Secure e-personal health care system

By 16BCE2226 and 17BBT0151

Table of contents:

| Heading | Page number |
| --- | --- |
| Motivation | 3 |
| Problem statement | 3 |
| Aim | 3 |
| Objectives | 3 |
| Literature survey along with summary | 4 |
| Overview of architecture | 6 |
| Flowchart | 7 |
| Code | 8 |
| Conclusion | 13 |
| References | 13 |

Motivation:  
We live in a world where information is available in our fingertips. This has made life easy not only for the everyday consumer but also for healthcare industry as it means that they are able to get the crucial information that they need within a few seconds rather than waste time manually going through a paper trail. We believe that there is a need for a system that provides timely information while at the same time protecting the confidentiality of the patient.

Problem statement:

With the increasing amount of personal information being stored in electronic format, there is a need to protect the information from unauthorised access. A patient’s health record is a confidential data which if unprotected can cause embarrassment for the patient and could also endanger him. It is also his right to decide whom the information is shared with. Thus it is crucial to have a system that not only enables authorised personnel to access the data but also protect the data from unauthorised access.

Aim:

To design an healthcare database management system that ensures that unauthorised personnel can't access it. It should also enable quick and swift access for the authorised entities. It must also have provisions for the patient to check up on his own data. It should also contain an log book of who and all had accessed the data.

Objectives:

* Store the patient data in an encrypted format.
* A dean should be able to add authorised personnel(doctors).
* Doctors and dean should be able to view and add patient records hasle freely.
* Patient should be able to view his own record.
* Dean should be able to see which doctors accessed which patient record.

Literature survey:

Articles reviewed:

* <https://watermark.silverchair.com/3-2-139.pdf?token=AQECAHi208BE49Ooan9kkhW_Ercy7Dm3ZL_9Cf3qfKAc485ysgAAAiEwggIdBgkqhkiG9w0BBwagggIOMIICCgIBADCCAgMGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMI3RqgriZhK-gdH9AAgEQgIIB1KpC3ge_B3FS8tPXrFbKUElXHXq0LicMV5-6q46amI1o_McnQXnEZMwAbs2vIhccP7tB3xNX_24CwmTWxQuTZgEJ-Wr8U0Fzg93mcj_auUZbGQ5WU76_iZCBih6GdU6hAyhWOrI96l1xgQoHLJ-K58LmXAjrXGpJLZuOE7LHvVIzw47oPyfJRvcS2wCGGqxBHAzZ7CStNsuaE4qX13zBpiZLLH7GiHC8A2Nffs1cQVyCXgXPHLYDb0f9M00WjlCZ33zX7DkWc7kzeeWo9ftZhHEzvdWDM-F0kBH-vuE65vYD95TT161Po8PVYotgN5XWr0C0kzuB4aJeUABJ3vG4ROaGUfWcHkP7u4z2BZIWKPgoZ2USXISjEbPAamm4syKrPNTyBKI9414Ndx2ZRkjsyWZ3suRaEY7nS-wVACsBgvJNpJL8vWFo8-gmMAttsLQB7HgipTLpujh0B9f8ZZ9B_hrXP6JErKNyPO_vJsUeB5LS9-Ds20gcfjwppW_mK2OwdetJ5RaOGAlPMnWK4MbcVc_pMEVAl69QttWGEwjxxjkRJlP8mlEMVv-pMtAeWxbX1zZO8DVHcClq14Fhzdp0OG5-RB1l_AGfdHTZRdGn9ndcTy3Dbg>
* <http://www.library.armstrong.edu/eres/docs/eres/MHSA8625-1_MCADAMS/862504McAFuturePart1.pdf>
* <https://journalofethics.ama-assn.org/article/electronic-health-records-privacy-confidentiality-and-security/2012-09>
* <https://ac.els-cdn.com/S1532046412001864/1-s2.0-S1532046412001864-main.pdf?_tid=f564fb64-7ad5-47a7-ba91-3270616247f2&acdnat=1536475024_1b6efecdc8b07e286a386cb983c9e4ae>
* <https://ac.els-cdn.com/S1532046414001403/1-s2.0-S1532046414001403-main.pdf?_tid=43f7b85a-a3ae-4035-8da4-215c9b21c9f6&acdnat=1536475284_d2ed1adcf889dcaf1cbd2c062ad63842>
* <https://www.nature.com/articles/gim200876>
* <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3757992/>

Gaps identified:

