A
Dissertation
On

An Access control mechanism to ensure privacy using Attribute based encryption

Submitted in Partial Fulfillment of the Requirement
For the Award of the Degree of

Master of Technology

in

Computer Science and Engineering

by
Radhika Kuchhal
Roll No. 2K14/CSE/14

Under the Esteemed Guidance of

Ms Divyashikha Sethia
Assistant Professor

Computer Science & Engineering Department, DTU

COMPUTER SCIENCE & ENGINEERING DEPARTMENT

DELHI TECHNOLOGICAL UNIVERSITY

DELHI - 110042, INDIA

2014-2016

ABSTRACT

PHR contains patient's personal data which should be protected from unauthorized access. There are laws such as The Health Insurance Portability and Accountability Act (HIPAA) which protects the health records of an individual. In this work we propose a novel approach, of how a patient can maintain the privacy of his health records. We use Ciphertext-Policy Attribute-Based Encryption (CP-ABE) for confidentiality of records. To make the system more secure, we have features like revocation. Whenever a Medical Personnel is found doing malicious behavior, we can revocate him from the system. Also, if the patient loses his attribute key, we can revocate the patient so that unauthorized access to the records can be avoided. We have also implemented the concept of break the glass in our system. In case of emergency, normal access policy is overridden and emergency staff is able to access patient's health records. To provide better health services, we have delegation in our system. A doctor of one hospital may like to consult a doctor of another hospital. For consultation purpose the doctor will delegate part of his key to another doctor, so that the other doctor can also decrypt the records. At any time, delegated key can be revoked. In the system, all the medical personnel are not given write access. Whenever a Medical Personnel wants to write on patient's records, he has to prove his write access to the patient, only then the patient will accept the updated records. All the reads and writes to the health records are audited so that any malicious behavior can be detected and avoided in the future. Extensive experimental results are presented which show the efficiency of our proposed system.

KEYWORDS: PHR, HIPAA, CP-ABE, revocation, delegation, break the glass

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Ms Divyashikha Sethia for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to her for the support, advice and encouragement she provided without which the project could not have been a success.

Secondly, I am grateful to Dr. OP Verma, HOD, Computer Science & Engineering Department, DTU for his immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Radhika Kuchhal

University Roll no: 2K14/CSE/14

M.Tech (Computer Science & Engineering)

Department of Computer Science & Engineering

Delhi Technological University

Delhi - 110042

TABLE OF CONTENTS

LIST OF FIGURES AND TABLE

CHAPTER 1

INTRODUCTION

Everything is getting digitized, so is the medical records. A patient can maintain his medical history on a USB, phone in the form of PHR when he moves from one hospital to another. [1]

PHR contains patient's personal data which should be protected from unauthorized access. There are laws such as The Health Insurance Portability and Accountability Act (HIPAA) which protects the health records of an individual [2].

In this work we propose a novel approach, of how a patient can maintain the privacy of his health records. To achieve confidentiality, patient transmits the records to Medical Personnel in encrypted format. We use Ciphertext-Policy Attribute-Based Encryption (CP-ABE) for encrypting the records. CP-ABE is used because it provides one to many communication and allows fine grained access control.

To make the system more secure, we have revocation. Whenever a Medical Personnel is found doing malicious behavior, we can revocate him from the system, so that he doesn't harm any other patients. Also, if the patient loses his attribute key, we need to revocate the patient so that unauthorized access to the records can be avoided. Revocation is also useful when a person changes his attributes. In such a scenario we revocate the previous key and issue the key new key to Medical Personnel. In our system, we use the concept of linear secret sharing to revocate users.

We have implemented the concept of break the glass in our system. In case of emergency, normal access policy is overridden and emergency staff is able to access patient's health records. An emergency key is issued to the emergency staff, which is revoked after 1 hour, that is, emergency staff can access the records only for an hour, after which they have to request for the key again.

To provide better health services, we have features like delegation in our system. A doctor of one hospital may like to consult a doctor of another hospital. For consultation purpose the doctor will delegate part of his key to another doctor, so that the other doctor can also decrypt the records. At any time, delegated key can be revoked.

In the system, all the medical personnel are not given write access. Whenever a Medical Personnel wants to write on patient's records, he has to prove his write access to the patient, only then the patient will accept the updated records. All the reads and writes to the health records are audited so that any malicious behavior can be detected and avoided in the future.

1.1 BACKGROUND

Today everything is getting digitized. We have moved from retail to e-retail. Today even railways don't need a printed ticket, if we have an e-ticket in our phone. Same is the case with medical health records. Now a patient can maintain his medical history easily on USB, phone etc in the form of Personal Health Records(PHR) [1].

Earlier individuals and family maintained paper documents like laboratory reports, medications, clinical notes from medical personnel in envelopes or loose leaf binders. Whenever a patient moves from one Medical personnel to another, he carries these documents along. As the number of documents increases, maintenance of these documents becomes difficult and in emergency situations like accidents it becomes difficult to obtain these documents. So digitizing of health records is very important to provide better healthcare services.

PHR is loosely defined as patient health data repository that can serve individuals to manage their own health. They empowers patient to play a more active role in their wellness and self care. Information from both patients and healthcare providers can be consolidated in PHR. Emergency situations like natural disasters have highlighted the importance of PHR in response and recovery efforts. [3,4]

A PHR consist of patient's medical history including test results, medication list, diagnosis, physician interaction, immunization and more. PHR can also provide services like requesting appointments, asking billing questions, insurance related etc. [5] They can help in bridging the gap between patient and medical professional by connecting both of them.

There are different PHR models. Standalone or free standing PHR helps patient to organize and maintain their medical data. This medical data can be accessed anywhere at anytime and thus enable easy information sharing with healthcare providers. In Standalone PHR, patient has to manually enter and update his records. The other type is Integrated, interconnected or network based PHR consisting of patient information from variety of sources like Electronic Health Records (EHR) of a hospital, pharmacy data, different health sensors, insurance claims etc. In this type of PHR, patient is allowed to enter information only at selected areas in PHR, thus may eliminate manual entry of data and reduce medical errors, improve quality and eliminate duplication. [4,5]

1.2 MOTIVATION

Medical Records of a patient include some of the most important details about person's life. They consist of patient's mental and physical health, and can also include information on personal relationship and social behavior [6]. Studies show that people are less likely to take

treatments for mental health and substance abuse if confidentiality of health services is not maintained.[7]

If Medical records are not kept securely, insurance companies may access the health information and accordingly give the insurance plans. Employers can limit the job opportunities on the basis of person's health information [6]. According to a research, 550 individuals were refused employment or denied life insurance based on their genetic constitution [8]. All this will create fear of getting economically harmed, and will prevent individual from seeking quality care by storing his medical records in digitized form.

There are laws like Health Insurance Portability and Accountability Act of 1996 (HIPAA) which protects the health records of an individual. HIPAA allows only authorized individuals to see stored data and no unauthorized tampering of data. Also one can only see the data if they need to use it for an authorized purpose.[2]

So there are needs to store medical records in a secure way so that a patient can utilize the benefits of technology and at the same time does not harm his interest.

1.3 PROBLEM STATEMENT

A patient maintains his medical records on a portable device such as USB, mobile phone, etc. when he moves from one hospital to another. The health records of the patient need to be protected. The system should be able to handle following objectives.

- A Patient health records should be accessed only by the Medical Personnels of the hospitals he approves and no one else.
- In case of emergency, emergency staff should be able to access the patient's health records to provide health care services.
- If a Medical Personnel is found doing malicious behavior, his power should be taken, so that he is not able to harm any other person.
- If a patient loses his attribute key, his records should be protected against unauthorized access.
- For better healthcare services, Medical Personnel should be able to consult others for their suggestions.
- Everyone should not be able to write on a patient's health records. Only authorized personnel should be able to write.
- All the reads and writes to the health records should be audited for future.

1.4 ORGANIZATION OF THESIS

Chapter 1 gives the Introduction. It describes the background, motivation behind the work and defines the problem statement. Chapter 2 talks about the research already done in the area. Chapter 3 gives the proposed system architecture. Chapter 4 talks about how we have implemented the system. Chapter 5 shows the experimental results obtained. Chapter 6 concludes the work and talks about future work that can be done.

CHAPTER 2

LITERATURE SURVEY

2.1 CIPHERTEXT - POLICY ATTRIBUTE – BASED ENCRYPTION (CP-ABE)

Cryptography is one of the most common method to achieve confidentiality of the message. It is the science of transforming message to make them secure and immune to attacks. All cryptosystems can be categorized into symmetric and asymmetric cryptosystems.

In symmetric key cryptography transmitter and receiver either share the same pair of information (key) or they share pairs of related keys easily computable from each other. Transmitter using the key transforms the message, so that any unauthorized receiver cannot understand the message. Only the authorized receiver can decipher the message using the shared key [9]. The disadvantage of using symmetric key cryptography is the number of keys involved. If there are n users and each has to communicate with each other, then each user has to maintain (n-1) keys, that is there are ((n)(n-1))/2 keys in the system [10].

In an asymmetric cryptosystem transmitter and receiver have different keys and it is computationally unfeasible to compute one from another. In traditional public key infrastructure [11] each user has a public key and a private key. To encrypt data, transmitter obtains the public key of the receiver and encrypts the data using the public key. Receiver decrypts the data using his private key. By this way one to one communication is possible in a secured manner. If we want to send a message to many users, then we will have to encrypt the same message again and again using different public keys and then send the message separately to the corresponding receivers. This will increase the processing overhead and requires more network bandwidth. Broadcast encryption can be used to solve above problem. But in Broadcast encryption, the encryptor must know the user's list before encrypting the data [12]. That is, we need to identify all the decryptors before encrypting the data.

Identity based cryptosystems [13] allow users to communicate securely without exchanging public keys. In this scheme, instead of generating random pairs of public and private keys, the user uses any combination of his name, address, social security number, etc as a public key. The combination should be such which a user cannot deny later and the combination should be readily available to other parties. The secret key corresponding to the public key is generated by trusted key generation authority and is given to the user in the form a smart card. This scheme basically resembles an electronic mail system. If we know someone's E-mail ID, then we can send him the message and the message can be opened only by him and no one else. Similarly,

when user A wants to send message to user B, he signs the message with the secret key present in his smart card, encrypt the message by using B's public key which is easily available, adds his own identity and send the message to B. When B receives the message, he decrypt the message using his secret key and verifies the signature by using sender's identity as a verification key. So by this scheme, we don't need any certificate management procedure and also there is no need to transfer certificates among the users which reduces the network bandwidth.

Sahai and Waters [14] introduces new type of identity based encryption called Fuzzy Identity Based Encryption (Fuzzy IBE) in which identities are viewed as a set of descriptive attributes. Fuzzy IBE can be regarded as the first concept of Attribute Based Encryption. In fuzzy IBE, a user with a secret key for the identity w is able to decrypt ciphertext encrypted with identity w', if and only if, w and w' are within certain distance to each other as per some metric. So the system allows for some amount of error tolerance in the identities. Fuzzy IBE has two very interesting applications. The first one uses biometric identities, that is, we use the user's biometric like an iris scan as a user identity and then use this identity for encryption. Secondly, fuzzy IBE can be used in Attribute Based Encryption. In this identities are attributes that describe the user. So while encrypting the message we define the attributes that can decrypt it. In attribute based encryption, we also define the threshold value d, which tells at least how much overlap should be there between user attribute and ciphertext attributes. The value d allows for a certain amount of error tolerance in the identities match. For example, if a patient wants all the doctors of Hospital A and Hospital B should be able to decrypt his records. For this we assume d value of the system as 2. Then the patient would encrypt as {doctor, hospitalA, hospitalB}. In this case any user who has at least 2 overlapping attributes will be able to decrypt the records.

In attribute based schemes, attributes play a very important role. Subsequent researches can be categorized into Key-Policy Attribute-Based Encryption (KP-ABE) or Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [15].

In KP-ABE [16], the tree access structure, where interior nodes consist of AND and OR gates and leaves have attributes specified in the private key of the user, while the ciphertext simply consist of set of descriptive attributes. So for the user to decrypt the ciphertext, access tree present in the user's private key should satisfy the set of attributes. For example, a patient wants to protect his medical records and want only doctors of hospital A and hospital B should be able to decrypt the records. The Patient will encrypt the records using the attributes {doctor, hospitalA, hospitalB}. Now the doctor should have satisfying access policy in his secret key to decrypt the records.

In KP-ABE, encryptor selects only the descriptive attributes for the data, he has no control over who accesses the data. Like in the above example, although encryptor wants only doctors of hospital A and hospital B should be able to decrypt the records but, in reality any doctor can decrypt the records. A person should just have doctor attribute in his secret key. CP-ABE helps to resolve this issue.

In CP-ABE access tree is present with the ciphertext, whereas the private key has descriptive attributes. CP-ABE is more towards role based access control than KP-ABE. Behencourt [17] framework is the basic CP-ABE. Later on more versions were suggested to this framework.

The algorithms used in CP-ABE by Behencourt is described below:

Setup : In the setup phase, master key and public key are generated. The algorithm will choose a bilinear group G_0 with generator g and prime order p. It will also choose two random exponents α and β that belongs to Z_p .

The public key has following components:

$$\mathrm{PK} = \mathbb{G}_0, g, h = g^{\beta}, f = g^{1/\beta}, e(g,g)^{\alpha}$$

And master key MK is published as (β , g^{α}).

Encrypt (PK, M, T) : This algorithm encrypts the Message M under the tree structure T. Each non-leaf node of the tree represents a threshold gate and leaves represent the attributes. Let num_x be number of children of node x and k_x be its threshold value, then

- When $k_x = 1$, the threshold gate is OR gate.
- When $k_x = num_x$, the threshold gate is AND gate.
- Leaf nodes have $k_x = 1$.

To encrypt the message, algorithm chooses a polynomial q_x for each node x in the access tree T. The polynomial are chosen in the top down manner, starting from the root node R. The degree d_x of the polynomial of each node is set to $d_x = k_x - 1$.

A random element s is chosen belonging to Z_p and we algorithm $q_R(0) = s$. Here s is our secret we need to reconstruct using Shamir linear secret sharing. Then the algorithm chooses d_R other points to define the polynomial completely. For any other node except the root node

$q_x(0) = q_{parent(x)}(index(x))$

where,

parent(x) denote the parent of the node x in the tree

index(x) returns the number associated with the node x. Every children of a node is numbered 1 to num_x . index(x) returns this number.

d_x other points of the polynomial are chosen randomly.

Let, Y be the set of leaf nodes in T. The corresponding ciphertext is constructed by giving the access structure T and computing

$$\mathrm{CT} = \big(\mathcal{T},\ \tilde{C} = Me(g,g)^{\alpha s},\ C = h^s,$$
$$\forall y \in Y: \quad C_y = g^{q_y(0)}, C'_y = H(\mathsf{att}(y))^{q_y(0)}\big).$$

KeyGen (MK, S) : This algorithm takes the attribute set S and outputs the key that identifies the set. The algorithm first chooses a random r and then a random r_j for each attribute j in S, all belonging to Z_p . The private key SK of the user would be computed as

$$\mathrm{SK} = \big(D = g^{(\alpha+r)/\beta},$$
$$\forall j \in S: \quad D_j = g^r \cdot H(j)^{r_j}, D'_j = g^{r_j}\big).$$

Decrypt(CT, SK) : Decryption algorithm takes as input ciphertext CT and a private key associated with set S of attributes. In this algorithm we use a recursive algorithm DecryptNode(CT, SK, x). Here x is the node of the access tree.

If x is the leaf node, then we set i = att(x) . If i is present in the attribute set present in the private key SK then

$$\begin{aligned}\mathrm{DecryptNode}(\mathrm{CT}, \mathrm{SK}, x) &= \frac{e(D_i, C_x)}{e(D'_i, C'_x)} \\ &= \frac{e\left(g^r \cdot H(i)^{r_i}, h^{q_x(0)}\right)}{e(g^{r_i}, H(i)^{q_x(0)})} \\ &= e(g,g)^{r q_x(0)}.\end{aligned}$$

If i is not present in the attribute set of SK then

$$\mathrm{DecryptNode}(\mathrm{CT}, \mathrm{SK}, x) = \perp$$

If x is not a leaf node, then the function DecryptNode is called for all the children z of x and the result obtained from the function DecryptNode is stored in F_z . We compute them as

$$
\begin{aligned}
F_x &= \prod_{z \in S_x} F_z^{\Delta_{i,S'_x}(0)}, \qquad \text{where } \substack{i=\mathsf{index}(z) \\ S'_x=\{\mathsf{index}(z):z\in S_x\}} \\
&= \prod_{z \in S_x} \left(e(g,g)^{r \cdot q_z(0)}\right)^{\Delta_{i,S'_x}(0)} \\
&= \prod_{z \in S_x} \left(e(g,g)^{r \cdot q_{\mathsf{parent}(z)}(\mathsf{index}(z))}\right)^{\Delta_{i,S'_x}(0)} \text{ (by construction)} \\
&= \prod_{z \in S_x} e(g,g)^{r \cdot q_x(i) \cdot \Delta_{i,S'_x}(0)} \\
&= e(g,g)^{r \cdot q_x(0)} \qquad \text{(using polynomial interpolation)}
\end{aligned}
$$

The Decryption algorithm begins by calling DecryptNode on the root node which recursively calls its children and so on.

If the tree satisfies the attribute set S present in the private key SK, we get

$$\text{DecryptNode}(\text{CT}, \text{SK}, r) = \quad e(g,g)^{rq_R(0)} = e(g,g)^{rs}$$

The algorithm now decrypt by computing

$$\tilde{C}/(e(C,D)/A) = \tilde{C}/\left(e\left(h^s, g^{(\alpha+r)/\beta}\right)/e(g,g)^{rs}\right) = M.$$

2.2 REVOCATION

Revocation is taking back previously granted privilege. It is very necessary for any multiuser encryption scheme to handle malicious behaviors. The revocation mechanism of ABE schemes is more complicated than that of traditional public key cryptosystems or IBE schemes [15]. This is because in ABE schemes different users may hold the same secret key, and also same attribute may be possessed by many users which makes design of revocation mechanism very difficult.

In attribute based setting, revocation mechanism is divided into two kinds: user revocation and attribute revocation. In user revocation, user is revoked from the system. For example, if some doctor is found doing some malicious work, we may revoke him from the system [18]. Whereas in attribute revocation, a particular attribute is revoked and not the entire user [19]. There are mainly two ways to realize revocation, one is direct revocation and the other one is indirect revocation.

2.2.1 DIRECT REVOCATION

In direct revocation method, revocation is enforced by the sender who specifies the revocation list while encrypting the ciphertext. Direct revocation method does not involve any key update phase or proxy re-encryption like indirect method but, in this sender has to maintain current revocation list whose management could be troublesome. Several ABE schemes that use direct mode of revocation have been proposed.

