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 $\mathcal{T}(\Sigma, \mathcal{X} \cup \mathcal{N})$.
Constructor terms (messages) are elements of $\mathcal{T}(\Sigma_c, \mathcal{N}) = \mathcal{M}$.

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 Estd is given by:

$$\mathrm{sdec}(\mathrm{senc}(x, y), y) =_{\mathsf{Estd}} x \qquad \mathrm{adec}(\mathrm{aenc}(x, \mathrm{pub}(y)), y) =_{\mathsf{Estd}} x$$

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

Proposition

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

- $u =_{\mathsf{Estd}} v$;
- $u \leftrightarrow^* v$;

Example: rewrite rules for standard primitives

Standard equational theory

The equational theory Estd is given by:

$$\mathsf{sdec}(\mathsf{senc}(x, y), y) =_{\mathsf{Estd}} x \qquad \mathsf{adec}(\mathsf{aenc}(x, \mathsf{pub}(y)), y) =_{\mathsf{Estd}} x$$

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

Proposition

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

- $u =_{\mathsf{Estd}} v$;
- $u \leftrightarrow^* v$;
- $u \rightarrow^* w \leftarrow^* v$ *for some* w;

Example: rewrite rules for standard primitives

Standard equational theory

The equational theory Estd is given by:

$$\mathsf{sdec}(\mathsf{senc}(x,y),y) =_{\mathsf{Estd}} x \qquad \mathsf{adec}(\mathsf{aenc}(x,\mathsf{pub}(y)),y) =_{\mathsf{Estd}} x$$

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

Proposition

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

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

Processes

Syntax

$$
\begin{array}{rcl}
P, Q, R & ::= & \mathsf{in}(c, x).P \quad | \quad \mathsf{out}(c, u).P \\
& | & \mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \\
& | & 0 \quad | \quad (P \mid Q) \quad | \quad \mathsf{new}\ x.P \quad | \quad !P
\end{array}
$$

Structural congruence

Let $\equiv$ be the least congruence such that:

$$0 \mid P \equiv P \qquad P \mid Q \equiv Q \mid P \qquad P \mid (Q \mid R) \equiv (P \mid Q) \mid R$$

Process semantics

Reduction semantics

Rules can be applied modulo $\equiv$:

$$\mathsf{in}(c,x).P \mid \mathsf{out}(c,u).Q \mid R \quad \rightsquigarrow \quad P[x := u] \mid Q \mid R \quad \text{when } u =_{\mathsf{E}} m \in \mathcal{M}$$

$$\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \mid R \quad \rightsquigarrow \quad P \mid R \qquad \text{when } u =_{\mathsf{E}} v$$

$$\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \mid R \quad \rightsquigarrow \quad Q \mid R \qquad \text{when } u \neq_{\mathsf{E}} v$$

$$(\mathsf{new}\ x.P) \mid Q \quad \rightsquigarrow \quad P[x := n] \mid Q \qquad \text{when } n \text{ if fresh}$$

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

Example: Needham-Schroeder

$\mathsf{I}(sk_a, pk_b)$	$\mathsf{R}(sk_b, n_b, honest)$
new n_a.	
out$(c, \mathrm{aenc}(\langle \mathrm{pub}(sk_a), n_a \rangle, pk_b))$.	in(c, y).
	let $pk_a = \mathrm{proj}_1(\mathrm{adec}(y, sk_b))$ in
	let $n_a = \mathrm{proj}_2(\mathrm{adec}(y, sk_b))$ in
in(c, x).	out$(c, \mathrm{aenc}(\langle n_a, n_b \rangle, pk_a))$.
if $n_a = \mathrm{proj}_1(\mathrm{adec}(x, sk_a)$ then	
out$(c, \mathrm{aenc}(\mathrm{proj}_2(\mathrm{adec}(x, sk_a)), pk_b))$	in(c, z).
	if $n_b = \mathrm{adec}(z, sk_b)$ then
	if $pk_a = honest$ then
	out$(c, \mathrm{senc}(secret, n_b))$

