Patient-Controlled Personal Health Record
Enforcing
Patient Privacy in Cloud based Healthcare System

1C. Menaka and 2R.S. Ponmagal
1Department of Computer Science, Bharathiar University, Coimbatore, India.
Menu_mca_03@yahoo.co.in
2Department of Computer Science and Engineering, Dr.M.G.R. Educational and Research Institute University, Maduravoyal, Chennai.
sponmagal@gmail.com

Abstract

Personal Health Record (PHR) through Cloud computing is a new technology reaching its fame off late. Cloud computing has done miracles in the world of computers and internet through the concept of virtualization. In the field of Personal Health Record (PHR), cloud server acts as a third party server to store the records. The major concerns involed here is with fine-grained access, cryptography, key management and diversifying the availability of information during on-demand user request. This paper provides a secure sharing of patient health information through cloud computing. Cloud-based PHR system allows patients to secure their sensitive data on a semi-trusted cloud based service provider with various other security protecting attributes. In Cloud computing the information is available to a large group of crowd and customizing them to specified users like doctors, nurses, friends, relatives or family members is highly possible. In our proposed paper the PHR owner has the rights to encrypt their data in public domain using AES Encryption Scheme and it is re encrypted using Elliptic Curve Cryptography algorithm to strengthen the security. We have also introduced Hybrid Crypto Secure Secret Sharing Algorithm when the owner sends the secret code to random shares to the requester who wants access to the PHR. The algorithm is designed to allow the user with the correct key to access the
decrypted data else only the encrypted version of the information is available. Our proposed algorithm not only allows the user to secure the information but also share the information to the concern professionals or users.

**Key Words:** Personal health record, cloud computing, AES encryption, elliptic curve cryptography algorithm, hybrid crypto secure secret sharing algorithm.

1. **Introduction**

Cloud computing is a new evolution in technology which introduces the concept of sharing resources on demand. It reduces the presence of physical devices and the traditional idea of holding and maintaining the computing systems. It provides a shared pool of resources through virtualization and many more. Hence it requires less or no maintenance and management services. Collaborating this sophisticated technology with Personal Health Record (PHR) is all together a new proposal. However there are several challenges attached to this proposed theorem. Data sharing is secured or rather encrypted twice using AES encryption method and Elliptic Curve Cryptography. Only the encrypted data is stored in the cloud server to prevent data loss. It is highly necessary to have the PHR completely filled with all the information in order to categorise and encrypt them. The frequently used fields are encrypted using AES like the name, age, sex, etc and the remaining fields through Elliptic Curve Cryptography.

PHR is a place to hold all the sensitive and personal information of an individual's medical history. Accessing the files through cloud computing is architecture with high risk intolerance. The file is encrypted through complex methods before it is shared or stored in the cloud server. As there are several black holes in Cloud computing which are still unresolved it is questionable to have personal health information secured in a third party server. Patient-centric framework is proposed in this paper in order to achieve cryptographically enforced data access control and fine grained security through AES encryption techniques and Hybrid Crypto Secure Secret Sharing Algorithm. Unlike the other proposed methods, we divide the PHR records of the users into different security domains and apply multi authority AES. Dynamic modification in the search filter is also possible in this framework.

On-demand user access and break-glass access procedure is also discussed in this paper work.

Cloud eases the access of information from anywhere at any time. Cloud technology also keeps the database up to date and scalable. When it comes to medical information, the history of the patient plays a vital role to decide the further progressment in the treatment. Hence PHR records hold highly sensitive data and needs to be stored at high efficiency. Hence encryption of data is a must in PHR and in our proposed method we use major encryption techniques
which are based on the field of information.

**Our Approach**

As mentioned earlier, we approach cloud computing for storing PHR information. PHR contains patients personal and health record details. This is encrypted using AES encryption algorithm and in turn reencrypted using Elliptic Curve Cryptography algorithm. The PHR data is collected, analyzed and stored in Cloud Storage system. In the process of storing the data in Cloud system there are different keys generated namely public, private and secret key for every PHR file. In layman terms, the method works as the below system representation in Figure.1.

When a person other than the PHR record holders wants to view the record, the person sends a request for access to the record holder or called as the patient. Once the patient receives the request, he/she sends a secret code with random shares and thresholds using Hybrid Crypto Secure Secret Sharing Algorithm. This algorithm is chosen over the other algorithms like Shamir's algorithms and Information dispersal Algorithms due to its efficiency and complexiting in breaking the security.