* The system should allow all medical professionals who need access to the document a way to access the medical details of a patient without issue. The system should also record who and all has viewed the system.It should have a mechanism to check if the data has been modified to ensure integrity of the document.The use of a asymmetric encryption is recommended.
* The main security concerns in personal clinical information are authenticating the sender and receiver, to establish audit trails, to ensure integrity of message along with its confidentiality and to make sure no unauthorised modification occurs.It also talks about various tools we can use to achieve this including smart cards,biometric solutions and public key encryptions. Another major issue is the need to have up to date documents.The changing definition of privacy is also a key factor with regard to patients consent.
* The paper stresses the need that the patient must hold control over the decision about which information is shared with whom. It also discusses that the audit must note down who accessed the document along with a timestamp. The systems must also have a backup to ensure that if anything fails, information critical to save a patient's life is still available .
* The paper discusses the need to use digital signatures to verify authenticity.It also supports that there must be an override switch to ensure that during emergencies information is available
* The system must ensure that it protects the user from identity,membership and attribute disclosure. It must also ensure that the bit size is same for each document, so that you cannot link a document to a personal.
* Genetic information should be given higher priority for protection.The level of security should also depend on the type of data.
* The system could be made into a cloud based system. The access can be role based, each personal is given access to only a part of the electronic health record.Firewalls can be installed to make sure that unauthorized persons are able to implant false information inside the database.

Overview of model:

The encryption algorithm:

We are using fernet symmetric key encryption as it combines encryption along with message authentication and a time stamp to ensure that the data is protected and that it has not been modified.

Fernet uses 128-bit AES in CBC mode, with HMAC using SHA256 for authentication.

Entities in the system:

* Dean
* Doctors
* Patients

Storage type:  
The datas are stored in encrypted binary files

Identity verification mechanism:

The dean and the doctors will have their passwords stored as hashes to prevent anyone who looks at the doctors database to be able to determine their password. Each doctor's’ record will contain his ID and the hash of his password.

The patient identity is verified by his/her knowledge of their patient ID number and birthdate

Health record:

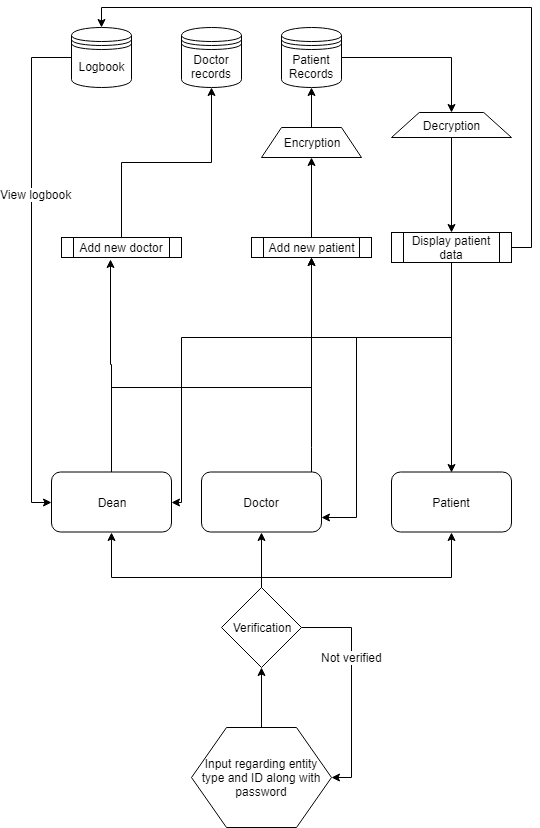
The health record of a patient will contain the following details

* ID number
* Name
* Blood type
* Gender
* Age
* Date of birth
* Height
* Weight
* Allergies
* Medications he/she is on
* Medical conditions he/she has
* Pathological test report
* Phone number
* Emergency phone number
* Remarks
* Details regarding who added the patient to the database and at what time he/she added

Logbook:

A log book will be maintained that records the details of who accessed which patient's record at what time. This is available only to the Dean of medicine.

Flowchart:



Programming language used:

The following code is run on python version 3.6.5

Code:

import os,pickle,hashlib

import time

from cryptography.fernet import Fernet as aes

def pat(T):

choice=1

while(choice!=0):

print("Enter 0 to exit")

print("Enter 1 to add patient record")

print("Enter 2 to display patient record")

choice=int(input("Enter choice: "))

M=0

if choice==1:

p1=medRecord()

p1.insRecord(T)

f=open("Patient records.bin","ab")

pickle.dump(p1,f)

f.close()

elif choice==2:

pid=input("Enter patient ID: ")

f=open("Patient records.bin","rb")

try:

while True:

p1=pickle.load(f)

if cipher.decrypt(p1.pid).decode()==pid:

if cipher.decrypt(p1.docID).decode()==T or T==("Dean"):

