MPRI 2.30, part I

Formal Proofs of Cryptographic Protocols

November 27, 2019

You have three hours. All documents are allowed, but electronic devices are forbidden.

1 Revisiting trace and may testing equivalences

In this exercise, we consider a fragment of the applied pi-calculus of the lecture notes, given by the following grammar:

$$P, P' ::= 0 \mid \mathsf{in}(c, x).P \mid \mathsf{out}(c, u).P \mid \mathsf{let}\ x = t\ \mathsf{in}\ P\ \mathsf{else}\ P'$$

In other words, we do not consider name creation, replication, and even parallel composition. As before, u denotes a constructor term, while t may feature destructors.

Semantics The definition of internal reduction is unchanged, except for the fact that several cases become useless. The labelled transition system has the standard rules for input, output, let, plus the following new rule:

$$(P, \Phi) \xrightarrow{\mathsf{store}(w,t)} (P, \Phi \uplus \{w \mapsto u\}) \text{ if } w \in \mathcal{W} \setminus \mathsf{dom}(\Phi), t \in \mathcal{T}(\mathsf{dom}(\Phi) \cup \mathcal{N} \setminus \mathsf{bn}(\Phi)) \text{ and } t\Phi \Downarrow u$$

Since a process cannot create new private names, the bound names of a frame (i.e. $\mathsf{bn}(\Phi)$) is never modified. We will thus assume in the rest of this exercise that all frames have the same bound names. This amounts to have a static distinction between public and private names, where only public names may appear in recipes: we thus have a set of public names $\mathcal{N}_{\mathsf{pub}}$ such that, for all Φ, $\mathcal{N}_{\mathsf{pub}} = \mathcal{N} \setminus \mathsf{bn}(\Phi)$. A recipe is simply a term of $\mathcal{T}(\mathcal{W} \cup \mathcal{N}_{\mathsf{pub}})$.

The only transitions labelled τ correspond to the evaluation of let constructs. The label $\mathsf{store}(w, t)$ is considered observable.

Trace equivalence We define as before $A \overset{\mathsf{tr}}{\Rightarrow} B$ when tr contains no τ action and $A \xrightarrow{\mathsf{tr}'} B$ for some tr' obtained from tr by inserting τ actions. We finally define trace equivalence as before: $A \approx B$ iff $\mathsf{Tr}'(A) = \mathsf{Tr}'(B)$ where

$$\mathsf{Tr}'(P, \Phi) \overset{def}{=} \{(\mathsf{tr}, \Phi'') : (P, \Phi) \overset{\mathsf{tr}}{\Rightarrow} (P', \Phi') \text{ and } \Phi' \sim \Phi''\}.$$

May testing We consider a simplified notion of may testing, relying on a restricted syntax for tests[1]. We first define a grammar for test contexts C (with a hole noted $[]$) and terminal tests D (where _ stands for a dummy variable that cannot be used elsewhere):

$$\begin{array}{rcl} C & ::= & \mathsf{in}(c, w).C \mid \mathsf{out}(c, w).C \mid \mathsf{let}\ w = t\ \mathsf{in}\ C\ \mathsf{else}\ 0 \mid [] \\ D & ::= & \mathsf{out}(\mathbb{T}, w) \mid \mathsf{let}\ \_ = t\ \mathsf{in}\ D\ \mathsf{else}\ 0 \mid \mathsf{let}\ \_ = t\ \mathsf{in}\ 0\ \mathsf{else}\ D \end{array}$$

As before, $\mathbb{T}$ is the special channel used to signal success. We then define tests as those processes that can be written $C[D]$ for some test context C and terminal test D, that do not feature private names, but may feature free variables in $\mathcal{W}$. Note that tests only use variables in $\mathcal{W}$, and that

[1] This is actually without loss of generality, but we won't show it here.

outputs in tests are restricted to variables. We may also assume that a test never features two bindings for the same variable w.

We say that a process Q succeeds when $Q \rightsquigarrow^* \mathsf{out}(\mathbb{T}, u)$ for some u, and we define:

$$\mathsf{T}(P, \Phi) = \{T : T \text{ is a test with } \mathsf{fv}(T) \subseteq \mathsf{dom}(\Phi) \text{ and } (T\Phi \mid P) \text{ succeeds}\}$$

Finally, A and B are may testing equivalent when $\mathsf{T}(A) = \mathsf{T}(B)$.

