International Journal of Engineering Research and General Science Volume 3, Issue 3, Part-2 , May-June, 2015
ISSN 2091-2730

# Patient-Centric Secure Sharing of Personal Health Records in Cloud Storage

Prajakta Solapurkar,Girish Potdar

Department of Computer Engineering, Pune Institute of Computer Technology, Pune, Maharashtra, India

Email-praj.solapurkar112@gmail.com

**Abstract**— In a modern healthcare environment, personal health record(PHR) owners are willing to store and share electronic medical records via the cloud because of its ubiquity and on-demand self service. Secure and efficient data sharing scheme enable patients to have full control over their PHRs and at the same time provide confidentiality and authenticity of personal health data. Selective data sharing requires different documents to be encrypted with different keys, which implies, patients to distribute to users a large number of keys and the authorized users have to securely store the received keys. The need for secure storage, communication and efficient key management renders the approach impractical. The current work focuses on reducing key management overhead by generating a single aggregate key, but does not provide, how it can satisfy the principles of efficient data sharing. So, we propose a scheme to achieve: confidentiality of personal health data, authenticity of personal health data, patient-centric fine-grained access control and revocation of access control using key-aggregate cryptosystem. Experimental results show that the proposed scheme reduces key-size and key management overhead.

**Keywords**— Key-aggregate Cryptosystem, Personal Health Records, Secure Sharing, Cloud Storage, Data sharing, Access Control, Security.

## INTRODUCTION

With the increasing popularity of electronic health records, personal health records (PHR) that include personal and medical information, insurance details, etc. have become increasingly important. Cloud computing has emerged as a promising solution for providing ubiquitous, convenient, and on-demand accesses to large amounts of data. Hence it is considered as an attractive option by many patients to store and share their health records and hence remove the geographical dependence between health care provider and patient. However, PHRs may contain sensitive data like patients' personal contact information, laboratory test reports, X-rays and so on. Due to data outsourcing, patients do not have full control privileges over the data which increases security and privacy risks.

To ensure data privacy and access control, PHR owners choose to encrypt their data before uploading it on the cloud and hence the data remains secure against the cloud providers and other malicious users. Suppose that patients want to share their PHRs with different data users, there are two ways to achieve this under traditional encryption scheme:

- Data owner (patient) encrypts all categories of personal health records with a single encryption key (symmetric key cryptosystem) and gives the data users like doctors, nurses the corresponding secret key directly.
- Data owner (patient) encrypts different categories of personal health records with different keys and sends the corresponding secret keys to data users like doctors, nurses, relatives etc.

Here, the first method is inappropriate since it also provides access to unauthorized categories of PHR. The second method is inefficient as the number of keys will be equal to the number of file categories. In practice, transferring these secret keys requires a secure communication channel, and storing of these keys requires an expensive secure storage. Therefore, the best solution for the above problem is that the data owner encrypts different categories of PHR with different public-keys, but is sent to data users like doctors, nurses etc. a single key for decryption. Small key-size is desired since the secure communication and storage is required. So to reduce key management overhead (key-size, key transfer, key storage) of data owner a concept of a key aggregate cryptosystem is used [1].

But secure and efficient data sharing scheme also enables a data owner to have full control over their PHRs.Many recent studies [3-5] discuss on how patients apply the encrypted-once and decrypted-many-times encryption technique to their PHR data using Attribute-Based Encryption (ABE) [8-10] or proxy re-encryption schemes [11-12]. But, if the access structure is too complicated, then ABE is insufficient for providing patients with full control over their data [2]. It is therefore necessary to explore how patients set up

access privileges for fine-grained access control of their PHR data, i.e., PHR data categories should be accessible only to those users who possess the corresponding decryption keys and the set of decryption keys should be kept confidential from others with the minimum key management overhead. Our proposed solution employs the concept of a key-aggregate cryptosystem and successfully resolves the problem of data access control in a health care setting.