Example: Needham-Schroeder

$\mathsf{I}(sk_a, pk_b)$	$\mathsf{R}(sk_b, n_b, honest)$
new n_a.	
out$(c, \mathsf{aenc}(\langle \mathsf{pub}(sk_a), n_a\rangle, pk_b))$.	in(c, y).
	let $pk_a = \mathsf{proj}_1(\mathsf{adec}(y, sk_b))$ in
	let $n_a = \mathsf{proj}_2(\mathsf{adec}(y, sk_b))$ in
in(c, x).	out$(c, \mathsf{aenc}(\langle n_a, n_b\rangle, pk_a))$.
if $n_a = \mathsf{proj}_1(\mathsf{adec}(x, sk_a)$ then	
out$(c, \mathsf{aenc}(\mathsf{proj}_2(\mathsf{adec}(x, sk_a)), pk_b))$	in(c, z).
	if $n_b = \mathsf{adec}(z, sk_b)$ then
	if $pk_a = honest$ then
	out$(c, \mathsf{senc}(secret, n_b))$

Scenario $(sk_a, sk_b, n_b \in \mathcal{N})$

out$(c, \langle \mathsf{pub}(sk_a), \mathsf{pub}(sk_b)\rangle)$. $(\mathsf{I}(sk_a, \mathsf{pub}(sk_b)) \mid \mathsf{R}(sk_b, n_b, \mathsf{pub}(sk_a)))$

Example: Needham-Schroeder

$\mathsf{I}(sk_a, pk_b)$	$\mathsf{R}(sk_b, n_b, honest)$
$\mathsf{new}\ n_a.$	
$\mathsf{out}(c, \mathrm{aenc}(\langle \mathrm{pub}(sk_a), n_a \rangle, pk_b)).$	$\mathsf{in}(c, y).$
	$\mathsf{let}\ pk_a = \mathrm{proj}_1(\mathrm{adec}(y, sk_b))\ \mathsf{in}$
	$\mathsf{let}\ n_a = \mathrm{proj}_2(\mathrm{adec}(y, sk_b))\ \mathsf{in}$
$\mathsf{in}(c, x).$	$\mathsf{out}(c, \mathrm{aenc}(\langle n_a, n_b \rangle, pk_a)).$
$\mathsf{if}\ n_a = \mathrm{proj}_1(\mathrm{adec}(x, sk_a)\ \mathsf{then}$	
$\mathsf{out}(c, \mathrm{aenc}(\mathrm{proj}_2(\mathrm{adec}(x, sk_a)), pk_b))$	$\mathsf{in}(c, z).$
	$\mathsf{if}\ n_b = \mathrm{adec}(z, sk_b)\ \mathsf{then}$
	$\mathsf{if}\ pk_a = honest\ \mathsf{then}$
	$\mathsf{out}(c, \mathrm{senc}(secret, n_b))$

Scenario $(sk_a, sk_b, n_b, sk_i \in \mathcal{N})$

$\mathsf{out}(c, \langle sk_i, \mathrm{pub}(sk_a), \mathrm{pub}(sk_b) \rangle).\ (\mathsf{I}(sk_a, \mathrm{pub}(sk_i)) \mid \mathsf{R}(sk_b, n_b, \mathrm{pub}(sk_a)))$

Secrecy

Definition

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

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

Secrecy

Definition

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

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

A lot of redundancy in that definition!