Question 1. We say that a frame Φ satifies a terminal test D, noted $\Phi \models D$, when $D\Phi$ succeeds. Show that, for all frames Φ and Ψ such that $\Phi \sim \Psi$, we have $\Phi \models D$ iff $\Psi \models D$.

Question 2. Define a translation $\mathsf{tr}(\_)$ from test contexts to traces such that, for all test context C and terminal test D, we have $C[D] \in \mathsf{T}(A)$ iff there exists A' such that $A \overset{\mathsf{tr}(C)}{\Rightarrow} A'$ and $\Phi(A') \models D$. Justify the two directions of this equivalence.

Question 3. Conclude that trace equivalence implies may testing equivalence[2].

For the rest of the exercise, we assume that we have a destructor implementing equality checking (i.e. $\mathsf{eq}(x, x) \to x$) so that $\Phi \sim \Psi$ is equivalent to the following condition alone:

$$\forall R \in \mathcal{T}(\mathcal{W} \cup \mathcal{N}_{\mathsf{pub}}), R\Phi \Downarrow \text{ iff } R\Psi \Downarrow$$

Question 4. Let Φ be a frame and $(\Psi_i)_{i \in [1;n]}$ be a finite collection of frames such that, for all $i \in [1; n]$, $\Phi \not\sim \Psi_i$. Show that there exists D such that $\Phi \models D$ but, for all $i \in [1; n]$, $\Psi_i \not\models D$.

Question 5. A configuration A is image finite if, for all observable traces tr, the set $\{\Phi(B) : A \overset{\mathsf{tr}}{\Rightarrow} B\}$ is finite up to static equivalence. Show that, for all image finite configurations A and B such that $\mathsf{T}(A) = \mathsf{T}(B)$, one has $A \approx B$. *Hint: proceed by contradiction.*

2 Bana-Comon logic: EUF and LAK

Consider a keyed hash function $H(\_, \_)$. We say that the it satisfies existential unforgeability (EUF) when, for any attacker $\mathcal{A}$, the probability $\Pr[\mathsf{EUF}(\mathcal{A}) = 1]$ is negligible in η, where the experiment EUF is defined as follows:

$$\begin{array}{lll} \mathsf{EUF}(\mathcal{A}) & \overset{def}{=} & k \overset{R}{\leftarrow} \{0,1\}^\eta \\ & & \mathcal{Q} \leftarrow \emptyset \\ & & m, s \leftarrow \mathcal{A}^{\mathcal{O}} \\ & & \mathsf{if}\ m \notin \mathcal{Q} \wedge s = H(m, k)\ \mathsf{then\ return}\ 1 \end{array} \qquad \begin{array}{lll} \mathcal{O}(x) & \overset{def}{=} & \mathcal{Q} \leftarrow \mathcal{Q} \cup \{x\} \\ & & \mathsf{return}\ H(x, k) \end{array}$$

Here, $\mathcal{O}$ is an oracle allowing the attacker to compute hashes using the (private) key k that is randomly sampled at the beginning of the experiment. The experiment succeeds if the attacker manages to compute the hash s of a message m without having called $\mathcal{O}(m)$.

We consider the language of the Bana-Comon logic. We write $t \doteq t'$ for $\mathsf{EQ}(t, t')$. We write $t \mathbin{\dot\wedge} t'$ for $\big(\mathsf{if}\ t\ \mathsf{then}\ (\mathsf{if}\ t'\ \mathsf{then\ true\ else\ false})\ \mathsf{else\ false}\big)$, and similarly for $t \mathbin{\dot\vee} t'$ and $\dot\neg\, t$.

Question 1. Show that the following rule is unsound:

$$\frac{t[b] \sim t' \quad a \sim b}{t[a] \sim t'}$$

[2]In the lecture notes, we have shown that this is not the case if non-deterministic computations are allowed: this problem disappears here, not due to abusive simplifications but mainly due to the store actions.