Ostrovsky etal [20] scheme can be used for direct revocation of users. Their scheme supports negative clauses. So to revoke a person, one just adds the AND of negation of revoked user identities. For example, if we want all the doctors of Hospital A should be able to access the health records, except the doctor with identity BA123. To achieve this while encrypting the records we will define access policy as

(DOCTOR AND HOSPITALA) AND (NOT BA123)

In this scheme size of the policy scales with the number of users revoked, thus the encryption and decryption time also increases making it inefficient.

Attrapadung and Imai [21] suggested a method which combines broadcast encryption along with ABE. In this scheme each user has ID as a unique serial number in their private key. So while encrypting the data, sender associates the set S=U/R where, U is the universe of user indexes and R is the set of revoked users along with the usual attribute based access policy. Now for a user to decrypt the data, his key attributes should satisfy the access structure and ID ε S. So by this scheme users in S can decrypt or not is a don't care condition, which is evaluated on the basis of attribute part, but the users in R cannot decrypt surely.

2.2.2 INDIRECT REVOCATION

The indirect revocation enforces revocation by the authority. In this sender need not know the list of revoked users. Several schemes have been proposed based on indirect revocation method.

One of the method of indirect revocation is setting expiration time on each attribute. For example, instead of using attribute ComputerScience we will use the attribute ComputerScience:Dec31,2014 [22]. This denote that the attribute will expire at the end of 2014. This method has several shortcomings. In this attributes have an exact date therefore encrypting party and key authority should have agreement on this date [17], Secondly, there is a scalability problem. The key authority has to periodically announce the key update material at each time slot so that all the non revoked users can update their keys. Also periodic re-encryption of data would be required so that the time slot matches with that in the key. This could be bottleneck for both key authority and non revoked users. Also, this method fails to achieve backward and forward secrecy. Backward secrecy means that any user who comes to hold an attribute should be prevented from accessing the plaintext of the previous data exchanged before he holds the attribute. On the other hand, forward secrecy means that any user who drops an attribute should

be prevented from accessing the plaintext of the subsequent data exchanged after he drops the attribute, unless the other valid attributes that he is holding satisfy the access policy [23].

Another method to achieve revocation is proxy re-encryption. In proxy re-encryption scheme, a semi trusted proxy converts ciphertext for one user into ciphertext for another user, without seeing underlying plaintext. In this, proxy does not know the secret keys of any of the users and also does not learn anything about the plaintext during the conversion. The proxy uses proxy keys or re-encryption keys to perform translation [24]. Now, whenever attribute revocation event occurs, the authority redefines the master key component of revoked attributes and corresponding public key components are also updated. Now the data will be encrypted using new public key so, the secret key of the users should also be updated. Authority generates proxy re-key for updated master key component. With these proxy re-keys the proxy server is able to update the secret keys to the latest version. With proxy re-encryption we can also re-encrypt existing ciphertext [25].

2.2.3 HYBRID REVOCATION METHOD

Both direct and indirect revocation method have some advantages and disadvantages. Hybrid tries combining the advantages of both direct and indirect method.

Attrapadung and Imai [26] proposed a new system called Hybrid Revocable Attribute Based System. In this system, encryptor decides the revocation mode, direct or indirect on the fly when encrypting the message whereas decryptor possesses only one key, but will be able to decrypt ciphertext that were encrypted in either modes. In this method, each user has a unique ID in his private key. When encryptor decides direct mode, he specifies the revocation list R along with the access policy. Decryptor is able to decrypt if, attributes possessed by him satisfies the access policy and ID does not belongs to R. In case of indirect mode, encryptor along with access policy defines the present time slot t. While decrypting in this mode, decryptor obtains key update material U(R,t) from the authority. This key update material contains revocation list R for the time slot t. A decryptor is able to decrypt the text if, its attributes satisfy the access policy and ID does not belong to Revocation list R obtained from authority for time slot t.

2.3 BREAK THE GLASS

Break the glass is controlled overriding of access control restrictions and is particularly important in emergency conditions. For example, a person might meet an accident and is taken to hospital whose Medical Personnel are not authorized to decrypt the health records. In such a situation, emergency access is required to give proper treatment to the patient.

Li [27] suggested how break the glass can be achieved in case of medical emergency. In his framework, an emergency attribute is defined for emergency access. The PHR owner delegates his access right to the emergency department. In case of emergency, emergency staff contacts the emergency department to get its identity and emergency situation verified. After verification, emergency staff obtains the temporary read keys. These keys can be revoked by the patient once the emergency is over.

Tong [28] proposed a scheme to use (k,n) threshold signature for emergency access. In (k,n) threshold scheme, all n users have a signature share. To generate a valid signature at least k valid signatures are required. In their framework, the user encrypts the data using IBE. The record owner is the trusted dealer and he assigns his trustful people valid signature shares. Now in case of emergency access, the threshold number of trustful people of record owner would be contacted to generate an IBE decryption key. Here each share of the decryption key serves as the signature from the authorized party.

Brucker [29] incorporated emergency access using emergency attribute. In his framework depending upon severity of an emergency, we have different access policies. If we have no emergency condition, regular access policy (p_{reg}) would be followed. If we have low emergency situation represented by e_{low} then access policy p_{low} would be followed and so on. The policy would be represented as

(p_{reg}) OR $(e_{low}$ AND $p_{low})$ OR ……

Basically, whenever a subject request an access, the system checks if regular access policy can fulfill the request. In case the request is denied, system verifies if this request can be fulfilled by break the glass policy [30].

2.4 RELATED WORK

The concept of patient centric health information system was introduced by Szolovits et al. [31]. In this work, the author proposed an individual health information system, which integrates all health related information of an individual and maintains them. Companies like Google [32] and Microsoft [33] have also introduces their patient centric web based electronic health record systems. These systems allow patients to maintain copy of their health records, access them whenever needed, share the records with the medical professional based on the access policy.

In Narayan framework [34], all patients store their health records on the cloud which are encrypted by the symmetric key. A table consisting of entries is also stored on the cloud. Each entry corresponding to the file consist of metadata describing the file, its location, symmetric key all in encrypted format. It also has an access policy in plaintext which specifies who can decrypt the data and a search index for keywords within the encrypted file. A user that satisfies the access policy can decrypt the metadata entry associated with the file and decrypt the file using

the symmetric key. In the framework access policy is identified by the data owner himself. At any time he can add or remove the healthcare provider access to the data. Whenever a Medical professional has to view the health records, he sends the access request to the data owner. The data owner provides the access by updating the access policy. Now the medical professional can view the records. In case of write to the records, medical professional send the updated file to the patient encrypted using patient's public key. The patient can decrypt the updated file and upload it preserving the existing access policy. In this framework there is lot of burden on data owner himself. He himself is responsible for revoking the users and has to maintain the revocation list.

Ibrami [35] proposed to divide the system into two security domains, social domain and professional domain. Social domain consists of family members, friends and their key would be managed by patient himself. Professional domain has medical professionals like doctor, nurse, etc. They will get the key according to their roles from the key authority. The access policy defined by the patient will be of the form $P = P_1$ OR P_2 where P_1 is for social domain and P_2 for professional domain. Either P_1 or P_2 must be satisfied in order to decrypt the records.

Li [27] also proposed to divide the system into 2 domains, public and private domain. In public domain there are multiple attribute authorities, each governing disjoint subset of attributes. Users in public domain obtain their keys from these authorities. Data owner uses KP-ABE to manage the access right of the users in private domain and key to users in this domain is assigned by data owner himself. For break the glass access, each owner PHR access right is also delegated to emergency department. Emergency staff needs to contact the emergency department to verify the identity and emergency situation and obtain temporary read keys. The revocation in this system is handled by combining proxy encryption and lazy revocation. Whenever a user attribute is revoked, the attribute authority who governs the attribute, distribute the key update to all the non revoked users. Attribute authority can delegate updation of secret key of unrevoked users and ciphertext updation to the proxy, thereby reducing its burden.

Hur and Nor [23] proposed the scheme for fine grained user revocation. It is an efficient revocation method that employs Binary tree introduced by Boldyreva []. In this method data is encrypted twice. In initial decryption, access policy has attributes and enforces access control. The second encryption enforces user level access control per each attribute group. Whenever the receiver receives the data, he has to perform the decryption twice. Outer decryption handles the revocation, that is, any revoked user will not be able to decrypt this layer. Inner layer handles the access control. In this method whenever the revocation occurs, we need to re-encrypt only the second layer. The re-encryption of the second layer can be outsourced to proxy as the honest but curious proxy won't be able to learn anything about the message due to the first layer of encryption.

Ibraimi [36] scheme divides the secret key of the user into two shares, one share for the mediator and the other for the user. Each user along with the attributes also has a unique identifier I_u. To decrypt the data, the user contacts the mediator along with his identifier. The mediator checks his

attribute revocation list for the identifier. If any of the attribute is revoked related to the identifier, the mediator refuses to issue the decryption token for the revoked attributes. Without the token, the user cannot decrypt the ciphertext, therefore the attribute is revoked implicitly. The technique of splitting the attribute component of the secret key into two shares, enables us to revoke an attribute from a single user without affecting other users and also enable us to revoke the attribute from the system where all users are affected. In this method, immediate revocation is done, without re-encrypting or proxy encryption of the data.

CHAPTER 3

SYSTEM ARCHITECTURE

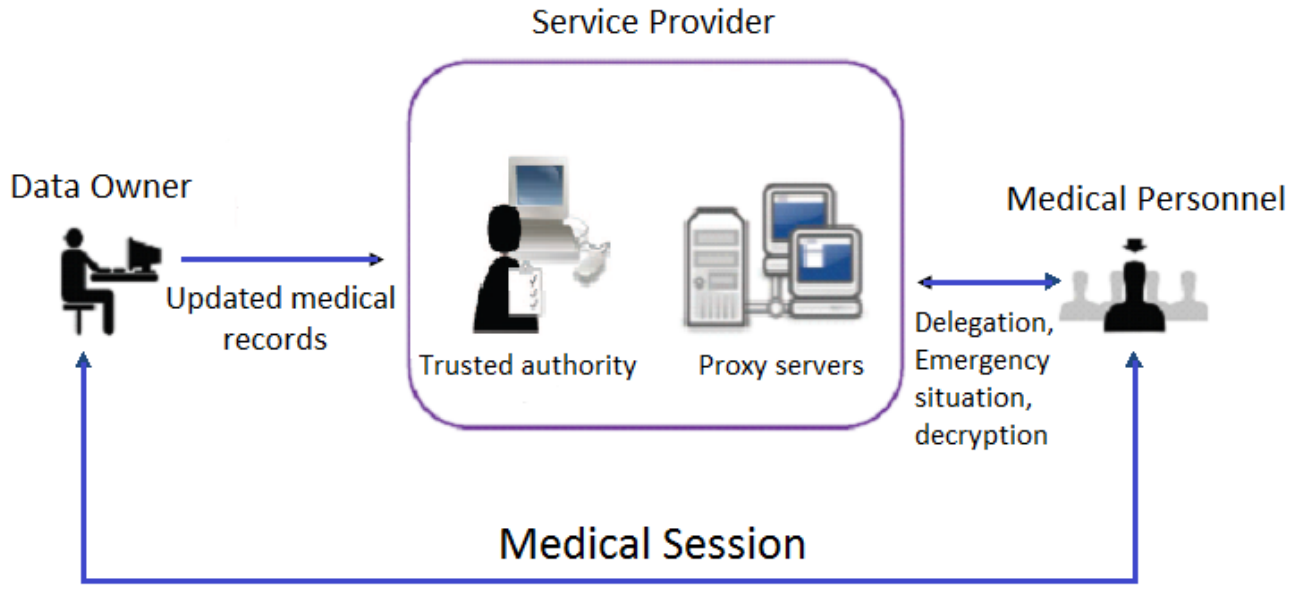

Figure 3.1: System architecture

The architecture of the system consists of the following entities:

- **Data owner** : This is a client who owns data, and wishes to establish a medical session with the Medical Personnel. The data owner is responsible for defining attribute policy and encrypting its data using this policy before the Medical Session. He is also responsible for updating the health records along with signature of Medical personnel and his own. Data owner also verifies the write access of the Medical Personnel before accepting the updated records. At any time, data owner can revoke his key.
- **Service Provider** : It is an entity that consist of Trusted Authority and proxy servers. Trusted authority is responsible for issuing attribute as well as private keys to the users in the system. It is also responsible for issuing emergency key in case of emergencies and revoke it after 1 hour from the time of issue. The trusted authority also looks after the delegation and revocation of malicious users. The authority has a database for storing the information and records for the users. Whereas the Proxy servers assist users in the decryption part. Whenever a decryption is to be performed by any user, he has to contact the proxy servers for obtaining the shares, without which decryption cannot be done. If

the user is revoked, he will not obtain appropriate shares and hence will not be able to decrypt the data.

- **Medical Personnel** : This is an entity who accesses the health records of the patients. Medical personnel possess a set of attributes in his attribute key. For decrypting or updating any records the attribute set in the key must satisfy the access policy with which the records are encrypted. Medical personnel can send the delegation request to the service provider to delegate part of his key to another personnel and can revoke the same at any time. In case of emergencies, he can obtain the emergency key. The Medical Personnel can be revoked by the service provider when found doing malicious behavior.

3.1 DATABASE DESIGN

The database consist of the following fields :

Table 3.1: Database design

Unique ID	LoginID	Password	KeyID	Emergency key ID	Source ID	Read logs	Write logs

- UniqueID : This is the unique ID that uniquely identifies the row in the database
- LoginID : This is set by the user while registration. LoginID is unique for every user.
- Password : This field is set by the user during registration. If he has right combination of login ID and password only then he can access the system.
- Key ID : Whenever the user registers, attribute key is issued to the user based on his role. The key ID is the id of the key issued to the user.
- Emergency Key ID : Whenever a key is issued to the user in case of emergency, the ID of the issued emergency key is stored here.
- Source ID : Whenever a user gets a delegated key, source ID stores the ID of the source from where he gets the delegated key.
- Read logs : This stores the ID of all those people who contacted the proxy for decryption of records. Along with ID, timestamp at which request is made is also stored.
- Write logs : This stores the ID along with signatures and timestamp of all those people who performed write on the records.

We use MongoDB for maintaining the database as it has following features which are missing in relational database [37] :

- Flexible data model **:** MongoDB has a flexible data model which enables applications to be easily build and evolve. The data of any structure can be combined and stored.
- High performance : MongoDB has support for embedded data models which reduces I/O activity on database system. It also has queries which supports faster queries.
- Rich Query language : It has rich query language to easily perform read and write operations.
- High availability : MongoDB has provisions for automatic failover and data redundancy.
- Horizontal Scalability : The data is distributed across cluster of machines by Sharding.
- The Nexus Architecture : MongoDB's architecture is based on combining the critical capabilities of relational databases with the NoSQL technologies.

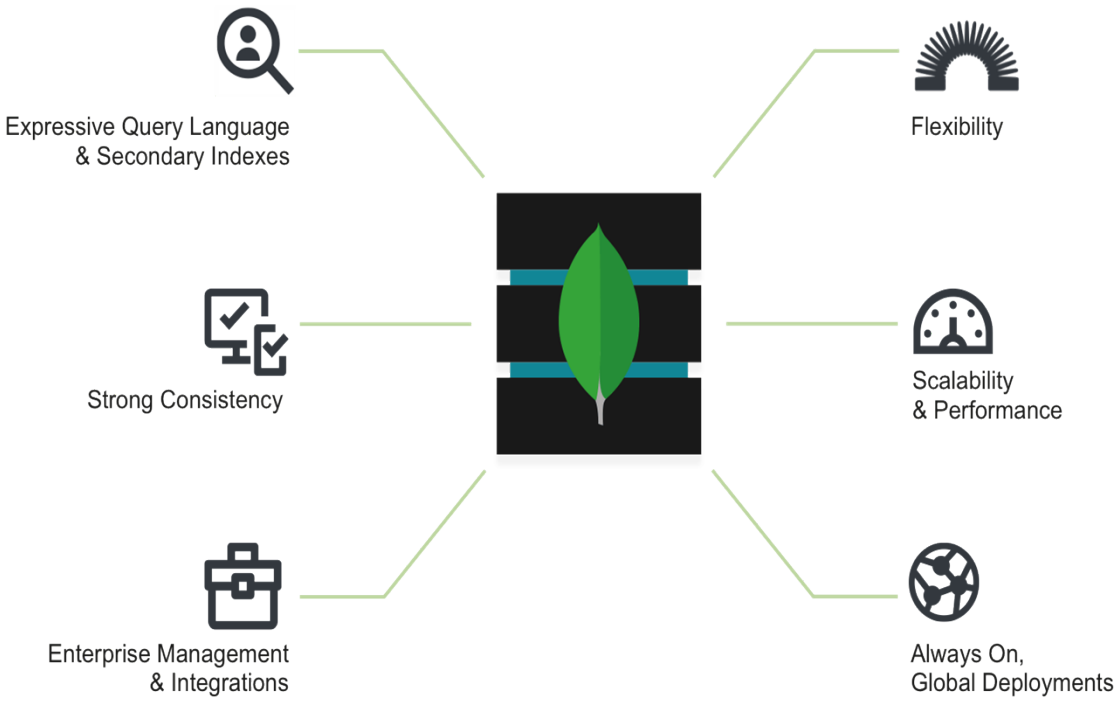

Figure 3.2 : MongoDB features

3.2 TOOLKIT USED

We use the toolkit provided by the [38] in our implementation. This toolkit has been derived from Bethencourt CP-ABE. In this we can revoke maximum t users, the total number of users in the system is not limited. Different functions used in the toolkit are:

Setup : The Key authority generates the polynomial of degree t, where t is the maximum number of users that can be revoked. In this P(0) is the secret to be used for revocation. α and β that belongs to Z_p are randomly chosen. The master and public key are defined as

$$PK = \mathbb{G}_1, \mathbb{G}_2, g_1, g_2, h = g_1^{\beta}, e(g_1, g_2)^{\alpha}$$
$$MK = \beta, g_2^{\alpha}, P$$

Encrypt(PK, M, T) : This algorithm works in the same way as that of basic CP-ABE. The algorithm encrypts the message M under the access structure T.

$$CT = (\tau, \tilde{C} = Me(g_1, g_2)^{\alpha s}, C = h^s = g_1^{\beta s},$$
$$\forall y \in Y : C_y = g_1^{q_y(0)},\ C'_y = H(att(y))^{q_y(0)} = g_2^{h_y q_y(0)})$$
$$\text{where } H : \{0,1\}^* \rightarrow \mathbb{G}_2 \text{ and } h_y = \log_{g_2} H(att(y))$$

KeyGen (MK , S) : The algorithm KeyGen generates the private key containing the attribute set S. This algorithm generates the secret key, blinded by P(0) from Master key. In this algorithm extra component D_j'' is added to the basic CP-ABE.

$$SK = (D, \forall j \in S : \langle D_j, D'_j, D''_j \rangle), \text{where}$$
$$D = g_2^{(\alpha + r)/\beta},$$
$$D_j = g_2^r H(j)^{r_j P(0)} = g_2^{r + h_j r_j P(0)},$$
$$D'_j = g_1^{r_j},$$
$$D''_j = (D'_j)^{P(u_k)} = g_1^{r_j P(u_k)}$$

ProxyRekey (PK, MK, RL) : Whenever the key authority wants to revoke some users, he generates a Revocation list containing their identities u_i and evaluates the corresponding $P(u_i)$. The revocation key PXK is generated using this. In case of no or fewer revocations than t, the key authority randomly generates (x, P(x)) other than the actual user identities to make PXK of length t.

$$PXK = \forall u_i \in RL : \langle u_i, P(u_i) \rangle$$

Convert : In this algorithm proxy uses the key PXK and decryptor identity u_k to calculations. Since the user secret key is blinded by P(0), the user also needs C_y'' for decryption.

$$\forall i, j \in \{1, \ldots, t\},\ k \notin \{1, \ldots, t\},$$
$$\lambda_i = \frac{u_k}{u_k - u_i} \cdot \prod_{j \neq i} \frac{u_j}{(u_j - u_i)},$$
$$\forall y \in Y : C''_y = (C'_y)^{\sum_{i=1}^{t} \lambda_i P(u_i)} = g_2^{h_y q_y(0) \sum_{i=1}^{t} \lambda_i P(u_i)}$$

Decrypt(CT, SK) : The decryption algorithm is same as in Bethencourt CP-ABE except one extra pairing is needed at each leaf node.

