Cryptography Progress Report

On

Proposing a Secure System for usage of Electronic Health Records

Submitted To: Dr Ashutosh Bhatia

As a partial fulfilment for the cryptography Course BITS F453 for 1st Semester 2019-2020, BITS Pilani, Pilani Campus.

Submitted By:
Sanjeev Singla
2017A7PS0152P

Project Proposal (As submitted before for reference):

Cryptography Project Proposal

Objective: To propose a solution to the below problem Statement.

Problem Statement: There are 2 mobile applications - Doctor Application and Patient App. The users use the apps to maintain their electronic health records. Patients organise all their documents through their app. They can choose to share their medical data with the doctor when they go for check-up. Doctors can use the doctor app to view their patients records and if patient allows can share those records with his/her colleagues if required. I have to design the securities behind these exchanges in the most efficient manner possible.

Plan:

(1) Design the basic structure and features offered by the application.

For this I have chosen 2 live application offered by Apollo and roundglass to get the abstract structure of the app.

This will be done inline the current ehr standards given by https://www.nhp.gov.in/ehr_standards_mtl_mtl

(2) Study of various techniques used for designing the security features.

example: In case the application uses 3rd party servers for storing the records(like AWS etc) , I will explore techniques like access - control mechanisms , online encryption and homomorphic encryption.

For studying about these techniques, I will use the textbook and resources like https://www.cybrary.it/ . I will use research papers I have collected for the same if time permits.

(3) Propose a system to ensure security of the EHR. This solution will try to answer questions like:

(1) How will the access key be shared?

(2) Does doctor get to make its own copy of the records?

(3) How to add more information to the records? Who will have the access?

(4) What if patient later decides to take back the access?

(5) What happens in case the user forgets the password (secret key)?

Outcome: A system design for the security of Electronic Health Records.

PROGRESS REPORT:

Objective: To propose a secure system for usage of electronic Health Records according the below **problem Statement:**

There are 2 mobile applications - Doctor Application and Patient App. The users use the apps to maintain their electronic health records. Patients organise all their documents through their app. They can choose to share their medical data with the doctor when they go for check-up. Doctors can use the doctor app to view their patients records and if patient allows can share those records with his/her colleagues if required. I have to design the securities behind these exchanges in the most efficient manner possible.

Motivation: My Practice School – I was at a healthcare firm developing similar mobile applications as above. By working on them, I realised there are practically no laws regarding the handling of the Electronic Health Records. As announced by the Government of India, a new bill is in making to tackle this issue. Therefore, I am attempting to design a secure system which can be used in such use cases.

Introduction:

The main aim of this project is to explore and design a security system for usage the Electronic Health Records. Healthcare Industry is booming lately especially because of the technical revolution in the current decade.

In this world of machine learning, data has surpassed oil in terms of value, but Medical data is still considered out of reach. This is mainly because of the stiff HIPPA regulation and sensitivity of the medical data in general.

This project is an attempt to develop systems for Applications for the medical domain which involve sharing of EHR as frequently as in a clinic / hospital.

For the Project I have used the Pallier Cypto-system which is partially homomorphic encryption scheme. The Paillier crypto system, invented by and named after Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing n-th residue classes is believed to be computationally difficult. The decisional composite residuosity assumption is the intractability hypothesis upon which this cryptosystem is based.

Methodology:

I divided the work into following 4 phases:

Phase 1: Research and Concepts:

This Phase includes the research and concept building required for the project.
Some of the concepts to be used:

(1) Homomorphic Encryption: This filed has recently gained attention due to the active need of workarounds around the HIPPA rules in the USA. These rules severely cripple any effect on applying the Deep Learning Models on any medical data.
(2) Reading about the current rules on the handling of Electronic Health Records. Due to the ambiguity in the Indian Laws, HIPPA rules are also taken into account.
(3) Access Control Mechanisms - This is required for verifying edit requests, change password, forgot password request.

