Université Libre de Bruxelles
INFO-F405 - Computer Security
Project 2: Granting access to Web Services

Assistant: Naïm Qachri

2013-2014

Introduction

You are asked to design and develop two kinds of servers : an Authorisation Server (AS) and two Web Services (WS).

The Authorisation Server authenticates the users and delivers short-term session keys to authorised users in order to allow them to securely contact a Web Service.

On the basis of the following notations, you are asked to implement the protocol described herehunder as well as the Authorisation Server and two Web Services. The Web Services are accessed through confidential channels. The first Web Service is a secure virtual blackboard on which authorised clients can publish posts. The second Web Service is a virtual keychain server in which authorised clients can securely store their passwords.

Notations

– $h(.)$ is a SHA-1 cryptographic hashing ;

– $E_{Client}(.)$, $E_{AS}(.)$ and $E_{WS}(.)$ are RSA encryptions using the public key (1024 bits) of respectively the Client, the Authorisation Server and a Web Service ;

– $E_{K_{A\_B}}(.)$ is an AES encryption using a 128-bits key K_{A_B} shared between Alice and Bob ;

– r_i is a random nonce.

The protocol

The aim of the protocol is to mutually authenticate the Clients, the Web Services and the Authorisation Server and to distribute the symmetric keys that will allow respectively the Authorisation Server and the Web Service as well as the Client and the Web Service to securely communicate. The protocol is based on the Needham-Schroeder protocol (seen in the theoretical course).

Web Services execute the protocol as it follows (in order to obtain a symmetric session key shared between each Web Service and the Authorisation Server) :

1. **Web Service → Authorisation Server** : $ID_{Web\ Service}, E_{AS}(ID_{Web\ Service}, r_1)$
2. **Authorisation Server → Web Service** : $E_{Web\ Service}(ID_{AS}, r_1, r_2)$
3. **Web Service → Authorisation Server** : r_2

If the verifications associated to these three steps succeed, the protocol distributes securely a session key as it follows :

4. **Authorisation Server → Web Service** : $E_{Web\ Service}(K_{AS\_WS}, r_1)$

If the verifications associated to this last step succeed, the Web Service accepts the received session key.

Registered users execute the protocol as it follows (here-under a user asks to gain access to the first Web Service) :

1. **Client → Authorisation Server** : $ID_{Client}, ID_{WS_1}, E_{AS}(ID_{Client}, ID_{WS_1}, r_3)$
2. **Authorisation Server → Client** : $E_{Client}(ID_{AS}, ID_{WS_1}, r_3, r_4)$
3. **Client → Authorisation Server** : r_4

If the verifications associated to these three steps succeed, and if, on the basis of an access control list that you have to manage, the Client can have access to the asked Web Service, the protocol distributes securely a session key $K_{C\_WS_1}$ with a given cryptoperiod [1] t, as it follows :

4. **Authorisation Server → Client** : $E_{Client}(K_{C\_WS_1}, t, r_3)$
5. **Authorisation Server → Web Service 1** : $ID_{AS}, E_{K_{AS\_WS_1}}(ID_{Client}, K_{C\_WS_1}, t)$

If the verifications associated to these two last steps succeed, the Client can access the Web Service :

6. **Client → Web Service 1** : $ID_{Client}, E_{K_{C\_WS_1}}(request)$
7. **Web Service 1 → Client** : *service*

Web interface of the Authorisation Server

The Authorisation Server needs a web interface to perform the following administration tasks :

1. *"A cryptoperiod is the time span during which a specific cryptographic key is authorized for use"*, see http ://en.wikipedia.org/wiki/Cryptoperiod.

– registration of new clients ;

– management of the access control list :

– distribution of the RSA public/private key pairs to the registered clients ;

– creation and distribution of the RSA public/private key pairs to Web Services ;

– revocation of keys.

Realization of the project and deliverables

The project must be done by groups of four students (the same than the first project).

You have to use PHP, OpenSSL, MySQL and Java.

All the needed files and information must be stored encrypted (on SQL servers or other). The communications and connections for the web interface must be made through encrypted channels (https).

The security choices for the development of those services must be lead by your threat model.

For the certificate used for the https access of the web interface, you will use an auto-signed certificate from OpenSSL as shown during the exercise course.

You will also generate the pairs of RSA keys for the protocol from OpenSSL.

The user interface for the administrator account must be usable, but must not necessarily look nice.

Some parts of the system have voluntarily not been completely described in this document in order to let you investigate how to implement them securely and efficiently.

Your report has to mention details about your choices of implementation, the pros and cons of these choices, and a complete threat modeling .

Do not forget to mention your names and your group number in the report. You have also to provide all the files needed to install and configure a complete operational Web Service.

Your complete report has to be submitted to the secretariat of the computer science department (2N8.104).

You have also to send an electronic version of your project to *infof405@lit.ulb.ac.be*. The object of the email must be "[INFO-F405] Project 2 Group x", where x is your group number. Your report, the implemented project and the different files must be compressed within a zip file that will be attached to this email. The name of your zip file must be "Secu+group number" (Secu3.zip for example).

Oral defenses will be organized during the examination session of January. The place and the date will be communicated later.

Project delivery instructions

To respect scrupulously !

1. Your project must include **your name** and **your group number**.
2. Your project must be **typewritten**. Handwritten project will **not be corrected**.
3. Your implementation must be **commented**.
4. Your must respect the constraints of programming languages and libraries.
5. You must respect the following terms :
   - Deadline : **The 2nd december 2013 before 4pm**
   - Where : **"Student" secretariat of the computer science department, local 2N8.104**

   The secretariat closes at 4pm. **After 4pm**, the project will be considered **overdue**, and the project will be scored 0. This constraint is also valid for the asked email.