For a leaf node recursive function, DecryptNode is defined as

$$
\begin{aligned}
&DecryptNode(CT, SK, x) \\
&= \frac{e(C_x, D_i)}{e(D_i'', C_x')^{\lambda_k} e(D_i', C_x'')} \\
&= \frac{e(g_1, g_2)^{rq_x(0)+h_i r_i P(0) q_x(0)}}{e(g_1, g_2)^{r_i h_i q_x(0) \lambda_k P(u_k)} e(g_1, g_2)^{r_i h_i q_x(0) \sum_{j=1}^{t} \lambda_j P(u_j)}} \\
&= \frac{e(g_1, g_2)^{rq_x(0)+h_i r_i P(0) q_x(0)}}{e(g_1, g_2)^{r_i h_i q_x(0) (\sum_{j=1}^{t} \lambda_j P(u_j) + \lambda_k P(u_k))}} \\
&= \frac{e(g_1, g_2)^{rq_x(0)+h_i r_i P(0) q_x(0)}}{e(g_1, g_2)^{r_i h_i q_x(0) P(0)}}, k \notin \{1, 2, \ldots, t\} \\
&= e(g_1, g_2)^{rq_x(0)}
\end{aligned}
$$

For a non leaf node, the recursive function is defined as

$$
\begin{aligned}
F_x &= \prod_{z \in S_x} F_z^{\lambda_i}, [i = index(z), \\
&\quad \lambda_i \text{ calculated over the indices of } z \in S_x] \\
&= \prod_{z \in S_x} (e(g_1, g_2)^{rq_z(0)})^{\lambda_i} \\
&= \prod_{z \in S_x} (e(g_1, g_2)^{rq_{parent(z)}(index(z))})^{\lambda_i} \\
&= \prod_{z \in S_x} (e(g_1, g_2)^{rq_x(i)})^{\lambda_i} \\
&= e(g_1, g_2)^{\sum_{z \in S_x} r \lambda_i q_x(i)} \\
&= e(g_1, g_2)^{rq_x(0)}
\end{aligned}
$$

If the access tree satisfy the access structure we have

$$A = e(g_1, g_2)^{rq_R(0)} = e(g_1, g_2)^{rs}$$

Decryption proceed as

$$\frac{\tilde{C}}{\frac{e(C,D)}{A}} = Me(g_1, g_2)^{\alpha s} \frac{e(g_1, g_2)^{rs}}{e(g_1, g_2)^{\alpha s + rs}} = M$$

Delegation : Delegation allows a user B to delegate some sunset of the attributes to user C. B randomizes its secret key for C for subset of attributes.

The delegated key is generated as:

$$\tilde{SK} = (D, \forall j \in \tilde{S} : \langle D_j, D'_j, \tilde{D''_j}, \tilde{D'''_j} \rangle), \text{where}$$
$$\tilde{D''_j} = (D''_j)^{1/P_B(0)}, \tilde{D'''_j} = (D''_j)^{P_B(C)/P_B(0)}$$

Where P_B is the polynomial present in the B's master key.

To decrypt the ciphertext using delegated key, we have function DecryptNode as

$$\begin{aligned}
&DecryptNode(CT, \tilde{SK}, x) \\
&= \frac{e(C_x, D_i)}{e(\tilde{D''_i}, C''_{xB})^{\lambda_B} e(\tilde{D'''_i}, C'_x)^{\lambda_B \lambda_C} e(D'_i, C''_{xA})} \\
&= \frac{e(C_x, D_i)}{e(g_0, g_1)^{r_i h_x q_x(0) X_B \lambda_B P_A(B)/P_B(0)}} \\
&\cdot \frac{1}{e(g_0, g_1)^{r_i h_x q_x(0) P_A(B) P_B(C)/P_B(0) \lambda_B \lambda_C} e(D'_i, C''_{xA})} \\
&= \frac{e(g_0, g_1)^{rq_x(0) + r_i h_x q_x(0) P_A(0)}}{e(g_0, g_1)^{r_i h_x q_x(0) \lambda_B P_A(B)} e(g_0, g_1)^{r_i h_x q_x(0) X_A}} \\
&= \frac{e(g_0, g_1)^{rq_x(0) + r_i h_x q_x(0) P_A(0)}}{e(g_0, g_1)^{r_i h_x q_x(0) P_A(0)}} \\
&= e(g_0, g_1)^{rq_x(0)}
\end{aligned}$$

CHAPTER 4

IMPLEMENTATION

4.1 USER REGISTRATION

4.1.1 Patient registration

A new patient registers by clicking on registration button and then fills up the details required by the service provider. The service provider checks the details filled by the patient and adds the entry to the database. At the same point service provider also generates the attribute key and private key for the patient and sends the key to the patient in secure manner.

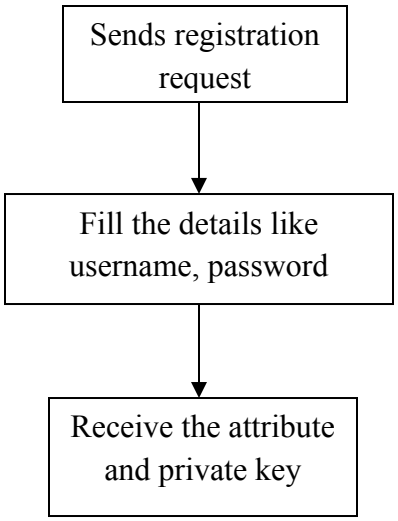

Figure 4.1:Flow chart showing patient registration

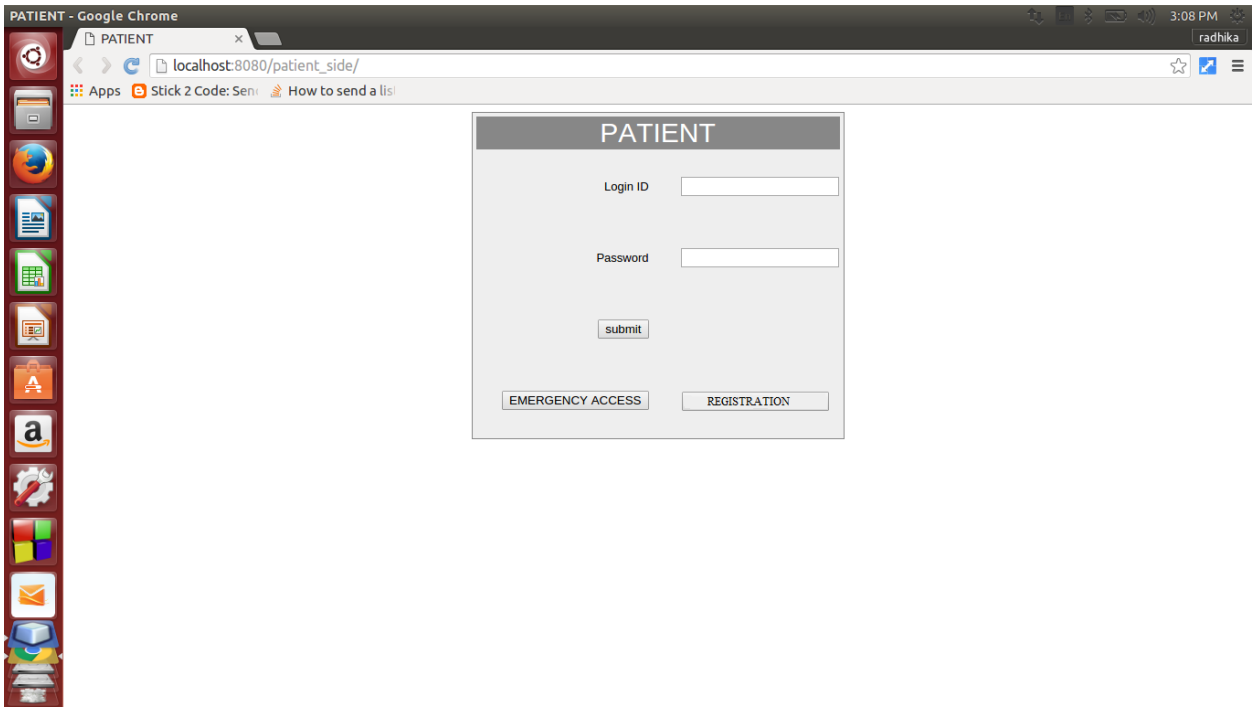

Figure 3.2: Screen snapshot of patient home screen

4.1.2 Medical Personnel Registration

The Medical Personnel registers by sending request to the Service provider. Upon receiving the request the service providers send the form to be filled out by Medical Personnel. The form contains details like the role of Medical personnel, i.e. a doctor, nurse, etc. Other details contained in the form are like the hospital name in which Medical Personnel is working, username, password. According to the form filled by the Medical Personnel, the Service provider generates the attribute key for the personnel. Service provider also generates the private key for the personnel and adds the entry to the database. The generated keys are then send to the Medical Personnel.

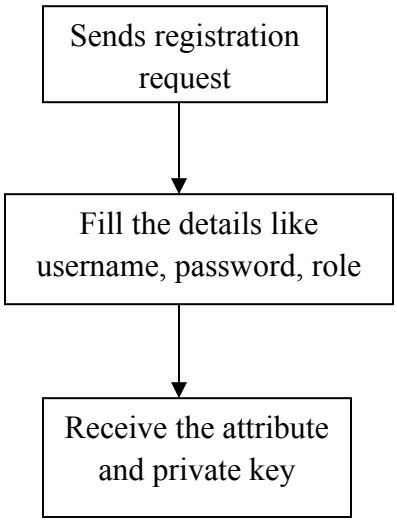

Figure 4.3: Medical Personnel registration

4.2 MEDICAL PERSONNEL AND PATIENT LOGIN

Both Medical Personnel and Patient can access the system by entering the right combination of login ID and password. Once they enter the right combination screen appears for Medical Personnel as in figure and for patient as in figure

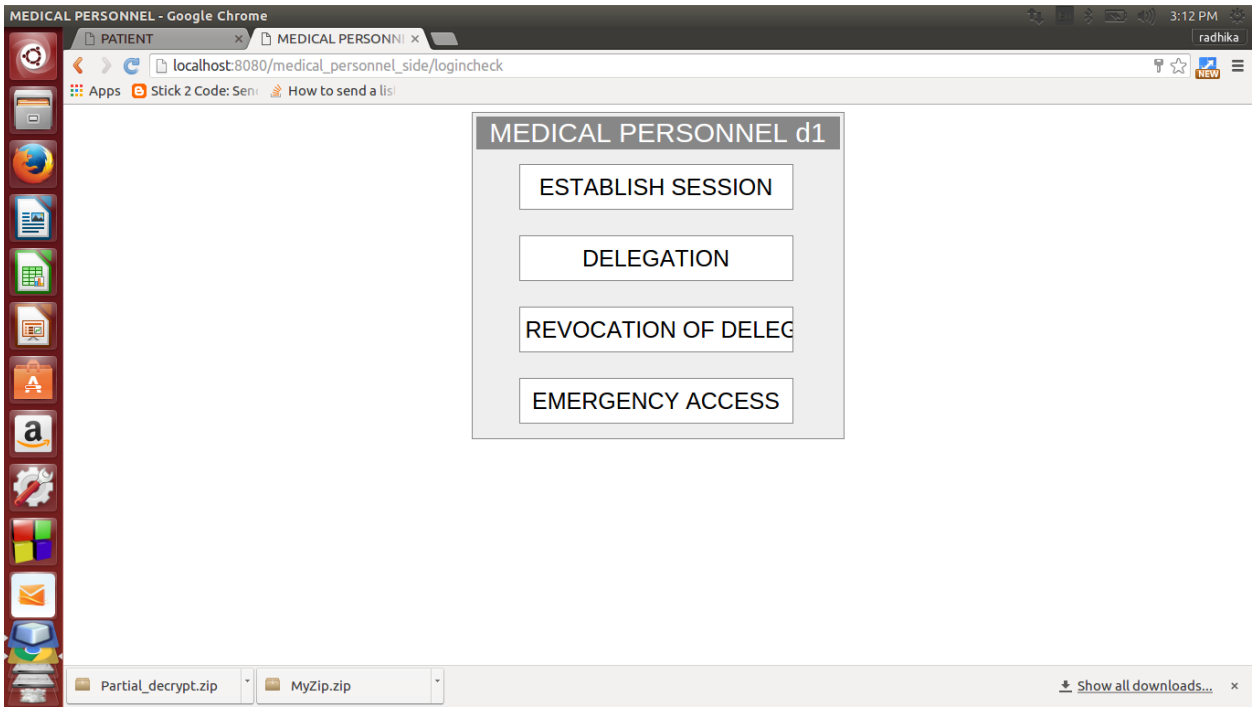

Figure 4.4: Screen snapshot of medical personnel after login

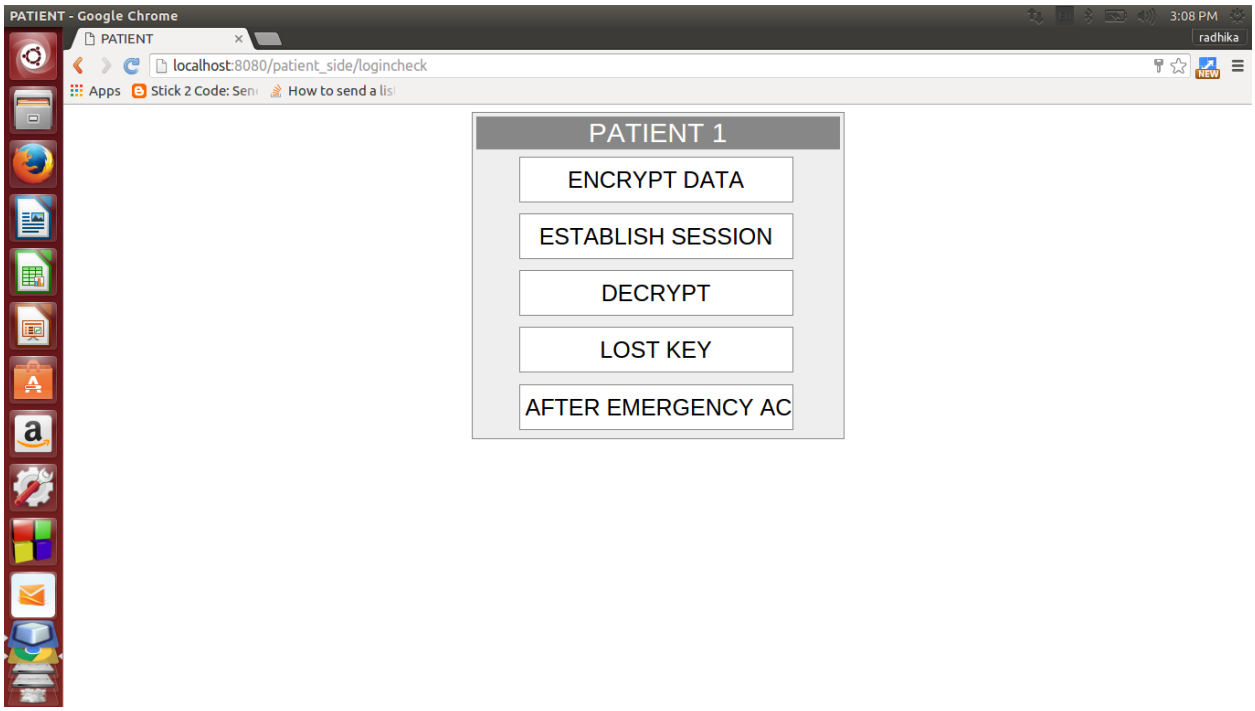

Figure 4.5: Screen snapshot of patient after login screen

4.3 ENCRYPTION OF HEALTH RECORDS

Whenever patient has to encrypt his health records, he sends the request to the Service Provider and then selects the file to be encrypted. After file selection, patient defines the hospitals whose Medical Personnel can decrypt his records. According to the selection made by the patient, Service provider encrypts the records and then the encrypted file can be downloaded by the patient.