**LITERATURE SURVEY**

Several recent studies have focused on the issue of secure sharing of electronic health records in the cloud.

Chen *et al.* [4] proposed an EHR solution, relying mainly on smart cards and RSA that enables patients to store their medical records on hybrid clouds. In this approach, patients' medical records are stored in two types of cloud: the hospital's private cloud and the public cloud. The authors discussed two usage cases. The first is that of the medical records being accessed by the owner of the data, *i.e.*, the doctor who created the records. They can directly access the records from their private cloud or from the public cloud. The second case is that of the medical records being accessed by other hospitals, who must seek permission from the data owner before they can access the records. The authors also provide a solution for emergency situations. However, the shortcoming of this approach is that data owners, *i.e.*, doctors have access control for the medical records and their computing load is heavy.

Leng *et al.* [5] proposed a solution that allows patients to specify a policy to support fine-grained access control. They primarily utilized Conditional Proxy Re-Encryption to enforce sticky policies and provided users with write privileges for PHRs. When users finish writing data to their PHRs, they sign the modified PHRs. However, users sign the PHRs using the signature key of the PHR owner and it is therefore difficult to correctly verify who signed the PHRs.

Kuo et al. [2] proposed a scheme for patient-centric access control over PHR data. The proposed scheme ensures the following security properties: (1) confidentiality of health data, (2) integrity of health data, (3) authenticity of health data, (4) patient-centric fine-grained access control, and revocation of access control using symmetric key cryptosystem and proxy re-encryption (PRE) scheme. But the main drawback of this scheme is, each file category is encrypted with distinct secret key so whenever a data user (e.g. Doctor or nurse) wants to update PHR categories, patient have to provide the corresponding secret keys. Besides this, the scheme is based on proxy re-encryption scheme which requires data owners to have too much trust on the proxy that it only converts cipher texts according to his instruction. A PRE scheme allows data owners to delegate to the proxy the ability to convert the cipher texts encrypted under his public key into ones for data users. Hence it is desired that proxy doesn't reside in the storage server. This increases communication overhead since every decryption requires separate interaction with the proxy.

Chu et al.[1] proposed a new public key cryptosystem which can aggregate any set of secret keys to generate a single compact aggregate key encompassing the power of all the keys being aggregated. But the work did not focus on how it can help patients to have fine grained access control and revocation of access control and at the same time ensuring confidentiality, authentication and integrity of their PHRs.

So in this paper, we redesign the scheme in [2] for patient-centric access control over PHR data belonging to the patient using the concept of a key-aggregate cryptosystem. Our solution ensures the following security properties: (1) confidentiality of personal health data, (2) integrity of personal health data, (3) authenticity of personal health data, (4) patient-centric fine-grained access control, and (5) revocation of access control.

**PROPOSED SCHEME**

A. Access control policy for PHR data

In the architecture of the proposed scheme, PHR data are divided into different categories and arranged in hierarchy as shown in Fig 1. PHR data may include several medical records like dental records, medical records and other categories like personal information, insurance policy information etc.

PHR owners specify policies for their PHR data to grant access privileges to each user. A policy may contain the following details:

(1) Role: users who are permitted to access the data, for example, the doctor, nurse, or insurance broker.
(2) Category of PHR data: Personal Information, Laboratory Test Reports, Medical History, etc.
(3) Permission: includes read, write, and even print.

