MPRI Exam 2-30 (part 1)
Cryptographic protocols: formal and computational proofs

Duration: 3h. All documents are allowed. Electronic devices are forbidden.

December, 3rd, 2014

Exercise 1 : Constraint solving

We consider the BAN-Yahalom protocol as described informally below. The purpose of this protocol is to establish a fresh session key K_{ab} between two participants A and B. This is done through a server S who shares a long-term symmetric key with each participant. The key K_{as} (resp. K_{bs}) is a symmetric key shared between A (resp. B) and S.

$$\begin{array}{ll}
1.\ A \to B: & A,\ N_a \\
2.\ B \to S: & B,\ N_b,\ \{A, N_a\}_{K_{bs}} \\
3.\ S \to A: & N_b,\ \{B, K_{ab}, N_a\}_{K_{as}},\ \{A, K_{ab}, N_b\}_{K_{bs}} \\
4.\ A \to B: & \{A, K_{ab}, N_b\}_{Kbs},\ \{N_b\}_{K_{ab}}
\end{array}$$

We consider the constraint system $\mathcal{C}$ given below with $T_0 = \{a,\ b, n_i, k_i\}$ and the classical inference system $\mathcal{I}_{\mathsf{DY}}$ to deal with symmetric encryption and pair.

$$\begin{array}{rcl}
T_1 \stackrel{\mathsf{def}}{=} T_0,\ \langle a, n_a\rangle & \stackrel{?}{\vdash} & \langle a, x\rangle \\
T_2 \stackrel{\mathsf{def}}{=} T_1, \langle b, \langle n_b, \mathsf{senc}(\langle a, x\rangle, k_{bs})\rangle\rangle & \stackrel{?}{\vdash} & \langle a, x'\rangle \\
T_3 \stackrel{\mathsf{def}}{=} T_2,\ \langle b, \langle n_b', \mathsf{senc}(\langle a, x'\rangle, k_{bs})\rangle\rangle & \stackrel{?}{\vdash} & \langle \mathsf{senc}(\langle a, \langle y, n_b\rangle\rangle, k_{bs}),\ \mathsf{senc}(n_b, y)\rangle \\
T_3 & \stackrel{?}{\vdash} & y
\end{array}$$

1. Explain the scenario encoded by the constraint system $\mathcal{C}$. Who are the agents involved in this scenario? What are the roles played by each agent? How many instances of each role are they playing? What is the security property under study?
2. Check that the substitution $\sigma = \{x \mapsto n_i;\ x' \mapsto \langle k_i, n_b\rangle;\ y \mapsto k_i\}$ is a solution of $\mathcal{C}$, *i.e.* give proof trees witnessing that σ is a solution of the constraint system $\mathcal{C}$. Explain the underlying attack on the protocol using the informal Alice & Bob notation.
3. Solve the deducibility constraint system $\mathcal{C}$, using the simplification rules of the lectures, and give all the solutions of the constraint system $\mathcal{C}$. *You may notice that the simplification rules can always be applied to the first unsolved deducibility constraint, according to the completeness proof. Moreover, when using the rules* R_2 *and* R_3*, you may assume that* t, u, t_1, t_2 *are neither variables nor of the form* $\langle v_1, v_2\rangle$*. Completeness is still true under these hypotheses. This avoids unnecessary branching.*
4. We propose to modify the protocol by adding *tags*. We consider public constants $\mathsf{c}_1, \mathsf{c}_2, \ldots$ and we add such a constant in each ciphertext. For instance, messages 2 and 3 will be modified as follows :

$$\begin{array}{ll}
2.\ B \to S: & B,\ N_b,\ \{\mathsf{c}_1, A, N_a\}_{K_{bs}} \\
3.\ S \to A: & N_b,\ \{\mathsf{c}_2, B, K_{ab}, N_a\}_{K_{as}},\ \{\mathsf{c}_3, A, K_{ab}, N_b\}_{K_{bs}}
\end{array}$$

Of course, when an agent receives such a message, he will check (after decrypting the ciphertext) that the plaintext starts with the expected constant. Modify the constraint system $\mathcal{C}$ to reflect the changes done on the protocol. Let $\mathcal{C}_{\mathsf{tag}}$ the resulting constraint system. Is $\mathcal{C}_{\mathsf{tag}}$ satisfiable ? Could you comment on the usefulness of such a tagging mechanism from a security point of view ?

