TD 2: Security properties in the symbolic model

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

November 22, 2024

Exercise 1: Security Properties

Try to find a way to express these security properties in protocol between two parties A and B:

1. **Secrecy of some name s:** This mean that the adversary cannot know the name s;
2. **Unlinkability:** Here there is multiple A (with the same process but different identities) and multiple sessions of each A (i.e. each identity can be executed multiple times), the goal of the adversary is to know one A talking multiple times or not;
3. **Authentication:** Here at some point B want to be sure that she has indeed talked to A;
4. **Mutual authentication:** Here at some point B (resp. A) want to be sure that she talked to A (resp. B);
5. **Strong secrecy of some name s:** The adversary cannot know *anything* about the name s.

Hint: you can assume there is some event in the process to help you.

Exercise 2: Secrecy

For each of the following processes, try to see if the secrecy of the secret n is ensured, otherwise exhibit an trace tr and a recipe R such that R can be reduce to n.

1. $P_1 = \textbf{new } k.\textbf{out}(c, \mathsf{senc}(n, k)).\textbf{out}(c, k)$
2. $P_2 = \textbf{in}(c, x).\textbf{out}(c, \mathsf{senc}(n, x))$
3. $P_3 = \textbf{out}(c, \mathsf{senc}(n, k)).\textbf{in}(c, x).\textbf{if } x = n \textbf{ then out}(c, k)$
4. $P_4 = \textbf{out}(c, \mathsf{senc}(n, k)).\textbf{in}(c, x).\textbf{if } x = k \textbf{ then out}(c, k)$
5. $P_5 = \textbf{in}(c, x).\textbf{let } y = \mathsf{adec}(x, k) \textbf{ in out}(c, k) \textbf{ else out}(c, \mathsf{aenc}(n, \mathsf{pk}(k)))$
6. $P_6 = !P_5$

Exercise 3: A flawed fix

Consider the following protocol between A and B defined as followed:

$$A \to B \ \ (\{A, \{s\}_{\mathsf{pk}(B)}\}_{\mathsf{pk}(B)})$$
$$B \to A \ \ (\{B, \{s\}_{\mathsf{pk}(A)}\}_{\mathsf{pk}(A)})$$

Show (informally) that the secrecy of s is not ensured.

Exercise 4: A flawed fix – follow up

Consider the following protocol between A and B defined as followed:

$$A \to B \ \ (\{A, \{s\}_{\mathsf{pk}(B)}\}_{\mathsf{pk}(B)})$$
$$B \to A \ \ (\{B, \{s\}_{\mathsf{pk}(A)}\}_{\mathsf{pk}(A)})$$

1. Write down the process calculus, explict the signature that you use.
2. Translate your process calculus intro an labelled transition system.
3. Write down formally the execution leading to an attack on the secrecy of s with the LTS.

Exercise 5: The Needham-Schroeder protocol

Consider the following protocol between A and B defined as followed:

$$A \to B \ \ \{A, N_A\}_{\mathsf{pk}(B)}$$
$$B \to A \ \ \{N_A, N_B\}_{\mathsf{pk}(A)}$$
$$A \to B \ \ \{N_B\}_{\mathsf{pk}(B)}$$

1. Write it down formally in the process calculus.
2. Translate your process calculus into an labelled transition system.
3. Write the property representing the secrecy of N_A, and the one representing the authentification of A regarding B.
4. Show that the secrecy of N_A is still not ensured. (Explicit the assumption you made on the interpretation to do it.)
5. Can you find a fix of this protocol to have the secrecy of N_A?

Exercise 6: A first fix: the Needham-Schroeder-Lowe protocol

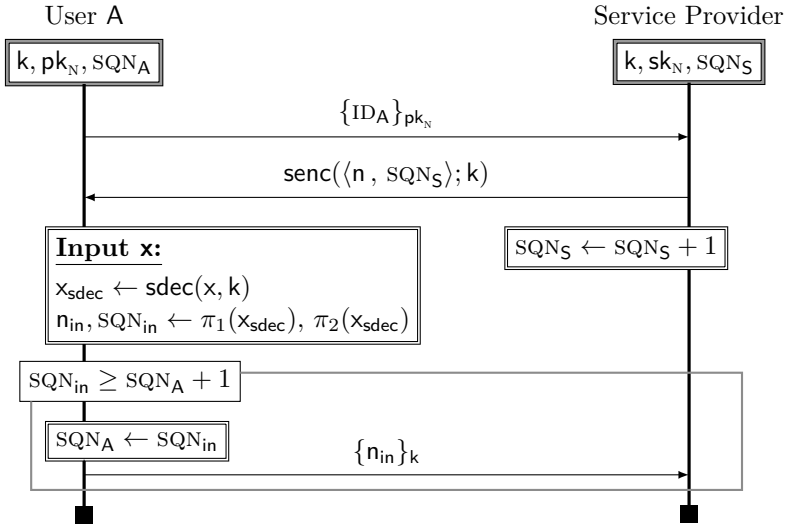
Consider the following protocol between A and B defined as followed:

$$A \to B \ \ \{A, N_A\}_{\mathsf{pk}(B)}$$
$$B \to A \ \ \{B, N_A, N_B\}_{\mathsf{pk}(A)}$$
$$A \to B \ \ \{N_B\}_{\mathsf{pk}(B)}$$

1. Write it down formally in the process calculus.
2. Show that the secrecy of N_A is finally ensured. (Explicit the assumption you made on the interpretation to do it.)

Exercise 7: AKA-

Consider the following protocol :



1. Write it down formally all the agents in the process calculus.
2. Translate your process calculus into an labelled transition system for the case where there is only one user and one session by user.
3. Translate your process calculus into an labelled transition system for the case where there is multiple users and one session by user.

Exercise 8: Process Calculus to LTS

1. With all the examples of process calculus to LTS (in previous exercises), can you came up with a way to translate the process calculus to a LTS?
2. Does the operational semantics of the process calculus match the semantics given by the translated LTS?