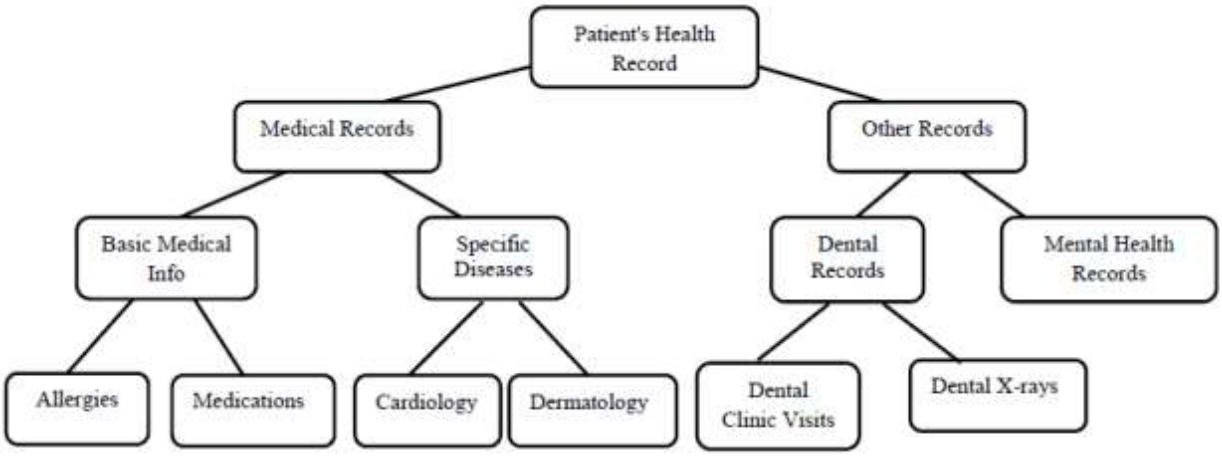


Figure 1. A hierarchical personal health record

B. Construction of Proposed Scheme

The architecture of proposed scheme consists of three roles PHR owners (patients), PHR users (doctors, nurses, insurance policy brokers etc.) and cloud server. The framework of the proposed scheme is shown in the Fig.2.It is as follows:

a) Setup ($1^k$, n):
This algorithm is executed by the patient to set up an account on an untrusted server. On input of security level parameter (k) which can be high, medium or low and the number of cipher text classes n (i.e., class index should be an integer between 1 and n), it outputs the public system parameter param which is set of public keys $p_1$, $p_2$...$p_n$.