Phase 2: App Design

This phase focuses on the abstract design of the application using the system. This is basically a sample use case for the system.

The goal here is not to code the application but to study its design for understanding the potential implementation constraints and possible loopholes.

I have used roundGlass reach and cross applications for this.
Main themes to be identified:

(1) Tech-stack example: Technologies used for web-development; storage services used like AWS etc.
(2) Design principles like sign in options, forgot password, options to share data with doctor, options to take access back, edit options, addition and deletion options etc.

Phase 3: Security Design

This phase deals with the Security design of the application. By security design, I mean the algorithms used on top of the application. This basically focuses on the homomorphic encryption techniques and their feasibility. This part also includes answering the questions like:

(1) How will the access key be shared?
(2) Does doctor get to make its own copy of the records?
(3) How to add more information to the records? Who will have the access?
(4) What if patient later decides to take back the access?
(5) What happens in case the user forgets the password (secret key)?

Phase 4: Coding the part

This phase deals with the coding part of the project. I will not be coding the application. The current plan is to simulate the design in a python script with following specs:

The current plan is to simulate the design in a python script with following specs:

(1) Doctor app and patient apps: Will be just two branches in a git repository
(2) Using git as the cloud and sync manager for the demo
(3) Git commits is used as edits and git pulls as request.
(4) Encryption and Decryption algorithms.

Work Progress:

Phase Wise:

Phase 1:

(1) For Homomorphic encryption , I have read parts of this paper : https://hal.archives-ouvertes.fr/tel-01918263/document

(2) As for access control mechanisms, I have not done any exhaustive research. This part will be done by this month end.

Completion: About 80%

Phase 2:

(1) Initially I was using two applications – roundGlass and Apollo. Since I didn't have a doctor's license, I was not able to explore it fully. Hence, I focused on the roundGlass application.

(2) The design structure is ready. The functionalities have been studied and incorporated in the design structure.

(3) I did some testing on online portals (I followed a YouTube tutorial for this)

Completion: About 90%

Phase 3:

(1) Answers to the posed questions have been articulated.

(2) The themes which were tried and tested.

(1) Using users' phone as the only the storage device.

(2) Maintaining only the transactions which occurred after a initial commit.

Completion: 80 %

Phase 4:

(1) Learnt python programming.

Sources : (1) Learn python the hard way (2) YouTube tutorials

(2) Learnt some python libraries like: Pyfhel ,Microsoft Seal ,Pyseal and paillier

Completion: 25%

Net Completion: 68.75 %

Outcome Based on Progress:

(1) Concepts involving Homomorphic Encryption.

(2) Python Programming and some relevant libraries.

(3) App Design Concepts and basis penetration testing concepts.

Remaining Work:

Phase wise:

Phase 1: Access Control Mechanisms

Phase 2: Some tech-stack related questions like AWS vs Azure and representation of the ideas formulated.

Phase 3: Some questions like – How often will the data by synced? How can we retrieve data in case of database failure? Are still unanswered.

Phase 4: Complete Coding is left. Some concepts like using command line command in python script are to be explored. I will try to include GUI if possible (I haven't learned GUI in python yet).

References:

(1) https://en.wikipedia.org/wiki/Paillier_cryptosystem

(2) https://pypi.org/project/pycryptodome/

(3) https://www.pycryptodome.org/en/latest/src/examples.html

(4) https://python-paillier.readthedocs.io/en/stable/phe.html

(5) https://www.nhp.gov.in/ehr_standards_mtl_mtl

(6) https://hal.archives-ouvertes.fr/tel-01918263/document

(7) https://www.youtube.com/playlist?list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU

(8) https://play.google.com/store/apps/details?id=glass.round.cross.doctor&hl=en_IN

(9) https://play.google.com/store/apps/details?id=glass.round.reach&hl=en_IN

(10) https://www.microsoft.com/en-us/research/project/microsoft-seal/

(11) https://github.com/Lab41/PySEAL

(12) https://github.com/ibarrond/Pyfhel