivations, the first premise of a decomposition must be derived by another decomposition or an axiom.
A non-repeating derivation of $T \vdash v$ may only contain subterms of either $T$ or $v$.
One can check in PTIME whether there exists a derivation of $S \vdash u$ featuring only subterms of $S$ and $u$. 
Deducibility constraints

**Definition**

A deducibility constraint system is either $\bot$ or a (possibly empty) conjunction of **deducibility constraints** of the form

\[ T_1 \vdash ? u_1 \land \ldots \land T_n \vdash ? u_n \]

such that

- $\emptyset \neq T_1 \subseteq T_2 \subseteq \ldots \subseteq T_n$ (monotonicity)
- for every $i$, $\text{fv}(T_i) \subseteq \text{fv}(u_1, \ldots, u_{i-1})$ (origination)

**Definition**

The substitution $\sigma$ is a **solution** of $C = T_1 \vdash ? u_1 \land \ldots \land T_n \vdash ? u_n$ when $T_i \sigma \vdash u_i \sigma$ for all $i$. 
Example: Needham-Schroeder

- $S_1 := \langle sk_i, \text{pub}(sk_a), \text{pub}(sk_b) \rangle, \text{aenc}(\langle \text{pub}(sk_a), n_a \rangle, \text{pub}(sk_i))$

$S_1 \vdash ? \ x$
Example: Needham-Schroeder

- \( S_1 := \langle sk_i, \text{pub}(sk_a), \text{pub}(sk_b) \rangle, \text{aenc}(\langle \text{pub}(sk_a), n_a \rangle, \text{pub}(sk_i)) \)

- \( S_1 \vdash? \text{aenc}(\langle x_a, x_{na} \rangle, \text{pub}(sk_b)) \)
Example: Needham-Schroeder

\[ S_1 := \langle sk_i, pub(sk_a), pub(sk_b) \rangle, \text{aenc}(\langle pub(sk_a), n_a \rangle, pub(sk_i)) \]
\[ S_1 \vdash? \text{aenc}(\langle x_a, x_{na} \rangle, pub(sk_b)) \]

\[ S_2 := S_1, \text{aenc}(\langle x_{na}, n_b \rangle, x_a) \]
\[ S_2 \vdash? \text{aenc}(\langle n_a, x_{nb} \rangle, pub(sk_a)) \]
Example: Needham-Schroeder

- $S_1 := \langle sk_i, \text{pub}(sk_a), \text{pub}(sk_b) \rangle, \text{aenc}(\langle \text{pub}(sk_a), n_a \rangle, \text{pub}(sk_i))$
  $S_1 \vdash ? \text{aenc}(\langle x_a, x_{na} \rangle, \text{pub}(sk_b))$

- $S_2 := S_1, \text{aenc}(\langle x_{na}, n_b \rangle, x_a)$
  $S_2 \vdash ? \text{aenc}(\langle n_a, x_{nb} \rangle, \text{pub}(sk_a))$

- $S_3 := S_2, \text{aenc}(x_{nb}, \text{pub}(sk_i))$
  $S_3 \vdash ? \text{aenc}(n_b, \text{pub}(sk_b))$
Example: Needham-Schroeder

- \( S_1 \) := \( \langle sk_i, \text{pub}(sk_a), \text{pub}(sk_b) \rangle, \text{aenc}(\langle \text{pub}(sk_a), n_a \rangle, \text{pub}(sk_i)) \)
  - \( S_1 \vdash ? \text{aenc}(\langle x_a, x_{na} \rangle, \text{pub}(sk_b)) \)

- \( S_2 \) := \( S_1, \text{aenc}(\langle x_{na}, n_b \rangle, x_a) \)
  - \( S_2 \vdash ? \text{aenc}(\langle n_a, x_{nb} \rangle, \text{pub}(sk_a)) \)

- \( S_3 \) := \( S_2, \text{aenc}(x_{nb}, \text{pub}(sk_i)) \)
  - \( S_3 \vdash ? \text{aenc}(n_b, \text{pub}(sk_b)) \)

- \( S_4 \) := \( S_3, \text{senc}(\text{secret}, n_b) \) and \( x_a = \text{pub}(sk_a) \)
  - \( S_4 \vdash ? \text{secret} \)
A system is solved if it is of the form

\[ T_1 \vdash ? x_1 \land \ldots \land T_n \vdash ? x_n \]

**Proposition**

*If \( C \) is solved, then it admits a solution.*
Solved form

A system is solved if it is of the form

\[ T_1 \vdash \exists x_1 \land \ldots \land T_n \vdash \exists x_n \]

Proposition

If \( C \) is solved, then it admits a solution.