The Hybrid Crypto Secure Scheme has added advantages which provides both security and performance improvement. Once the user receives the secret code they can access the decrypted PHR.

There is another detailed procedure involved after the generation of the secret code. Once the secret code is inserted in the system, it in turn generates a private key for that particular PHR. This private key will help to generate the specific public and secret key of the concerned PHR. Only after which the user can download or view the requested PHR file.
Figure 1: System Architecture
2. Related Work

Query Auditing

HIPAA- Health Insurance Portability and Accountability Act (HIPAA) of United States- states that users have rights to know who accessed their health record through auditing. Electronic Medical Record- EMR systems is built on the security guidelines of HIPAA [3]. The auditor tracks the suspect who disclosed the user details.

We have a similar auditing done in cloud computing as well which uses SQL statement: AUDIT audit list FROM table list WHERE condition list. audit list refers to the list of columns or privacy attributes; table list refers to the various tables of attributes in audit list and condition list refers to the list of conditions used to describe the patient of the audit.

In reference paper [4] a detailed audit procedure for complex SQL statement is described. Paper [5] helps in analyzing the privacy disclosure using multiple SQL queries. In paper [6] an audit method using XML data query is discussed. The literature work in [7] studies in detail about the audit method implemented on a wrongly set access control policies. In [8] an audit method for data retention is proposed. All the earlier discussed methods study the various audit methods used in EMR systems where the audit methods are used by professional audits. Though the above proposed methods can be implemented on PHR system, it is not user friendly for inexperienced users or in other words for layman users. Hence our main approach in proposing this paper work is to help the common users to access the audit method in PHR systems.

Recommender Systems

Recommender system is a commonly used method in electronic commerce and social network. It helps to generate a list of recommended items for the user based on the previous search history or browse history of that particular user. The recommender system can be implemented using these four approach systems [9]: collaborative filtering, network structure based filtering, content-based filtering and social network based filtering. Among these four methods, collaborative filtering method is in widely used. This method simply uses the concept of collaborating the interest between two common users. If two users have similar taste of interest towards some items then their evaluation towards other items are also to be similar. Hence in applying this concept in PHR it simply means, if a doctor has consulted two users with similar disease, then their requirements towards medication, consultation, diet and check up routines will also be similar.

Hence these two users can be collaborated under one heading. But the challenge is the above recommendation methods cannot be applied directly in the PHR system since the method implemented for finding the similar users in PHR system and that of the traditional recommender environment is different. In our
proposed method we analyse the similarity between two users using doctor's historical queries which is similar to the similarity calculation method in recommender system.

Background

The PHR system in cloud computing is simply having the information stored in the web-based tool. It allows the user to store, capture, update and edit the individuals health records in a secured approach at a central place [1], [2], [3]. The PHR is owned by the individual who has all the rights to edit, delete and share the information. The PHR system is a convenient tool to store all the information in one central storage which helps the user to efficiently manage the records and use them as and when required. Based on the severity of the disease and the need for cautious health checkups the PHR system allows the user to set reminders for taking checkups, medications, schedule appointments and so on. It also helps the user to take notes on the daily diet, symptoms, track pain, side effects of medication. It also helps to track your previous diseases, allergies, procedures, test results, immunizations and medications [5], [6].

PHR in cloud helps to maintain the record with current information and handy to carry the records everywhere. The users can also provide access to health care providers to add their inputs in the medical history. This approach helps in speeding the healing process and optimises the diagnosis process. It also helps other healthcare providers to have a detailed study about the patient and avoid the reputation of some medical tests and medications too [7], [8]. There are several ways through which the patients and the health care providers can benefit out of PHR through Cloud access [13], [14], [15], [16], [11], [7]. Some of the advantages are:

- Scalability.
- Interoperability.
- Reduced cost.
- Ease of use.
- Improved continuity of care.

3. Algorithms for Cloud Security

Advanced Encryption Standard (AES)

The earlier method of 3DES had drawbacks in implementing when it comes to the bit size and the security attributes attached to it. The 3DES which was recommended by the Federal Information Processing Standard FIPS PUB 46-3 introduced a new standard with 168-bit key in 1999. But due to the 64-bit block size and the inefficiency of the system it led to the larger block size usage.