Labelled transition system

A configuration is a pair (P, Φ) where

- P is a ground process; (processes still identified up to $\equiv$)
- $\Phi \subseteq \mathcal{M}$ is called a frame. (attacker's knowledge)

$$(\mathsf{out}(c,u).P \mid Q, \Phi) \xrightarrow{\mathsf{out}(c,u)} (P \mid Q, \Phi \cup \{u\}) \qquad \text{where } u =_\mathsf{E} v \in \mathcal{M}$$

$$(\mathsf{in}(c,x).P \mid Q, \Phi) \xrightarrow{\mathsf{in}(c,u)} (P[x := u] \mid Q, \Phi)$$
$$\text{where } u \in \mathcal{M},\ u =_\mathsf{E} t \text{ for some } t \in \mathcal{T}(\Sigma, \Phi)$$

$$(\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \mid R, \Phi) \xrightarrow{\tau} (P \mid R, \Phi) \qquad \text{when } u =_\mathsf{E} v$$

$$(\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \mid R, \Phi) \xrightarrow{\tau} (Q \mid R, \Phi) \qquad \text{when } u \neq_\mathsf{E} v$$

$$((\mathsf{new}\ x.P) \mid Q, \Phi) \xrightarrow{\tau} (P[x := n] \mid Q, \Phi) \qquad \text{for some fresh } n$$

$$(!P \mid Q, \Phi) \xrightarrow{\tau} (P \mid !P \mid Q, \Phi)$$

Reduction semantics vs. LTS

Theorem

P does not ensure the secrecy of u iff

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

Reduction semantics vs. LTS

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

Theorem

P does not ensure the secrecy of u iff

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

More generally, the following are equivalent:

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

Reduction semantics vs. LTS

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

Theorem

P does not ensure the secrecy of u iff

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

More generally, the following are equivalent:

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

$$S \vdash u \quad \text{iff} \quad u \in \mathcal{M} \text{ and there exists } t \in \mathcal{T}(\Sigma, S) \text{ such that } t =_{\mathsf{E}} u.$$

Modified LTS

$$(\mathsf{in}(c,x).P \mid Q, \Phi) \xrightarrow{\mathsf{in}(c,u)} (P[x := u] \mid Q, \Phi) \text{ when } \Phi \vdash u$$

Example: Deduction system for standard primitives

$$\frac{u \quad v}{\langle u, v\rangle} \qquad \frac{\langle u, v\rangle}{u} \qquad \frac{\langle u, v\rangle}{v} \qquad \frac{u}{\mathrm{pub}(u)}$$

Example: Deduction system for standard primitives

$$\frac{u \quad v}{\langle u, v\rangle} \qquad \frac{\langle u, v\rangle}{u} \qquad \frac{\langle u, v\rangle}{v} \qquad \frac{u}{\mathsf{pub}(u)}$$

$$\frac{u \quad v}{\mathsf{senc}(u, v)} \qquad \frac{\mathsf{senc}(u, v) \quad v}{u} \qquad \frac{u \quad v}{\mathsf{aenc}(u, v)} \qquad \frac{\mathsf{aenc}(u, \mathsf{pub}(v)) \quad v}{u}$$

Terminology: composition and decomposition rules.

Example: Deduction system for standard primitives

$$\frac{u \quad v}{\langle u, v\rangle} \qquad \frac{\langle u, v\rangle}{u} \qquad \frac{\langle u, v\rangle}{v} \qquad \frac{u}{\mathrm{pub}(u)}$$

$$\frac{u \quad v}{\mathrm{senc}(u, v)} \qquad \frac{\mathrm{senc}(u, v) \quad v}{u} \qquad \frac{u \quad v}{\mathrm{aenc}(u, v)} \qquad \frac{\mathrm{aenc}(u, \mathrm{pub}(v)) \quad v}{u}$$

Terminology: composition and decomposition rules.

Lemma

For all $S \subseteq \mathcal{M}$,

$$S \vdash_{\mathrm{std}} u \quad \textit{iff} \quad u \in \mathcal{M} \textit{ and } \exists t \in \mathcal{T}(\Sigma_{\mathrm{std}}, S) \textit{ such that } t =_{\mathrm{Estd}} u.$$

The insecurity problem

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

The insecurity problem

Given some (P, Φ) and $u \in \mathcal{M}$,
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 $S \subseteq \mathcal{M}$ and $u \in \mathcal{M}$, 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 $\perp$ or a (possibly empty) conjunction of deducibility constraints of the form

$$T_1 \vdash^? u_1 \wedge \ldots \wedge T_n \vdash^? u_n$$

such that

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

Definition

The substitution σ is a solution of $\mathcal{C} = T_1 \vdash^? u_1 \wedge \ldots \wedge T_n \vdash^? u_n$ when $T_i\sigma \vdash u_i\sigma$ for all i.