Theorem

There exists a terminating relation \( \rightsquigarrow \) such that for any \( C \) and \( \theta \), \( \theta \in \text{Sol}(C) \) iff there is \( C \rightsquigarrow^* C' \) solved and \( \theta = \sigma \theta' \) for some \( \theta' \in \text{Sol}(C') \).
Simplification of constraint systems

Here systems are considered modulo AC of $\land$.

$$(R_1) \quad C \land T \vdash? u \rightsquigarrow C \quad \text{if } T \cup \{x \mid (T' \vdash? x) \in C, T' \subset T\} \vdash u$$

$$(R_2) \quad C \land T \vdash? u \rightsquigarrow \sigma C\sigma \land T\sigma \vdash? u\sigma \quad \text{if } \sigma = \text{mgu}(t, u), t \in \text{st}(T), t \neq u, \text{ and } t, u \notin X$$

$$(R_3) \quad C \land T \vdash? u \rightsquigarrow \sigma C\sigma \land T\sigma \vdash? u\sigma \quad \text{if } \sigma = \text{mgu}(t_1, t_2), t_1, t_2 \in \text{st}(T), t_1 \neq t_2$$

$$(R_4) \quad C \land T \vdash? u \rightsquigarrow \bot \quad \text{if } \text{fv}(T \cup \{u\}) = \emptyset, T \not\vdash u$$

$$(R_f) \quad C \land T \vdash? f(u_1, \ldots, u_n) \rightsquigarrow C \land \bigwedge_i T \vdash? u_i \quad \text{for } f \in \Sigma_c$$

$$(R_{\text{pub}}) \quad C \rightsquigarrow C[x := \text{pub}(x)] \quad \text{if } \text{aenc}(t, x) \in T \text{ for some } (T \vdash? u) \in C$$
Examples of simplifications

1. \( \text{senc}(n, k) \vdash? \text{senc}(x, k) \)

2. \( \text{senc}(\text{senc}(t_1, k), k) \vdash? \text{senc}(x, k) \) (two opportunities for \( R_2 \))

3. \( S \vdash? x \land S, n \vdash? y \land S, n, \text{senc}(m, \text{senc}(x, k)), \text{senc}(y, k) \vdash? m \)

4. \( S \vdash? x \land S \vdash? \langle x, x \rangle \)

5. \( n \vdash? x \land n \vdash? \text{senc}(x, k) \)
Proposition (Validity)

If $C$ is a deducibility constraint system, and $C \xrightarrow{\sigma} C'$, then $C'$ is a deducibility constraint system.
Proposition (Validity)

If $C$ is a deducibility constraint system, and $C \leadsto_{\sigma} C'$, then $C'$ is a deducibility constraint system.

Proposition (Soundness)

If $C \leadsto_{\sigma} C'$ and $\theta \in \text{Sol}(C')$ then $\sigma\theta \in \text{Sol}(C)$.

Proposition (Termination)

Simplifications are terminating, as shown by the termination measure $(v(C), p(C), s(C))$ where:

- $v(C)$ is the number of variables occurring in $C$;
- $p(C)$ is the number of terms of the form $aenc(u, x)$ occurring on the left of constraints in $C$;
- $s(C)$ is the total size of the right-hand sides of constraints in $C$. 
Left-minimality & Simplicity

A derivation $\Pi$ of $T_i \vdash u$ is left-minimal if, whenever $T_j \vdash u$, $\Pi$ is also a derivation of $T_j \vdash u$.

A derivation is simple it is non-repeating and all its subderivations are left-minimal.

Proposition

If $T_i \vdash u$, then it has a simple derivation.

Lemma

Let $C = \bigwedge_j T_j \vdash^? u_j$ be a constraint system, $\theta \in \text{Sol}(C)$, and $i$ be such that $u_j \in X$ for all $j < i$.

If $T_i\theta \vdash u$ with a simple derivation starting with an axiom or a decomposition, then there is $t \in \text{subterm}(T_i) \setminus X$ such that $t\theta = u$. 

David Baelde (ENS Saclay & Inria Paris)
Constraint simplification proof (3)

Lemma

Let $\mathcal{C} = \bigwedge_j T_j \vdash u_j$, $\sigma \in \text{Sol}(\mathcal{C})$.
Let $i$ be a minimal index such that $u_i \notin \mathcal{X}$.
Assume that:

- $T_i$ does not contain two subterms $t_1 \neq t_2$ such that $t_1\sigma = t_2\sigma$;
- $T_i$ does not contain any subterm of the form $\text{aenc}(t, x)$;
- $u_i$ is a non-variable subterm of $T_i$.

Then $T'_i \vdash u_i$, where $T'_i = T_i \cup \{x \mid (T \vdash x) \in \mathcal{C}, T \subsetneq T_i\}$. 
Lemma

Let $C = \bigwedge_j T_j \vdash^? u_j$, $\sigma \in \text{Sol}(C)$. Let $i$ be a minimal index such that $u_i \not\in \mathcal{X}$. Assume that:

- $T_i$ does not contain two subterms $t_1 \neq t_2$ such that $t_1\sigma = t_2\sigma$;
- $T_i$ does not contain any subterm of the form $a_{\text{enc}}(t, x)$;
- $u_i$ is a non-variable subterm of $T_i$.

Then $T_i' \vdash u_i$, where $T_i' = T_i \cup \{x \mid (T \vdash^? x) \in C, T \subsetneq T_i\}$.

Proposition (Completeness)

If $C$ is unsolved and $\theta \in \text{Sol}(C)$, there is $C \rightsquigarrow_\sigma C'$ and $\theta' \in \text{Sol}(C')$ such that $\theta = \sigma\theta'$.
Concluding remarks

Improvements

- A complete strategy can yield a polynomial bound, hence a small attack property
- Equalities and disequalities may be added
- Several variants and extensions may be considered: sk instead of pub, signatures, xor, etc.

We have not answered the original question yet!

- Symbolic semantics, (dis)equality constraints
- The enumeration of all interleavings is too naive

Complexity

- Deciding whether a system has a solution is NP-hard
- Reminder: for a general theory, security is undecidable