Implementing a Health Information Exchange (HIE) System

Course Project, CS463 Spring 2011

1 Introduction

The goal of this project is for you to develop a working Secure Health Information Exchange System (HIE). The HIE system allows various parties in the health-care system to exchange information securely. It also allows a later auditing of these interactions.

The HIE System has different kinds of users. You will need to support three distinct kinds of users in the system.

- **Patients:** This consists of individual patients who are part of the HIE system. The data corresponding to a patient is stored in an Electronic Health Record (EHR). Each EHR corresponds to a single patient in the system. A patient accesses his health records using a Personal Health Record (PHR) provider. We assume that patients access the PHR using a web based access.
- **Healthcare Professionals:** This category of users consists of professionals in the system. In this project we assume that there are two kinds of professionals; doctors and nurses. They access the EHR database using a Health Information Service Provider (HISP). You can assume that professionals access the EHR data either using a web-based access or using standalone applications.
- **Researchers:** This category of users accesses anonymized data of a large number of patients in the database. They access the data using a Research Aggregator (RA).

Besides the users and their agents, we also have two other components in our HIE System.

- **Data Store:** A Data Store (DS) is a third party agent whose whole responsibility is to store encrypted EHR records (and their metadata). Multiple agents (such as PHR, HISP, RA) can save their data at a single DS. Note that the DS should be running on a *different* machine compared to the agents. You can store the individual EHRrs in a database or as flat files (your performance may be affected by your design choices).
- **Key Store:** A Key Store (KS) stores the keys of the individual EHR records (as well the key-id to key mapping).

For the sake of ease of implementation, we have divided the project delivery into two distinct phases. The requirements for Phase II builds upon the deliverables on Phase I. Therefore, please consider the requirements for Phase II while writing code for Phase I.

2 Requirements: Phase I

Phase I is due after the spring break (March 28th 2011, Monday).

In Phase I, you are required to support only three components. You need to design and implement PHR agent, HISP agent and DS. You may assume that the KS is locally accessible to both PHR agent and HISP agent (with no access control done required). Note that you need at least two machines to make this design to work (the Data Store must be running on a different machine compared to the HISP,PHR agents).

You need to support the following interactions:

- Doctors should be able create EHR records for patients.
- Doctors with access to EHR can allow other other doctors to view/modify the EHR records. He/She may give read only (or read-write) access to Nurses in the system.
- Users can only read the EHR records, not modify them.
- The originating Doctor (of an EHR) may revoke access to nurses/doctors who have access to the EHR.
- Make sure that the protocol used between DS and agents (HISP, PHR) does not compromise confidentiality of either patients or doctors. Also, ensure that it is resistant to replay attacks.

3 Phase II

Phase II is due on the last day of instruction (May 4th 2011, Wednesday).

In Phase II, you are required to support everything in Phase I as well as KS and RA. Furthermore, you are required to ensure that the KS, DS and agents are on different machines (which means at least three machines assuming all the agents are running on a single machine). You also need to support auditing of the interactions at HISP and RA. You are *not* required to support any client for analyzing the audit logs (a simple file listing the user/type of interaction/time is enough)

You also need to support the following interactions:

- A researcher can access de-identified EHR records. Note that you need to support easy access of potentially thousands of records per researcher.
- Support access control policies at RA for a given researcher. For instance, one researcher may only have access to age, diagnosis of a patient; whereas another researcher may be able to access prescription and the physician data as well. For extra credit, you can be creative and support more interesting access control policies for extra credit (for instance, policies that give access depending on the data values as well; say only allow access to patients between age 20,50).
- Note that you can cache de-identified EHR data at RA. This may be done for efficiency reasons. However you will need to ensure that you keep the data in sync at periodic intervals.

4 Deliverables

For both Phase I and Phase II, you are required to submit a report and code (in Java). The report should include details about the design of your code as well as some discussion about the various security mechanisms that you have used. In particular, any messaging protocols that you have used should be documented. Document any access control policy formats that you have used as well.

Some pointers about your deliverables.

- The report should be no more than 3 pages (11pt) (4 for Phase II). It should be in pdf.
- The code should compile and run on linux (please contact us if you cannot meet this requirement)
- Include a readme.txt about how to perform some of the common actions in the system.
- Document your code.
- In the project, make sure you mention the contribution of each individual student (a line each).
- For Phase II, include some evaluation of the time taken for some common operations.

5 Extra Credit

You have various ways of getting extra credit on your project.

- A good portion of your grading will depend on the report. Make sure you include details regarding
  - Protocols used for communication between various entities in the system. In particular, how are you taking care of replay attacks? Is your system susceptible to DoS attacks? What is the protocol overhead on the messages?
  - What attack model are you considering in the system? Are there malicious insiders or only third-party intruders?
- Perform good evaluation of various common operations such as EHR creation, EHR deletion, Access control overhead etc.,(How long does it take? How many messages?)
- Support multiple agents (multiple HISP, PHRs, multiple DS)
- Support single-sign on
- Better ways of ensuring privacy when releasing de-identified data.

You can also contact us to discuss any extra additions you may want to add for extra credit.

Good Luck!