Exercise Class January 7, 2021

This is a slightly edited and re-formated transcript of the live notes taken in class.

Exercise 7

Task: the FFGG prototocol: too complicated?

looking for: simple "alice instance," so that:

- a protocol with a single instance is secure,
- a protocol with "more of the same" instances is insecure.

(protocol may contain more participants than Alice).

Alice $\epsilon \to \mathsf{enc}^{\mathsf{a}}_{k_B}\left(N^1_A, N^2_A, N^3_A, N^4_A, FAIL\right)$
Bob $\mathsf{enc}^{\mathsf{a}}_{k_B}\left(x_1, ...x_4, y\right) \to (x_1, \mathsf{enc}^{\mathsf{a}}_{k_B}\left(x_2, \ldots, x_4, y, x_1\right))$

Second version:

Alice $\epsilon \to \mathsf{enc}^{\mathsf{s}}_{k_{AB}}\left(\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(FAIL)\right)$
Bob $\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(x) \to x$

Algorithm idea: Use n copies of Alice/Bob, where n is something like

- maximal depth of a term in the protocol,
- maximal length of a term,
- ...

Any such idea fails, since it contradicts the undecidability proof given in the lecture.
question: How does Bob access messages?
idea: adversary can start as many "copies" of the protocols as she likes, cp. ProVerif notation
!Alice || !Bob

- start Alice1: $\epsilon \to \mathsf{enc}^{\mathsf{s}}_{k_{AB}}\left(\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(FAIL)\right)$
- activate Alice, get $\mathsf{enc}^{\mathsf{s}}_{k_{AB}}\left(\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(FAIL)\right)$,
- start Bob1: $\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(x_1) \to x_1$
- active Bob1: deliver $\mathsf{enc}^{\mathsf{s}}_{k_{AB}}\left(\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(FAIL)\right)$, gets $\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(FAIL)$
- start Bob2: $\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(x_2) \to x_2$
- active Bob2: deliver $\mathsf{enc}^{\mathsf{s}}_{k_{AB}}(FAIL)$, gets $FAIL$

Task: unbounded instances formalization

The situation we want to cover is: An unlimited number of "people" run an unlimited number of "copies" of the protocol. Question: Is there an attack in any of these setting? Start with specifications of the protocol roles, i.e., the initiator and the responder role. Let the adversary control the setting. Then, what does the adversary need to be able to do?

- start copies of the initiator, with chosen identities for initiator and responder.
- start copies of the responder, with chosen identities for initiator and responder.

only interesting cases: adversary choses initiator and/or responder to be alice or bob.
What about sessions running between Charlie and Dave, both controlled by the adversary?
adv can: start responder session between charlie and dave, (both of whom adv controls) if responder session contains a break/fail rule, then ?

simple example responder Alice: $\mathsf{sig}_{k_A}(BREAK) \rightarrow FAIL$
unbounded setting: (ProVerif: !)
adv may start arbitrarily many instances of this.
many copies of $\mathsf{sig}_{k_A}(BREAK) \rightarrow FAIL$ don't help her.
adv can can also start protocol with a different identity!
Adv has Charlie's private key, and starts:
responder Charlie: $\mathsf{sig}_{k_C}(BREAK) \rightarrow FAIL$
adv can derive $\mathsf{sig}_{k_C}(BREAK)$, and get FAIL.

Task: Rusinowitch-Turuani with specified maximal number of sessions