14-741/18-631
Protocol Verification with ProVerif

1 Introduction

ProVerif is a software tool for automatically analyzing the security of cryptographic protocols and it is designed to verify protocol in the so-called Dolev-Yao attacker model. Briefly speaking a Dolev-Yao attacker can intercept all communications and inject his own messages into the network but the cryptographic primitives (e.g. encryption algorithm) used in the protocol are assumed to be perfect. That is, the attacker can only decrypt an encrypted message if he knows the required key. This is an approximation of what an attacker can do in real life but analysis in the Dolev-Yao model can already cover a rich class of attacks for example the man-in-the-middle attack you learned in *Unit5: Security Protocol*.

This document includes an installation guide and lists a few resources and pointers that can help you start modeling and analyzing protocols in ProVerif. This document does not cover modeling details of ProVerif for which you should read Chapters 2 and 3 (excluding Chapter 3.4) of ProVerif's manual.

2 Installation

2.1 Installation via OPAM

ProVerif is compatible with all major operating systems (Linux, Mac, and Windows). The easiest way to install ProVerif is to use the package manager OPAM; you need to first install OPAM on your machine if it is not already installed (instructions to install OPAM: https://opam.ocaml.org/doc/Install.html). Once OPAM is installed, run the following commands to install ProVerif:

```
opam init
opam update
opam install proverif
```

2.2 Use the provided Dockerfile

We also provide a Dockerfile if you have trouble installing ProVerif on your machine. You need to make sure Docker is installed (installation guide: https://docs.docker.com/get-docker/) on your system before proceeding.

- Build the a Docker image from the provided Dockerfile (you should issue the following command while in the directory where the Dockerfile is located).

  ```
  docker build -t docker-proverif .
  ```

- Run the image inside a Docker container.

  ```
  docker run --rm -it -v "$PWD:/models" -w "/models" docker-proverif
  ```

The command above will mount you current directory to `/models` in the container. That is, you can work on your ProVerif file in you host machine with your favorite editor and invoke ProVerif's binary to prove your model inside the container.

3 Using ProVerif

To analyze a protocol in ProVerif, you will need to model the protocol and the properties in ProVerif's input language (the typed pi-calculus) see Section 4 for some useful resources. Once you have a model you can execute ProVerif with

```
./proverif <filename>.pv
```

4 Useful Resources

- ProVerif has a comprehensive user manual (https://bblanche.gitlabpages.inria.fr/proverif/manual.pdf) that contains everything you need to know about ProVerif. However, you only need to read Chapters 2 and 3 (excluding Chapter 3.4) to complete your homework.
- There is a online demo (http://proverif16.paris.inria.fr/) for ProVerif that you can play around with it. You may also find it helpful to look through the example ProVerif models there.

5 Advice for Homework

- Read the selected sections (Chapters 2 and 3 excluding 3.4) of the manual especially the examples.
- You only need to use secrecy and authentication (as correspondence assertions). You don't need to worry about ProVerif's advance features such as proving observational equivalence.
- You should allow infinite replications of protocol agents in your model (see the starter code `protocol.pv` in your handout).
- Make sure your model is executable from start to the finish (you can check this by put an *event* at the end of the protocol and make a query to see if that *event* is reachable).
- We will go over modeling details and common pitfalls in recitation class but please start early.
- If you stuck somewhere, feel free to make a post (public or private) on Piazza.
- Good luck!