In 1997, the NIST- National Institute of Standards and Technology proposed the Advanced Encryption Standard (AES). There were 15 algorithms proposed for the first round of evaluation. The criteria for the new method is that it should be efficient, secured with 128 bit size. In the second round of evaluation there were
5 algorithms and finally NIST published the AES algorithm proposed by Rijndael in November 2001. Dr. Joan Daemen and Dr. Vincent Rijmen were the researchers of AES from Belgium.

AES Evaluation

Security – 128 minimum bit size which provides better security system than 3DES.

Cost- It has high computational efficiency.

Final NIST Evaluation of Rijndael (October 2, 2000)

<table>
<thead>
<tr>
<th>General Security</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rijndael has no known security attacks. Rijndael uses S boxes as nonlinear components. Rijndael appears to have an adequate security margin, but has received some criticism suggesting that its mathematical structure may lead to attacks. On the other hand, the simple structure may have facilitated its security analysis during the timeframe of the AES development process.</td>
<td></td>
</tr>
</tbody>
</table>

Software Implementations

- Rijndael performs encryption and decryption very well across a variety of platforms, including 8-bit and 64-bit platforms, and DSPs. However, there is a decrease in performance with the higher key sizes because of the increased number of rounds that are performed.
- Rijndael’s high inherent parallelism facilitates the efficient use of processor resources, resulting in very good software performance even when implemented in a mode not capable of interleaving. Rijndael’s key setup time is fast.

Restricted-Space Environments

- In general, Rijndael is very well suited for restricted-space environments where either encryption or decryption is implemented (but not both). It has very low RAM and ROM requirements. A drawback is that ROM requirements will increase if both encryption and decryption are implemented simultaneously, although it appears to remain suitable for these environments. The key schedule for decryption is separate from encryption.

Hardware Implementations

- Rijndael has the highest throughput of any of the finalists for feedback modes and second highest for non-feedback modes. For the 192 and 256 bit key sizes, throughput falls in standard and unrolled implementations because of the additional number of rounds. For fully pipelined implementations, the area requirement increases, but the throughput is unaffected.

There are a number of parameters considered depending on the key length (Table 5.3).

In our proposed method we consider the key length to be 128 bits.
A fixed single 128 bit block is taken as input to both encryption and decryption algorithm in FIPS PUB 197. This 128 block is illustrated as a square matrix of bytes. This block is represented as State array and is modified in every step of encryption and decryption. The 128 bit is converted into an array of key where each word is 4 bytes and the total key schedule is 44 words for the 128-bit key. The bytes are arranged in column in the matrix.

![Figure 3: Overall Structure of AES](image)

Figure 3: Overall Structure of AES

This prototype does not follow the Feistel structure. The input key is converted into an array of 44 words (32-bits each) and 4 distinct words (128 bits) will be...
the round key in each round. The first step is permutation and the third is the substitution. Now every byte present in the S-box will be mapped to its multiplicative inverse in GF ($2^8$), and the value {00} will be mapped to itself. Let's consider that each byte in the S box has 8 labelled bits namely (b7,b6,b5,b4,b3,b2,b1,b0). Then the transformation is applied to all the byte in S.

$$b'_i = b_i \oplus b_{(i+4) \mod 8} \oplus b_{(i+5) \mod 8} \oplus b_{(i+6) \mod 8} \oplus b_{(i+7) \mod 8} \oplus c_i$$

where $c_i$ is the $i$-th bit of byte $c$ with the value {63}, that is, $(c7c6c5c4c3c2c1c0)=(01100011)$. The prime (') means that the value has to be replaced with the value on the right.

**Elliptic Curve Cryptography**

Let's discuss on how Elliptic Curve is used in the context of cryptography and for which we need to understand the concept of Elliptic curves. Elliptic Curve is made up of a set of points that lies on the curve. $y^2 = x^3 + ax + b$.

Where $a$ and $b$ are real numbers.

**Elliptic Curve Groups:** The points which satisfy the elliptic curve equation along with the point $\infty$ form the additive group. The summing up of two points on the elliptic curve can be defined as below.

The complexity of an Elliptic Curve Encryption Algorithm depends on the
complexity of calculating $kP$. $k$ is a product of two large primes and $P$ is an element on the ECG. To add $P$ to itself, first draw a tangent line from that point and the line will intersect the curve only at one point, now the addition $2P$ is said to be the negative of the point of intersection as shown above.