M=1

break

except EOFError:

print("Patient doesn't exist")

else:

if M==1:

p1.printRec()

g=open("logbook.bin","ab")

q=("Doctor ID " +str(T),str(time.asctime(time.localtime(time.time()))),"Patient ID :" + str(pid))

pickle.dump(q,g)

g.close()

else:

print("Access denied")

fernet\_key=b'rrm-9Rx\_5eeVLJQRehibrO\_AwjazFV\_mEb7RrzcHans='

cipher=aes(fernet\_key)

#patient

class medRecord:

def \_\_init\_\_(self):

self.pid=""

self.name=""

self.btype=""

self.gender=""

self.age=0

self.dob=""

self.height=0

self.weight=0

self.allergies=[]

self.medications=[]

self.conditions=[]

self.pTestRep=""

self.phone=""

self.emerno=""

self.remarks=[]

self.time=""

self.docID=""

def insRecord(self,d):

self.pid=cipher.encrypt(input("Enter patient ID: ").encode())

self.name=cipher.encrypt(input("Enter patient name: ").encode())

self.btype=cipher.encrypt(input("Enter patient blood type: ").encode())

self.gender=cipher.encrypt(input("Enter patient gender: ").encode())

self.age=cipher.encrypt(input("Enter patient age: ").encode())

self.dob=cipher.encrypt(input("Enter patient's DoB: ").encode())

self.height=cipher.encrypt(input("Enter patient's height: ").encode())

self.weight=cipher.encrypt(input("Enter patient's weight: ").encode())

n=int(input("Enter no: of allergies: "))

for i in range(0,n):

self.allergies+=[cipher.encrypt(input("Enter allergy: ").encode())]

n=int(input("Enter no: of mdedications: "))

for i in range(0,n):

self.medications+=[cipher.encrypt(input("Enter medication: ").encode())]

n=int(input("Enter no: of medical conditions: "))

for i in range(0,n):

self.conditions+=[cipher.encrypt(input("Enter medical condition: ").encode())]

self.pTestRep=cipher.encrypt(input("Enter pathological test report: ").encode())

self.phone=cipher.encrypt(input("Enter phone no.: ").encode())

self.emerno=cipher.encrypt(input("Emter emergency no.: ").encode())

n=int(input("Enter no: of remarks: "))

for i in range(0,n):

self.remarks+=[cipher.encrypt(input("Enter remarks: ").encode())]

self.time=cipher.encrypt(str(time.asctime(time.localtime(time.time()))).encode())

self.docID=cipher.encrypt(d.encode())

def printRec(self):

print("\n")

print("Patient ID: ",cipher.decrypt(self.pid).decode())

print("Patient name: ",cipher.decrypt(self.name).decode())

print("Patient blood type: ",cipher.decrypt(self.btype).decode())

print("Patient gender: ",cipher.decrypt(self.gender).decode())

print("Patient age: ",cipher.decrypt(self.age).decode())

print("Patient's DoB: ",cipher.decrypt(self.dob).decode())

print("Patient height: ",cipher.decrypt(self.height).decode())

print("Patient weight",cipher.decrypt(self.weight).decode())

print("Patient allergies:")

for i in self.allergies:

print("\t-",cipher.decrypt(i).decode())

print("Patient medications:")

for i in self.medications:

print("\t-",cipher.decrypt(i).decode())

print("Patient medical conditions:")

for i in self.conditions:

print("\t-",cipher.decrypt(i).decode())

print("Pathological test report: ",cipher.decrypt(self.pTestRep).decode())

print("Patient phone no.: ",cipher.decrypt(self.phone).decode())

print("Patient emergency no.: ",cipher.decrypt(self.emerno).decode())

print("Remarks:")

for i in self.remarks:

print("\t-",cipher.decrypt(i).decode())

print("Patient since : " ,cipher.decrypt(self.time).decode())

print("\n")

#Doctor class

class doc:

def \_init\_(self):

self.did=""

self.hash=""

def insrec(self):

self.did=input("Enter doctor ID: ")

P1=input("Enter new password: ")

self.hash=((hashlib.sha256(P1.encode())).hexdigest())

#Dean hash

DH="5fc4da096f5f9dbad39f3fb48072121d82c18efcafdb4604cd6a86c975928850"

c0=1

while c0!=0:

print("Enter 0 to exit")

print("Enter 1 for dean login")

print("Enter 2 for doctor login")

print("Enter 3 for patient login")

c0=int(input("Enter choice: "))

#Dean

if c0==1:

P=input("Enter password: ")

deanh=((hashlib.sha256(P.encode())).hexdigest())

