IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 41-50
www.iosrjournals.org

Securing Phr Using Different Attribute Based Encryption

Jeena Kuriakose[1], Gayathry K. V[2], Dr Suvanam Sasidhar Babu[3]
[1.] *Department of CSE, SNGCE, Kadayiruppu, Kerala, India*
[2.] *Department of CSE, SNGCE, Kadayiruppu, Kerala, India,*
[3.] *Department of CSE, SNGCE, Kadayiruppu, Kerala, India,*

Abstract: *PHR system allows patient to share their personal health records in a centralized way. PHR records are outsourced to any third party server so that these will be easily accessible to the users. Patients normally share the records with wide range of users including relatives, friends, doctors, nurses and so on. In order to avoid the complexity of handling users, the system divides the users into two different domains namely personal domain and public domain. The former consists of family and friends while the latter consists of doctors, pharmacists, researchers etc. Since the data is outsourced to third party such as cloud providers, data security and privacy remains main issue. Security of data is enhanced by encrypting the data before being outsourced. In this paper two types of encryption were proposed, HMASBE to encrypt the records shared to public domain users and AMIRBE to encrypt records shared to personal domain users. Data anonymization is used to ensure the privacy of shared data.*

Keywords: *Anonymization, Encryption, Personal domain, Public domain, PHR, Privacy, Security*

I. Introduction

Personal health records PHR has got much interest in our information society. PHR enable patient to create and share their sensitive health records which is normally outsourced to third party providers. According to Kaelber, personal health record is a set of computer based tools that allow people to access and coordinate their lifelong health information and make appropriate parts of it available to those who need it [1]. PHRs are not restricted by any file formats or local issues. PHR services are outsourced to many third party providers because of high cost of building specialized data centers. Security and privacy remains the main issues regarding the adoption of PHRs by common people despite of the standards and numerous initiatives by industry to provide the interoperability across different PHR and EHR services. Patients always had a fear of their sensitive health data being leaked or misused.

PHR data consists of sensitive and confidential data of a patient regarding diseases, treatments, results etc. Evenif the health care regulations such as HIPPA exists, many CSPs are not covered entities and also they are being directly targeted by malicious attackers through which they can expose the PHR data. For example, an employee without authorization took the data from Department of Veterans Affairs database containing sensitive data of 26.5 million military veterans including their social security numbers and health problems [2]. In order to protect the sensitive data in PHR records, best way is to encrypt the data before being outsourced to the cloud.

PHR owner can decide how to encrypt the data and to whom the data should be shared. Only users having correct decryption key can access the files, while for others they remain confidential. Users need the PHR either for professional purpose or for personal purpose. The system divides the set of users into two domains namely public and personal. Friends and relatives belong to personal domain and doctors, pharmacists and researchers belong to public domain. In the simplest method [3] PHR can be encrypted using symmetric key encryption scheme such as AES and users who are authorized to access their data will get the symmetric key. However,certificate management is complex and costly in these method.

The main focus of the paper is to provide secure sharing of personal health records stored on semitrusted servers along with addressing of complicated key management issues. Two different encryption techniques were adopted for encrypting the data on public and personal domains. For encrypting records of personal domain, anonymous multi-receiver identity-based encryption (AMRIBE) scheme is used. With this encryption technique, only authorized users whose identities belong to dedicated identities can decrypt the data. Hierarchical and multi-authority attribute-sets based encryption (HM-ASBE) is used to encrypt the records shared to public domain users. Patients selectively share their PHR data with the users using the access policies which are expressed based on the attributes of users or data.

Data anonymization is used to provide privacy to shared data. Through anonymization, data can be made worthless to anyone except the owner of the data. It transforms data in such a way that the detecting the key information will be prevented. As so much of data is stored on the cloud, Data Aggregation and

Deduplication algorithm is used. Data aggregation is an information mining process that searches, gathers and presents a summarized report to achieve specific business objectives. Data deduplication is a data compression method that removes duplicate copies of repeated data. Security is ensured by encrypting the PHRs before outsourcing to the cloud.

The rest of the paper is organized as follows. Section 2 will give a brief study on the related work. Section 3 describes existing system. The details of proposed system will be taken on Section 4 and Section 5 will conclude the discussion.