Question 2. Let h be a binary function symbol and k a name, which will be used as a key for h. Let m and s be two terms in which k only appears as second argument of h. Let $E = \{m' : h(m', k) \text{ is a subterm of } m \text{ or } s\}$. Consider the following inference rule[3]:

$$\frac{}{\mathsf{if}\ \dot{\bigvee}_{m' \in E}\, m \doteq m'\ \mathsf{then\ false\ else}\ s \doteq h(m,k) \sim \mathsf{false}}$$

Show that the conclusion formula is satisfied in all computational interpretations where the interpretation of h satisfies EUF. In other words, the rule is sound wrt. this class of models.

In order to derive equivalences, we will consider the previous rule, together with the following ones, which are also sound:

$$\frac{}{t \sim t} \qquad \frac{t[b] \sim t' \quad a \doteq b \sim \mathsf{true}}{t[a] \sim t'} \qquad \frac{\mathsf{if}\ b\ \mathsf{then}\ t'\ \mathsf{else}\ e \sim \mathsf{true} \quad \mathsf{if}\ t'\ \mathsf{then}\ t\ \mathsf{else\ true} \sim \mathsf{true}}{\mathsf{if}\ b\ \mathsf{then}\ t\ \mathsf{else}\ e \sim \mathsf{true}}$$

$$\frac{t \sim t'}{\dot{\neg}\, t \sim \dot{\neg}\, t'} \qquad \frac{}{t \doteq t \sim \mathsf{true}} \qquad \frac{}{\pi_1(\langle t, t' \rangle) \doteq t \sim \mathsf{true}} \qquad \frac{}{\pi_2(\langle t, t' \rangle) \doteq t' \sim \mathsf{true}} \qquad \frac{u \equiv v}{u\theta \doteq v\theta \sim \mathsf{true}}$$

In the last rule $u \equiv v$ means that u and v are logically equivalent expressions built from (boolean) variables and conditionals $\mathsf{if}\ \_\ \mathsf{then}\ \_\ \mathsf{else}\ \_$. For example, that rule allows to derive all statements of the form $\big(\mathsf{if}\ b\ \mathsf{then}\ c\ \mathsf{else}\ d\big) \doteq \big(\mathsf{if}\ \mathsf{not}(b)\ \mathsf{then}\ d\ \mathsf{else}\ \mathsf{not}(\mathsf{not}(c))\big) \sim \mathsf{true}$.

Finally, equivalences $t \sim t'$ and equalities $t \doteq t'$ may be considered implicitly modulo symmetry.

Consider the following frames, resulting from an execution of the LAK protocol:

$$\begin{aligned} \Phi_1 &\stackrel{def}{=} nR \\ \Phi_2 &\stackrel{def}{=} \Phi_1, \langle n_T, h(\langle g_1(\Phi_1), n_T \rangle, k) \rangle \end{aligned}$$

We then consider the following boolean terms:

$$\begin{aligned} t_a &\stackrel{def}{=} \pi_2(g_2(\Phi_2)) \doteq h(\langle n_R, \pi_1(g_2(\Phi_2)) \rangle, k) \\ t_h &\stackrel{def}{=} g_1(\Phi_1) \doteq n_R \mathbin{\dot{\wedge}} \pi_1(g_2(\Phi_2)) \doteq n_T \\ t_c &\stackrel{def}{=} \langle g_1(\Phi_1), n_T \rangle \doteq \langle n_R, \pi_1(g_2(\Phi_2)) \rangle \end{aligned}$$

The first term corresponds to the *a*cceptance test that the reader would evaluate when receiving $g_2(\Phi_2)$ after having sent n_R. The second term expresses that the interaction is *h*onest enough, and more precisely that the tag and reader agree on their respective nonces.

Question 3. Derive $\mathsf{if}\ t_a\ \mathsf{then}\ t_c\ \mathsf{else\ true} \sim \mathsf{true}$ using the above rules.

Question 4. Conclude that there exists a derivation of the following equivalence, which intuitively expresses that if the reader accepts an interaction, then the interaction is honest:

$$\mathsf{if}\ t_a\ \mathsf{then}\ t_h\ \mathsf{else\ true} \sim \mathsf{true}$$

Question 5. We modify our frames to reason about an execution involving more reader messages, with a subterm m that will be defined below:

$$\begin{aligned} \Phi_1 &\stackrel{def}{=} nR, nR' \\ \Phi_2 &\stackrel{def}{=} \Phi_1, \langle n_T, h(\langle g_1(\Phi_1), n_T \rangle, k) \rangle \\ \Phi_3 &\stackrel{def}{=} \Phi_2, h(\langle m, n_R \rangle, k) \end{aligned}$$

[3] If E is empty, the disjunction $\dot{\bigvee}_{m' \in E}\, m \doteq m'$ is simply false.