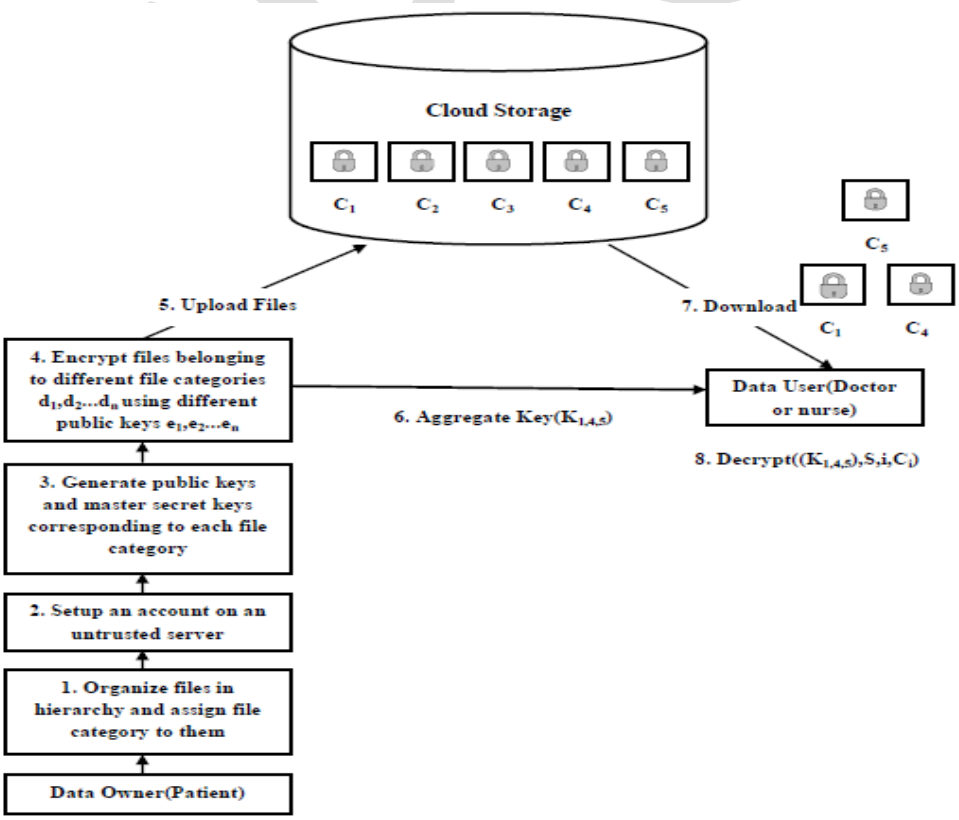


Figure 2. Framework of Proposed System

b) Key Generation (pk, msk):
This algorithm is executed by the patient to randomly generate a public/master-secret key pair (pk, msk).

c) Encrypt (pk, i, d):
This algorithm is executed by anyone who wants to encrypt data. On input a public-key pk, an index i denoting the PHR category, and a document d, it outputs a ciphertext C.

d) Extract (msk, r, S):
This algorithm is executed by the patient for delegating the decrypting power for a certain set of ciphertext classes to a delegatee. On providing input of the master secret key msk, an access right r and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by KS.

e) Decrypt (KS, S, i, C):
This algorithm is executed by a data user who received an aggregate key KS generated by Extract. On input KS, the set S, an index i denoting the file category the ciphertext C belongs to, and C, it outputs the decrypted result, if $C \in S$.

**SOLUTION ANALYSIS**

A. Confidentiality of Personal Health Data:

Before uploading a health record i, it is encrypted using the product of $(pk)_i$ and (msk)i. The master secret key is kept secret. When patient generates an aggregate key which is the product of master secret keys, data user or an interceptor cannot obtain each multiplier from the product. Hence, even $pk_i$ component of the encryption key is publicly available, other component $msk_i$ is hidden hence confidentiality of personal health data is ensured.

B. Authenticity of Personal Health Data:

In our proposed scheme owner of the PHR generates an aggregate key. At the time of decryption,an aggregate key successfully decrypts the authorized set of cipher text. This verifies the authenticity of personal health data.

C. Patient-centric fine-grained access control:

In our scheme, the PHR owner generates a value representing a particular access right when generating the aggregate key. The PHR owner can therefore control access privileges for every user.

D. Revocation of access control:

If the PHR owner wishes to revoke some users in a certain category, then they need only to replace the aggregate key $K_s$ with $K_{s1}$. The small aggregate key size minimizes the communication overhead for transferring the new key. Revocation therefore, is easily achieved.

**EXPERIMENTAL WORK AND RESULTS**

A. Platform and Technology

The experimental setup in our proposed system will be configuration of OpenStack, an open source cloud platform using DevStack module.

Platform and technology used are:

a) O.S: Ubuntu 12.04
b) Database: OpenStack MySQL
c) Web Server: Apache
d) Network: OpenStack Nova
e) Hypervisor: KVM
f) Language: Java
g) Browser: Mozilla Firefox, Google Chrome etc.

B. Datasets

We conducted experimental evaluation of the proposed system on the basis of performance parameters mentioned in subsection C on a real world dataset [14].We selected the records and arranged them into a hierarchical tree structure with different heights as per the requirements of different patients.

C. Performance Parameters

Performance is the accomplishment of a given task measured against preset known standards of accuracy, completeness, cost, and speed [13]. Following is the list of parameters that we are going to evaluate:

a) Key-assignment Ratio (Ratio of number of keys granted to total number of keys granted in traditional (one-to-one) approach and proposed approach.
b) Amount of compression with respect to key size.