II. Related Work

A. Boldyreva et al [4] proposed identity based encryption to realize fine grained access control for outsourced data. Identity (ID)-based encryption, is an exciting alternative to public-key encryption, which eliminates the need for a Public Key Infrastructure (PKI) that makes publicly available the mapping between identities, public keys, and validity of the latter.

The senders using an IBE do not need to look up the public keys and the corresponding certificates of the receivers, because the identities (e.g. emails or IP addresses) together with common public parameters are sufficient for encryption. The private keys of the users are issued by a trusted third party called the private key generator (PKG). In terms of encryption and decryption, their construction is slightly less efficient than the existing IBE schemes.

In S. Yu et al [5], they propose Ciphertext-Policy Attribute Based Encryption. In CP-ABE, each user is associated with a set of attributes and data are encrypted with access structures on attributes. A user is able to decrypt a ciphertext if and only if his attributes satisfy the ciphertext access structure. Beside this basic property, practical applications usually have other requirements. In this paper they focus on an important issue of attribute revocation which is cumbersome for CP-ABE schemes. These solutions can just disable a user secret key at a designated time, but are not able to revoke a user attribute/key on the ad hoc basis.

Lewko and Waters et al [6] propose a Multi-Authority Attribute-Based Encryption (ABE) system. In the system, any party can become an authority and there is no requirement for any global coordination other than the creation of an initial set of common reference parameters. A party can simply act as an ABE authority by creating a public key and issuing private keys to different users that reject their attributes. A user can encrypt data in terms of any Boolean formula over attributes issued from any chosen set of authorities. Finally, system does not require any central authority. They thus avoid the performance bottleneck incurred by relying on a central authority, which makes our system more scalable. There is a crucial improvement for efficiency as well as security, since even a central authority that remains uncorrupted may occasionally fail for benign reasons, and a system that constantly relies on its participation will be forced to remain stagnant until it can be restored. In the system, authorities can function entirely independently, and the failure or corruption of some authorities will not affect the operation of functioning, uncorrupted authorities. This makes our system more robust.
However, this construction does not lend itself to a proof under a non-interactive assumption.

III. Existing System

Existing system considers a PHR system where the user can share their data with those they need it. CP-ABE is utilized to encrypt the data before being outsourced. Main actors of PHR system are doctors, hospital and user. Admin will control the overall actions. He/She is responsible for approving hospital, adding state or country, approving the request in case of emergency login and so on. Users can upload and download the records once they are logged in. Uploading consists of generation of policies which are used to construct the access structure. Based on this, the permission for downloading will be granted.

User and owner can login to the phr system with a valid username and password. New user can be registered to the system too. Owner can open, update and upload data to the cloud storage. User can search data on the cloud. Since the data in the cloud are encrypted for security, the user needs a key to decrypt the data. User gets the secret key from the owner in 2 ways, either the owner send the key to the users at the time of encryption itself or the user gets the key on request. If the key is valid, then the user can access the data, otherwise request is rejected. In case of emergency situation, it is necessary to violate the regular access policy which is specified by the break access process.

Data can be encrypted using CP-ABE scheme as follows. Access policy is defined over a set of attributes and the data owner encrypts the data based on the access policy generated. Those users whose secret key associated with set of attributes satisfying the access policy can only decrypt the encrypted data. Keys are associated with set of attributes and ciphertexts are associated with access policy.

The system make use of a single trusted authority. Because of this the system suffers from key escrow problem as the TA can access all the encrypted data in addition of load bottleneck. Delegating the tasks such as certifying the user attributes or roles and generating secret keys to one authority is not always practical.

IV. Proposed System

The proposed system provide secure patient centric PHR access and efficient key management at the same time. In order to avoid key management issue, the users of the system is divided into two domains public and personal. Users who have some contacts to the PHR owner belongs to personal domain, while for those related to profession belongs to public domain. HM-ASBE [7] is used in public domain in which TA is responsible for system initialization and key distribution and management of subordinate domain authorities. Attribute authority belonging to each domain is responsible for managing the user's attributes. AMRIBE [8] is used to encrypt data in personal domain.

4.1 Using HM-ASBE in the public domain

If user attribute set is $A_u = \{ A_1 , A_2 , \ldots , A_n \}$, system will automatically divides it into K mutually disjoint subsets corresponding to each AA and it satisfies:

$$\sum_{k=1}^{K} A_u^k = A_u$$

For example, the set is {Name: Jack, ID: 30202, {Age: 34:Sex: Male, {Location: USA, Job: Physician}}}, then {Name: Jack, ID: 30202} will be managed by TA, {Age: 34: Sex: Male} and {Location: USA, Job: Physician} will be distributed to corresponding AAs.

