TD 6: Deducibility Constraints in the symbolic setting

Margot Catinaud margot.catinaud@lmf.cnrs.fr
Théo Vignon theo.vignon@lmf.cnrs.fr

December 20, 2024

1 Going back to TD3 : locality

Recall the definition of locality for a deduction system and the intruder deduction problem:

Definition: Locality

A deduction system $\mathcal{D}$ is local if for all T finite set of terms and term s such that $\mathsf{T} \vdash \mathsf{s}$, there is a proof tree Π of that fact such that $\mathsf{Terms}(\Pi) \subseteq \mathsf{st}(\mathsf{T} \cup \{\mathsf{s}\})$.
Where $\mathsf{Terms}(\Pi)$ is the set of terms that appear in a given proof tree Π
and $\mathsf{st}(\mathsf{E})$ is the set of sub-terms of terms in E.

Definition: Intruder deduction problem

Let $\mathcal{I}$ be an inference system. The intruder deduction problem is:

Input: a finite set of terms T and a term s

Output: whether $\mathsf{T} \vdash_{\mathcal{I}} \mathsf{s}$

Exercise 1: Dolev-Yao locality

Consider the following deduction system:

Definition: Dolev-Yao inference system $\mathcal{I}_{\mathcal{DY}}$

$$\frac{x \quad y}{\langle x \, , \, y \rangle} \qquad \frac{\langle x \, , \, y \rangle}{x} \qquad \frac{\langle x \, , \, y \rangle}{y}$$

$$\frac{x \quad y}{\mathsf{senc}(x, y)} \qquad \frac{\mathsf{senc}(x, y) \quad y}{x}$$

$$\frac{x \quad y}{\mathsf{aenc}(x, y)} \qquad \frac{\mathsf{aenc}(x, \mathsf{pk}(y)) \quad y}{x}$$

1. Show that it is local
   Hint: for that, a valid proof tree to consider is any minimal (in the number of terms (with multiplicity)) proof tree.
2. Conclude that the intruder deduction system is decidable on this inference system

Exercise 2: Blind signature and intruder deduction problem

In this exercise, we consider *blind signatures* represented by the following inference system:

Definition: Blind signatures inference system $\mathcal{I}_{\mathsf{blind}}$

$$\frac{x \quad y}{\langle x \,,\, y\rangle} \qquad \frac{\langle x \,,\, y\rangle}{x} \qquad \frac{\langle x \,,\, y\rangle}{y}$$

$$\frac{x \quad y}{\mathsf{blind}(x, y)} \qquad \frac{x \quad y}{\mathsf{sign}(x, y)}$$

$$\frac{\mathsf{sign}(\mathsf{blind}(x, y), z) \quad y}{\mathsf{sign}(x, z)} \qquad \frac{\mathsf{blind}(x, y) \quad y}{x}$$

1. Find a set of messages T and a name n such that $\mathsf{T} \vdash n$ with this inference system but $\mathsf{T} \nvdash n$ with the inference system of the previous exercise
2. Show that this inference system is not local
3. Provide an algorithm to decide the intruder deduction problem for this inference system
Hint: you can "adapt" the definition of locality with an extended notion of subterms: the set $\mathsf{st}_{\mathsf{ext}}(t)$ is the smallest set such that
   - $\mathsf{st}(t) \subseteq \mathsf{st}_{\mathsf{ext}}(t)$
   - if $\mathsf{sign}(\mathsf{blind}(x, y), z) \in \mathsf{st}_{\mathsf{ext}}$ then $\mathsf{sign}(x, z) \in \mathsf{st}_{\mathsf{ext}}(t)$

2 Deducibility constraints

Recall the definition of deducibilty constraints system:

Definition: Deducibility constraints system

A *deducibility constraint* is an expression of the form $T \vdash^?_{\mathcal{I}} u$ where T is a non-empty set of terms, u a term, $\mathcal{I}$ (often omitted) is the deduction system used.