Exercise 2 : Blind signatures

We want to study the intruder deduction problem for the inference system $\mathcal{I}_{\mathsf{sign}}$ given below :

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

We consider two binary function symbols sign and blind. Intuitively, the term $\mathsf{sign}(m,k)$ represents the signature of the message m with the key k, and the term $\mathsf{blind}(m,r)$ represents the message m *hidden* with the random factor r. The blind signature primitive has the following property : given the signature of a blind message, *i.e.* $\mathsf{sign}(\mathsf{blind}(m,r),k)$, and the blinding factor r, it is possible to compute the signature of the message m, *i.e.* $\mathsf{sign}(m,k)$. This additional ability is also given to the intruder. This is the purpose of the last inference rule of the system $\mathcal{I}_{\mathsf{sign}}$.

1. Show that the inference system $\mathcal{I}_{\mathsf{sign}}$ is not *local* w.r.t. the notion of syntactic subterm that we have seen during the lectures.

   We consider a notion of extended subterms defined as follows : $st_{ext}(t)$ is the smallest set such that $st(t) \subseteq st_{ext}(t)$ and

   $$\text{if } \mathsf{sign}(\mathsf{blind}(u_1,u_2),u_3) \in st_{ext}(t) \text{ then } \mathsf{sign}(u_1,u_3) \in st_{ext}(t).$$

   This notion is extended to sets of terms as follows : $st_{ext}(T) = \bigcup_{t \in T} st_{ext}(t)$.
2. Let $T_0 = \{\mathsf{blind}(m,r),\ k,\ r\}$ and $u = \mathsf{sign}(m,k)$. Give two different proof trees Π_1 and Π_2 of $T \vdash u$ in $\mathcal{I}_{\mathsf{sign}}$ having minimal size (size = number of nodes).
3. Show that the inference system $\mathcal{I}_{\mathsf{sign}}$ is local w.r.t. the notion of extended subterms.
4. Show that the intruder deduction problem is decidable
   Input : a finite set T of terms, and a term u ;
   Output : Is u deducible from T in $\mathcal{I}_{\mathsf{sign}}$?

   Justify the termination of your algorithm.
5. We consider the simplification rules seen during the lectures where the underlying deduction relation used in R_1 is $\mathcal{I}_{\mathsf{sign}}$, and the notion of subterm used in R_2 and R_3 is the notion of extended subterm mentioned above. We consider the following theorem :

   Termination : There is no infinite chain $\mathcal{C} \rightsquigarrow_{\sigma_1} \mathcal{C}_1 \ldots \rightsquigarrow_{\sigma_n} \mathcal{C}_n$.

   Correctness : If $\mathcal{C} \rightsquigarrow^*_{\sigma} \mathcal{C}'$ for some constraint system $\mathcal{C}'$ and some substitution σ and if θ is a solution of $\mathcal{C}'$ then $\sigma\theta$ is a solution of $\mathcal{C}$.

   Completeness : If θ is a solution of $\mathcal{C}$, then there exist a solved constraint system $\mathcal{C}'$ and substitutions σ, θ' such that $\theta = \sigma\theta'$, $\mathcal{C} \rightsquigarrow^*_{\sigma} \mathcal{C}'$ and θ' is a solution of $\mathcal{C}'$.

   Say whether each of these statements is true or not. You will provide a short explanation to justify a positive answer, and a counter-example to illustrate a negative answer.

Exercice 3 : Encrypted Key Exchange protocol

The EKE protocol is designed to solve the problem of authenticated key exchange while being resistant against dictionay attacks. EKE is a password-only protocol : the password pw is the only shared data between the two participants A and B.

$$\begin{array}{ll} 1.\ A \to B: & \{pkey\}_{pw} \\ 2.\ B \to A: & \{\{R\}_{pkey}\}_{pw} \\ 3.\ A \to B: & \{N_a\}_R \\ 4.\ B \to A: & \{N_a, N_b\}_R \\ 5.\ A \to B: & \{N_b\}_R \end{array}$$

First, A generated a fresh private/public key pair, and then sends the public key encrypted with the password pw shared with B. Then B extracts the public key, generates a fresh session key R, encrypts it with the public key, and encrypts the result with the password. B sends this message to A. Then, nonces N_a and N_b are exchanged to perform the "hand-shaking" necessary to defend again replay attacks.

