Exercise Class December 3, 2020

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

Exercise Sheet 3

Task: "Weird Operator"

Looking for: crypto operator where we need "weird" derivations, i.e., a composition rule where we compose a term t that does not show up as a subterm in either

- S (the set we're deriving from), or
- m (the message we want to derive).

Idea: decomposition that requires composition. We construct a "weird hash function:" From $\mathsf{hash}(t)$, you cannot get t (as usual), but we can get t from $\mathsf{hash}(\mathsf{hash}(t))$. This fits the idea.

DY rules for this "weird hash function" (T is the closed set):

- if T contains $\mathsf{hash}(\mathsf{hash}(t))$, then T contains t
- if T contains t, then T contains $\mathsf{hash}(t)$

composition/decomposition rules:

- $L_c(hash(t))$: $t \rightarrow hash(t)$
- $L_d(hash(hash(t)))$: $hash(hash(t)) \rightarrow t$

base set and target message for derivation:

- base set to derive the mesage from: $S = \{\mathsf{hash}(N_A)\}$
- $m = N_A$

m is derivable, but only by adding one layer of hash functions. Used rules:

- composition: we compose $hash(hash(N_A))$, no subterm of either S (stating set) nor m (target message)
- decomposition: we decompose $hash(hash(N_A))$, no subterm of either S (stating set) nor m (target message)

Since the Lemma says that we only use composition rules to compose subterms of terms in $S \cup \{m\}$, and only use decomposition rules to decompose subterms of S, this "weird hash function" contradicts both components of the lemma. The reason is of course that this hash function does not follow the 1-step-pattern discussed in the video.

Exercise Sheet 4

Woo-Lam protocol

- why no Alice session in the attack version of the protocol?
- why does Bob release FAIL when the protocol completes?

Alice is not active in the attack, so no need to model her steps, if Bob accepts in Alice's session, the attack is successful.

conclusion The analysis given by the Rusinowitch-Turuani theorem is not quite the "push-button tool" we wanted for automatic analysis of cryptographic protocols. Possible and impossible extensions of the Rusinowitch-Turuani methods will be discussed soon in the lecture.