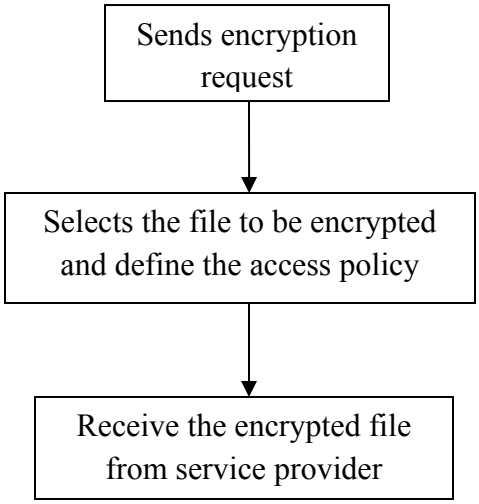

Figure 4.6: Encryption of Health records

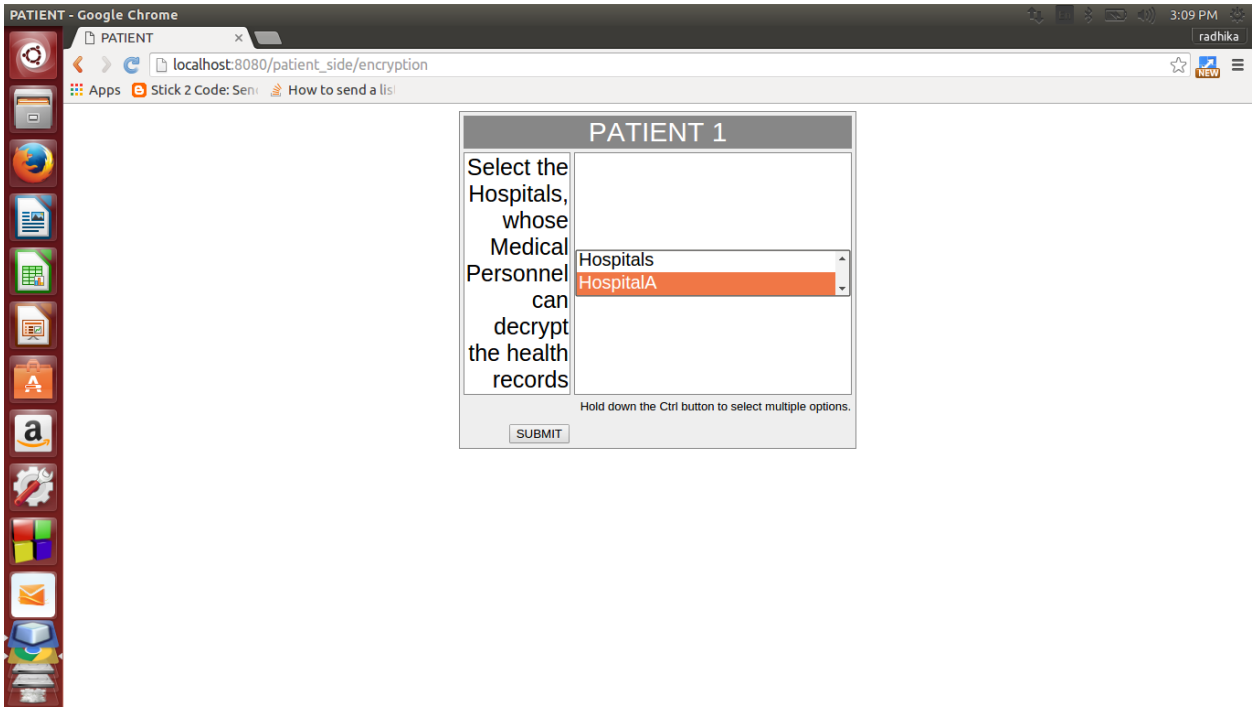

Figure 4.7: Screen snapshot of defining access policy for encryption

4.4 SESSION ESTABLISHMENT

Whenever patient wants to connect to the Medical personnel, he establishes connection with the Medical Personnel. He sends the encrypted file and random number to the Medical Personnel and waits for the reply from the Medical Personnel. On receiving the file and random number, Medical Personnel decrypts them and if, he wants to write on the records he request for write access from the patient or the session is completed from both the ends.

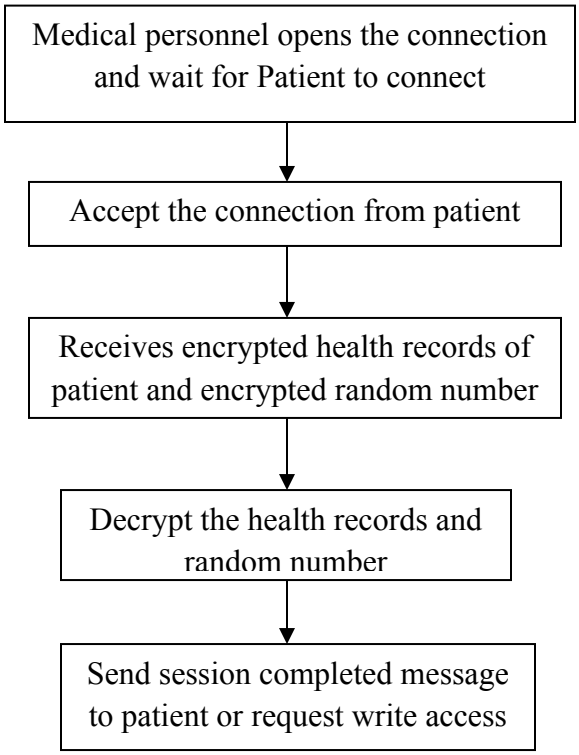

Figure 4.8: Session establishment between Medical Personnel and Patient

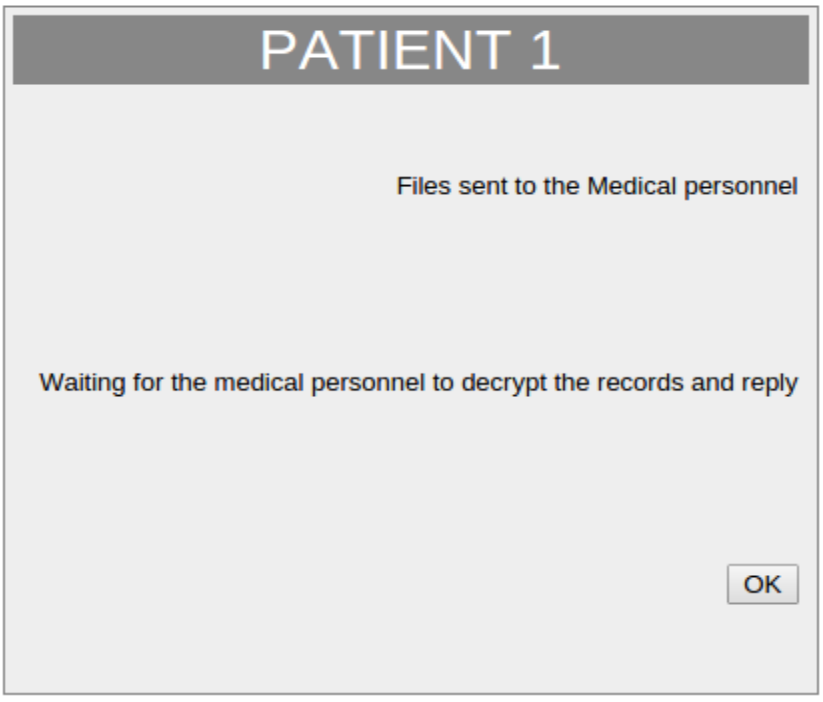

Figure 4.9: Snapshot showing health records sent to Medical Personnel

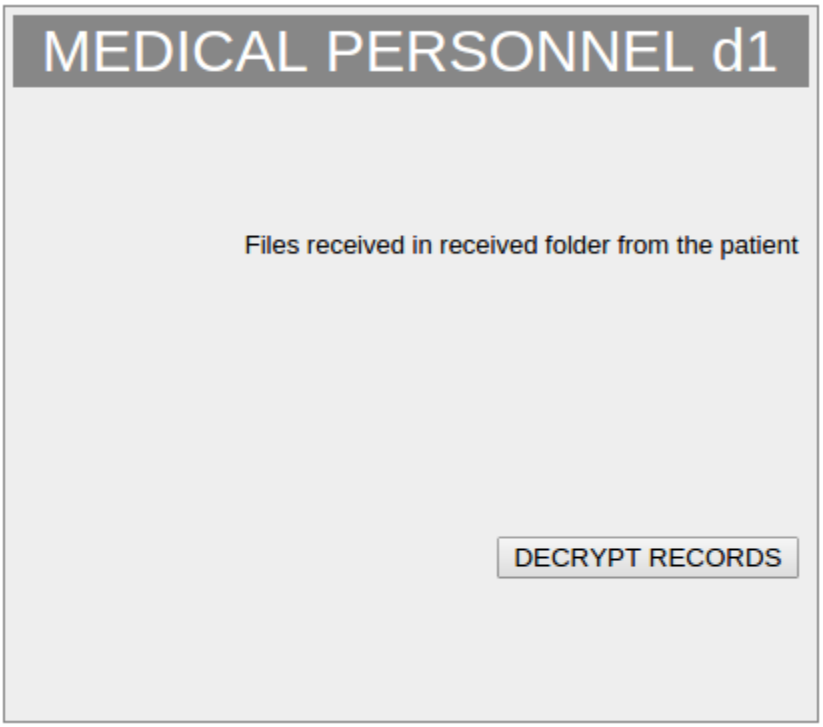

Figure 4.10: Snapshot showing health records received by Medical Personnel

4.5 PATIENT'S LOST KEY

Whenever patient loses his key, to protect his health records he sends the lost key request to the Service provider. Upon receiving the request, service provider ask for the key ID of the lost key. When Service provider receives the key ID he checks whether key ID provided by the patient actually belongs to him or not. Upon verification, service provider revocates the key by generating the new revoke key. Once revocation is done successfully, revocation successful message is sent to the patient.

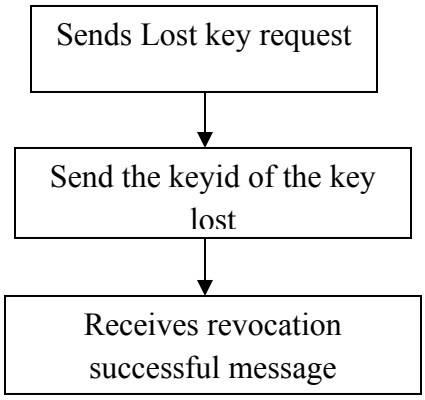

Figure 4.11: Patient lost key flow diagram

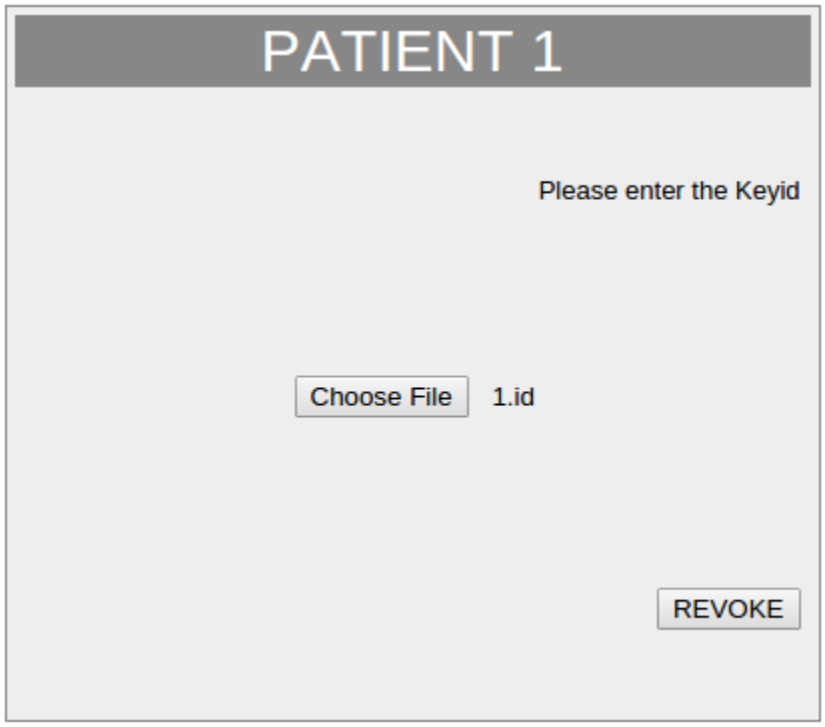

Figure 4.12: Snapshot asking keyid of the lost key

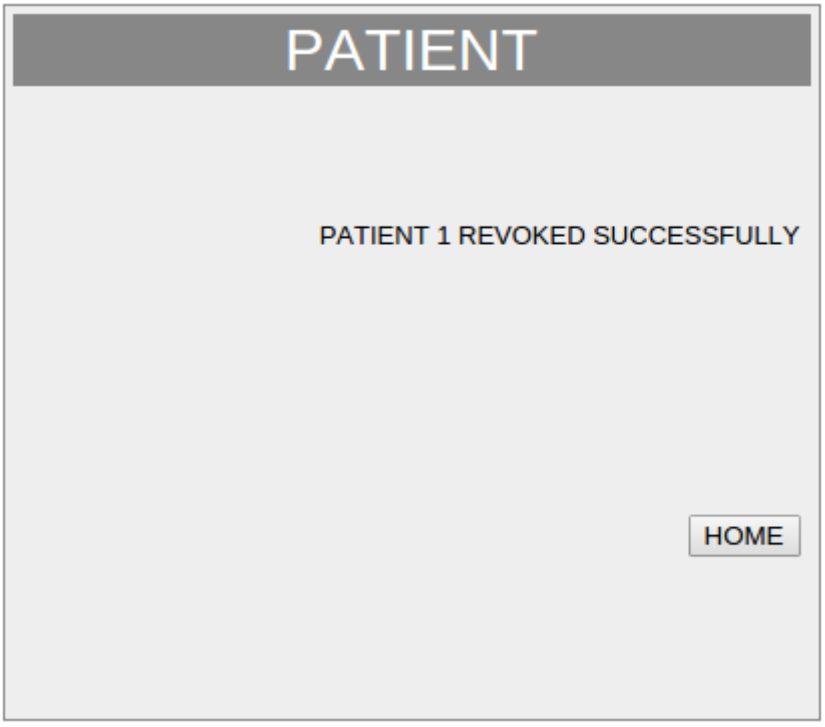

Figure 4.13: Snapshot showing lost key successfully revoked

4.6 DELEGATION

Whenever a Medical Personnel wants to delegate his key, he sends the delegation request to the service provider. Service provider upon receiving the request, ask for the key and attributes to be delegated. Once the Medical Personnel has responded back, service provider generates the new key and adds the entry of delegation to the database. Then the delegated key is send to the Medical Personnel.

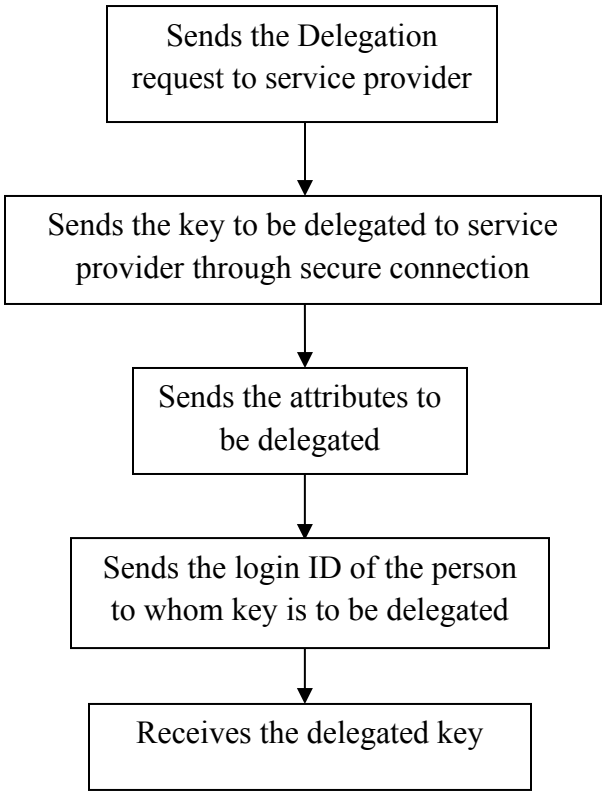

Figure 4.14: Delegation of the key flow diagram

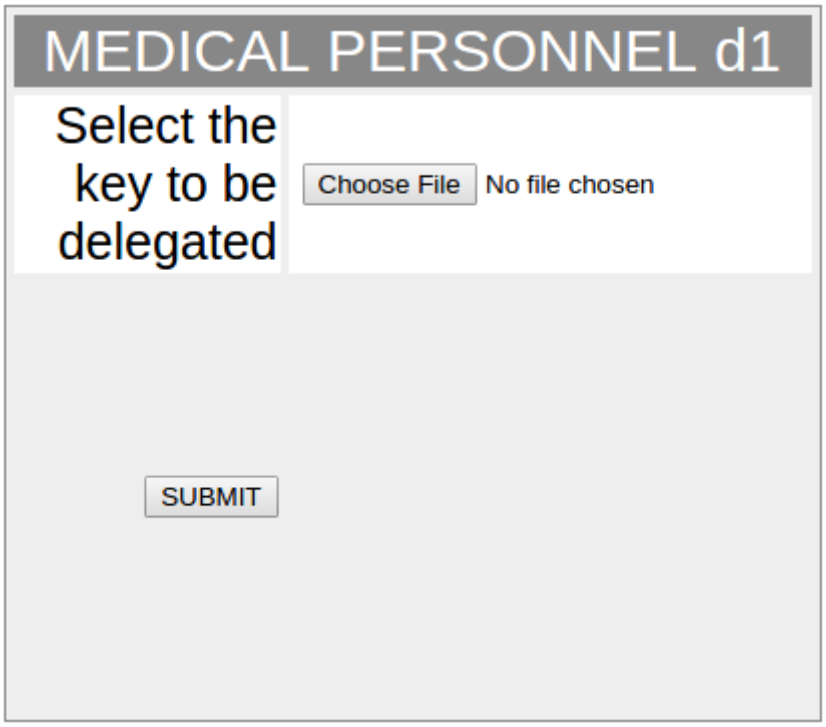

Figure 4.15: Snapshot showing selection of key to be delegated

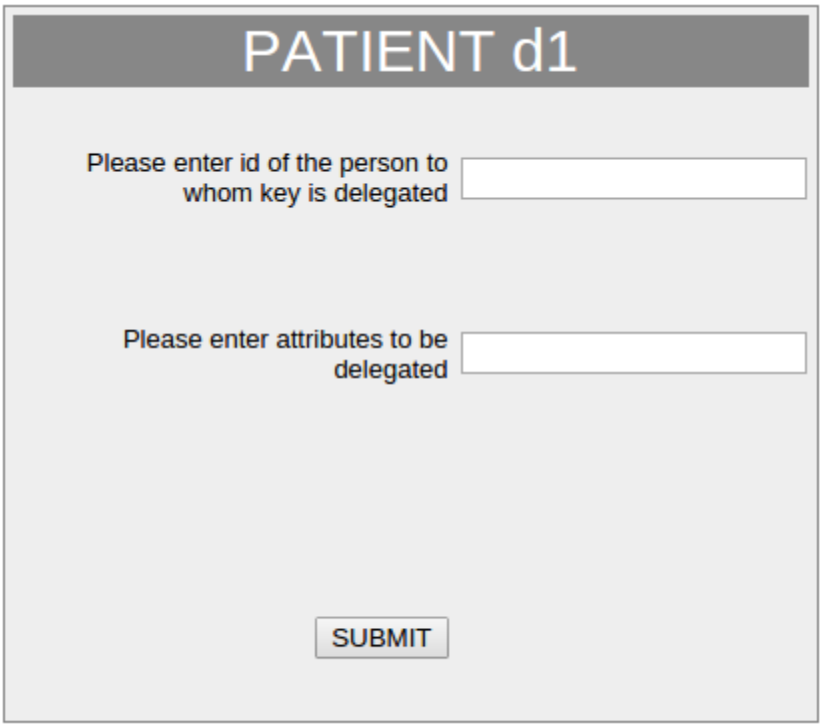

Figure 4.16: Snapshot showing details of attribute set in the delegated key

4.7 REVOCATION OF DELEGATED KEY

Whenever the Medical Personnel wants to revoke the delegated key, he sends the request to the Service provider. Upon receiving the request, Service provider enquires about the ID of the person to whom the key was delegated. Service provider checks the response received in the database, and if the key is delegated, service provider removes the entry from the database and then the revocation successful message is sent.

