EIT060 2nd project

Tommy Olofsson <ada09tol@student.lu.se>
Erik Westrup <ada09eww@student.lu.se>
Gustaf Waldermarson <ada09gwa@student.lu.se>
Erik Jansson <ada09eja@student.lu.se>

1 Description of the system

The system should handle medical records for patients. It consists of multiple entities including patients, doctors, and agencies.

The entities are as follows:

- Patient: A patient can have one or more records in the system and he will at all time have read access to them. He will have to use private and public keys and a software client to obtain them from a server.
- Doctor: A doctor works at a division and will have read, write and create access for records to all patients registered to him. Also all records associated with the same division will be readable.
- Nurse: A nurse works under a division will have read and write access to records for records to all patients registered to him/her. Also all records associated with the same division will be readable.
- Medical record: A medical record is stored on a server. Each record is assosiacted with one patient, one doctor, one nurse and one division.
- Agency: An agency will have the ability to read and delete records at will.
- Certificate Authority: All certificates are signed by a root CA and every entety trusts this CA, this CA is separate from the hospital.
- The server sits in a locked room and it accessible only by trusted technicans. Is has an audit log containing all access to the patient records.

2 Setup process

The first step in setting the system up (appart from server installation and hardware distribution) is to generate a keypair and from this a Certificate Signing Request (CSR) for the public key. The CSR is transfered over sneakernet to the CA where it is signed and take back and stored.

A patient will need to obtain the client software from the "socialstyrelsen" office. They will there get their signed certificate stored together with the client on some form of read only memory (e.g compact disc or thumbdrive).

In a further extension of this system, it should be possible to add new doctors and nurses. This, however, is out of the scope of this project.

3 Protocol

3.1 Read

Read will return the requested record if the user is

- Patient: A patient can read a specific record with this command by supplying a record id. Returns the record or error if not found.
- Doctor, Nurse, Agency: A parameter specifying which record to obtain is needed. Returns the record or an error if not found or access denied.

3.2 Append

- Patient: Access always denied with a message "You shall not pass!".
- Doctor, Nurse: Two parameters with record id and text to append is submitted. Responds with success or denied. Note that a doctor could mistype the id and accidentally write to another patients journal. A final product would include a GUI which would eliminate this problem.
- Agency: Access always denied with a message "You shall not pass!".

3.3 Create

Creates a new record for a patient.

- Patient: Access always denied with a message "You shall not pass!".
- Doctor: Parameters are patient id, nurse.
- Nurse: Access always denied with a message "You shall not pass!".
- Agency: Access always denied with a message "You shall not pass!".

3.4 Delete

Removes a particular record.

- Patient: Access always denied with a message "You shall not pass!".
- Doctor: Access always denied with a message "You shall not pass!".
- Nurse: Access always denied with a message "You shall not pass!".
- Agency: Parameter with journal id to delete. Returns success.

3.5 List

- Patient: If called by a patient with no parameters, it will list all records pertaining to the patient.
- Doctor, Nurse, Agency: If called by a doctor or a nurse with a patient id as a parameter, it will list all records belonging to that patient. If no parameter is specified, an error is displayed.

4 SSL/TLS

In our system we are using SSL/TLS-technology for authentication and establishment of a secure connection between the communicating parties. The Secure Socket Layer

[1] [2]

[1] Computer Security 3ed, Dieter Gollman, Wiley, page 310-314

[2] Java®Secure Socket Extension (JSSE) Reference Guide, accessed 2012-02-09, http://docs.oracle.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html