D. Results

Section (C) identifies performance parameters of our proposed system. We have implemented the parameters and their results are as follows: Table I. Shows compression with respect to the number of keys granted in our proposed system to the total number of keys generated as per cipher text classes (Key Assignment Ratio) and compression with respect to the key size.

| Total Number of Cipher Text Classes | 511 records | | | |
|---|---|---|---|---|
| Delegation Ratio | Compression ratio (no. of keys granted) | Existing Scheme [2] Key-Size in bytes | Proposed Scheme Key-Size in bytes | Compression achieved in bytes |
| 0.1 | 0.66 | 3366 | 969 | 2397 |
| 0.2 | 0.71 | 6732 | 1938 | 4794 |
| 0.3 | 0.75 | 10098 | 2907 | 7191 |
| 0.4 | 0.80 | 13464 | 3876 | 9588 |
| 0.5 | 0.88 | 16896 | 4864 | 12032 |
| 0.6 | 0.90 | 20262 | 5833 | 14429 |
| 0.7 | 0.91 | 23628 | 6802 | 16826 |
| 0.8 | 0.92 | 26994 | 7771 | 19223 |
| 0.9 | 0.93 | 30360 | 8740 | 21620 |
| 1 | 0.97 | 33726 | 9109 | 24617 |

TABLE I. Compression for Different Delegation Ratios and Cipher Text Classes

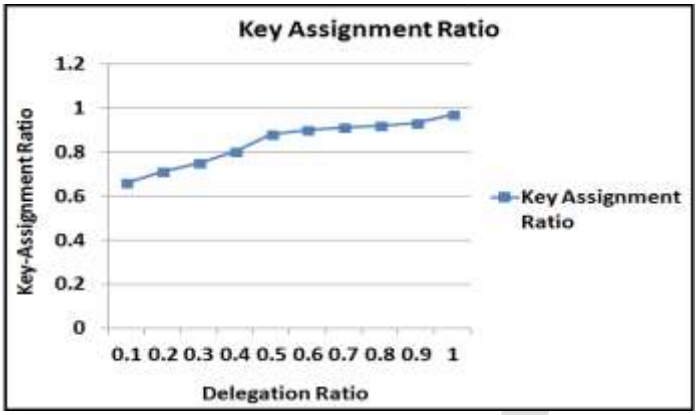


Figure 3. Key Assignment Ratio

Fig. 3 illustrates the relationship between Key Assignment Ratio and Delegation Ratio. The X-axis represents delegation ratio and Y-axis represents Key assignment Ratio (Ratio of number of keys granted to total number of keys granted in traditional(one-to-one approach). We observe that the high compression ratio can be achieved when the delegation ratio is 1.

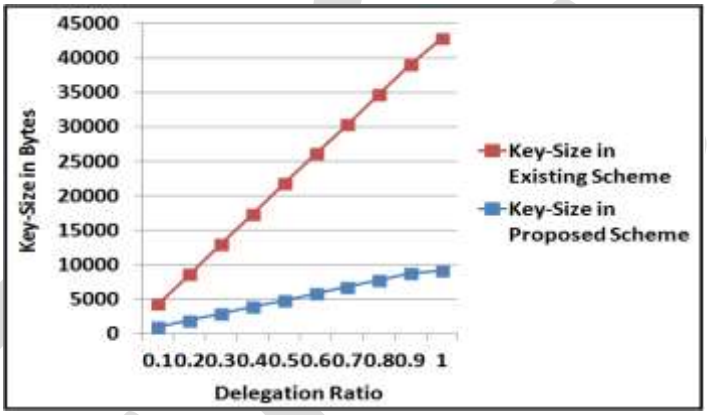


Figure 4. Key Size in Different Approaches

Fig. 4 illustrates the key-size for different approaches in the case of 511 records. The X-axis represents delegation ratio and the Y-axis represents key-size in bytes for different approaches.