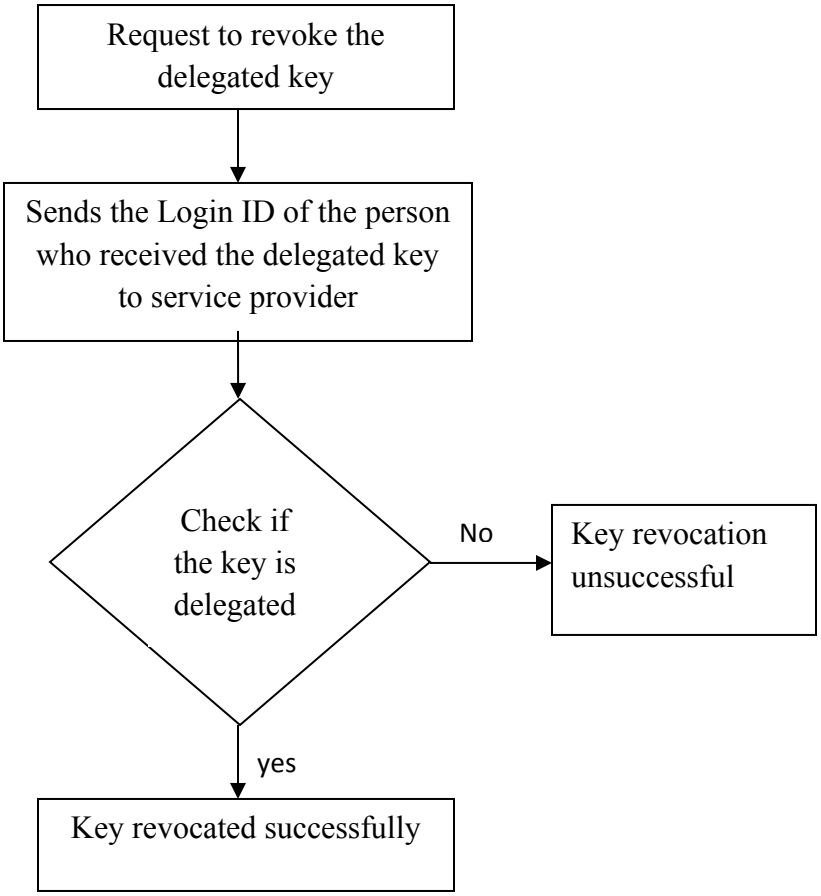

Figure 4.17: Revocation of delegated key flow diagram

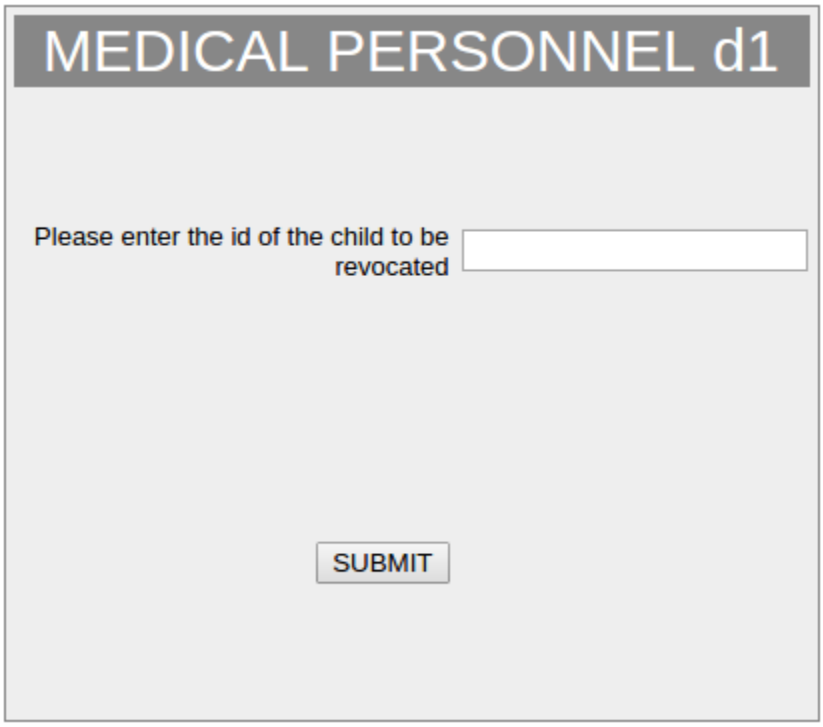

Figure 4.18: Screen snapshot showing revocation of delegated key

4.8 REVOCATION OF MALICIOUS USERS

Whenever a Medical personnel is found doing some malicious work, he is revoked from the system so that he is not able to harm any other person. The revocation is done by the service provider by revoking his key. To revoke the key, service provider generates a new revoke key and adds the key ID of the revoked personnel to the revocation list.

4.9 DECRYPTION OF HEALTH RECORDS

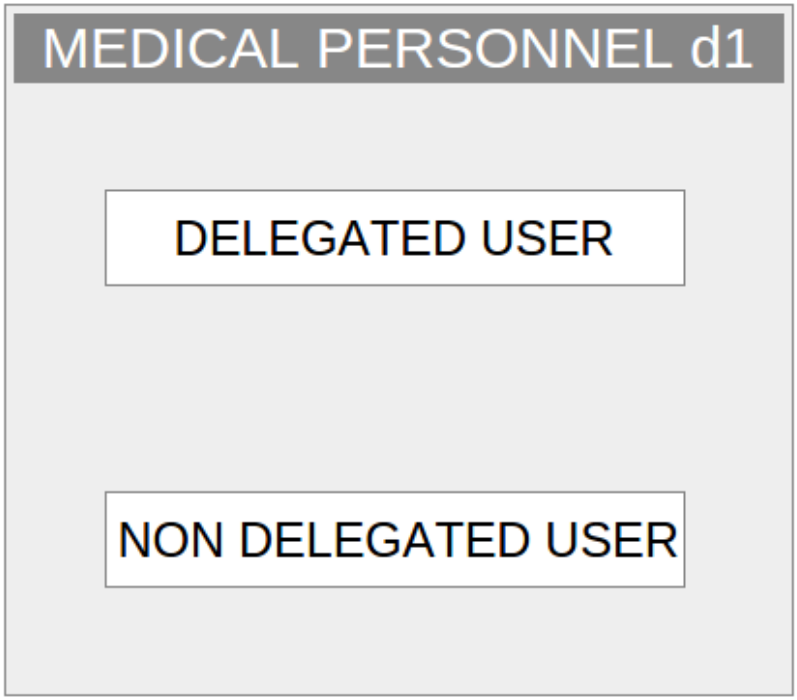

Figure 4.19 : Screen snapshot showing selection between delegated or non delegated user

4.9.1 Non delegated user

To decrypt the records, the user sends the file to be decrypted to the service provider, who using the revoke key tries to generate the shares and adds them to the file and file is renamed to .proxy file. lamdba is also generated. Service provider then sends both the things to the user and also adds the entry to the database for auditing purposes. Now the user tries to decrypt the received .proxy file with his attribute key. If the user is not revoked he will be able to decrypt the records else he will not be able to decrypt the records.

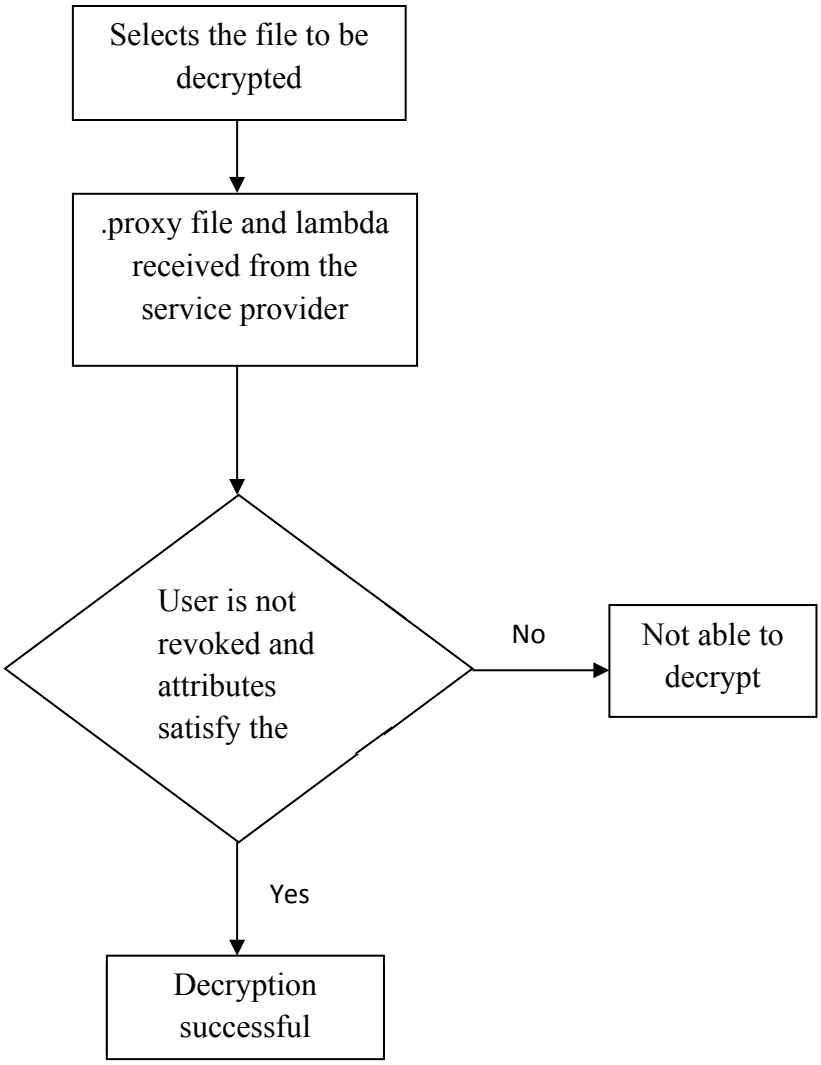

Figure 4.20: Flowchart depicting decryption flow

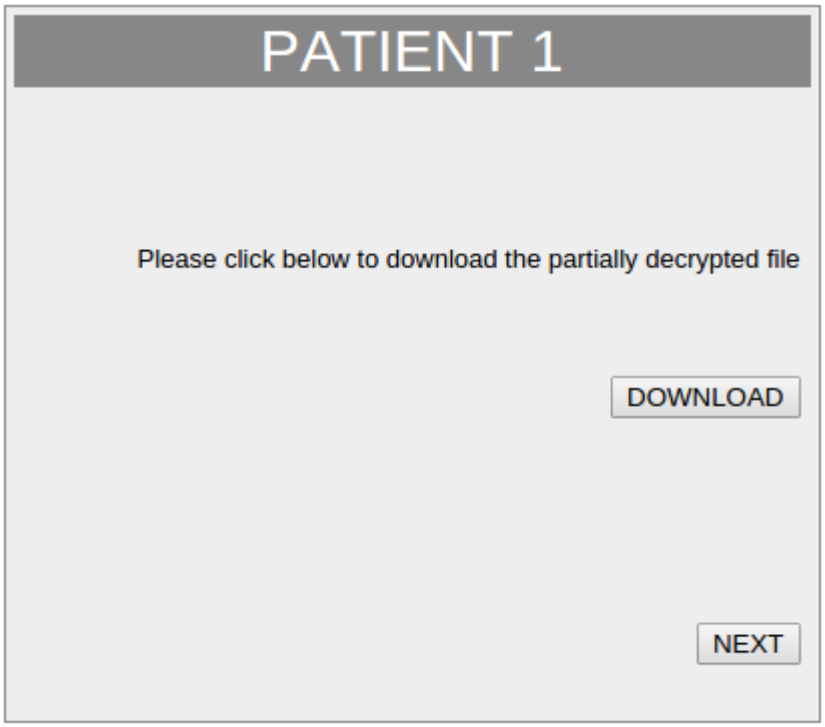

Figure 2.21: Screen snapshot showing file converted to .proxy by proxy server

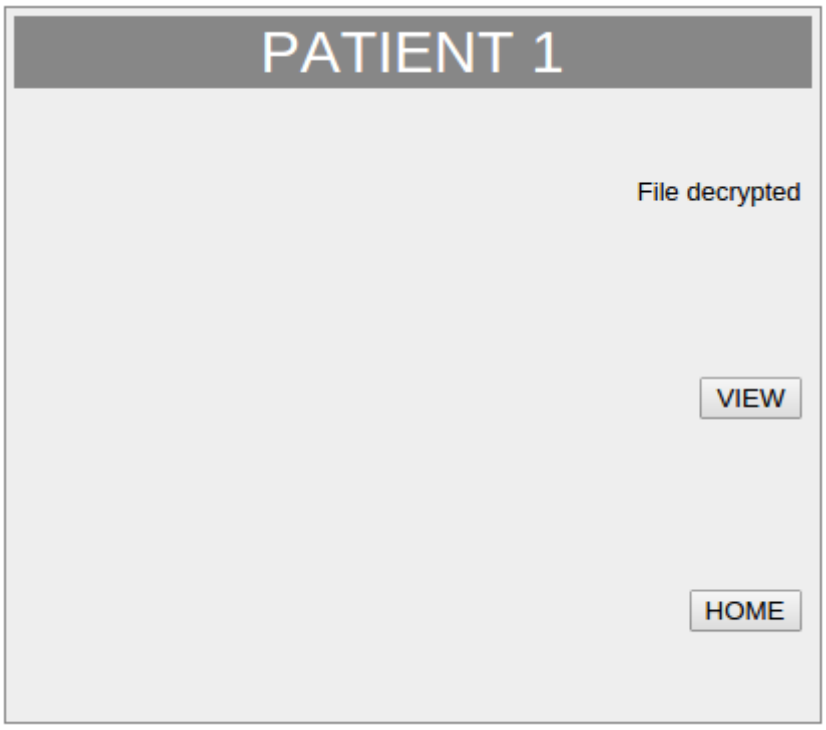

Figure 4.22: Snapshot showing file decrypted successfully

4.9.2 Delegated User

To decrypt the records, delegated user sends the request to the Service provider and then uploads the file to be decrypted. Delegated user also tells the ID of the source of the delegated key. Service provider after receiving all this information verifies whether the key is delegated by the source or not. On verification, Service provider generates the .proxy file and lambda for the user and the source and send all the things to the user. Now the user can decrypt the records using his delegated key.

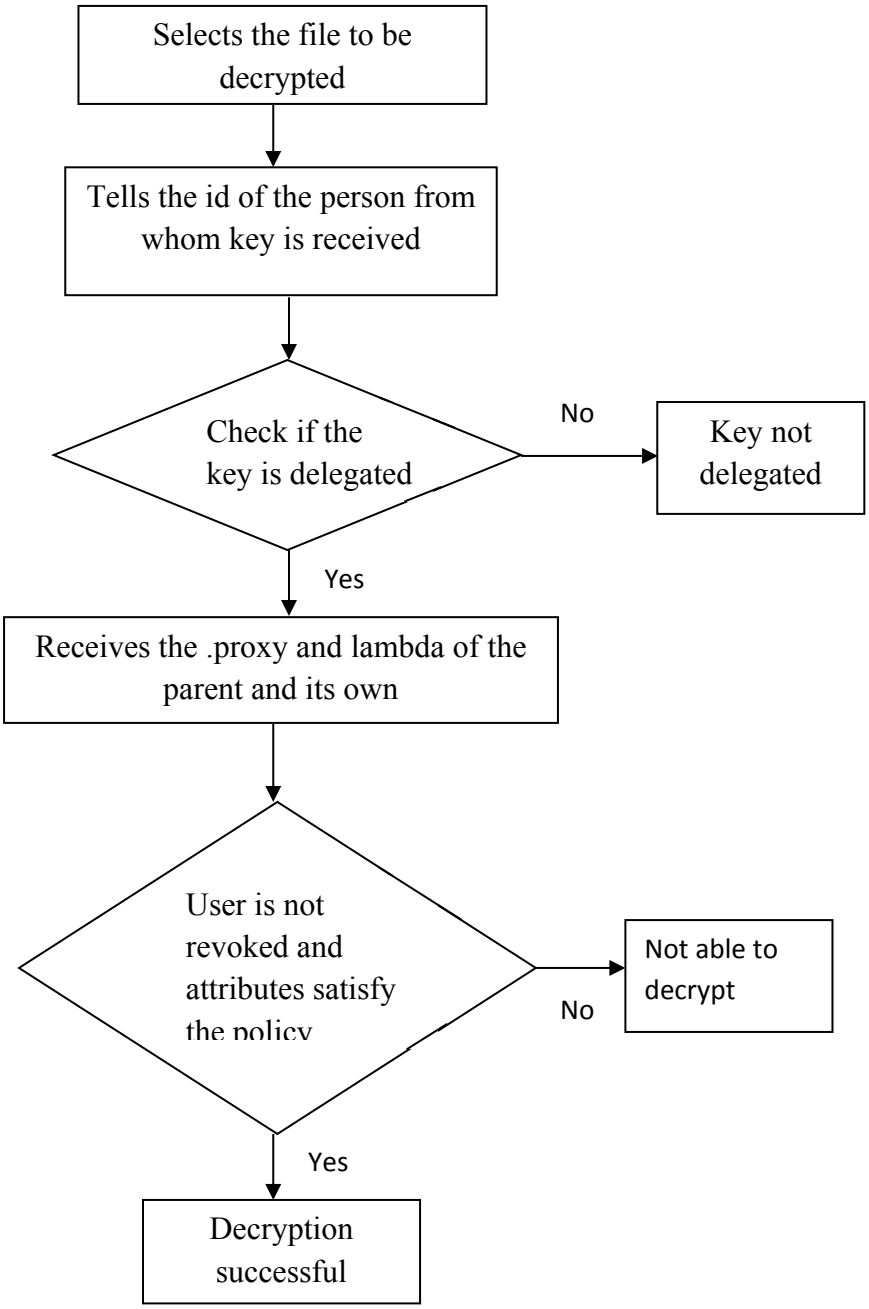

Figure 4.23 : Decryption for a delegated user flow diagram

4.10 WRITE ACCESS

Whenever the Medical Personnel wants to write on the patient records, he has to get his write access verified, i.e. whether he can write on the records or not. When the Medical Personnel establishes connection with the patient along with encrypted file he also receives the encrypted random number. If the Medical Personnel is able to decrypt the random number, it means he has the write access. So to prove his write access, Medical Personnel decrypts the random number and encrypts the updated records with this random number and send the updated encrypted records to the patient. The patient on receiving the records, decrypts them and if he is able to decrypt them successfully, means Medical Personnel has write access. So the patient accept the changes made to the records.