But Elliptic curve groups are not applicable with real numbers when it comes to cryptography since it has tedious calculations involved and round-off error is possible. Hence we consider Elliptic Curve over Finite Fields. An elliptic curve is drawn in a finite field $F_p$ with variable $a$ and $b$ in the field. So let's describe the elliptic curve with points $(x, y)$ as $y^2 = x^3 + ax + b$ modulo $p$. $x, y \in F_p$; together with a special point $\infty$.

If $x^3 + ax + b$ has no repeated factors, or if $4a^3 + 27b^2 \equiv 0 \pmod{p}$, then these points form a group. It is a known factor that EGC (the Elliptic Curve Group) is an additive abelian group with $\infty$ serving as its identity element.

![Elliptic curve equation: $y^2 = x^3 + x$ over $F_{23}$](image)

Example: In the ECG of $y^2 = x^3 + x^2$ over the field $F_{23}$ the point $(9,5)$ satisfies the equation $y^2 \equiv x^3 + x^2 \pmod{23}$ as. The elements of this ECG are given in the pictured below.

$$f(x) = a_0 + a_1 x + \ldots + a_{t-1} x^{t-1}$$

It is obvious from the above picture that we cannot derive a curve to define our addition geometrically. So in order to translate the addition geometrically we define $F_p$ as

$$y_3 = \lambda(x_1 - x_2) - y_2$$

Now consider $P(x_1, y_1)$ and $Q(x_2, y_2)$ be elements of the ECG. So $P + Q = (x_3, y_3)$, and
Though the equations look tedious they are easy to be computed with computers. Some complex computational problems like to find \( k \) in a field \( \mathbb{F}_p^* \), such that \( r = qk \pmod{p} \). In a similar way the Elliptic Curve Discrete Logarithm Problem is to find \( k \) in an ECG with points \( P \) and \( Q \) where \( Pk = Q \). Where \( k \) is called the discrete log of \( Q \) to the base \( P \).

**Hybrid Crypto Secure Secret Sharing Algorithm**

We have proposed Hybrid Crypto Secure Secret sharing algorithm for cloud computing. This proposed scheme is based on polynomial interpolation. Consider \( D \) who can broadcast the secret value \( s \) to \( n \) number of players, considering that \( t < n \) players is required to reconstruct the secret code. The protocol used here is Information Theoretically Secure which means without all the \( t \) players it is not possible to construct the secret code.

**The Sharing Protocol**

**Goal:** For \( D \) to share the secret \( s \) to \( P_1, P_2, \ldots, P_n \), where \( t \) players are needed to reconstruct the secret. \( D \) will create a random polynomial \( f(x) \) with \( t-1 \) as the degree of the equation and with a constant term \( s \). The polynomial function is constructed over a finite field where coefficient \( a_0 \) will be the secret \( s \) and the other coefficients are random elements in the field. Now \( D \) will choose \( n \) as a random distinct evaluation point: \( X_j \neq 0 \), and it gets secretly distributed to all the players \( P_j \),

\[
f(1), \ldots, f(j), \ldots, f(n) \quad \text{share}_j(s) = (X_j, f(X_j)), j = 1, \ldots, n.
\]

(Remark: The randomly selected evaluation point \( X_j \) is value known to all and for ease of calculation we represent this as, \( X_{j\cdot} \))

**The Reconstruction Protocol**

**Goal:** Now to reconstruct the secret from \( t \) shares out of \( n \) shares. We consider the subset \( f(1), \ldots, f(t) \) and using Lagrange interpolation we find the unique polynomial \( f(x) \) where \( \deg f(X) < t \), \( f(j) = \text{share}_j(s) \) for \( j = 1, 2, \ldots, t \). Reconstruct the secret to be \( f(0) \).

**Interpolation Property**

With \( t \) pairs of \( (i, f(i)) \), where all the \( i \)'s are unique, there is a polynomial \( f(X) \) of degree \( t-1 \) which passes through all these points. This function can be computed using \( (i, f(i)) \).
Lagrange interpolation:

\[ f(x) = \sum_{i=1}^{t} f(i) \cdot L_i(X) \quad \text{where } L_i(X) \text{ is the Lagrange polynomial:} \]

\[ L_i(X) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)} \]

which has value 1 at \( x_i \), and 0 at every other \( x_j \).

Figure 5: Graphic-Representation of a Degree-2 Polynomial and Its Share

**Properties of Hybrid Crypto Secure Secret Sharing**

**Perfect Security:** The Secure secret sharing system paves a secured way in sharing the information rightly. With the given \( t \) shares it is possible to compute the secret \( a_0 \) as the polynomial is uniquely determined. But if given \( t-1 \) shares or any lesser number of shares than \( t \) then the computation becomes difficult. The secret element can be placed anywhere in the finite field.

