CSCI-531 Spring 2023 Semester Project

Designing a Secure Decentralized Audit System

1. The nature of the assignment

The semester project gives each student the opportunity to use and illustrate the concepts from the course in an applied manner. You are to report the design, analysis, and prototype of a secure electronic voting system.

You can work as **a team of two students or you can work alone**. Completion of the semester project is to be an independent effort for each team.

The code and project report you submit must be entirely your team's own work, and you are bound by the Honor Code. You may discuss the conceptualization of the project, but you may not look at any part of someone else's solution or collaborate with anyone other than your partner. You may consult published references, provided that you appropriately cite them.

Attempting to benefit from work of other students, past or present and similar behavior that defeats the intent of an assignment is unacceptable to the University. Such behavior will be treated as a violation of USC academic integrity standards, which are summarized in the on-line tutorial available at http://www.usc.edu/libraries/about/reference/tutorials/academic_integrity/index.php

In this project, you will have an option to either concentrate more on the on the detailed design of the system or on the implementation of your system prototype.

Prepare a report in PDF format with a font size of 12 points, single-spaced, single column. Not less than 10 and no more than 20 pages. Figures, tables, screenshots and the like are not included in the 20-page maximum page count. You will not be penalized for exceeding the page limit, but text beyond the 20-page limit will not be considered in grading.

On the first page of the report clearly state **the name of your partner**. Upload your report **and your code** in electronic form (as a zip file) to D2L "Semester Project" folder. For each team, submit the required documents only once. For the second team member submit a file in PDF format that contains only the name of the parter.

Note: this project was developed specifically for this course, you are not allowed to make the code for this project public.

2. Project description

Electronic health record (EHR) systems have gradually replaced traditional paper-based health record systems in the United States. Audit logs serve multiple functional and regulatory purposes in EHR systems. When patient records are accessed for some reason, the history of all such events must be recorded in a log file for later audit on access histories. The log file is used for reconstructing the past state of medical records, and it can be used as a legal evidence in medical malpractice cases.

Your job is to produce design and prototype a system that meets the following goals:

1. *Privacy*. Patient privacy should be maintained. Unauthorized entities should not be able to access audit records.

2. *Identification and authorization*. All system users must be identified and authenticated. All requests to access the audit data should be authorized.

3. *Queries.* Only authorized entities should be able to query audit records.

Immutability. No one should be able to delete or change **existing** audit records without detection. Any modifications/deletions of the audit records should be detected and reported. You can focus on attackers who are internal to the system modifying the audit data after it has already been received by the system. Note that you do not need to protect against modification, it is sufficient to just detect and report any unauthorized modifications to the audit data.

4. *Decentralization*. The system should not rely on a single trusted entity to support immutability.

The description of the system is deliberately underspecified so that you have the intellectual freedom to consider various possibilities for how such a system should operate.

Some useful references are listed on the last page. You may use web sites and other aids to help you with this project, be sure to **list your references** in your report.

3. Deliverables

This project will let you explore the implementation of a simplified secure electronic audit system. You can limit your system to ten patients and three audit companies who attempt to access the audit data.

You do not need to implement a component that controlls access to the EHR data. However, you will need to implement a component that generates the audit data when the EHR data is accessed. Jason format for audit records is recommended.

An audit record should include the following:

- Date and time of logged event
- Patient ID whose record was accessed
- User ID who performed the logged event
- Action type (create, delete, change, query, print, or copy)

Implementation of a practical audit system should be scalable and distributed and run over the network. You are welcome to implement a scalable, distributed, decentralized solution with a web-based interface. You will get **extra points** for such implementation.

However, we do not expect everyone interested in applied cryptography to have network programming experience. For this reason, your implementation can run on a single machine and we do not require socket programming. Message exchange can be implemented by writing and reading to/from a file. You can implement the required functionality as a library that can be called from a client or a server. However, you need to implement client and server stubs to demonstrate the required functionality. In data exchange scenarios there are sender/client and receiver/server roles, you may implement both sides by respective client.py and server.py.

You need to prototype some of the system components discussed in your system design. You can use a programming language of your choice. You are also allowed to use code developed by others. Some examples:

- Flask webserver
- Harmony
- Hyperledger

You must clearly specify which parts of your implementation were developed by you and which parts you acquired from some sources (e.g., GitHub). You must provide explicit references to the code you obtained from the external sources (including all imported packages) as well as description of functionality the external code implements. Do not use any proprietary code in this project!