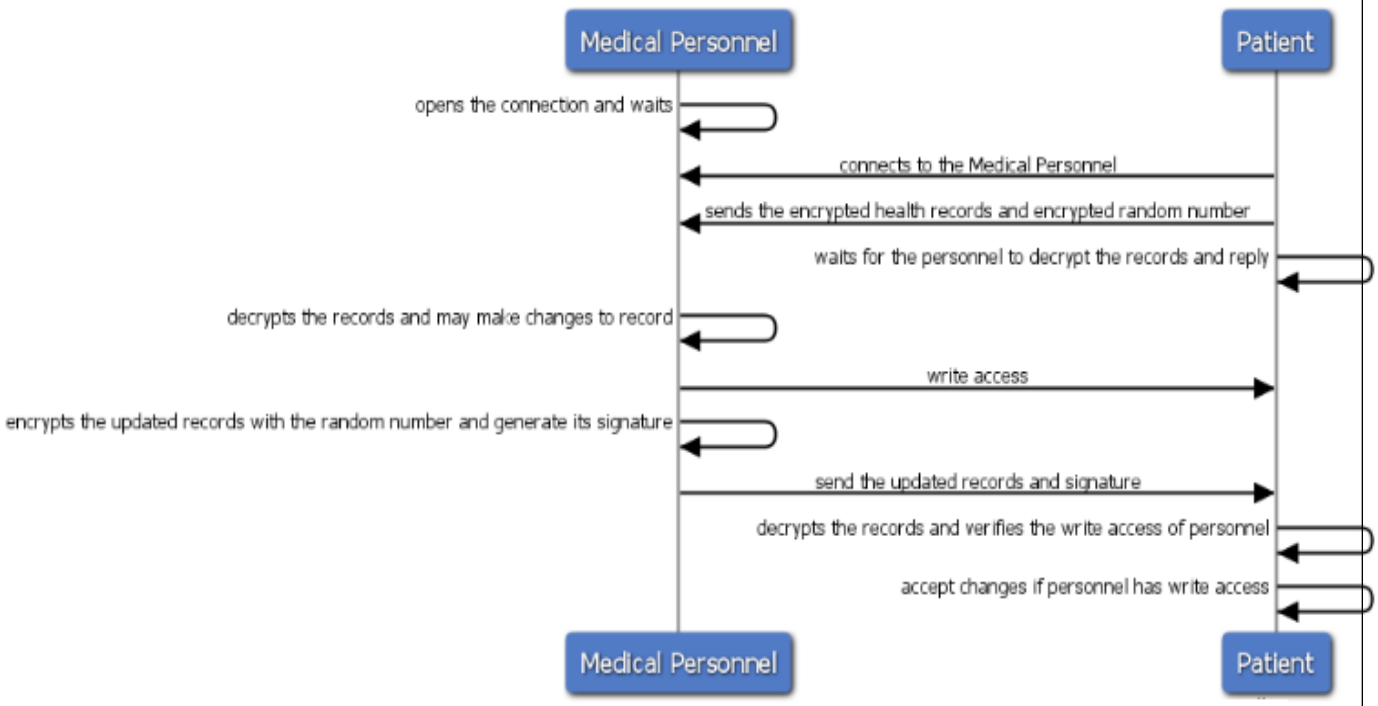

Figure 4.24 : Write access check

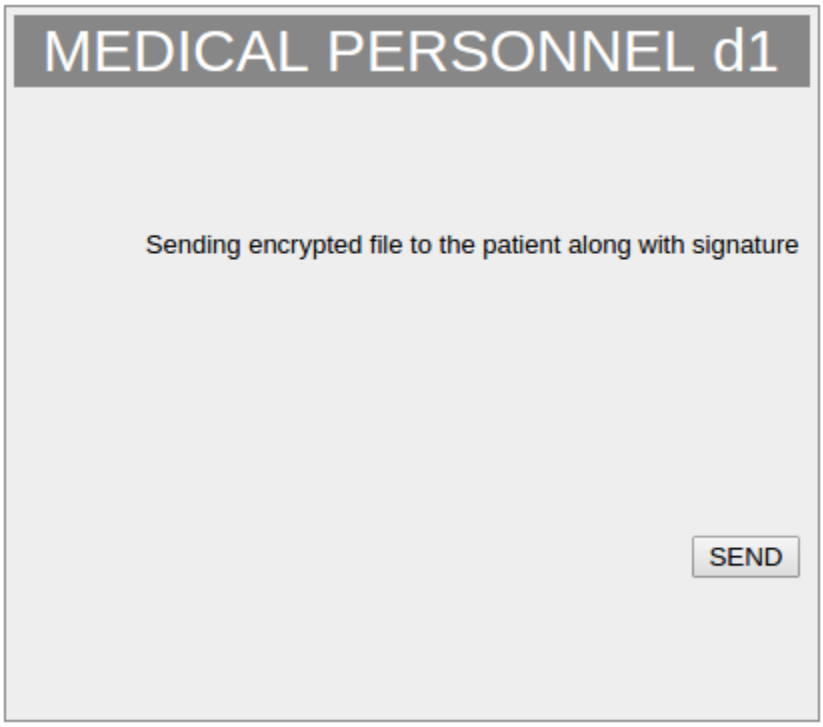

Figure 4.25 : Snapshot sending file along with signature to patient

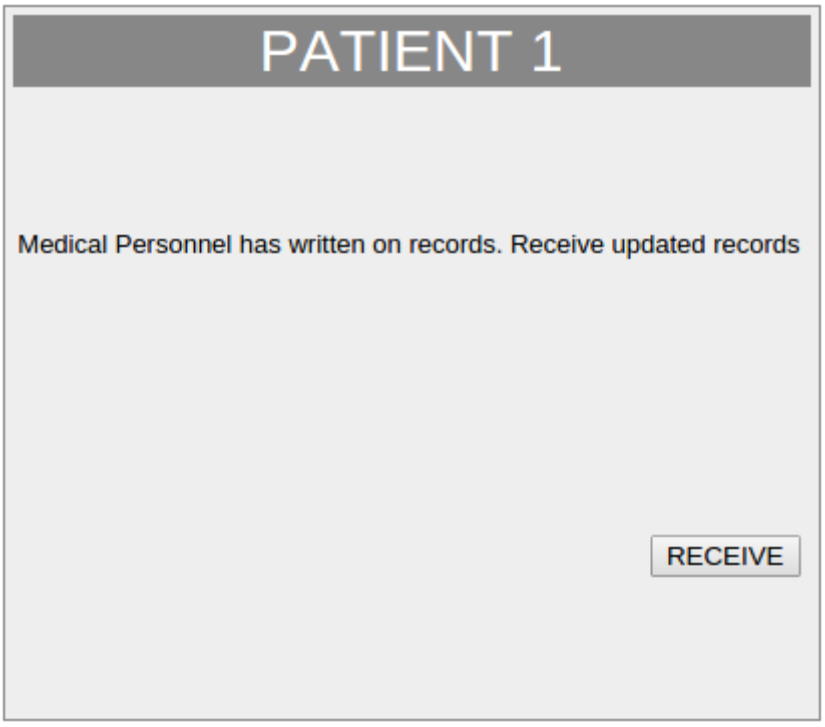

Figure 4.26 : Screen snapshot showing receiving updated file from the Medical Personnel

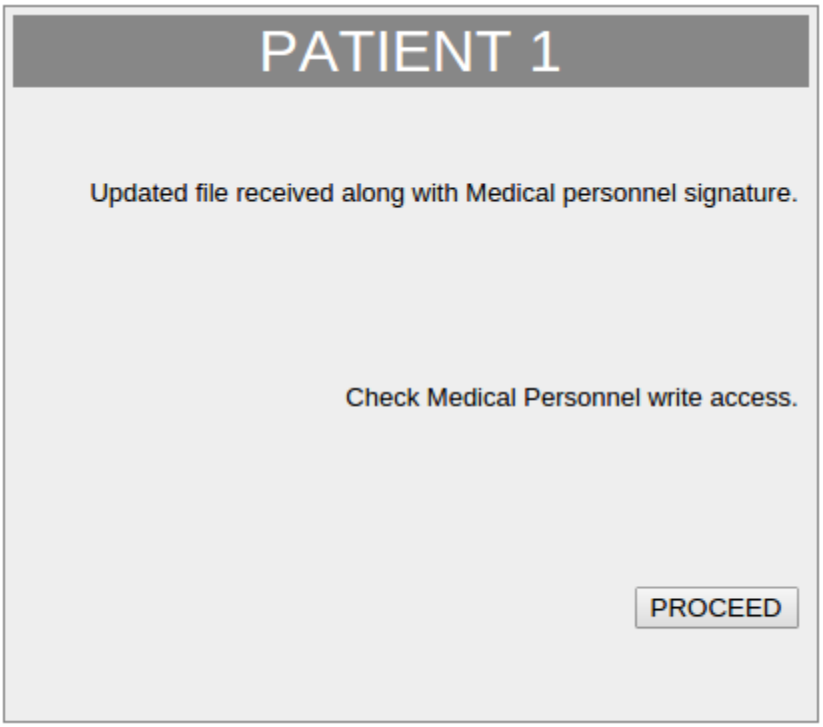

Figure 4.27: Screen snapshot checking write access of the Medical personnel

4.11 AUDITING

We maintain the read and write audits to detect any malicious behavior. Whenever a medical personnel contacts the proxy a read audit is added to the database. In case of write, when Medical personnel sends the updated records along with his signature, patient after verifying write access of the Medical Personnel adds his own signature. Then updated file, Medical personnel signature, patient signature along with timestamp are added to the database.

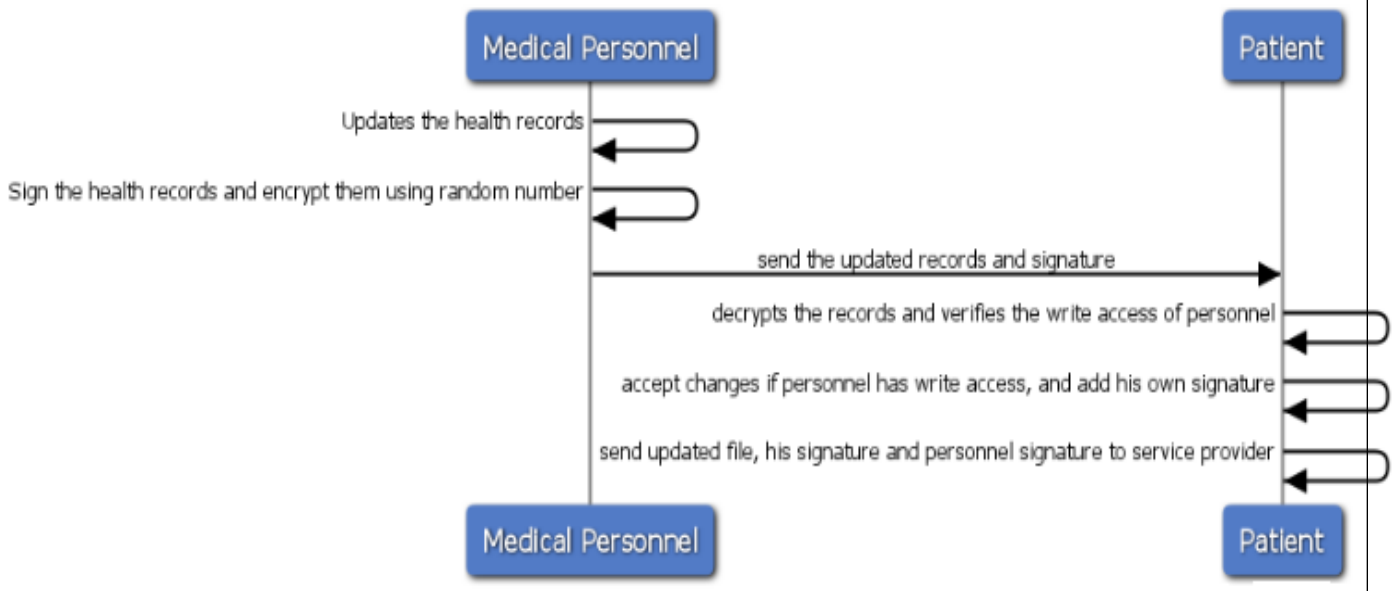

Figure 4.28: Auditing

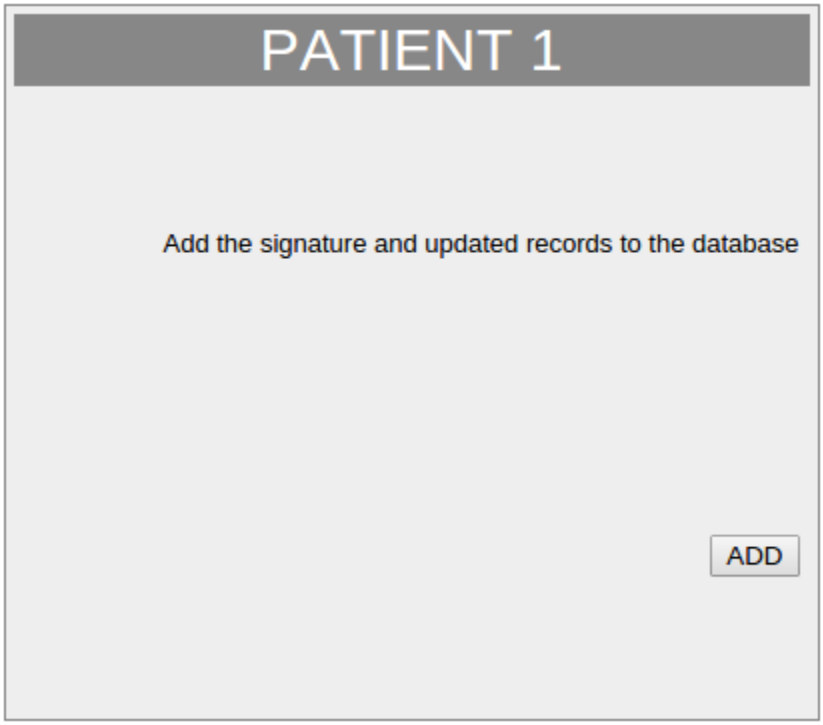

Figure 4.29: Snapshot adding updated records and signature to database

4.12 EMERGENCY ACCESS

In case of emergency, Medical Personnel request the Service provider for emergency access. Service provider verifies the emergency condition and generates a new emergency key and adds the entry to the database. Service provider then send the emergency key to the Medical Personnel and revokes the key after 1 hour. Medical personnel on receiving the key, decrypts the record using this key. Auditing and write access remains the same like in normal condition.

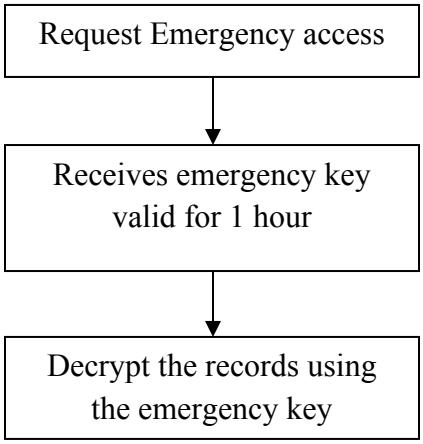

Figure 4.30: Emergency access

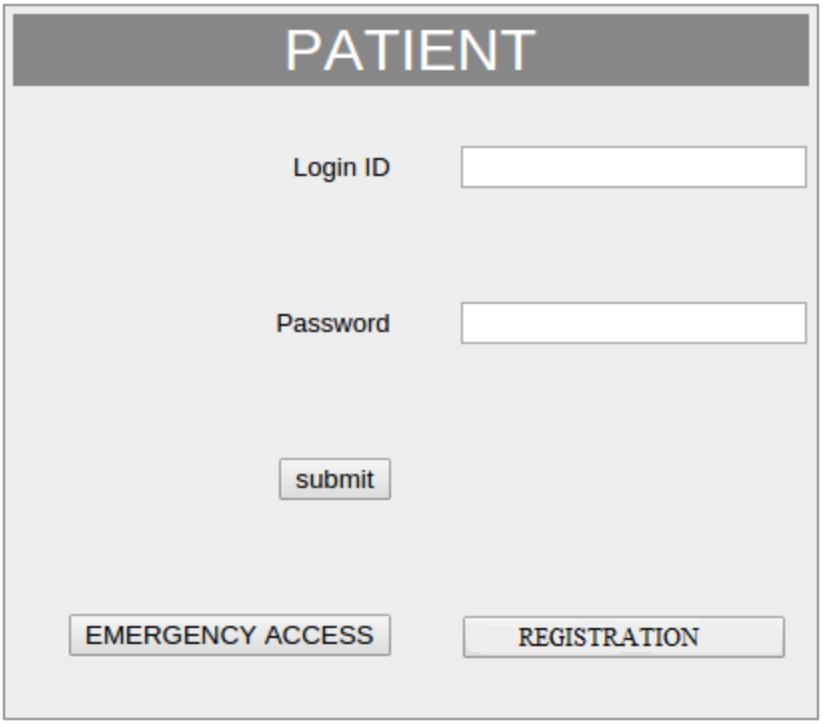

Figure 4.31: Snapshot of emergency access screen for patient

CHAPTER 5

RESULTS

Figure 5.1 shows the encryption time when the policy used has OR's between the attributes. We can see as the number of attributes increases, time taken to encrypt the file also increases. Maximum number of users that can be revoked have no effect on the encryption time.