Example: Needham-Schroeder

- $S_1 := \langle sk_i, \mathsf{pub}(sk_a), \mathsf{pub}(sk_b)\rangle, \mathsf{aenc}(\langle \mathsf{pub}(sk_a), n_a\rangle, \mathsf{pub}(sk_i))$
  $S_1 \vdash^? x$

Example: Needham-Schroeder

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

Example: Needham-Schroeder

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

Example: Needham-Schroeder

- $S_1 := \langle sk_i, \mathsf{pub}(sk_a), \mathsf{pub}(sk_b)\rangle, \mathsf{aenc}(\langle \mathsf{pub}(sk_a), n_a\rangle, \mathsf{pub}(sk_i))$
  $S_1 \vdash^? \mathsf{aenc}(\langle x_a, x_{na}\rangle, \mathsf{pub}(sk_b))$
- $S_2 := S_1, \mathsf{aenc}(\langle x_{na}, n_b\rangle, x_a)$
  $S_2 \vdash^? \mathsf{aenc}(\langle n_a, x_{nb}\rangle, \mathsf{pub}(sk_a))$
- $S_3 := S_2, \mathsf{aenc}(x_{nb}, \mathsf{pub}(sk_i))$
  $S_3 \vdash^? \mathsf{aenc}(n_b, \mathsf{pub}(sk_b))$

Example: Needham-Schroeder

- $S_1 := \langle sk_i, \mathrm{pub}(sk_a), \mathrm{pub}(sk_b)\rangle, \mathrm{aenc}(\langle \mathrm{pub}(sk_a), n_a\rangle, \mathrm{pub}(sk_i))$
  $S_1 \vdash^? \mathrm{aenc}(\langle x_a, x_{na}\rangle, \mathrm{pub}(sk_b))$
- $S_2 := S_1, \mathrm{aenc}(\langle x_{na}, n_b\rangle, x_a)$
  $S_2 \vdash^? \mathrm{aenc}(\langle n_a, x_{nb}\rangle, \mathrm{pub}(sk_a))$
- $S_3 := S_2, \mathrm{aenc}(x_{nb}, \mathrm{pub}(sk_i))$
  $S_3 \vdash^? \mathrm{aenc}(n_b, \mathrm{pub}(sk_b))$
- $S_4 := S_3, \mathrm{senc}(\mathrm{secret}, n_b)$ and $x_a = \mathrm{pub}(sk_a)$
  $S_4 \vdash^? \mathrm{secret}$

Constraint resolution

Solved form

A system is solved if it is of the form

$$T_1 \vdash^? x_1 \wedge \ldots \wedge T_n \vdash^? x_n$$

Proposition

If $\mathcal{C}$ is solved, then it admits a solution.

Constraint resolution

Solved form

A system is solved if it is of the form

$$T_1 \vdash^? x_1 \wedge \ldots \wedge T_n \vdash^? x_n$$

Proposition

If $\mathcal{C}$ is solved, then it admits a solution.

Theorem

There exists a terminating relation $\rightsquigarrow$ such that for any $\mathcal{C}$ and θ,
$\theta \in \mathrm{Sol}(\mathcal{C})$ iff there is $\mathcal{C} \rightsquigarrow^*_\sigma \mathcal{C}'$ solved and $\theta = \sigma\theta'$ for some $\theta' \in \mathrm{Sol}(\mathcal{C}')$.