Required prototype implementation (**you need to implement this functionality even if you select option 1**):

- Implement routines that support “Privacy” goal. In particular, confidentiality and integrity protection of sensitive data in transit and at rest. You should identify which data managed by your system requires such protection.
- Implement routines to support “Queries” goal. Patients can query the system to monitor usage of **only their own** EHR data. This means that patients should be able to issue queries over the audit data to see

who accessed their EHR data. Audit companies can query the system to monitor usage of EHR data of **all** patients.

- Implement routines to support "Immutability" goal. Demonstrate how your system enforces immutability buy implementing a scenario where an attacker tampers with some audit data and the system reports the attack. For this task just detection of tempering is enough.

You must provide a comprehensive written description of the design and implementation of your system. The report should include the following:

1. **System workflow**
   - ✓ Describe a general workflow for your system: the tasks to be accomplished and steps that are necessary to complete a specific task
2. **System architecture**
   - ✓ Describe the system components (e.g., authentication server, audit server, query server, etc.) and their functionality
   - ✓ Describe the communication patterns among the components (e.g., requests and responses)
3. **Cryptographic components**
   - ✓ Discuss appropriate choice cryptographic primitives to ensure the system supports the goals outlined above
   - ✓ Describe the concrete encryption schemes and key management approaches used in your system
4. **How system meets the outlined requirements**
   - ✓ Clearly describe (in writing) and demonstrate (in your demo) how your system **meets the five goals** discussed above
5. **System assumptions and limitations**
   - ✓ Clearly state your assumptions and discuss limitations of the system
     - In particular, discuss security challenges were not addressed
   - ✓ We will not require rigorous proofs of correctness.
6. **Implementation**
   - ✓ Provide a detailed written description of the design of your programs and a screen capture of a session demonstrating that your programs work.
   - ✓ Clearly explain how your programs can be executed, describe expected inputs and outputs.
   - ✓ Submit your code
7. **Demo recording**
   - ✓ Demonstrate how your system works. Save the video of your demo in MP4 format.

You can choose **one of the two options** to complete the project:

1. Explore applicability of various cryptographic schemes (e.g., homomorphic encryption, attribute-based encryption, zero knowledge proofs, etc.) to improve the system security properties. You need to discuss **in depth** at least two cryptographic schemes not studied in class and the additional security benefits they provide. Give **specific examples** how this additional functionality will improve your system.

2. Implement **extended** prototype. In particular:

   - ✓ Implement routines to support "Identification and authorization" goal.
   - ✓ Implement routines to support "Decentralization" goal. This means that there is no single entity (organization) that controls the audit logs. Blockchain-based technology works well for such systems.

4. Grading

The total of 100 points for the project will be allocated for **Option 1** as follows:

1. [5 points] System workflow
2. [15 points] System architecture
3. [25 points] Discussion of cryptographic components
4. [10 points] Discussion of how system meets the requirements

5. [5 points] Discussion of assumptions and system limitations
6. [30 points] Implementation
7. [10 points] Demo recording
8. **Extra 5% of the total score or this class** for realistic implementation: scalable, distributed, and decentralized with web-based interface.

The total of 100 points for the project will be allocated for **Option 2** as follows:

1. [5 points] System workflow
2. [15 points] System architecture
3. [5 points] Discussion of cryptographic components
4. [10 points] Discussion of how system meets the requirements
5. [5 points] Discussion of assumptions and system limitations
6. [50 points] Implementation
7. [10 points] Demo recording
8. **Extra 5% of the total score or this class** for realistic implementation: scalable, distributed, and decentralized with a web-based interface.

References

Charalampos Stamatellis, Pavlos Papadopoulos, Nikolaos Pitropakis, Sokratis Katsikas, William J Buchanan, A Privacy-Preserving Healthcare Framework Using Hyperledger Fabric, arXiv - CS - Cryptography and Security, 2020.

D Tith, JS Lee, H Suzuki, W Wijesundara, N Taira, T Obi, N Ohyama, Application of Blockchain to Maintaining Patient Records in Electronic Health Record for Enhanced Privacy, Scalability, and Availability, Healthcare Informatics Research 26 (1), 3-12, 2020.

M. M. Madine et al., Blockchain for Giving Patients Control Over Their Medical Records, in IEEE Access, vol. 8, 2020.