

Modern Cryptography Introduction

1st Semester 2008/2009

P. Lafourcade.

SET 7

Date: 17.11.2008

All files needed for this session are available on the lecture web page:

http://www-verimag.imag.fr/~plafourc/teaching/L2_Magistere_2008_2009

For using Avispa the command line is `avispa`, for scyther is `scyther-guy.py` and for proverif is `analyzer`. With Avispa you need to specify the tool used for instance by typing “`avispa toto.txt -ofmc`”. You should use proverif in horn clause using the following command “`analyzer -in horn toto.txt`”.

Exercise 1

AVISPA:

1. Use AVISPA (<http://www.avispa-project.org/>) to analyze Needham Schroder protocol, called NSPK with the 4 Tools (OFMC, Cl-Atse, SATMC and TA4SP) using the web interface and also the command line.

Hint: For the command line use `avispa -help` in an xterm.

2. Understand the different results given by the Tools.
3. Copy the NSPK protocol in a file called NSPK-Lowe and correct by yourself the protocol with Lowe correction.
4. Check your corrected version and compare your result with NSPK-fix given on AVISPA web site.

Exercise 2

SCYTHER:

Now open in Scyther the file `protocol1.spdl` downloaded on the lecture webpage using:

```
scyther-gui.py protocol1.spdl
```

- a) Verify the security claims in the protocol using Scyther (Press F1). You find several attacks. Explain the attacks: why is the property violated in each case?
- b) Copy the protocol file to your own directory, and call it `protocol1fixed.spdl`. In the file, make sure you change ‘`protocol1`’ to ‘`protocol1fixed`’. Improve the protocol such that the property now holds, and use Scyther to show that your improved protocol indeed meets the requirements.

Hint: Examine the first message: $\{\mathbf{R}, ni\}_{pk(R)}$ or compare the protocol to the fixed Needham-Schroeder protocol shown in the lecture.

Exercise 3

SCYTHER: Open in Scyther the file `protocol2.spdl` downloaded on the lecture webpage. This protocol contains a rather large messages and contains many random numbers (nonces) and hash functions. Not all of these elements are necessary to guarantee the correctness of the protocol.

- Suggest five efficiency improvements (in terms of message size or complexity) for the protocol, and motivate your choice. Test each suggested improvement using Scyther. If any of your suggestions fails, explain why.

Exercise 4

Proverif with Horn Clauses:

1. Check the file `needham.horn` with `proverif` (cf my web page).

Hint use `analyzer -help` to get some help in a terminal:

```
analyzer -in horn needham.horn
```

2. Understand the attack found
3. Correct it.