1. Give a signature $\mathcal{F}$ and an equational theory E suitable to model this protocol in a reasonnable way. We will assume that the operations symmetric encryption and decryption are commutative, *i.e.* $\mathsf{senc}(\mathsf{sdec}(x, y), y) = x$ is one of the equations in E. In the following, all the function symbols will be assumed to be public. So, your model is supposed to be reasonnable in this setting.
2. Write the processes $P_A(pw)$ and $P_B(pw)$ to model one session of the role A and one session of the role B.

   In the following, we consider the frame $\phi_0 = \mathsf{new}\, pw.\mathsf{new}\, \tilde{n}.\sigma_0$ obtained after a normal execution of one session this protocol. In other words, the attacker does not try to intercept, modify, or inject some messages during this execution.
3. Write the substitution σ_0, and the names $\tilde{n}$ that represent such an execution. We are interested in the following static equivalence (modulo the theory E you defined above) :

   $$\mathsf{new}\, \tilde{n}.\sigma_0 \overset{?}{\sim} \mathsf{new}\, pw.\mathsf{new}\, \tilde{n}.\sigma_0$$

   Does this static equivalence hold or not ? Justify your answer.

   Hint : You could rely on the algorithm seen during the lectures.
4. Using the result obtained at the previous question, deduce whether ϕ_0 is resistant to dictionary attack against pw or not ? Justify your answer.
5. Now, we assume that the asymmetric cryptosystem that is used to implement this protocol is *which-key revealing*, allowing an attacker to deduce if two ciphertexts were encrypted under the same key. Propose a signature $\mathcal{F}^+$ and a set of equations E^+ to reflect this new attacker model. Is ϕ_0 resistant to dictionary attacks against pw (considering the signature $\mathcal{F}^+$ and the equational theory E^+) ?

Exercise 1 : Constraint solving

1. Actually a is involved to send the first message and then only b is involved in this scenario (in 2 different sessions). The agent b plays two instances of the role B (and only the two first actions for the second instance). This role is made up of one input (message 1) followed by one output (message 2), and then the reception of a last message (message 4). If the message has the expected form, then the agent accepts the key he received during this exchange. This constraint system encodes the secrecy of the key K_{ab} as received by the agent b who plays the role B.

2. We apply the substitution σ on $\mathcal{C}$ and we check that each rhs is indeed deducible from the lhs.

$$\Pi^2_{n_b} = \begin{cases} \dfrac{\dfrac{\langle b, \langle n_b, \mathsf{senc}(\langle a, k_i\rangle, k_{bs})\rangle\rangle}{\langle n_b, \mathsf{senc}(\langle a, k_i\rangle, k_{bs})\rangle}}{n_b} \end{cases}$$

We have the following prooftrees :

$$\dfrac{a \quad n_i}{\langle a, n_i\rangle} \qquad \dfrac{a \qquad \dfrac{k_i \quad \Pi^2_{n_b}}{\langle k_i, n_b\rangle}}{\langle a, \langle k_i, n_b\rangle\rangle} \qquad \dfrac{\dfrac{\dfrac{\langle b, \langle n'_b, \mathsf{senc}(\langle a, \langle k_i, n_b\rangle\rangle, k_{bs})\rangle\rangle}{\langle n'_b, \mathsf{senc}(\langle a, \langle k_i, n_b\rangle\rangle, k_{bs})\rangle}}{\mathsf{senc}(\langle a, \langle k_i, n_b\rangle\rangle, k_{bs}),} \qquad \dfrac{\Pi^2_{n_b} \quad k_i}{\mathsf{senc}(n_b, k_i)}}{\langle \mathsf{senc}(\langle a, \langle k_i, n_b\rangle\rangle, k_{bs}), \mathsf{senc}(n_b, k_i)\rangle} \qquad k_i$$

The attack can be informally described as follows :

$$\begin{array}{ll} 1.i \ \ I(A) \to B : & A, N_i \\ 2.i \ \ B \to (S) : & B, N_b, \{A, N_i\}_{K_{bs}} \\[4pt] 1.ii \ \ I(A) \to B : & A, \langle K_i, N_b\rangle \\ 2.ii \ \ B \to (S) : & B, N'_b, \{A, \langle K_i, N_b\rangle\}_{K_{bs}} \\[4pt] 4.i \ \ I(A) \to B : & \{A, \langle K_i, N_b\rangle\}_{K_{bs}},\ \{N_b\}_{K_i} \end{array}$$