HM-ASBE scheme consists of the following four procedures: System Setup, Key Generation, Encryption Algorithm and Decryption Algorithm.

4.1.1 System Setup

Choose a bilinear group G_0 of prime order p and let g be a generator of G_0. The bilinear map e: $G_0 \times G_0 = G_1$ .Every user and authority in our system is assigned a unique ID called GID and AID.

User's digital signature is usually used to represent GID, so that every AA can verify the identity of users.

AID is defined as follow:

AID(A) = {ParentA(AA),index(AA)},

ParentA(AA)$\in Z_P$ means the parent of AA and index(AA)$\in Z_P$ means the random sequence distributed by TA.

The Setup algorithm sets the level of recursive times and chooses random exponents a , $\{ \beta_1 , \beta_2$

$, \ldots , \beta_{depth} \} \in Zp$. To simplify the problem we set level = 2, the algorithm sets the public key PK_0 and master key MK_0 as:

$$PK_0 = \{G_0, g, f_1 = g^{\frac{1}{\beta_1}}, h_1 = g^{\beta_1}, f_2 = g^{\frac{1}{\beta_2}}, h_2 = g^{\beta_2}, e(g,g)^{\alpha}\} \quad (1)$$

$$MK_0 = \{\beta_1, \beta_2, g^{\alpha}\}$$

First-level Authority Authorization: A first-level authority is associated with a unique *AID* and an attribute set $\Lambda = \{A_1,A_2,\ldots A_n\}$ with A_0 being the first level and $Ai, 1 \leq i \leq n$ being the second level, $A_i = \{A_{i,1},A_{i,2},\ldots A_{i,m}\}$. When TA authorizes the AA, $r \in Z_P$ is randomly chosen to

represent the set Λ, $r_i \in Z_P$ being $A_i \in \Lambda$, and $r_{i,j}$

$\in Z_P$ being $a_{i,j} \in Ai$, $0 \leq i \leq n$, $1 \leq j \leq m$. The first-level authority's master key is:

$$MK = \{\Lambda, D = g^{\frac{\alpha+r}{\beta_1}}, D_{i,j} = g^{r_i} \cdot H(a_{i,j})^{r_{i,j}},$$
$$D'_{i,j} = g^{r_{i,j}}, E_i = g^{\frac{r+r_i}{\beta_2}} (0 \le i \le n, 1 \le j \le m)\} \quad (2)$$

In the above function,E_i is for translation from r'_i to r_i. Elements E_i and E'_i can be used as E_i / E'_i to translate r'_i to r_i at the translating nodes.

Subordinate Authority Authorization: When a new subordinate domain authority denoted as

AA_{k+1} wants to join the system, the administrating AA denoted as AA_k will verify

the AA's AID before authorization. Let Λ and Λ^ be the AA_k's and AA_{k+1}'s attribute sets. AA_k randomly chooses ŕ ∈ ZP for Λ^, ŕ i ∈ Z_P for $\hat{A}_i$

∈ Λ^ , and ŕ i,j ∈ Z_P for $\hat{a}_{i,j} \in \hat{A}_{i,}$, 0≤i≤n, 1≤j≤m .Then the master key of *AAk+1* is
computed as follows:

$$MK_{k+1} = \{\hat{\Lambda}, \hat{D} = D \cdot f_1^{\hat{r}}, \hat{E}_i = E_i \cdot f_2^{\beta+r_i},$$
$$\hat{D}_{i,j} = D_{i,j} \cdot g^{\hat{r}_i} \cdot H(\hat{a}_{i,j})^{\hat{r}_{i,j}}, \quad (3)$$
$$\hat{D}'_{i,j} = D'_{i,j} \cdot g^{\hat{r}_{i,j}} (0 \le i \le n, 1 \le j \le m)\}$$

In the above MK_{k+1} , D ,$D_{i,j}$ and E_i are corresponding to the items in MK_k of AA_k .