#Dean basic

if deanh==DH:

print("Verified")

c1=1

while c1!=0:

print("Enter 0 to exit")

print("Enter 1 to add new doctor credentials")

print("Enter 2 to access patient record")

print("Enter 3 to access log book")

c1=int(input("Enter choice"))

if c1==2:

pat("Dean")

elif c1==3:

w=open("logbook.bin","rb")

try:

while True:

e=pickle.load(w)

print(e)

except EOFError:

w.close()

print()

elif c1==1:

d1=doc()

d1.insrec()

f=open("Doctor records.bin","ab")

pickle.dump(d1,f)

f.close()

else:

print("Access denied")

#Doctor

if c0==2:

did=input("Enter doctor ID: ")

f=open("Doctor records.bin","rb")

dip=input("Enter password :")

dih=((hashlib.sha256(dip.encode())).hexdigest())

try:

while True:

p1=pickle.load(f)

if p1.did==did:

if p1.hash==dih:

print("Verified")

pat(str(did))

else:

print("Access denied")

break

except EOFError:

print("Doctor does not exist")

#Patient

if c0==3:

pid=input("Enter patient ID: ")

pdob=input("Enter date of birth : ")

f=open("Patient records.bin","rb")

try:

while True:

p1=pickle.load(f)

if cipher.decrypt(p1.pid).decode()==pid:

if cipher.decrypt(p1.dob).decode()==pdob:

p1.printRec()

break

else:

print("Access denied")

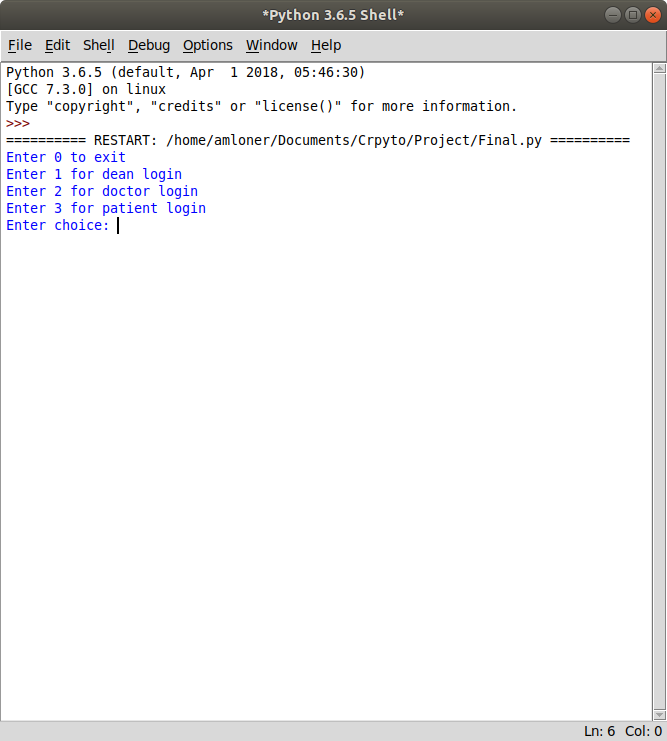
break

except EOFError:

print("Patient doesn't exist")

print("\n")

User interface(Output):



Conclusion:

We were able to create a program that could protect a patient's privacy while at the same time ensuring the information is available to the required personnel when required. It is vulnerable if the source code is exposed and modified.

Improvements proposed:

* Cloud based server development
* Mechanism to provide data for research without revealing patients personal data
* Giving dean the power to re-assign a patient to a new doctor

References:

* McGuire, A. L., Fisher, R., Cusenza, P., Hudson, K., Rothstein, M. A., McGraw, D., ... & Henley, D. E. (2008). Confidentiality, privacy, and security of genetic and genomic test information in electronic health records: points to consider. *Genetics in Medicine*, *10*(7), 495.
* Hesse, B. W., Nelson, D. E., Kreps, G. L., Croyle, R. T., Arora, N. K., Rimer, B. K., & Viswanath, K. (2005). Trust and sources of health information: the impact of the Internet and its implications for health care providers: findings from the first Health Information National Trends Survey. *Archives of internal medicine*, *165*(22), 2618-2624.
* Harman, L. B., Flite, C. A., & Bond, K. (2012). Electronic health records: privacy, confidentiality, and security. *Virtual Mentor*, *14*(9), 712.
* Fernández-Alemán, J. L., Señor, I. C., Lozoya, P. Á. O., & Toval, A. (2013). Security and privacy in electronic health records: A systematic literature review. *Journal of biomedical informatics*, *46*(3), 541-562.
* Gkoulalas-Divanis, A., Loukides, G., & Sun, J. (2014). Publishing data from electronic health records while preserving privacy: A survey of algorithms. *Journal of biomedical informatics*, *50*, 4-19.
* Rodrigues, J. J., De La Torre, I., Fernández, G., & López-Coronado, M. (2013). Analysis of the security and privacy requirements of cloud-based electronic health records systems. *Journal of medical Internet research*, *15*(8).