3. Starting with $\mathcal{C}$, and following the strategy suggesting in the question, we work on the first constraint and apply $\mathsf{R_f}$ followed by $\mathsf{R_1}$. This leads to the constraint system $\mathcal{C}_1$:

$$\mathcal{C}_1 = \begin{cases} T_1 \overset{?}{\vdash} x \\ T_2 \overset{?}{\vdash} \langle a, x'\rangle \\ T_3 \overset{?}{\vdash} \langle \mathsf{senc}(\langle a, \langle y, n_b\rangle\rangle, k_{bs}), \mathsf{senc}(n_b, y)\rangle \\ T_3 \overset{?}{\vdash} y \end{cases}$$

Then, since the first constraint is now solved, we work on the second one and we apply again $\mathsf{R_f}$ followed by $\mathsf{R_1}$. This leads us to the system $\mathcal{C}_2$:

$$\mathcal{C}_2 = \begin{cases} T_1 \overset{?}{\vdash} x \\ T_2 \overset{?}{\vdash} x' \\ T_3 \overset{?}{\vdash} \langle \mathsf{senc}(\langle a, \langle y, n_b\rangle\rangle, k_{bs}), \mathsf{senc}(n_b, y)\rangle \\ T_3 \overset{?}{\vdash} y \end{cases}$$

Then, we apply $\mathsf{R_f}$ on the third constraint and we obtain $\mathcal{C}_3$:

$$\mathcal{C}_3 = \begin{cases} T_1 & \overset{?}{\vdash} & x \\ T_2 & \overset{?}{\vdash} & x' \\ T_3 & \overset{?}{\vdash} & \mathsf{senc}(n_b, y) \\ T_3 & \overset{?}{\vdash} & \mathsf{senc}(\langle a, \langle y, n_b\rangle\rangle, k_{bs}) \\ T_3 & \overset{?}{\vdash} & y \end{cases}$$

Again, the only option is to apply $\mathsf{R_f}$ on the third constraint and after applying $\mathsf{R_1}$ to get rid of $T_3 \overset{?}{\vdash} n_b$, we obtain $\mathcal{C}_4$:

$$\mathcal{C}_4 = \begin{cases} T_1 & \overset{?}{\vdash} & x \\ T_2 & \overset{?}{\vdash} & x' \\ T_3 & \overset{?}{\vdash} & y \\ T_3 & \overset{?}{\vdash} & \mathsf{senc}(\langle a, \langle y, n_b\rangle\rangle, k_{bs}) \end{cases}$$

Then, on the fourth constraint, there are four possible options. We may apply :
(a) $\mathsf{R_2}$, and we obtain $\mathcal{C}_{41}$ (where $\sigma = \{x' \mapsto \langle y, n_b\rangle\}$) ;
(b) $\mathsf{R_f}$, and we obtain $\mathcal{C}_{42}$;
(c) $\mathsf{R_2}$, and we obtain $\mathcal{C}_{43}$ (where $\sigma = \{x \mapsto \langle y, n_b\rangle\}$) ;
(d) $\mathsf{R_3}$, and we obtain $\mathcal{C}_{44}$ (where $\sigma = \{x' \mapsto x\}$)
We only develop the two first cases. Actually, after some steps, we $\mathcal{C}_{43}$ and $\mathcal{C}_{44}$ are reduced to $\perp$ (no solution).

$$\mathcal{C}_{41} = \begin{cases} T_1 & \overset{?}{\vdash} & x \\ T_2 & \overset{?}{\vdash} & \langle y, n_b\rangle \\ T_3\sigma & \overset{?}{\vdash} & y \\ T_3\sigma & \overset{?}{\vdash} & \mathsf{senc}(\langle a, \langle y, n_b\rangle\rangle, k_{bs}) \end{cases} \qquad \mathcal{C}_{42} = \begin{cases} T_1 & \overset{?}{\vdash} & x \\ T_2 & \overset{?}{\vdash} & x' \\ T_3 & \overset{?}{\vdash} & y \\ T_3 & \overset{?}{\vdash} & k_{bs} \\ T_3 & \overset{?}{\vdash} & \langle a, \langle y, n_b\rangle\rangle \end{cases}$$