4.1.2 Key Generation

Let A_u be the user's attribute set, and it is managed by K authorities. Let $Au^{(k)}$={$Au_0^{(k)}$, $Au_1^{(k)}$,…… $Au_n^{(k)}$ } be the attribute set on the

kth(k ≤ K) authority. First of all, AA_k uses pseudo random function (PSK) to generate the key component $au_{(k)} = P_{sk}(u)$ for user according to user's GID and AA_k's AID_k. Then AA_k randomly $ru^{(k)} \in ZP$ for $Au^{(k)}$, $ru_i^{(k)} \in ZP$ for $Au_i^{(k)}$ $\in Au^{(k)}$, and $ru_{i,j}^{(k)} \in ZP$ for $au_{i,j}^{(k)} \in Au_i^{(k)}$
.Finally the key component on is generated as:

$$SK_u^{(k)} = \{VS^{(k)} = g^{index(AA_k)}, Au^{(k)},$$
$$Du^{(k)} = g^{\frac{au_{(k)}+ru^{(k)}}{\beta_{k,1}}},$$
$$Du_{i,j}^{(k)} = g^{ru_i^{(k)}} \cdot H(au_{i,j}^{(k)})^{ru_{i,j}^{(k)}}, \quad (4)$$
$$Du'^{(k)}_{i,j} = g^{ru_{i,j}^{(k)}}, E_i^{(k)} = g^{\frac{ru^{(k)}+ru_i^{(k)}}{\beta_{k,2}}}$$
$$(0 \le i \le n, 1 \le j \le m)\}$$
$$\beta_{k,1} = \frac{\beta_1}{\alpha + r + \hat{r}}, \beta_{k,2} = \frac{\beta_2}{r + r_i + \hat{r} + \hat{r}_i}.$$

So the user's private key is described as follow:

$$SK_u = \{\{SK_u^{(k)}\}_{k=1}^K, D_{user} = g^{(\alpha+\sum_{w=1}^{W} au_{(w)})/\sum_{w=1}^{W}\beta_{w,1}}\},$$

where D_{user} is user's decryption key distributed by TA.

$$CT^{(w)} = \{VC^{(w)} = g^{index(AA_w)}, \Gamma^{(w)}, C^{(w)} = h_{w,1}^{\theta},$$
$$\overline{C}^{(w)} = h_{w,2}^{\theta}, \forall y^{(w)} \in Y^{(w)} : C_y^{(w)} = g^{q_y(0)},$$
$$C_y^{r(w)} = H(attr(y^{(w)}))^{q_y(0)},$$
$$\forall x^{(w)} \in X^{(w)} : \hat{C}_x^{(w)} = h_{w,2}^{q_x(0)}\} \quad (5$$

4.1.3 Encryption Algorithm

Data owner should encrypt the data M before uploading it to cloud system. Firstly the system will divide the access control strategy into W pieces according to the attribute sets each authority managed.

Let $\{\Gamma^{(w)}\}^{W}_{w=1}$ be the set of access control trees. At first a random number $\theta \in Z_P$ is chosen and the ciphertext is computed as $\hat{C} = M.e(g,g)^{a\theta}$. Let

$\Gamma^{(w)}$ be the access control tree on $AA_{(w)}$, for each node $x^{(w)}$ in $\Gamma^{(w)}$, is associated with a polynomial

q_x. For converging node, the order of q_x is $d_x=k_x -1$ where k_x is the threshold and for leaf node, the order is $d_x=0$. For each non-root node, $q_x(0)$

$= q_{parent(x)}(index(x))$ and the other d_x points of q_x are randomly chosen. For root node, $q_R(0) = \theta$, $\theta \in Z_P$ and other d_R points of q_R are randomly

chosen. Lagrange polynomial is used to compute the q_x. Let $Y^{(w)}$ be the set of all leaf node $y^{(w)}$ and $X^{(w)}$ be the set of all converging node $x^{(w)}$,
the cipher on $AA_{(w)}$ is generated as :

The encryption algorithm is repeated on the rest of (W-1) authorities and the ciphertext of M and its access control strategy is generated as follows:

$$C = \{\{\Gamma^{(w)}\}_{w=1}^{W}, \tilde{C} = M \cdot e(g,g)^{\alpha\theta}, \{CT^{(w)}\}_{w=1}^{W}\}$$

4.1.4 Decryption Algorithm

Assume that access control strategy is distributed on W AAs and user has K components of the key. Only when $K \geq W$ can the algorithm continue.