**Ideal:** All the shares are evenly sized as the secret.

**Extendable:** The method is scalable and additional shares can be added as required by just computing the polynomial for that additional point.

**Flexible:** It can assign different weights to different shares or authorities.

**Homomorphic Property**

Homomorphic property in Hybrid Crypto Secure secret sharing scheme means considering two secrets being hidden. For example consider two secrets \( S \) and \( T \) and they are hidden using Hybrid Crypto Secure secret sharing scheme in their corresponding functions namely. \( f(0), \ldots, f(n)) \) for the polynomial \( f(X) \), and \( g(1), \ldots, g(n)) \) for the polynomial \( g(X) \).

Now consider the \( i \)th shareholder of the sum of the two functions.
\[ h(i) = f(i) + g(i) \quad (i \in [1..n]) \]

Hence the above function can be simply said as the sum of the two secret functions is in turn another polynomial by itself. Hence it can be written as \( h(X) = f(X) + g(X) \). And therefore \( h(0) = S + T \)

**Efficient Distributed Mechanism For Arithmetic Calculations** – for example multiplication by a constant: every function will be multiplied by a constant term.

4. **Comparative Study**

**AES Scheme**

The previously proposed methods with cloud computing use RSA encryption technique and the output achieved through RSA is poor in accuracy. The encryption result is different each time and it also says no record available even when the data is present in the database. Hence we have proposed cloud computing with AES which is efficient in searching and fetching the right result. It is quick and fast in retrieving data.

![Graph showing Key Generation Time](image)

**Key Generation Time**

Key generation is an important process when it comes to encryption and decryption. The efficiency and security of an algorithm completely depends on the key generation factor. There are a number of attributes to decide the key generation efficiency.

There are various key generation methods like the KP-EEC (Key Policy- Elliptic Curve Cryptography based encryption), in this prototype a key is generated during registration of each authority and its too long and take time to generate the key. In case of MA- ABE (Multi authority attribute based encryption) each authority has a different key. In KP-ABE only a single key is generated for all authorities and in MA- ABE the key is generated only for the registering authority and not for all authorities.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Key length</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA (Rivest Shamir Aldeman)</td>
<td>&gt;1024 bits</td>
</tr>
<tr>
<td>AES (Advanced Encryption Standard)</td>
<td>56 bit</td>
</tr>
<tr>
<td>DES (Data Encryption Standard)</td>
<td>138, 192, 256 bits</td>
</tr>
</tbody>
</table>

### Encryption Time

In general, encryption time is long when it comes to RSA. In RSA there is a need for encryption of two keys namely public and private key. Since RSA uses the concept of asymmetric key the computation time to encrypt two key is too long to generate the key. Whereas in our approached method we use RSA+AES+DES. As stated earlier AES and DES uses symmetric key which means a single key for both encryption and decryption process. Hence it takes less time when compared with the simple RSA process.

### Decryption Time

When it comes to RSA the decryption process is highly time consuming as it needs both public and private key. Hence it consumes a lot of time to generate both the keys. Whereas in our method RSA+AES+DES algorithm, the concept of symmetric key reduces this effort as it needs to generate only a single key at any point of process execution.
The above table talks about the comparison between AES, DES and 3DES dividing them into nine factors of analysis. This includes cipher type, key length, possibility key, resistance, block size, possible ASCII printable character keys, cryptanalysis resistance, time required for the checking process. It is obvious from the above table that AES is far better than the other schemes in all the nine factors.

**Hybrid Crypto Secure Secret Sharing Shame:** The below table distinguishes the characteristics of various schemes used for secret sharing process. It also shows the methods which support multiple secret sharing schemes. It is visibly noted that Hybrid Crypto Secure Secret Sharing scheme is highly secured and consumes less execution time when compared to other schemes such as Li Bai’s Secret Sharing Schemes, IDA and Shamir in cloud computing.
5. Conclusion

In this research paper we have implemented PHR through cloud computing. We have implemented encryption of data before the information is shared or stored through cloud computing. The attribute based encryption technique has improved and strengthened the security of information to its highest level secrecy. The simulations also prove that the proposed method adheres to the break glass procedure as well. Hence its rightly proved that a simple user friendly interface can still be efficient in delivering timely, securely and dynamic data to the users across the internet through cloud computing.

References