Since, no rule can be applied on the first unsolved constraint of $\mathcal{C}_{42}$, we deduce that $\mathcal{C}_{42}$ has no solution. Then, we apply $\mathsf{R_f}$ on the second constraint of $\mathcal{C}_{41}$ followed by $\mathsf{R_1}$ on the resulting constraint. We get $\mathcal{C}_5$:

$$\mathcal{C}_5 = \begin{cases} T_1 & \overset{?}{\vdash} & x \\ T_2 & \overset{?}{\vdash} & y \\ T_3\sigma & \overset{?}{\vdash} & y \\ T_3\sigma & \overset{?}{\vdash} & \mathsf{senc}(\langle a, \langle y, n_b\rangle\rangle, k_{bs}) \end{cases}$$

Then, we may apply $\mathsf{R_1}$ on the two last constraints, and we obtain at the end

$$T_1 \overset{?}{\vdash} x \ \wedge \ T_2 \overset{?}{\vdash} y$$

Thus, any substitution such that x is instantiated by a term t_x deducible from T_1, y is instantiated by a term t_y deducible from $T_2\{x \mapsto t_x\}$, and x' is mapped to $\langle t_y, n_b\rangle$ is a solution. These are the only solution of such a constaint system $\mathcal{C}$.

4. The constraint system $\mathcal{C}_{\mathsf{tag}}$ is as follows where $T_0^+ = T_0 \cup \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \mathsf{c}_4\}$.

$$\mathcal{C}_{\mathsf{tag}} = \begin{cases} T_1^+ \stackrel{\mathsf{def}}{=} T_0^+,\ \langle a, n_a\rangle & \stackrel{?}{\vdash} & \langle a, x\rangle \\ T_2^+ \stackrel{\mathsf{def}}{=} T_1^+, \langle b, \langle n_b, \mathsf{senc}(\langle \mathsf{c}_1, \langle a, x\rangle\rangle, k_{bs})\rangle\rangle & \stackrel{?}{\vdash} & \langle a, x'\rangle \\ T_3^+ \stackrel{\mathsf{def}}{=} T_2^+,\ \langle b, \langle n_b', \mathsf{senc}(\langle \mathsf{c}_1, \langle a, y\rangle\rangle, k_{bs})\rangle\rangle & \stackrel{?}{\vdash} & \langle \mathsf{senc}(\langle \mathsf{c}_3, \langle a, \langle z, n_b\rangle\rangle\rangle, k_{bs}), \mathsf{senc}(\langle \mathsf{c}_4, n_b\rangle, z)\rangle \\ T_3^+ & \stackrel{?}{\vdash} & y \end{cases}$$

Following the same strategy as the one described previously, we reach the system $\mathcal{C}_4^{\mathsf{tag}}$

$$\mathcal{C}_4^{\mathsf{tag}} = \begin{cases} T_1^+ & \stackrel{?}{\vdash} & x \\ T_2^+ & \stackrel{?}{\vdash} & x' \\ T_3^+ & \stackrel{?}{\vdash} & y \\ T_3^+ & \stackrel{?}{\vdash} & \mathsf{senc}(\langle \mathsf{c}_3, \langle a, \langle y, n_b\rangle\rangle\rangle, k_{bs}) \end{cases}$$

The option of applying R_2 is not possible anymore. The only possibility is to apply R_f, and this leads to a constraint system that is not satisfiable. Hence, we conclude that $\mathcal{C}_{\mathsf{tag}}$ is not satisfiable. The presence of such tags improves the security of the protocol by avoiding type confusion attacks as the one described in this exercice. This avoids confusion between different ciphertexts that are used at different places in the protocol.