The algorithm checks if $VS=VK$. Let $CT^{(w)}$ be the ciphertext and $SK_u^{(w)}$ be the user's key on wth

AA. Decryption algorithm runs recursive function *Tree*(*Au*) to verify that the $SK_u^{(w)}$ satisfies the $\Gamma^{(w)}$. Each node x in the access control tree is associated with a set S_x of labels that returned by *Tree*(*Au*). If $Au^{(w)}$ does not satisfy the $\Gamma^{(w)}$, then *Tree*(*Au*) returns $\perp$, otherwise the decryption algorithm selects one of the labels, $i \in S$, and calls a recursive function on the root node of the tree denoted as DecryptNode($CT^{(w)}$, $SK_u^{(w)}$, x, i) [9]. The bottom-up recursion returns the result of DecryptNode($CT^{(w)}$, $SK_u^{(w)}$, R, i) on root node *R*.

$$F_R^{(w)} = \begin{cases} e(g,g)^{ru_i^{(w)} \cdot q_R(0)} = e(g,g)^{ru_i^{(w)} \cdot s}, i \neq 0 \\ e(g,g)^{ru^{(w)} \cdot q_R(0)} = e(g,g)^{ru^{(w)} \cdot s}, i = 0 \end{cases}$$

When $i \neq 0$, $F_R^{(w)}$ is computed as follows:

$$F^{(w)} = \frac{e(\hat{C}_r^{(w)}, E_i^{(w)})}{F_R^{(w)}} = e(g,g)^{ru^{(w)} \cdot \theta} \qquad (6)$$

$$Q = \prod \frac{e(C^{(w)}, Du^{(k)})}{(\ )} = e(g,g)^{\sum au_{(\ )}} \qquad (7)$$

If the user satisfies all W access control strategies on each authority, which means $\{F^{(w)}\}^{W}_{w=1}$ has no null value, then the algorithm proceeds as follows:

And $e(g,g)^{a\theta}$ is computed as follows:

$$e(g,g)^{\alpha\theta} = \frac{e(\prod_{w=1}^{W} C^{(w)}, D_{user})}{Q} \qquad (8)$$

Finally the correctly decrypted data can be obtained:

$$M = \frac{\tilde{C}}{e(g,g)^{\alpha\theta}} = \frac{M \cdot e(g,g)^{\alpha\theta}}{e(g,g)^{\alpha\theta}} = M \qquad (9)$$

4.2 Using AMRIBE in personal domain

The TA acts as the root of trust and is responsible for generating system parameters, issuing attribute private key or identity private key for PHR owners and PHR users. The CSP manages a cloud to provide data storage service. It is important to assume that CSP is semi-trusted, which means the CSP will try to find out as much secret information in the stored PHR data as possible, but it will honestly follow the protocol in general. PHR data for the personal domain is encrypted using AMRIBE scheme [10]. Only authorized PHR users whose identities belong to dedicated identities can decrypt the encrypted PHR data.

4.2.1 Setup:

The TA performs the following steps.

1) Define the universe of attributes

$\Omega = \{a_1, a_2, \ldots, a_n\}$.

2) Run the bilinear group generator $G(1^k)$ to get a prime order bilinear group $(p, \mathbf{G}1, \mathbf{G}2, \hat{e}, g)$.

3) Choose $x \leftarrow^{\$} \mathbf{Z_p}^*$ and $y \leftarrow^{\$} \mathbf{Z_p}^*$

4) Compute $h = g^x$ and $Y = \hat{e}(g, g)^y$.

5) Choose $h_i \leftarrow^{\$} \mathbf{G_1}$ for $1 \leq i \leq n$.

6) Pick three secure cryptographic hash functions H_1, H_2 and H_3, and a semantically secure symmetric encryption scheme with the encryption algorithm E_k and decryption algorithm D_k, where the key $k \in \mathbf{K} = \{0, 1\}^k$.

7) TA publishes the system parameters as $mpk =$

$\{\mathbf{\Omega}, p, \mathbf{G_1}, \mathbf{G_2}, \hat{e}, g, h, Y, h_1, \ldots, h_n, H_1, H_2, H_3, E_k, D_k\}$

8) TA sets the master secret key as $msk = \{x, g^y\}$.

4.2.2 KeyGen:

Given a user's identity ID, and a set of attributes

$\mathbf{S} \subseteq \mathbf{\Omega}$ belonging to the user, the TA performs the following steps.