Simplification of constraint systems

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

$$(R_1) \quad \mathcal{C} \wedge T \vdash^? u \ \rightsquigarrow \ \mathcal{C} \qquad \text{if } T \cup \{x \mid (T' \vdash^? x) \in \mathcal{C}, T' \subsetneq T\} \vdash u$$

$$(R_2) \quad \mathcal{C} \wedge T \vdash^? u \ \rightsquigarrow_\sigma \ \mathcal{C}\sigma \wedge T\sigma \vdash^? u\sigma$$
$$\text{if } \sigma = \mathrm{mgu}(t, u), t \in \mathrm{st}(T), t \neq u, \text{ and } t, u \notin \mathcal{X}$$

$$(R_3) \quad \mathcal{C} \wedge T \vdash^? u \ \rightsquigarrow_\sigma \ \mathcal{C}\sigma \wedge T\sigma \vdash^? u\sigma$$
$$\text{if } \sigma = \mathrm{mgu}(t_1, t_2), t_1, t_2 \in \mathrm{st}(T), t_1 \neq t_2$$

$$(R_4) \quad \mathcal{C} \wedge T \vdash^? u \ \rightsquigarrow \ \bot \qquad \text{if } \mathrm{fv}(T \cup \{u\}) = \emptyset, T \not\vdash u$$

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

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

Examples of simplifications

1. $\mathsf{senc}(n, k) \vdash^? \mathsf{senc}(x, k)$
2. $\mathsf{senc}(\mathsf{senc}(t_1, k), k) \vdash^? \mathsf{senc}(x, k)$ (two opportunities for R_2)
3. $S \vdash^? x \;\wedge\; S, n \vdash^? y \;\wedge\; S, n, \mathsf{senc}(m, \mathsf{senc}(x, k)), \mathsf{senc}(y, k) \vdash^? m$
4. $S \vdash^? x \wedge S \vdash^? \langle x, x \rangle$
5. $n \vdash^? x \wedge n \vdash^? \mathsf{senc}(x, k)$

Constraint simplification proof (1)

Proposition (Validity)

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

Constraint simplification proof (1)

Proposition (Validity)

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

Proposition (Soundness)

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

Proposition (Termination)

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

- *$v(\mathcal{C})$ is the number of variables occurring in $\mathcal{C}$;*
- *$p(\mathcal{C})$ is the number of terms of the form $aenc(u, x)$ occurring on the left of constraints in $\mathcal{C}$;*
- *$s(\mathcal{C})$ is the total size of the right-hand sides of constraints in $\mathcal{C}$.*

Constraint simplification proof (2)

Left-minimality & Simplicity

A derivation Π of $T_i \vdash u$ is left-minimal if, whenever $T_j \vdash u$, Π 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 $\mathcal{C} = \bigwedge_j T_j \vdash^? u_j$ be a constraint system, $\theta \in \mathrm{Sol}(\mathcal{C})$,
and i be such that $u_j \in \mathcal{X}$ for all $j < i$.
If $T_i\theta \vdash u$ with a simple derivation starting with an axiom or a decomposition, then there is $t \in \mathrm{subterm}(T_i) \setminus \mathcal{X}$ such that $t\theta = u$.

Constraint simplification proof (3)

Lemma

Let $\mathcal{C} = \bigwedge_j T_j \vdash^? u_j$, $\sigma \in \mathsf{Sol}(\mathcal{C})$.
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* $aenc(t, x)$;
- u_i *is a non-variable subterm of* T_i.

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

Constraint simplification proof (3)

Lemma

Let $\mathcal{C} = \bigwedge_j T_j \vdash^? u_j$, $\sigma \in \mathsf{Sol}(\mathcal{C})$.
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* $aenc(t, x)$;
- u_i *is a non-variable subterm of* T_i.

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

Proposition (Completeness)

If $\mathcal{C}$ *is unsolved and* $\theta \in \mathsf{Sol}(\mathcal{C})$, *there is* $\mathcal{C} \rightsquigarrow_\sigma \mathcal{C}'$ *and* $\theta' \in \mathsf{Sol}(\mathcal{C}')$ *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