

# 18739- Foundations of Security and Privacy: Homework 1 (Mur $\Phi$ Model Checker)

September 6, 2011

Course: 18739, Carnegie Mellon University  
Instructor: Professor Anupam Datta  
Teaching assistant: Divya Sharma

## 1 Overview

The purpose of this assignment is to get you started with the Mur $\Phi$  verifier. You are required to extend the provided Mur $\Phi$  model of the Needham-Schroeder and SSL protocol with new rules and invariants. You are also required to run the verifier and interpret the results of the verification. The assignment is for 10 points.

## 2 Where to find the required files

All the files are provided on the course webpage at <http://www.ece.cmu.edu/ece739/schedule.html>. There is only one zip file containing the model files relevant for the homework (18739fall2011hw1.zip).

### 2.1 Building the tool

There are two versions of the tool - one for gcc 4.2.x and another for gcc 4.3.x. Check your gcc version using the command “g++ -dumpversion” and use the appropriate version. The Readme file provides enough information on how to build the tool (also the demo shown in class should be helpful).

## 3 Submission

You should zip your answers into a single zip file and submit by accessing the CMU blackboard site. Also, mention your group number in the name of the zip file itself. The following files are expected:

1. All files relevant to the homework, as mentioned in the assignment below.
2. A PDF write-up explaining the changes done in the assignment.

The assignment is due 12:01 AM (PDT), September 20.

## 4 General Description

### 4.1 NS

In the first 3 problems of the assignment we will look at two similar protocols, the Needham-Schroeder protocol (NS), and its fixed version the Needham-Schroeder-Lowe (NSL) protocol. Using the informal arrows-and-messages diagrams, they can be described as follows:

Needham-Schroeder protocol:

$$\begin{aligned} A &\rightarrow B : \{|A, N_A|\}_{K_B} \\ B &\rightarrow A : \{|N_A, N_B|\}_{K_A} \\ A &\rightarrow B : \{|N_B|\}_{K_B} \end{aligned}$$

Needham-Schroeder-Lowe protocol:

$$\begin{aligned} A &\rightarrow B : \{|A, N_A|\}_{K_B} \\ B &\rightarrow A : \{|B, N_A, N_B|\}_{K_A} \\ A &\rightarrow B : \{|N_B|\}_{K_B} \end{aligned}$$

Mur $\Phi$  models of both protocols are given in the file ns.m in the assignment directory. Boolean variable FIXED is used to switch between the two. The model also contains the actions of the intruder who can intercept messages and generate new messages, using observed data and initial knowledge. In each of the problems you are asked to check if some protocol invariant is satisfied in the Mur $\Phi$  model. If an invariant fails, you are required to write down the sequence of rules fired in the violating trace and a corresponding message-and-arrows diagram describing the attack. If an invariant is satisfied, write down the information about the number of states explored and the time needed for verification. For the purpose of this assignment, you can use the default parameters in the model (one initiator, one responder, one intruder), but you are encouraged to run the verifier with larger parameters.

Extra credit will be given for finding a previously undiscovered bug in the protocol.

### 4.2 SSL

The model of SSL protocol is given in the file ssl.m. This is much more complicated than the NS protocol and has been written in an incremental way as described in the paper “Finite State Analysis of SSL 3.0.” All fixes for intermediate protocols (named version A,B) are implemented and can be turned on/off by using the Boolean variables defined at the start of the file. The provided file does not check for the version rollback attack for a new session. In part (a) of the last problem your task is to write an invariant to check that the version rollback attack can be performed in this model and then provide the sequence of rules that leads to the failure of this invariant. In part (b) you have to fix the protocol so that this invariant is satisfied. You can use the default parameters in the model (one initiator, one responder, one intruder).

(Hints are provided in the file ssl.m itself)

## 5 Problems

### 5.1 Problem 1

(2.0 points)

### 5.1.1 Part a

Two invariants in the model specify the mutual authentication property we are interested in verifying. When translated to English. “initiator correctly authenticated” invariant states that whenever a responder  $i$  completes a session apparently with some initiator  $j$ , then it must be that  $j$  has completed a session, apparently with  $i$ . The meaning of the other invariant is analogous. For both invariants (independently), determine if they are satisfied in the provided Mur $\Phi$  model of the NS protocol.

### 5.1.2 Part b

Look up the definition of authentication as matching records in the paper by Diffie et al. (W. Diffie, P.C. van Oorschot, M.J. Wiener. *Authentication and authenticated key exchanges, Designs, Codes and Cryptography*, vol.2 (1992), pp.107-125. <http://people.scs.carleton.ca/paulv/papers/sts-final.pdf>) mentioned on the discussion board on Blackboard (Sep 1, 2011) and comment on its relation to the invariants in the Murphi model.

## 5.2 Problem 2

(2.0 points) In this problem we investigate whether NS or NSL can be used as a key exchange protocol. Specifically, we want to check if the nonces exchanged in the protocol remain secret. All the responders are assumed to be honest. (a) Write down the Mur $\Phi$  invariant “initiator secrecy” modeling the following property: if some initiator  $i$  completes a session with an honest responder then the intruder does not know the initiator’s nonce. Also, write down the analogous invariant “responder secrecy.” (b) For both NS and NSL protocol, test if these invariants are satisfied.

## 5.3 Problem 3

(2.0 points) In this problem we look at a scenario where a malleable encryption scheme is used in the NSL protocol. We say that an encryption scheme is malleable if, under some circumstances, an intruder can predictably modify the contents of an encrypted message without knowing the corresponding decryption key. (a) Assume that, if the intruder knows a message of the form  $\{|Y, \text{data}|\}_{K_X}$ , where  $X$  and  $Y$  are agent names, then he can generate a message of the form  $\{|Z, \text{data}|\}_{K_X}$  where  $Z$  is an arbitrary agent name. Write down a Mur $\Phi$  rule which models this capability of the intruder in the NSL protocol. (b) Test if the authentication and secrecy invariants hold in the NSL protocol with the improved intruder.

## 5.4 Problem 4- SSL

(4.0 points)

### 5.4.1 Part a

In the given `ssl.m` file, the invariant for detecting a version rollback attack is missing. Read the paper “Finite State Analysis of SSL 3.0” and write two invariants to detect a version rollback attack. These invariants should fail for now and the fix will be done by you in part (b).

Hint: At the end of the file `ssl.m` there are comments about the two invariants that are to be written.

### 5.4.2 Part b

A boolean variable `ServerVersionCheckInKey` is assigned the value *false*, but this variable is never used anywhere in the file `ssl.m`. Use the variable `ServerVersionCheckInKey` at appropriate places to check for version. After the proper check is put in the file, the invariants written in part (a) will be satisfied.

Hint: The check is needed only in one ruleset.