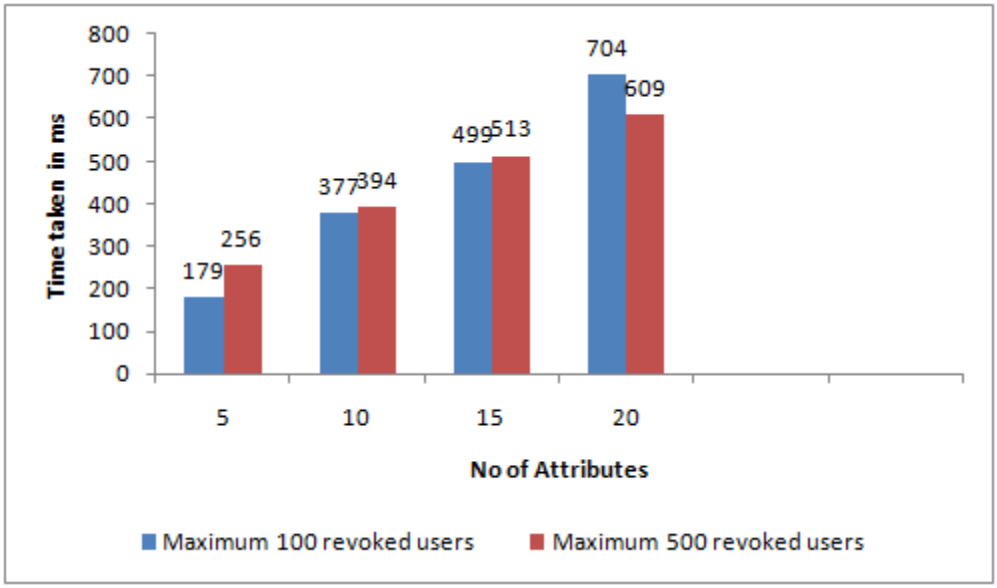

Figure 5.1: Encryption time with OR policy

Figure 5.2 shows the time taken by algorithm convert when the access policy has OR's between the attributes. As the number of attributes increases, time to perform convert function also increases. Also as we increase the maximum number of users that can be revoked the convert time also increases.

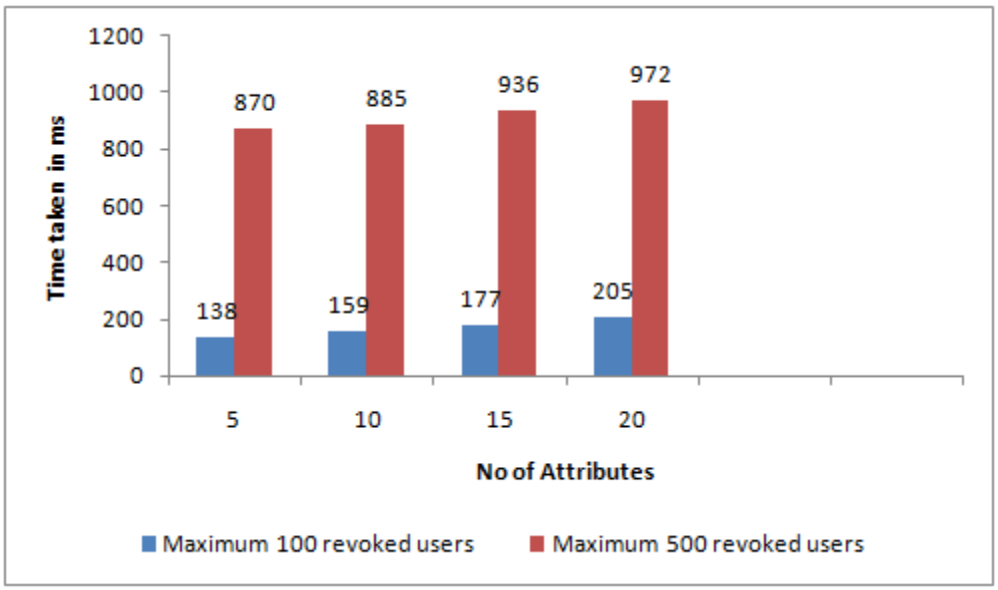

Figure 5.2: Convert time with OR policy

Figure 5.3 shows the decryption time when we use the policy which has OR between all the attributes. We can see the number of attributes have no effect on the decryption time. Also the maximum number of users that can be revoked in the system have no effect on the decryption time.

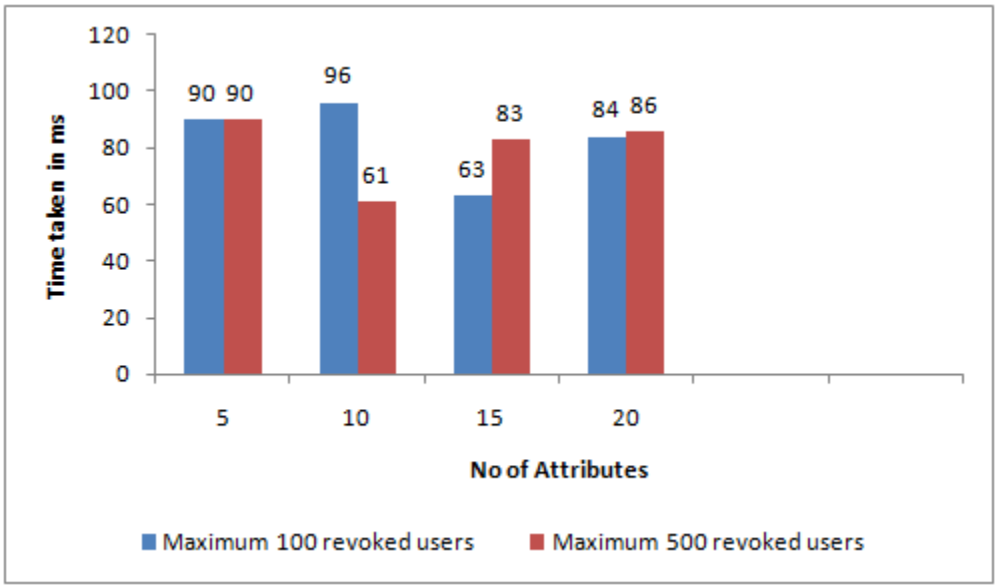

Figure 5.3: Decryption time with OR policy

Figure 5.4 shows the encryption time taken when the policy has AND between the attributes. Results show that when the number of attributes present in the access policy increases, the encryption time also increases but maximum number of users that can be revoked have no effect on the encryption time.

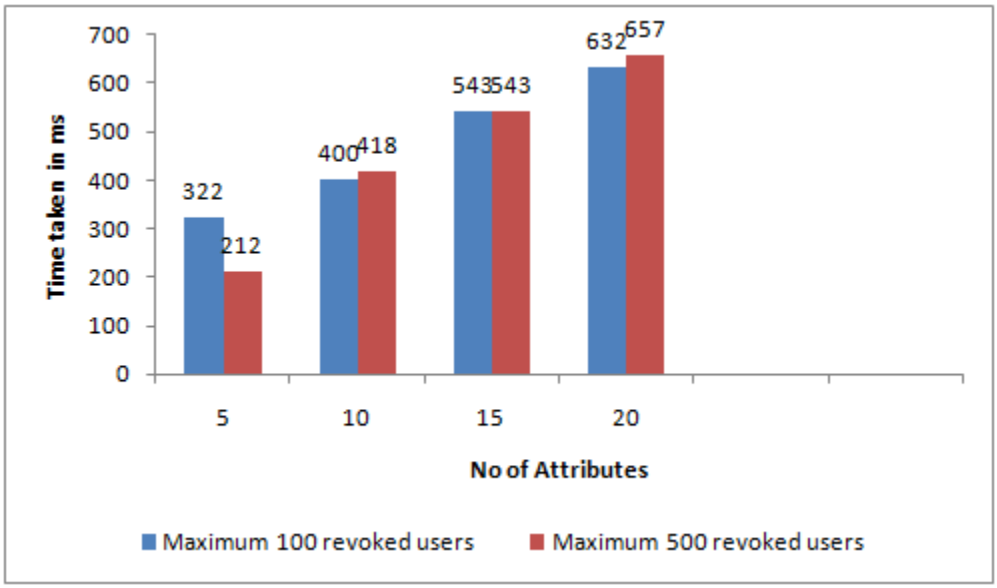

Figure 5.4: Encryption time with AND policy

Figure 5.5 shows the time taken by the convert algorithm when the access policy has all AND between the attributes. Results show that as the number of attributes increases, the time taken by the convert algorithm also increases. Also, when we increase the maximum number of users that can be revoked in the system, the time taken by convert algorithm increases.

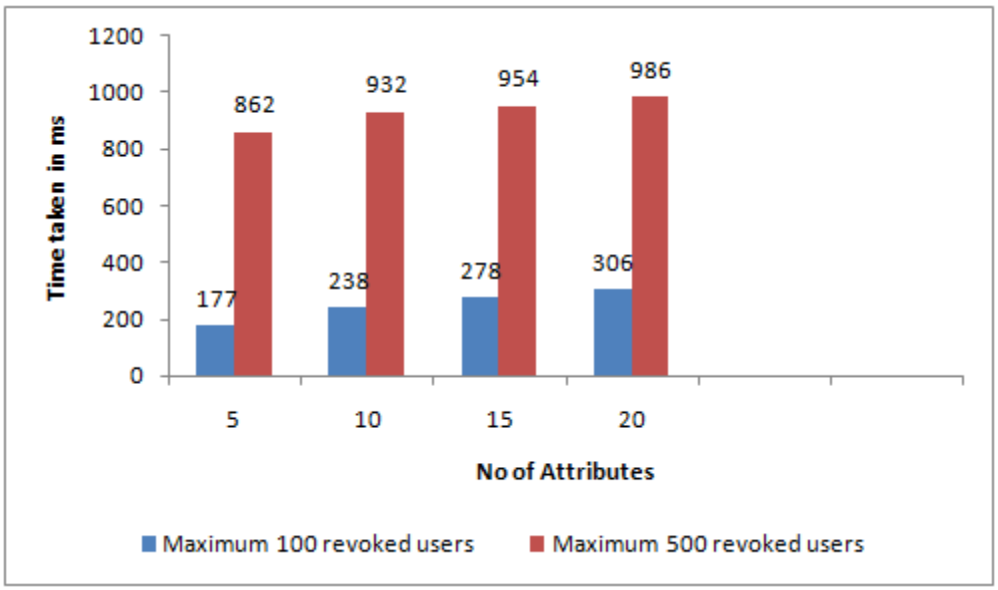

Figure 5.5: Convert time with AND policy

Figure 5.6 shows time taken in decryption when the access policy has all AND between the attributes. Results show that decryption time increases as number of attributes in the policy increases. It also shows that decryption time is independent of the maximum number of users that can be revoked in the system.

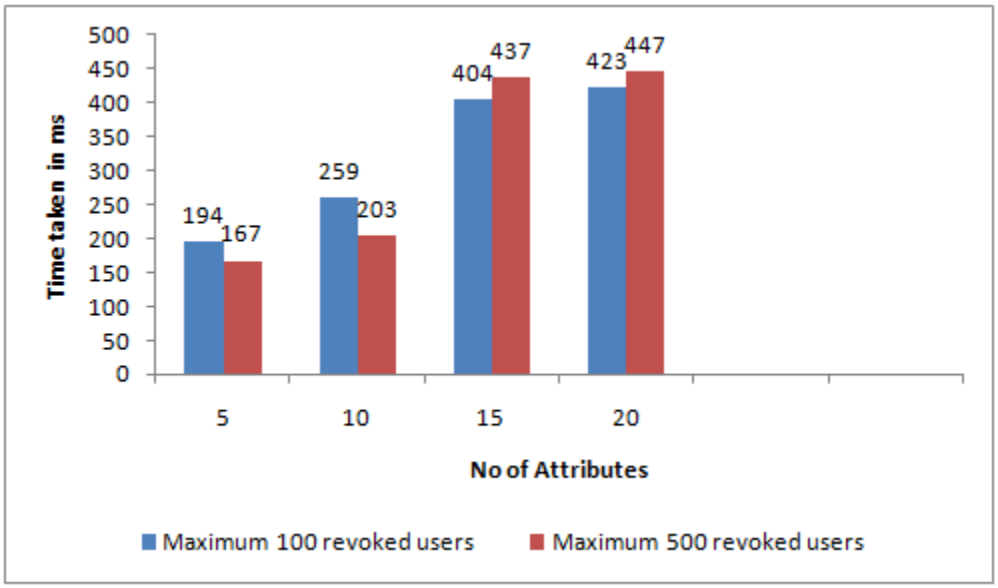

Figure 5.6: Decryption time with AND policy

Figure 5.7 shows the time taken in generating key when we change the number of attributes present in the attribute set. With increasing number of attributes, key generation time also increases but maximum number of users that can be revoked have no effect on the key generation time.

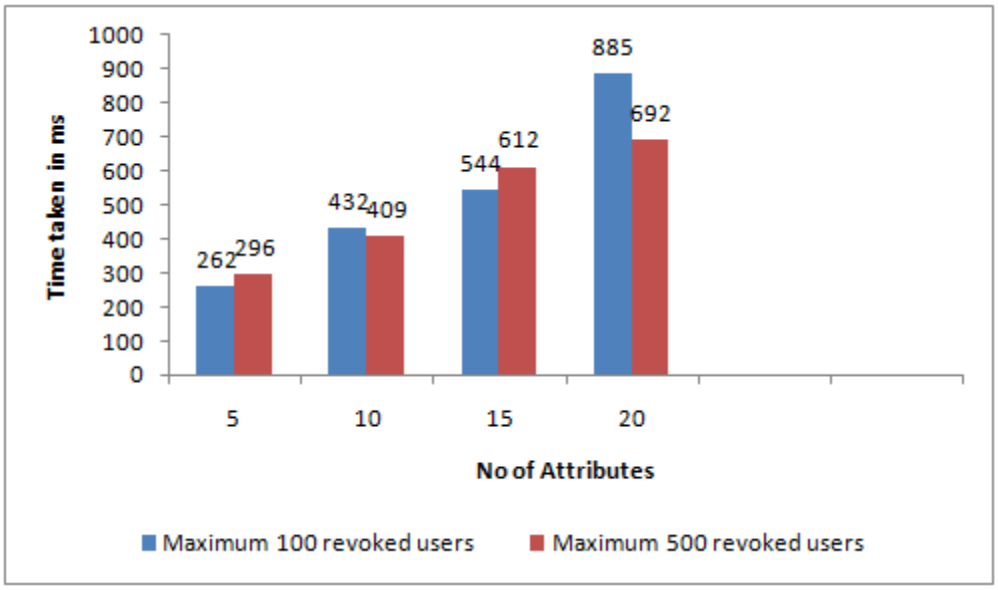

Figure 5.7: Key Generation time

Figure 5.8 shows the time taken to generate the delegated key. Results show that time taken to generate the delegated key is independent of the number of attributes in the key and maximum number of users that can be revoked from the system.

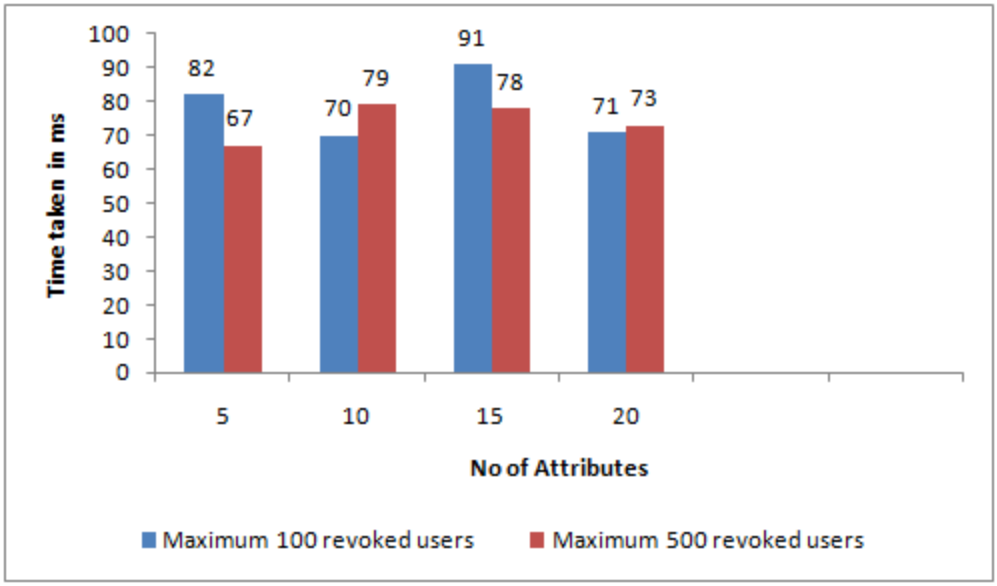

Figure 5.8: Delegated key generation time

Figure 5.9 shows the encryption time varying file size. We see that the encryption time does not depends upon the file size and is also independent of the maximum number of users that can be revoked from the system.

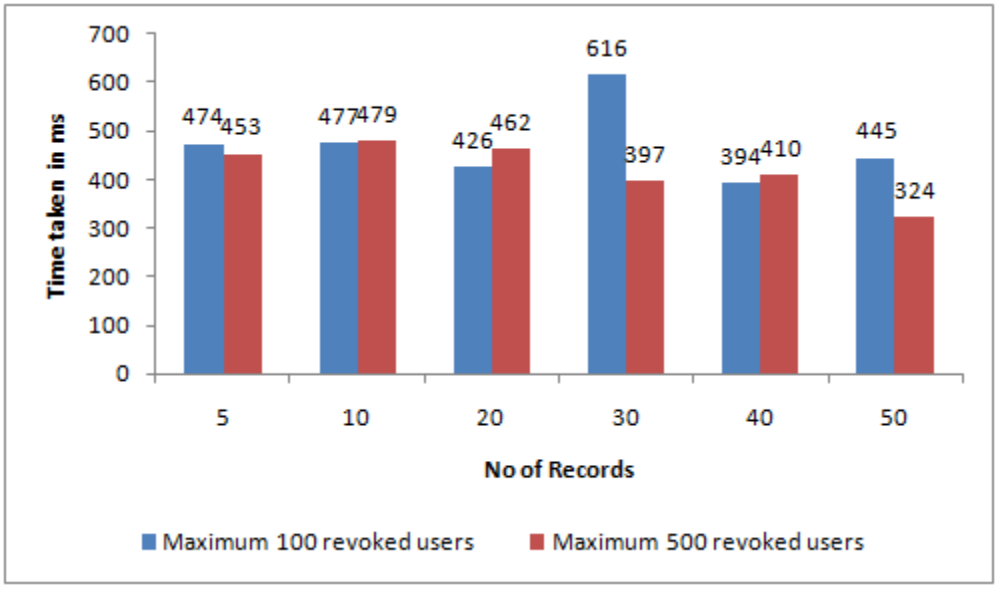

Figure 5.9: Encryption time with varying number of records

Figure 5.10 shows the time taken by convert function with varying no of records in the file. The convert function time does not depend on the file size but it does depend on the maximum

number of users that can be revoked. As the number of users that can be revoked from the system increases, time taken by convert function also increases.