Exercise 2 : Blind signatures

1. Let $T = \{\mathsf{sign}(\mathsf{blind}(\mathsf{blind}(m, r_1), r_2), k);\ r_2;\ r_1\}$ and $u = \mathsf{sign}(m, k)$. We have that $T \vdash u$ and any proof of this fact used the term $\mathsf{sign}(\mathsf{blind}(m, r_1), k)$ as an intermediate node. The term $\mathsf{sign}(\mathsf{blind}(m, r_1), k)$ is not a subterm of $T \cup \{u\}$.
2. We consider the proof trees Π_1 and Π_2 below :

$$\Pi_1 = \left\{ \dfrac{\dfrac{\mathsf{blind}(m, r) \quad r}{m} \quad k}{\mathsf{sign}(m, k)} \right. \qquad\qquad \Pi_2 = \left\{ \dfrac{\dfrac{\mathsf{blind}(m, r) \quad k}{\mathsf{sign}(\mathsf{blind}(m, r), k)} \quad r}{\mathsf{sign}(m, k)} \right.$$

3. Given a prooftree Π, we define its *size* as its number of nodes plus the number of instances of the "special" rule that occurs in it (*e.g.* $\mathit{size}(\Pi_1) = 5$ whereas $\mathit{size}(\Pi_2) = 6$).

   Let T be a set of terms, and u be a term such that $T \vdash u$. Let Π be a prooftree witnessing this fact whose size is minimal. We show by induction on Π that Π only contains terms in $st_{ext}(T \cup \{u\})$. Moreover, if Π is reduced to a leaf or ends with a "decomposition rule" (*i.e.* one of the two last rules), then Π only contains terms in $st_{ext}(T)$. We do the proof by case analysis on the last rule of the prooftree.
   - Π is reduced to a leaf : the result trivially holds.
   - Π ends with an instance of the first inference rule : $u = \mathsf{blind}(u_1, u_2)$ and we denote Π_1 and Π_2 the two direct sub-prooftrees of Π. By induction hypothesis, we have that Π_1 (resp. Π_2) only contains terms in $st_{ext}(T \cup \{u_1\})$ (resp. $st_{ext}(T \cup \{u_2\})$), and thus Π only contains terms in $st_{ext}(T \cup \{u\})$ since $u = \mathsf{blind}(u_1, u_2)$ and $u_1, u_2 \in st_{ext}(u)$. The case of the second inference rule is similar.
   - Π ends with an instance of the third inference rule with $\mathsf{blind}(u, v)$ and v as hypotheses. We denote by Π_1 and Π_2 the two direct sub-prooftrees of Π. By minimality, we know that Π_1 is either reduced to a leaf or ends with an instance of third rule. Thus, by induction

hypothesis, we have that Π_1 only contains terms in $st_{ext}(T)$, and thus $\mathsf{blind}(u,v) \in st_{ext}(T)$. By induction hypothesis on Π_2, we deduce that Π_2 only contains terms in $st_{ext}(T \cup \{v\}) \subseteq st_{ext}(T)$ (since $\mathsf{blind}(u,v) \in st_{ext}(T)$). We have also that $u \in st_{ext}(T)$, and this allows us to conclude that Π only contains $st_{ext}(T)$.

– Π ends with an instance of the fourth rule with $\mathsf{sign}(\mathsf{blind}(u_1,v),u_2)$ and u_2 as hypotheses. We denote by Π_1 and Π_2 the two direct sub-prooftrees of Π. We have that Π_1 is either reduced to a leaf, or ends with an instance of the 2nd, 3rd, or 4th inference rule. Actually, thanks to minimality, an instance of the 2nd rule is not possible (indeed a smaller proof will be possible in this case). Thus, we have that Π_1 only contains terms in $st_{ext}(T \cup \{\mathsf{sign}(\mathsf{blind}(u_1,v),u_2)\})$, thus v and $u = \mathsf{sign}(u_1,u_2)$ are in $st_{ext}(T)$, and this allows us to conclude that Π only contains terms in $st_{ext}(T)$.

Of course, this result allows us to conclude.

4. The saturation algorithm seen during the lectures applies. Correction comes form the fact that we only add deducible terms in the saturation set. Completeness is derived from the locality result. Regarding termination, we have to ensure that the set $st_{ext}(T \cup \{u\})$ is finite. We have that $|st_{ext}(t)| \leq |t| + |t|_{\mathsf{blind}}$ (where $|t|$ is the number of symbols occurring in t and $|t|_{\mathsf{blind}}$ is the number of occurrence of the symbols blind in t). This allows us to conclude.
5. Regarding termination, the same measure as the one seen during the lectures allows us to conclude $(\#vars(\mathcal{C}), \Sigma_{u \in \mathsf{rhs}(\mathcal{C})}|u|)$ with a lexicographical order. Regarding correctness, the same arguments also apply. In particular, correcteness for the rule R_1 can be established as during the lectures. However, completeness is wrong :

$$m,\ r \overset{?}{\vdash} x \qquad \mathsf{sign}(x,k),\ r \overset{?}{\vdash} \mathsf{sign}(m,k) \text{ with } \theta = \{x \mapsto \mathsf{blind}(m,r)\}.$$

There is no simplication rule that we can applied to progress towards this solution and the system is not in solved form.