1) Compute $g_{ID} = H_1(ID)$ and $K_{ID} = g^{x}_{ID}$. 2) Set the user's private key as

$$sk_{ID,\mathbf{S}} = \langle K_{ID}, K, L, \{K_i\}_{\mathbf{a}_i \in \mathbf{S}} \rangle .$$

3) Finally, TA sends $sk_{ID,S}$ to the user via a secure channel.

4.2.3 Encrypt:

Given an original PHR data $msg \in \{0, 1\}^*$ to be encrypted, a LSSS access structure A = ($\mathbf{M}_{l*n}$, ρ) and a list of identities $\mathbf{ID_R}$ = ($ID_1, \ldots, ID_m$), PHR owner performs the following steps:

1) Choose a random session key $k_2 \leftarrow^{\$} \{0, 1\}^k$, and compute $E_2 = E_{k2}$ (msg).

$$\begin{aligned} g_{ID_i} &= H_1(ID_i) \\ v_i &= H_2(\hat{e}(g_{ID_i}, h)^s) \end{aligned}$$

2) For $ID_i \in \mathbf{ID_R}$, compute

3) Construct the following polynomial $f(x)$ with degree m, and set $C_2' = (c_0, c_1, \ldots c_{m-1})$

$$\begin{aligned} f(x) &= \prod_{i=1}^{m} (x - v_i) + k_2 \bmod p \\ &= c_0 + c_1 x + \cdots + c_{m-1} x^{m-1} + x \end{aligned}$$

4) Set the ciphertert

$$CT = \langle E_2, C', \mathrm{C}_2' \rangle .$$

5) Finally, PHR owner uploads the ciphertext to the CSP along with a description of access

policy ($\mathbf{M}_{l*n}$, ρ) and a set of identities of designated recipients $\mathbf{ID_R}$.

4.2.4 Decrypt:
Given a ciphertext CT along with a description of access policy A = ($\mathbf{M}_{l*n}$, ρ) and a set of identities $\mathbf{ID_R}$, a PHR user performs following steps:

1) Compute

$v_i = H_2(\hat{e}(K_{IDi}, C'))$
$k_2 = f(v_i) \bmod p$

$= c_0 + c_1 v_i + \ldots + c_{m-1} v_i^{m-1} + v_i^m$

2) Decrypt $msg = D_{k2}(E_2)$.

4.3 Data Anonymization

For example, consider Table I to be the PHR to be shared. The records in Table I can be categorized into three namely explicit identifiers, quasi identifiers and sensitive identifiers [11]. The explicit identifier contains attribute which can uniquely identify a person (for eg: Name), the quasi identifiers are attributes that helps to identify a person when it is linked with other records (for eg: gender, age, zip code) and sensitive identifiers are attributes that contain sensitive information (for eg: disease). The hospital wants to share these records in such a way that it remains practically useful at the same time identity of a person should not be leaked. Therefore it is necessary to anonymize the data to be shared. Privacy of the patient is preserved by using simple techniques like generalization and suppression. The generalization is a process of replacing the quasi

identifiers with less specific values and suppression is a process of removing the fields to reduce the information content. The Table II shows the result of anonymization [12].

Patient Name	Gender	Age	Zip code	Health Problem
Kumar	Male	36	400071	Viral Infection
Logesh	Male	37	440121	Heart Problem
Pankaj	Male	37	400182	Cancer
Suresh	Male	38	400096	Viral Infection
Shesha	Male	43	400022	Cancer
Uma	Female	45	400135	Flu
Sanjay	Male	46	400135	Heart Problem
Sharan	Male	52	400493	Viral Infection
Ramya	Female	53	440672	Flu
Pallavi	Female	58	440123	Flu
Naresh	Male	59	400135	Viral Infection

Table 4.1: Original data

Gender	Age	Zip code	Health Problem
Male	31<=40	400*	Viral Infection
Male	31<=40	440*	Heart Problem
Male	31<=40	400*	Cancer
Male	31<=40	400*	Viral Infection
Male	41<=50	400*	Cancer
Female	41<=50	400*	Flu
Male	41<=50	400*	Heart Problem
Male	51<=60	400*	Viral Infection
Female	51<=60	440*	Flu
Female	51<=60	440*	Flu
Male	51<=60	400*	Viral Infection

Table 4.2: Anonymized data

4.4 Data Aggregation and Deduplication (DAD)

In general data aggregation is an information mining process that searches, gathers and presents a summarized report to achieve specific business objectives. Data deduplication is a data compression method that removes duplicate copies of repeated data.

