CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL
Institut für Informatik · AG Software Engineering
Priv.-Doz. Henning Schnoor

Engineering Secure Software Systems Winter 2020/21
Exercise Sheet 7

issued: December 15, 2020 **due:** January 7, 2021

Exercise 7.1, the FFGG prototocol: too complicated? (10 Points)

Can you come up with a simpler protocol that is secure when only one session is running, but becomes insecure if the adversary can start as many instances as she wishes? Is there an "advantage" of the ffgg protocol (as an example illustrating the need for the analysis of parallel sessions) over your example?

Solution A very simple protocol that requires two sessions in order to be insecure is given by the following receive/send rules:

Alice:	ϵ	$\rightarrow$	$\mathsf{enc}^{\mathsf{s}}_{k_{AB}}\left(\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(\mathsf{FAIL})\right)$
Bob:	$\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(x)$	$\rightarrow$	x

Clearly, if the adversary has only one Bob-instance available, she cannot obtain the FAIL-constant, but with two instances, there is an obvious attack. The example can be generalized to requiring n instances of Bob by using encryption nested n times.

The disadvantage of this protocol is that here is is very obvious how to proceed, and simple preprocessing based on the depth of the involved terms could add the additionally required instances. Since in general, insecurity for an arbitrary number of sessions is undecidable (as proven in the lecture), such a simple "preprocessing" does not work in the general case. An example where this does not suffice is of course the protocol used in the lecture's proof of the undecidability result.

Exercise 7.2, unbounded instances formalization (10 Points)

Specify the Needham-Schroeder protocol as an instance of the decision problem UNBOUNDED-INSECURE, and show that it is insecure in this formalization. Discuss the differences between expressing the protocol using this formalism compared to the earlier formalization using the decision problem INSECURE.

Note: You do not need to make your constructions formal. The goal of this exercise is to get a good understanding on how a formal definition of INSECURE (which we did not fully state in the lecture) would look like.

Solution We repeat the specification of the protocol:

$A \rightarrow B$	$\mathsf{enc}^{\mathsf{a}}_{k_B}(A, N_a)$
$B \rightarrow A$	$\mathsf{enc}^{\mathsf{a}}_{k_A}(N_a, N_b)$
$A \rightarrow B$	$\mathsf{enc}^{\mathsf{a}}_{k_B}(N_b)$

An input to UNBOUNDED-INSECURE needs to contain the following:

- the specification of a "target session" that contains the FAIL-constant,
- the specification of (replicable) sessions for Alice and Bob.

We first specify Bob's session, the one under analysis. This is the same as Bob's usual specification, with the FAIL-rule added. Note that we can assume, without loss of generality, that Bob in this session is interacting with honest Alice.

$$\begin{array}{lcl} \mathsf{enc}^{\mathsf{a}}_{k_B}(A,x) & \rightarrow & \mathsf{enc}^{\mathsf{a}}_{k_A}(x,N_B) \\ [\mathsf{BREAK},x,N_B] & \rightarrow & \mathsf{FAIL} \end{array}$$

We now specify "replicable" sessions for Alice and Bob. Note that we can assume, without loss of generality, that Alice and Bob are the only honest principals, as this is a maximally pessimistic assumption. We define a "pattern" for an initializer session:

parameters running number i, public key k_{init} of the identity running this session, name id_{init} matching to the public key, public key k_{resp} of the instance that this initializer wants to connect to. Note that we can assume that k_{init} is Alice's or Bob's key, as the adversary can run Charlie-controlled sessions himself

session

$$\begin{array}{lcl} \epsilon & \rightarrow & \mathsf{enc}^{\mathsf{a}}_{k_{resp}}\left(id_{init},N^i_{init}\right) \\ \mathsf{enc}^{\mathsf{a}}_{k_{init}}\left(N^i_{init},y^i\right) & \rightarrow & \mathsf{enc}^{\mathsf{a}}_{k_{resp}}\left(y^i\right) \end{array}$$

Similarly, we define a "replicable" responder session:

parameters public key k_{resp} of the identity running this session, name id_{init} and key k_{init} of the corresponding communication partner, running number i

session $\mathsf{enc}^{\mathsf{a}}_{k_{resp}}\left(id_{init},x^i\right) \rightarrow \mathsf{enc}^{\mathsf{a}}_{k_{init}}\left(x^i,N^i_{resp}\right)$

The attack can now easily be represented in this formalization:

- The attacker starts one instance of the initiator session, using parameters $k_{init} = k_A$, $id_{init} = A$, $k_{resp} = k_C$ and $i = 1$.
- The attacker also interacts with Bob's (explicit) session.
- The attacker can now deliver the messages analogous to the standard representation of the protocol.

The main difference between this formalization and the one as input to INSECURE is that in this version, we do not build any information about the attack into the input, but let the adversary start arbitrarily many sessions.

Exercise 7.3, Rusinowitch-Turuani with specified maximal number of sessions (10 Points)

We saw in the lecture that the "unbounded session" version of INSECURE is undecidable. A weaker version of that problem can be obtained by allowing instances to INSECURE to be accompanied by a maximal number of copies in which the adversary may start the corresponding protocol instance (we assume a mechanism that automatically renames variables to ensure that they are "local" to the copy in which they are used). Does the "positive" part of the Rusinowitch-Turuani theorem still hold for this generalization?

Hint: You are not expected to give a formal proof of your conjectures, an informal justification suffices. Also, be explicit about how the "maximal number of copies" is specified in the input to your generalized problem.