Exercise 3 : Encrypted Key Exchange protocol

1. We consider the signature $\mathcal{F} = \{\mathsf{senc}/2, \mathsf{sdec}/2, \mathsf{aenc}/2, \mathsf{adec}/2, \mathsf{pk}/1 \langle\ \rangle/2, \mathsf{proj}_1/1, \mathsf{proj}_2/1\}$, and the equational theory generated by the following equations :

$$\mathsf{sdec}(\mathsf{senc}(x,y),y) = x \quad \mathsf{adec}(\mathsf{aenc}(x,y),y) = x \quad \mathsf{proj}_1(\langle x,y\rangle) = x \quad \mathsf{proj}_2(\langle x,y\rangle) = y$$

Note : we may consider in addition the equation $\mathsf{senc}(\mathsf{sdec}(x,y),y) = x$.

2. $P_A(pw) = \mathsf{new}\ sk.\mathsf{out}(c, \mathsf{senc}(\mathsf{pk}(sk), pw)).\mathsf{in}(c, x_1).$
   $\mathsf{let}\ x_R = \mathsf{adec}(\mathsf{sdec}(x_1, pw), sk)\ \mathsf{in}$
   $\mathsf{new}\ n_a.\mathsf{out}(c, \mathsf{senc}(n_a, x_R)).\mathsf{in}(c, x_2).$
   $\mathsf{if}\ \mathsf{proj}_1(\mathsf{sdec}(x_2, x_R)) = n_a\ \mathsf{then}\ \mathsf{out}(c, \mathsf{senc}(\mathsf{proj}_2(\mathsf{sdec}(x_2, x_R)), x_R)).$

   $P_B(pw) = \mathsf{in}(c, y_1).\mathsf{let}\ y_{\mathsf{pub}} = \mathsf{sdec}(y_1, pw)\ \mathsf{in}$
   $\mathsf{new}\ r.\mathsf{out}(c, \mathsf{senc}(\mathsf{aenc}(r, y_{\mathsf{pub}}), pw)).\mathsf{in}(c, y_2).$
   $\mathsf{let}\ y_{na} = \mathsf{sdec}(y_2, r)\ \mathsf{in}\ \mathsf{new}\ n_b.\mathsf{out}(c, \mathsf{senc}(\langle n_a, n_b\rangle, r))$
   $\mathsf{in}(c, y_3).\mathsf{if}\ y_3 = \mathsf{senc}(n_b, r)\ \mathsf{then}\ 0.$
3. We have that

$$\phi_0 = \mathsf{new} pw.\mathsf{new} sk, r, n_a, n_b.\{x_1 \mapsto \mathsf{senc}(\mathsf{pk}(sk), pw);\ x_2 \mapsto \mathsf{senc}(\mathsf{aenc}(r, \mathsf{pk}(sk)), pw);$$
$$x_3 \mapsto \mathsf{senc}(n_a, r);\ x_4 \mapsto \mathsf{senc}(\langle n_a, n_b\rangle, r);\ x_5 \mapsto \mathsf{senc}(n_b, r)\}$$

Any test that is satisfied by $\mathsf{new}\ pw.\mathsf{new}\ \tilde{n}.\sigma_0$ will be also true in $\mathsf{new}\ \tilde{n}.\sigma_0$. Thus, to check whether static equivalence holds between these two frames, we will focus on computing the sets $\mathsf{sat}(\phi)$ and $\mathsf{Eq}(\phi)$ when $\phi = \mathsf{new}\,\tilde{n}.\sigma_0$. Then, we will check whether all these tests are also valid in $\mathsf{new}\,pw.\mathsf{new}\,\tilde{n}.\sigma_0$.

