ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

S. Mödersheim
Ch. Sprenger
AS 2009

Formal Methods for Information Security

Exercise Sheet 1

Hand-in date: Oct 5, 2009

Exercises *do not count* towards the final grade for the course, but you can hand them in to have them corrected in order to get the feedback. This is optional but recommended.

Assignment 1.1: Applying Formal Methods.

In this exercise, assume you've been hired to do some consulting work applying formal methods. The Swiss government wants you to evaluate their new software which is designed to make referenda (*Volksabstimmungen*) automated and online. Their contractors, Spitzenprotokoll AG, have designed the following protocol:

$$\begin{array}{lll} 1. & A \to S: & A \\ 2. & S \to A: & \{Q, N_S\}_{\mathsf{pk}(A)} \\ 3. & A \to S: & \{Ans_Q, N_S\}_{\mathsf{pk}(S)} \end{array}$$

where A is a voter, S is the voting server, N_S is a nonce sent by the server to ensure freshness, Q is a referendum question, and Ans_Q is A's answer to that question. Assume S and A share their respective public keys in advance.

Your job is to compare this protocol with the existing, physical voting method, which we assume is secure. We make the standard Dolev-Yao assumptions presented in the lecture.

(a) Does this protocol provide anonymity? Can an attacker tell who has voted?

(b) Does it provide confidentiality? Can an attacker find out, for a given A, how he or she voted?

(c) Does the protocol provide authentication? Can S be sure that the answer came from A?

(d) Can each voter vote at most once?

(e) Is availability guaranteed? Can A be sure that she can vote if she wants to?

(f) Is integrity provided? Does S know that a given answer hasn't been modified.

Assignment 1.2: Attack-Preserving Assumptions.

In protocol analysis, making assumptions or abstracting certain things away can be very helpful. Some assumptions, however, can exclude attacks at analysis time. We call these assumptions *non-attack-preserving*. Using non attack-preserving assumptions is a tradeoff.

(a) What are some arguments for and against the use of non-attack-preserving assumptions or abstractions?

(b) In the lecture (Module 2, Slide 10) we have made the assumption that when A and B receive messages, they "know" what protocol they belong to.

Do you think this assumption is reasonable? Do you think it is attack-preserving?

(c) What common mechanisms do we use in practice to try to realize this assumption?
Hint: On a Linux system, check out the file `/etc/services`.

Assignment 1.3: Diffie-Hellman

In module 2, we have built key-establishment protocols using an honest key-server S who has a shared key $\mathsf{sk}(A, S)$ with every agent A.

(a) Combine this schema with the Diffie-Hellman key-exchange, using the key-server to authenticate the exchange.
Hint: Use the structure of the protocol on slide 28.

(b) Argue why your Diffie-Hellman based protocol offers stronger security than the key-exchange protocols of module 2 in a certain situation.
Hint: Suppose the intruder is able at some point to compromise the honest key-server and find out all long-term keys $\mathsf{sk}(A, S)$.