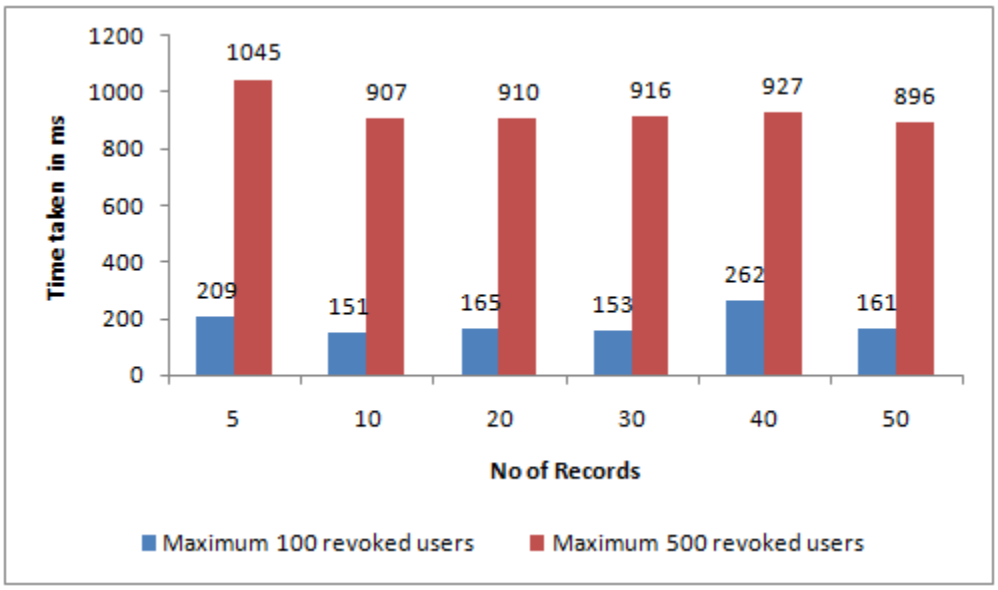

Figure 5.10: Convert time with varying number of records

Figure 5.11 shows the decryption time. Decryption time is independent of the file size and maximum number of users that can be revoked from the system.

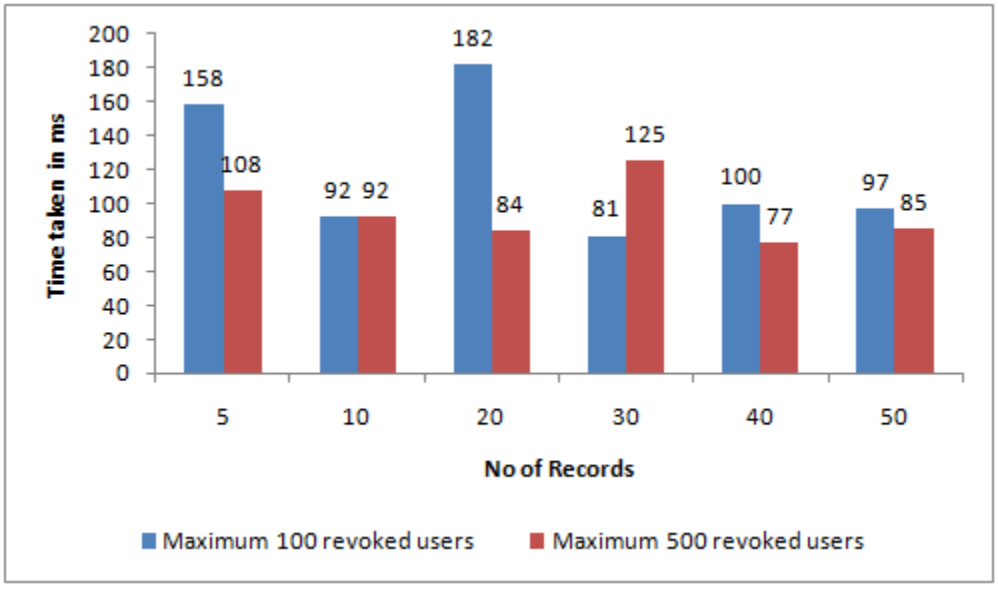

Figure 5.11: Decryption time with varying number of records

Figure 5.12 shows the time taken to generate revoke key. The time taken to generate the revoke key does not depends upon the number of users revoked but the time increases with the increase in number of maximum users that can be revoked from the system.

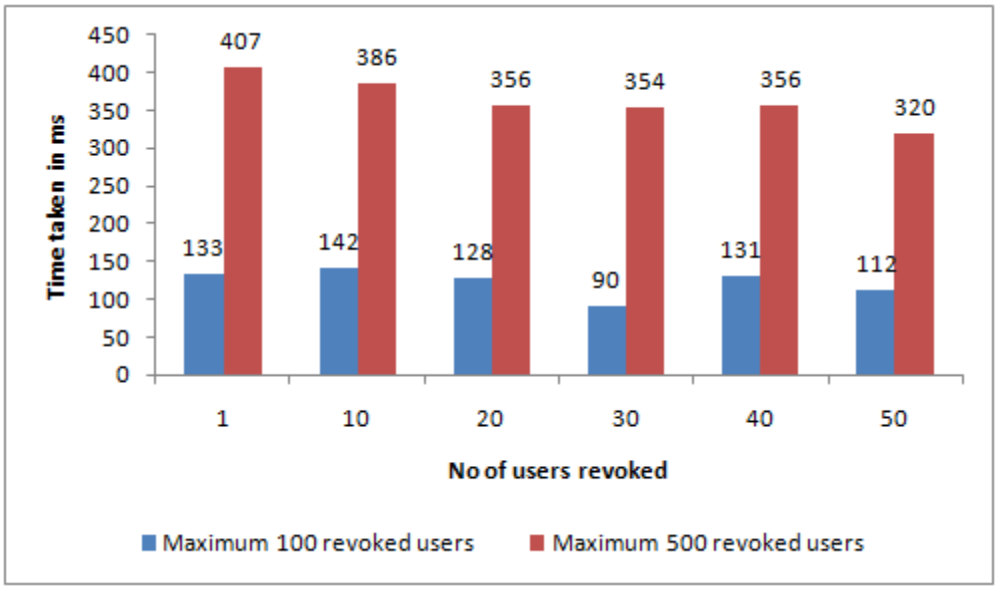

Figure 5.12: revocation key generation time

Figure 5.13 shows the time taken by the convert algorithm. The number of users revoked have no offect on the convert time but, as we increase the number of maximum users that can be revoked, convert time increases.

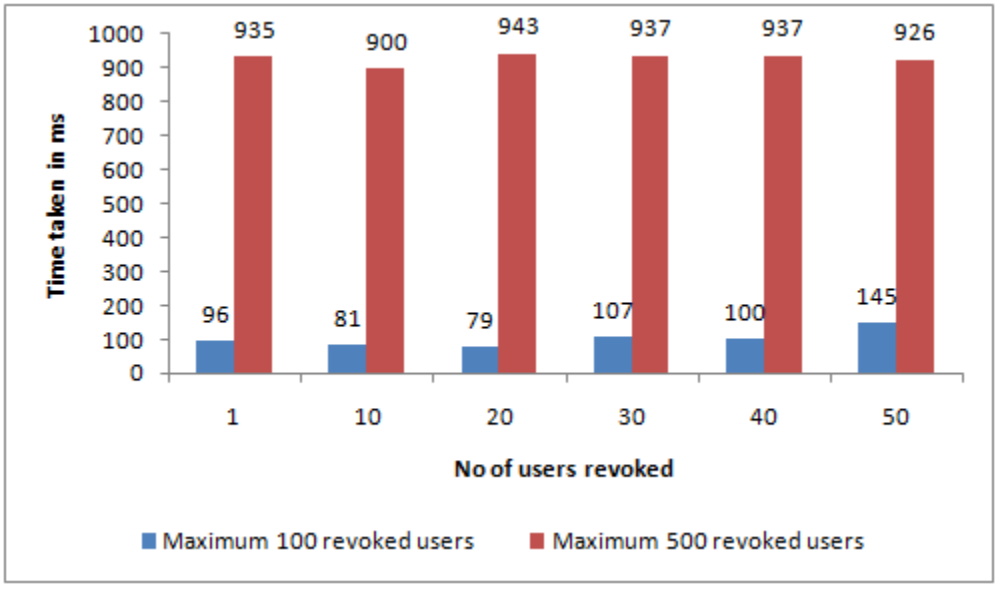

Figure 5.13: Convert time with varying number of users revoked

Figure 5.14 shows the decryption time. Results show that decryption time is independent of the number of users revoked from the system and maximum number of users that can be revoked from the system.

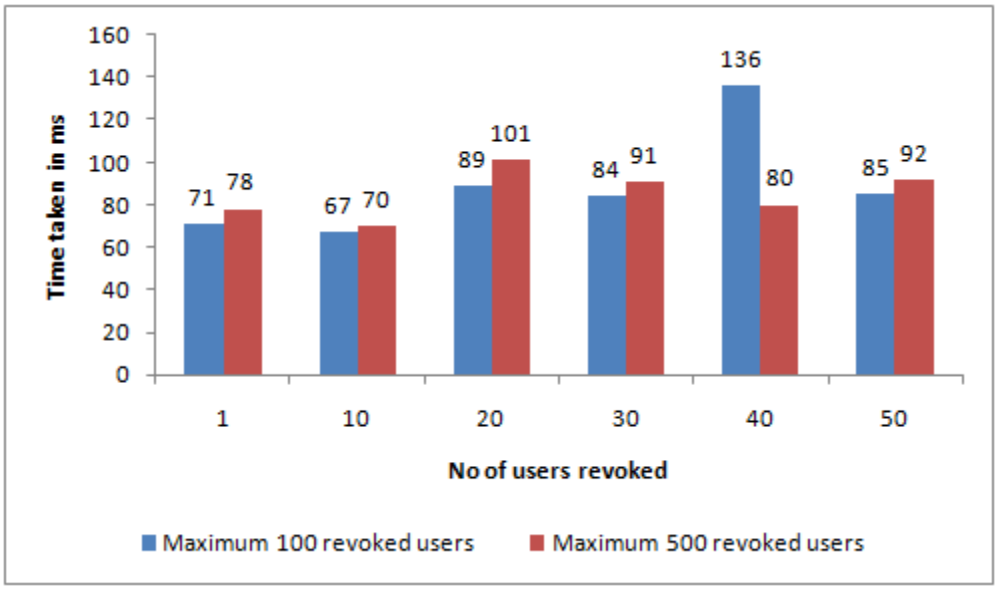

Figure 5.14: Decryption time with varying number of users revoked

CHAPTER 6

CONCLUSION AND FUTURE WORK

One of the most challenging issue in e-Health is keeping the health records of the patient secure. In our work, we proposed a method for secure sharing of health records. In the proposed scheme, the records are encrypted using ciphertext policy attribute based encryption. This cryptographic approach enforces fine grained access control on the data. Along with this approach we added features like revocation, break the glass, delegation and auditing which makes the system even more secure. With the help of revocation, we removed users from the system who were found doing malicious behavior and it also helped in protecting patient records in case of theft to the key. Break the glass is useful in emergency situation where the patient needs immediate attention. Delegation is useful when the medical personnel needs help from another Medical personnel. It helps in providing better healthcare services to the patient. Auditing helps detect malicious behavior, so that proper action can be taken against the attacker. In our system, everyone is not authorized to write on the patient's health records. Patient accepts updated records only from those users who are authorized to update the records.

In the future, we plan to improve the auditing in our system. Some better cryptographic method can be used for generating digital signatures. We also plan to divide the health records of the patient according to different departments and have different access policies according to the department. We also plan to make the system more scalable.

REFERENCES

[1] Sethia, Divyashikha, Daya Gupta, Tanuj Mittal, Ujjwal Arora, and Huzur Saran. "NFC based secure mobile healthcare system." In *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1-6. IEEE, 2014.

[2] *Health Information Privacy.* http://www.hhs.gov/hipaa/ [Accessed January 2016]

[3] Sprague, Lisa. "Personal health records: the people's choice." *NHPF Issue Brief* 820 (2006): 1-13.

[4] Raymond, B., 2007. Realizing the Transformative Potential of Personal Health Records. *Kaiser Permanente Institute for Health Policy, In Focus*

[5] Detmer, Don, Meryl Bloomrosen, Brian Raymond, and Paul Tang. "Integrated personal health records: transformative tools for consumer-centric care."*BMC medical informatics and decision making* 8, no. 1 (2008): 1.

[6] Nass, Sharyl J., Laura A. Levit, and Lawrence O. Gostin. "The Value and Importance of Health Information Privacy." (2009)

[7] Roback, Howard B., and Mary Shelton. "Effects of confidentiality limitations on the psychotherapeutic process." *The Journal of psychotherapy practice and research* 4, no. 3 (1995): 185.

[8] National Bioethics Advisory Commission. "Research involving human biological materials: ethical issues and policy guidance." (1999).

[9] Simmons, Gustavus J. "Symmetric and asymmetric encryption." *ACM Computing Surveys (CSUR)* 11, no. 4 (1979): 305-330.

[10] Agrawal, Monika, and Pradeep Mishra. "A comparative survey on symmetric key encryption techniques." *International Journal on Computer Science and Engineering* 4, no. 5 (2012): 877.

[11] Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM* 21, no. 2 (1978): 120-126.

[12] Fiat, Amos, and Moni Naor. "Broadcast encryption." In *Annual International Cryptology Conference*, pp. 480-491. Springer Berlin Heidelberg, 1993.

[13] Shamir, Adi. "Identity-based cryptosystems and signature schemes." In*Workshop on the Theory and Application of Cryptographic Techniques*, pp. 47-53. Springer Berlin Heidelberg, 1984.

[14] Sahai, Amit, and Brent Waters. "Fuzzy identity-based encryption." In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457-473. Springer Berlin Heidelberg, 2005.

[15] Pang, Liaojun, Jie Yang, and Zhengtao Jiang. "A survey of research progress and development tendency of attribute-based encryption." *The Scientific World Journal* 2014 (2014).

[16] Goyal, Vipul, Omkant Pandey, Amit Sahai, and Brent Waters. "Attribute-based encryption for fine-grained access control of encrypted data." In*Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89-98. Acm, 2006.

[17] Bethencourt, John, Amit Sahai, and Brent Waters. "Ciphertext-policy attribute-based encryption." In *2007 IEEE symposium on security and privacy (SP'07)*, pp. 321-334. IEEE, 2007.

[18] Wang, Guojun, Qin Liu, Jie Wu, and Minyi Guo. "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers."*computers & security* 30, no. 5 (2011): 320-331.

[19] Liu, Dongxiao, Hongwei Li, Yi Yang, and Haomiao Yang. "Achieving multi-authority access control with efficient attribute revocation in smart grid." In*2014 IEEE International Conference on Communications (ICC)*, pp. 634-639. IEEE, 2014.

[20] Ostrovsky, Rafail, Amit Sahai, and Brent Waters. "Attribute-based encryption with non-monotonic access structures." In *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 195-203. ACM, 2007.

[21] Attrapadung, Nuttapong, and Hideki Imai. "Conjunctive broadcast and attribute-based encryption." In *International Conference on Pairing-Based Cryptography*, pp. 248-265. Springer Berlin Heidelberg, 2009.

[22] Pirretti, Matthew, Patrick Traynor, Patrick McDaniel, and Brent Waters. "Secure attribute-based systems." *Journal of Computer Security* 18, no. 5 (2010): 799-837.

[23]Hur, Junbeom, and Dong Kun Noh. "Attribute-based access control with efficient revocation in data outsourcing systems." *IEEE Transactions on Parallel and Distributed Systems* 22, no. 7 (2011): 1214-1221.

[24] Green, Matthew, and Giuseppe Ateniese. "Identity-based proxy re-encryption." In *Applied Cryptography and Network Security*, pp. 288-306. Springer Berlin Heidelberg, 2007.

[25] Yu, Shucheng, Cong Wang, Kui Ren, and Wenjing Lou. "Attribute based data sharing with attribute revocation." In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 261-270. ACM, 2010.

[26] Attrapadung, Nuttapong, and Hideki Imai. "Attribute-based encryption supporting direct/indirect revocation modes." In *IMA International Conference on Cryptography and Coding*, pp. 278-300. Springer Berlin Heidelberg, 2009.

[27] Li, Ming, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou. "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption." *IEEE transactions on parallel and distributed systems* 24, no. 1 (2013): 131-143.

[28] Tong, Yue, Jinyuan Sun, Sherman SM Chow, and Pan Li. "Towards auditable cloud-assisted access of encrypted health data." In *Communications and Network Security (CNS), 2013 IEEE Conference on*, pp. 514-519. IEEE, 2013.

[29] Brucker, Achim D., Helmut Petritsch, and Stefan G. Weber. "Attribute-based encryption with break-glass." In *IFIP International Workshop on Information Security Theory and Practices*, pp. 237-244. Springer Berlin Heidelberg, 2010.

[30] Carminati, Barbara, Elena Ferrari, and Michele Guglielmi. "Secure information sharing on support of emergency management." In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pp. 988-995. IEEE, 2011.

[31] P. Szolovits, J. Doyle, W. J. Long, I. Kohane, and S. G. Pauker. Guardian angel: Patient-centered health information systems. Technical report, 1994.

[32] Google Inc. Google health. https://www.google.com/health/, 2009 [Accessed April, 2016].

[33]Microsoft. Microsoft healthvault. http://www.healthvault.com/personal/ websites-overview.html, 2009 [Accessed April, 2016].

[34] Narayan, Shivaramakrishnan, Martin Gagné, and Reihaneh Safavi-Naini. "Privacy preserving EHR system using attribute-based infrastructure." In*Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, pp. 47-52. ACM, 2010.

[35] Ibraimi, Luan, Muhammad Asim, and Milan Petković. "Secure management of personal health records by applying attribute-based encryption." In*Wearable Micro and Nano Technologies for Personalized Health (pHealth), 2009 6th International Workshop on*, pp. 71-74. IEEE, 2009.

[36] Ibraimi, Luan, Milan Petkovic, Svetla Nikova, Pieter Hartel, and Willem Jonker. "Mediated ciphertext-policy attribute-based encryption and its application." In *Information security applications*, pp. 309-323. Springer Berlin Heidelberg, 2009.

[37]Introduction to MongoDB. https://docs.mongodb.com/manual/introduction/. [Accessed March, 2016]

[38] Jahid, Sonia, and Nikita Borisov. "Piratte: Proxy-based immediate revocation of attribute-based encryption." *arXiv preprint arXiv:1208.4877* (2012).