We list below the elements in $\mathsf{sat}(\phi)$ together with their associated recipe, and the rule applied according to the definition of frame saturation.

$\mathsf{senc}(\mathsf{pk}(sk), pw)$	x_1	(1)
$\mathsf{senc}(\mathsf{aenc}(r, \mathsf{pk}(sk)), pw)$	x_2	(1)
$\mathsf{senc}(n_a, r)$	x_3	(1)
$\mathsf{senc}(\langle n_a, n_b\rangle, r)$	x_4	(1)
$\mathsf{senc}(n_b, r)$	x_5	(1)
pw	pw	(3)
$\mathsf{pk}(sk)$	$\mathsf{sdec}(x_1, pw)$	(3)
$\mathsf{aenc}(r, \mathsf{pk}(sk))$	$\mathsf{sdec}(x_2, pw)$	(3)

In case, we have the equation $\mathsf{senc}(\mathsf{sdec}(x, y), y) = x$ in our model, all the equalities that we can infer are actually trivial, and thus static equivalence holds. Otherwise, there is a test $\mathsf{senc}(\mathsf{sdec}(x_1, pw), pw) = x_1$ which holds in ϕ, and does not hold in $\mathsf{new}, pw.\phi$ (note that we have first to rename pw with pw').

4. Actually, we have $\mathsf{new}\tilde{n}.\sigma \sim \mathsf{new}\,pw.\mathsf{new}\,\tilde{n}.\sigma$ if, and only if, $\mathsf{new}\,pw.\mathsf{new}\,\tilde{n}.\sigma$ is resistant to dictionary attack against pw with the definition seen during the lectures, *i.e.* if and only if :

$$\mathsf{new}\,pw.\mathsf{new}\tilde{n}.(\sigma \cup \{x \mapsto pw\}) \sim \mathsf{new}\,pw.\mathsf{new}\,pw'.\mathsf{new}\,\tilde{n}.(\sigma \cup \{x \mapsto pw'\})$$

We show the following equivalence :

$$\mathsf{new}\,\tilde{n}.\sigma \sim \mathsf{new}\,pw.\mathsf{new}\,\tilde{n}.\sigma$$
if, and only if,
$$\mathsf{new}\,\tilde{n}.\sigma \sim \mathsf{new}\,pw'.\mathsf{new}\,\tilde{n}.(\sigma\{pw \mapsto pw'\}) \text{ by } \alpha\text{-conversion.}$$
if, and only if,
$$\mathsf{new}\,pw, \mathsf{new}\,\tilde{n}.(\sigma \cup \{x \mapsto pw\}) \sim \mathsf{new}\,pw.\mathsf{new}\,pw'.\mathsf{new}\,\tilde{n}.(\sigma\{pw \mapsto pw'\} \cup \{x \mapsto pw\})$$

Note that for this equivalence, it is easy to see that any test that distinsguishes the two frames can be transformed in another test that distinsguishes the two resulting frames (and conversely) by replacing the use of pw with x (or the converse).

Lastly, this equivalence holds if, and only if,

$\mathsf{new}\,pw, \mathsf{new}\tilde{n}.(\sigma \cup \{x \mapsto pw\}) \sim \mathsf{new}\,pw'.\mathsf{new}\,pw.\mathsf{new}\tilde{n}.(\sigma \cup \{x \mapsto pw'\})$ by α-conversion.

Hence we conclude that ϕ_0 is resistant to dictionary attacks against pw if, and only if, the static equivalence studied at the previous question holds.

5. We consider two additional function symbol $\mathsf{samekey}/2$ and $\mathsf{ok}/0$, as well as, the equation $\mathsf{samekey}(\mathsf{senc}(x_1, y), \mathsf{senc}(x_2, y)) = \mathsf{ok}$. Clearly, ϕ_0 is not resistant against dictionary attack on pw. It suffices to consider the test $\mathsf{samekey}(x_1, \mathsf{senc}(n, x)) \stackrel{?}{=} \mathsf{ok}$. This test holds on the frame $\mathsf{new}\,pw.\mathsf{new}\,\tilde{n}.(\sigma \cup \{x \mapsto pw\})$ but does not hold in $\mathsf{new}\,pw.\mathsf{new}\,pw'.\mathsf{new}\,\tilde{n}.(\sigma \cup \{x \mapsto pw'\})$.