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

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

such that

- *monotonicity:* $T_1 \subseteq T_2 \subseteq \cdots \subseteq T_n$
- *origination:* for all i, $\mathsf{fv}(T_i) \subseteq \mathsf{fv}(u_1, \ldots, u_{i-1})$

[a] a empty conjunction is equivalent to $\top$

Our goal is to try to solve a deducibility constraint system:

Definition: Solution of a constraint system

A subtitution σ is a *solution* of $\mathcal{C} = T_1 \vdash^? u_1 \wedge \cdots \wedge T_n$, a deducibility constraint system if for all $i \in [\![1; n]\!]$, there exist a proof of $T_i\sigma \vdash u_i$

Exercise 3: Form protocols to constrains system

Recall the Needham-Schroeder protocol:

$$\begin{aligned} B &\to A : \mathsf{pk}(B) \\ A &\to B : \{A, N_A\}_{\mathsf{pk}(B)} \\ B &\to A : \{N_A, N_B\}_{\mathsf{pk}(A)} \\ A &\to B : \{N_B\}_{\mathsf{pk}(B)} \end{aligned}$$

We know (from TD2) that there is an attack on N_B in this protocol, the goal of this exercise is to find it using constraints system.
To do this, we need to translate the protocol to some constraints systems. For that, we first express the protocols as a set of rules.
The idea behind those rules is to represent one possible transition of the protocol. For example, from the first two interactions of the protocol

$$\begin{aligned} B &\to A : \mathsf{pk}(B) \\ A &\to B : \{A, N_A\}_{\mathsf{pk}(B)} \end{aligned}$$

For the first interaction we write the rule:

$$\to \mathsf{pk}(B) \tag{B.1}$$

This rule represents the fact that without any input (there is nothing at the left side of the arrow), B sends its public key.
We can now write (A.1), representing the message that A sends to answer this (x here represents the fact that A is willing to talk to anyone. So on input x, it sends back $\{A, N_A\}_x$).

$$x \to \{A, N_A\}_x \tag{A.1}$$

1. Write $(B.2)$ and $(A.2)$ representing the two other messages.
   Hint: You can restrict the input using pattern matching to caracterize the inputs. For example, B only answers back when the message it gets as input as this form: $\{(\mathsf{pk}(A), y)\}_{\mathsf{pk}(B)}$

Now, we want to transform this rules into a constraint system. For that, we need an ordering on rules.

2. First, is there any restriction made by the protocol on the ordering of the rules ?
3. Respecting those restrictions, find a ordering of the rules that leads to an attack

To transform this ordering into a constraint system, we need to know the initial knowledge of the adversary T_0, and then use the ordering

$$\begin{aligned} u_1 &\to v_1 \\ &\vdots \\ u_n &\to v_n \end{aligned}$$

Leads to the constraints system $\mathcal{C}$:

$$\begin{aligned} &T_0 && \vdash^? u_1 \\ &T_0, v_1 && \vdash^? u_1 \\ & && \vdots \\ &T_0, v_1, \ldots, v_{n-1} && \vdash^? u_n \\ &T_0, v_1, \ldots, v_n && \vdash^? N_B \end{aligned}$$

Exercise 3: Form protocols to constrains system

since here, we want to show (actually break) the secrecy of N_B

4. Write down the initial knowledge of the adversary T_0
5. Write down the constraint system $\mathcal{C}$
6. Complete the subsitution $\sigma = \{x \rightarrow \mathsf{pk}(C), y \rightarrow N_A\}$ to make it a solution and prove that it is indeed a solution.

Exercise 4: Constraints solving

First, we want a class of constraint system where it is “easy” to show that they have a solution

Definition: Solved constraint system

A constraint system is said to be *solved* if it is in the form

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

where for all i, x_i is a variable.

1. Show that any solved constraint system have a solution

We are now interested in a algorithm to find a solution (if there is any) to a given deducibility constraints system.