For example consider Table II, here records 1and 4, 8 and 11, 9 and 10 are identical. To save the cost of cloud storage and minimize the information loss, DAD algorithm compares each record with all other records. If repeated records are found during comparison then they are eliminated and their count is incremented to the corresponding record as shown in Table III. Since we are dealing with huge data, DAD algorithm reduces the cost of cloud storage to a great extent [12].

Gender	Age	Zip code	Health Problems	Count
Male	31<=40	400*	Viral infection	2
Male	31<=40	440*	Heart Problem	1
Male	31<=40	400*	Cancer	1
Male	41<=50	400*	Cancer	1
Female	41<=50	400*	Flu	1
Male	41<=50	400*	Heart Problem	1
Male	51<=60	400*	Viral Infection	2
Female	51<=60	440*	Flu	2

Table 4.3: Result of Data Aggregation and Deduplication

Algorithm: DAD Algorithm Input: Anonymized dataset
Output: Aggregated and deduplicated dataset

Process:

1. for i {1…...n} do // n - No. of Records
2. for j {i+1…..n}do
3. flag=0;
4. for k {1….f} do // f – No of fields
5. if (data[i][k]!=data[j][k])
6. set flag=1;
7. end
8. end
9. if (flag==0)
10. data[i].count++;
11. remove data[j];
12. end
13. end
14. End

V. Conclusions

PHR system help the users to share their medical records to any one in more easier way. In order to avoid the complexity of handling the wide set of users, the whole system is divided into two domains namely public and personal domain where former consists of doctors,nurses, pharmicists etc and latter consists of relatives, friends. Security of the uploaded records are ensured by encrypting the data before being outsourced. HMASBE and AMRIBE are used as the main encryption technique. Privacy is enhanced with anonymization. The proposed framework reduces key complexity along with ensuring security and privacy.

References

[1]. D. C. Kaelber, A. K. Jha, D. Johnston, B. Middleton, and D. W. Bates, "Viewpoint paper:Aresearch agenda for personal health records (PHRs)," J. Amer. Med. Inform. Assoc., vol. 15, no. 6, pp. 729–736, 2008.
[2]. "At Risk of Exposure - in the Push for Electronic Medical Records, Concern Is Growing About How Well Privacy Can Be Safeguarded," http://articles.latimes.com/2006/jun/26 /health/ he-privacy26, 2006.
[3] E. Smith and J. Eloff, "Security in health-care information systems - current trends," International Journal of Medical Informatics, vol. 54, no. 1, pp. 39–54, April 1999
[4]. A. Boldyreva, V. Goyal, and V. Kumar, "Identity-Based Encryption with Efficient Revocation," Proc. 15th ACM Conf. Computer and Comm. Security (CCS), pp. 417-426, 2008.
[5]. S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. Fifth ACM Symp. Information, Computer and Comm. Security (ASIACCS '10), 2010.
[6]. A. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption," EUROCRYPT: Proc. 30th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology, pp. 568-588, 2011.
[7]. CHEN Danwei, CHEN Linling, "Securing Patient-Centric Personal Health Records Sharing System in Cloud Computing", China Communications●Supplement No.1 2014
[8]. Chang-Ji Wang, Xi-Lei Xu, "An Efficient Cloud-based Personal Health Records System Using Attribute-Based Encryption and

[9]. Anonymous Multi-Receiver Identity-Based Encryption", Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2014
[10]. Y. Tseng, Y. Huang, and H. Chang, "Cca-secure anonymous multi-receiver id-based encryption," in 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA). Fukuoka: IEEE, March 2012, pp. 177–182.
[11]. Y. Tseng, Y. Huang, and H. Chang, "Cca-secure anonymous multi-receiver id-based encryption," in 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA). Fukuoka: IEEE, March 2012, pp. 177–182.
[11]. Apeksha Sakhare and Swati Gana, "Anonymization: A Method To Protect Sensitive Data In Cloud," International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May-2013.
[12]. G.Logeswari, D.Sangeetha," A Cost Effective Clustering based Anonymization Approach for Storing PHR's in Cloud", International Conference on Recent Trends in Information Technology, 2014