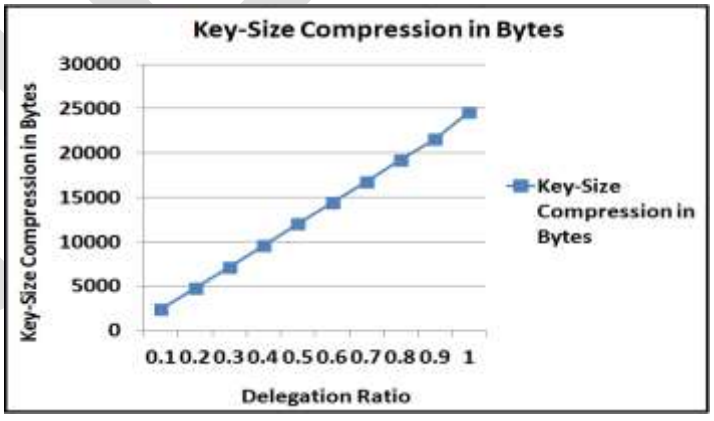


Figure 5. Compression in Key Size in Bytes

Fig. 5 illustrates the relationship between Compression in Key Size and Delegation Ratio. We observe that as delegation ratio increases and approaches to 1 (i.e. Entire PHR can be accessed by delegating single root key) the amount of key-size compression increases.

**CONCLUSION**

With the increasing popularity of modern healthcare systems based on cloud storage, how to protect PHRs stored in the cloud is a central question. Cryptographic techniques are getting more versatile and often involve multiple keys for a single application which increases the key management overhead. In this article, we consider how to generate a single compact aggregate key, but encompassing the power of all keys being aggregated and generate compressed secret keys in public-key cryptosystem. We also discuss how the confidentiality, and authentication of PHRs can be achieved using a key aggregate cryptosystem. This system also enables a patient to exercise complete control over their PHRs and perform revocation of access rights.

## REFERENCES:

[1] C. Chu, S. Chow, and W. Tzeng, "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, 2014, 25 (2): 468- 477.

[2] Kuo-Hsuan Huang, En-Chi Chang, and Shao-Jui Wang, "A Patient-Centric Access Control Scheme for Personal Health Records in the Cloud", Fourth International Conference on Networking and Distributed Computing, 2014.

[3] Dixit, G. N. "Patient Centric Frame Work For Data Access Control Using Key Management In Cloud Server", International Journal of Engineering, *2* (4), 2013.

[4] Chen, Y. Y., Lu, J. C., & Jan, J. K. "A secure EHR system based on hybrid clouds," Journal of medical systems, 36 (5), 3375-3384, 2012.

[5] Leng, C., Yu, H., Wang, J., & Huang, J. "Securing Personal Health Records in the Cloud by Enforcing Sticky Policies," TELKOMNIKA Indonesian Journal of Electrical Engineering, 11 (4), 2200-2208, 2013.

[6] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records", Proc. ACM Workshop Cloud Computing Security (CCSW 09), pp. 103-114, 2009.

[7] Chen Danwei, Chen Linling, Fan Xiaowei, He Liwen, Pan Su, and Hu Ruoxiang "Securing Patient-Centric Personal Health Records Sharing System in Cloud Computing", China Communications, Supplement No.1, 2014.

[8] Ming Li, Shucheng Yu, and Yao Zheng, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption", IEEE Transactions on Parallel and Distributed Systems, 24(1), pp. 131-143, 2013.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data", Proc. 13th ACM Conf. Computer and Comm. Security (CCS 06), pp. 89-98, 2006.

[10] M. Chase, and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption", Proc. ACM Conf. Computer and Comm. Security, pp. 121-130. 2009.

[11] R. Canetti, and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re- Encryption", Proc. 14th ACM Conf. Computer and Comm.Security (CCS 07), pp. 185-194, 2007.

[12] S.S.M. Chow, J. Weng, Y. Yang, and R.H. Deng, "Efficient Unidirectional Proxy Re-encryption", Proc. Progress in Cryptology AFRICACRYPT, vol. 6055, pp. 316-332, 2010.

[13] Mell, Timothy Grance, "The NIST Definition of Cloud Computing", NIST Special Publication 800-1145, Sept. 2011.

[14] https://catalog.data.gov/dataset/va-personal-health-record-non identifiable-data