(a) We want Φ_3 to model the situation where the first reader has accepted its input, and responded with its second output. Give the term m corresponding to that situation, relying on $g_2(\Phi_2)$ to model the reader's input.

We define t_a^1 as t_a above, except for the modification of Φ_1 which now features two nonces. We similarly adapt t_h into t_h^1. It is easy to see that we still have $\mathsf{if}\ t_a^1\ \mathsf{then}\ t_h^1\ \mathsf{else}\ \mathsf{true} \sim \mathsf{true}$.

(b) Now the second reader may receive a message $g_3(\Phi_3)$. Give the boolean term t_a^2 corresponding to the acceptance condition of the second reader.

(c) Let $t_h^2 \stackrel{def}{=} g_1(\Phi_1) \doteq n'_R \mathbin{\dot\wedge} \pi_1(g_3(\Phi_3)) \doteq n_T$. Does $\mathsf{if}\ t_a^1 \mathbin{\dot\wedge} t_a^2\ \mathsf{then}\ t_h^2\ \mathsf{else}\ \mathsf{true} \sim \mathsf{true}$ hold? No formal derivation is required, but an argument using the EUF assumption should be provided.

3 Constraint solving and resolution for a naive protocol

Consider the following protocol, where a and b are respective secrets of agents A and B, keys $\mathsf{pk}(a)$ and $\mathsf{pk}(b)$ are public, and n is a nonce generated by A:

$$\begin{aligned} A \to B &: \langle \mathsf{pk}(a), \mathsf{aenc}(n, \mathsf{pk}(b)) \rangle \\ B \to A &: \langle \mathsf{pk}(b), \mathsf{aenc}(n, \mathsf{pk}(a)) \rangle \end{aligned}$$

The naive idea is that A sends a first message containing a component that identifies the sender, and one that transmits n securely to the intended recipient B. Upon receipt of a message of the form $\langle x, \mathsf{aenc}(y, \mathsf{pk}(b)) \rangle$, B obtains n and sends back the last message to x as an acknowledgment. Obviously, there is an attack: an outsider can learn n.

Question 1. We want to model this protocol using the applied pi-calculus. We model pairs and asymmetric encryption using destructors. Give two processes $A(sk_a, pk_b)$ and $B(sk_b)$ modelling the two roles of the protocol, where A only uses channel c_A and B only uses c_B. Process A should have n as a free name.

Question 2. Let $P \stackrel{def}{=} \big(A(a, \mathsf{pk}(b)) \,|\, B(b)\big)$ and $\Phi \stackrel{def}{=} (n, a, b).\{w_a \mapsto \mathsf{pk}(a), w_b \mapsto \mathsf{pk}(b)\}$. Give the symbolic configuration corresponding to the execution of each of the following traces, starting with the configuration $(P, \Phi, \emptyset)$:

$$\mathsf{tr}_1 \stackrel{def}{=} \mathsf{out}(c_A, w).\mathsf{in}(c_B, x) \qquad \mathsf{tr}_2 \stackrel{def}{=} \mathsf{tr}_1.\mathsf{out}(c_B, w')$$

You should make the most general choices (e.g. in the symbolic evaluation steps) so that the resulting symbolic configuration accounts for all concrete configurations, in the sense of the completeness result of the symbolic semantics.

Question 3. Let $(\_, \Psi, \mathcal{C})$ be the last configuration obtained above. Show that the constraint system $\mathcal{C} \wedge \Psi \vdash^? n$ can be simplified into a solved constraint system. Conclude that n is not secret.

Question 4. Give the clauses that Proverif would generate for P and the theory of pairs and asymmetric encryption.

Question 5. We consider resolution with selection, with the selection strategy where any hypothesis that is not of the form $\mathsf{att}(x)$ is selected. Show how this rule can be used to derive, from the previous clauses, a solved clause with conclusion $\mathsf{att}(n)$. The hypotheses of a clause may be treated as a set of atoms, e.g. multiple hypotheses can be merged.