Definition: Simplification rules for constraints system

We will consider a set of simplification rules for constraints system: [a]

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

$$\mathcal{C} \wedge T \vdash^? u \rightsquigarrow_\sigma \mathcal{C}\sigma \wedge T\sigma \vdash^? u\sigma \qquad \text{if } t \in \mathsf{st}(T), \sigma = \mathsf{mgu}(t,u), t \neq u \text{ and } t, u \text{ not variables} \qquad (R_2)$$

$$\mathcal{C} \wedge T \vdash^? u \rightsquigarrow_\sigma \mathcal{C}\sigma \wedge T\sigma \vdash^? u\sigma \qquad \text{if } t, v \in \mathsf{st}(T), \sigma = \mathsf{mgu}(t,v), t \neq v \qquad (R_3)$$

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

$$\mathcal{C} \wedge T \vdash^? f(u_1, \ldots, u_n) \rightsquigarrow \mathcal{C} \wedge \bigwedge_i T \vdash^? u_i \qquad \text{if } f \text{ is a constructor symbol} \qquad (R_f)$$

[a] In this exercise session the constructor symbols are senc, aenc, (_, _), blind and sign

2. Show that those simplifications rules are valid, meaning that for $\mathcal{C}$ a constraint system, if $\mathcal{C} \rightsquigarrow_\sigma \mathcal{C}'$ then $\mathcal{C}'$ is also a constraint system.
3. Show that the constraint system simplification terminates, meaning that there is not infinite sequence $\mathcal{C}_0 \rightsquigarrow \mathcal{C}_1 \rightsquigarrow \ldots$.

Exercise 4: Constraints solving

Hint: Use a lexicographical order on (v, s) where v is the number of variables and s is the size of the constraints system, for some good notion of size to be defined.

4. Show that constrain system simplification is sound, meaning that if $\mathcal{C} \rightsquigarrow_\sigma \mathcal{C}'$ and θ is a solution of $\mathcal{C}'$ then $\sigma\theta$ is a solution of $\mathcal{C}$

Remark: The constraint simplification rules is also complete : if θ is a solution of $\mathcal{C}$, then there is exists a solved constraints system $\mathcal{C}'$ and substitutions σ, θ', such that $\theta = \sigma\theta'$, $\mathcal{C} \rightsquigarrow^*_\sigma \mathcal{C}'$ and θ' is a solution of $\mathcal{C}'$.

Exercise 5: Examples of simplification

On all those examples, try to apply the simplification rules as much as you can:

1. $\mathsf{senc}(n, k) \vdash^? \mathsf{senc}(x, k)$
2. $\mathsf{senc}(\mathsf{senc}(t_1, k), k) \vdash^? \mathsf{senc}(x, k)$
3. $T \vdash^? x \wedge T, n \vdash^? y \wedge T, n, \mathsf{senc}(m, \mathsf{senc}(x, k)), \mathsf{senc}(y, k) \vdash^? m$
4. $T \vdash^? x \wedge T \vdash^? (x, x)$
5. $n \vdash^? x \wedge n \vdash^? \mathsf{senc}(x, k)$

Exercise 6: Needham-Schroeder simplification

Apply the simplification rules on the constraint system given for the Needham-Schroeder protocol at the previous exercise to get back σ

Exercise 7: Wide-mouthed frog

Recall the example of the *wide-mouthed-frog* protocol:

$$\begin{aligned} A \rightarrow S &: A, \{B, s, m_1\}_{\mathsf{k}_{AS}} \\ S \rightarrow B &: \{A, s, m_2\}_{\mathsf{k}_{BS}} \end{aligned}$$

Here, we are interested in the case where A agrees to talk to both B and C (a dishonest agent).

1. Write the protocol as rules with input and output
2. Show that the associated constraints system does not have a solution, you can use the simplification rules to do so.

Recall the broken example of the *wide-mouthed-frog* protocol:

$$\begin{aligned} A \rightarrow S &: A, B, \{s\}_{\mathsf{k}_{AS}} \\ S \rightarrow B &: A, \{s\}_{\mathsf{k}_{BS}} \end{aligned}$$

3. Write down the protocol as rules with input and output
4. Show that the associated constraints system does have a solution. Exhibit a solution using the constraint solving simplification.