ISSN No. 0976-5697

Volume 5, No. 5, May-June 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Privacy Preserving PHR System in Cloud using Attribute-Based Encryption

K.Sudha
M.Tech Scholar, Prist University, Kumbakonam,
Tamil Nadu, India.

K.Saminathan
Assistant Professor,
Prist University, Kumbakonam,
Tamil Nadu, India.

Abstract: An important application of data sharing in cloud environment is the storage and retrieval of Patient Health Records (PHR) that maintain the patient's personal and diagnosis information. These records should be maintained with privacy and security for safe retrieval.The data are allowed to be accessed only by authorized persons. Each party is assigned with access permission for a set of attributes. Data owners update the patient data into cloud servers.To ensure the patient's control over access to their own PHRs, it is the best method to encrypt the PHRs before outsourcing. The attribute based encryption (ABE) scheme is used to secure these patient records. Multiple owners are allowed to access the PHRs.we propose patient- centric framework for secure sharing of PHRs under the multi owner settings by using multi-authority attribute-based encryption (MA-ABE)which guarantees the high degree of patient's privacy. To reduce the key distribution complexity, we divided the users in the PHRs system into multiple security domains. This scheme also enables dynamic modification of access policies (or) file attributes, supports efficient on-demand user/attribute revocation and break-glass access under emergency scenarios.

Keywords: Cloud computing, Personal health records, Attribute-based encryption, Fine-grained access control

I. INTRODUCTION

The demand for outsourcing data storage and management has increased dramatically in the last decade. A PHR service allows a patient to create, manage, and control her personal health data in one place through the web, which has made the storage, retrieval, and sharing of the medical information more efficient[1]. In this model each patient is allowed to control access rights as her medical records and can share her health data with a wide range of users, including healthcare providers, family members or friends.

Instead of building and maintaining specialized data centers which cost more, third party servers can be used. Successful examples are Amazon's EC2and S3 [2], Google App Engine, and Microsoft Azure which provide users with scalable resources in the pay-as-you use fashion at relatively low prices. One of the biggest challenges raised by data outsourcing is confidentiality. Data confidentiality is not only a privacy issue, but also of juristic concerns. In healthcare application scenarios use and disclosure of protected health information (PHI) should meet the requirements of Health Insurance Portability and Accountability Act (HIPAA), and keeping user data confidential against the storage servers is not just an option, but a requirement. Due to the high value of the sensitive personal health information (PHI), the third-party storage servers are often the targets of various malicious behaviours which may lead to exposure of the PHI. To ensure patient-centric privacy control over their own PHRs, it is essential to have fine-grained data access control mechanisms that work with semi-trusted servers.

To deal with the potential risks of privacy exposure, instead of letting the PHR service providers encrypt patients' data, PHR services should give patients (PHR owners) full control over the selective sharing of their own PHR data. To this end, the PHR data should be encrypted in addition to traditional access control mechanisms provided by the server. A PHR file should only be available to the users who are given the corresponding decryption key, while remain confidential to the rest of users. Due to the high cost of building and maintaining specialized data centers, many PHR services are outsourced to or provided by third-party service providers, for example, Microsoft HealthVault1. Recently, architectures of storing PHRs in cloud computing have been proposed. While it is exciting to have convenient PHR services for everyone, there are many security and privacy risks which could impede its wide adoption.

The main concern is about whether the patients could actually control the sharing of their sensitive personal health information (PHI), especially when they are stored on a third-party server which people may not fully trust. On the one hand, although there exist healthcare regulations such as HIPAA which is recently amended to incorporate business associates. Cloud providers are usually not covered entities. On the other hand, due to the high value of the sensitive personal health information (PHI), the third-party storage servers are often the targets of various malicious behaviours which may lead to exposure of the PHI. A feasible and promising approach would be to encrypt the data before outsourcing. Basically, the PHR Owner herself should decide how to encrypt her files and to allow which set of users to obtain access to each file. A PHR file should only be available to the users who are given the corresponding decryption key, while remain confidential to the rest of users. Furthermore, the patient shall always retain the right to not only grant, but also revoke access privileges when they feel it is necessary.

II. RELATED WORK

A. *Single trusted authority:*

For the secured sharing of personal health record, the data is stored in cloud server and the key management is provided by the single trusted authority [3]. It not only leads to load bottleneck, but also creates the key escrow problem.

As it is a single trusted authority there may be user collision due to the confusion in key distribution. It is not secured to delegate the key management for all attributes to the single trusted authority. There is a need to divide the users into public and professional users based on the divide and rule, for the secure sharing of PHR. Both the user and owner can manage the minimal keys under a set of attributes. For the key management under the set of attributes we propose the Multiple Authority-ABE (MA- ABE).

B. *Data access control:*

For outsourcing the data in the cloud server there existing a work to realize data access control for the outsourceddata.They use the cipher text-ABE (CP-ABE)[4] for the direct revocation and the cipher text length increases with the number of unrevoked users. In the existing system for the secure sharing of PHR they apply CP-ABE technique.

But there exists drawbacks such as the use of single trusted authority and lack of on-demand user revocation.

In the existing system they provide the time limit for the decryption keys. So it is difficult to access the data for long time with that key and there is a need for re-encryption by providing dummy attributes additionally[5].

C. *Key-Policy ABE (KP-ABE):*

In these schemes, the secret keys are associated with an access structure, while the ciphertext is labeled with a set of attributes [6].

D. *Ciphertext-Policy ABE (CP-ABE):*

In these schemes, the ciphertext is associated with an access structure, while the secret keys are labeled with a set of attributes.

M¨uller, Katzenbeisser and Eckert proposed a distributed CP-ABE scheme [7], where the pairing operations executed in the decryption stage are constant. This scheme was proven to be secure in the generic group instead of reducing to a complexity assumption. Furthermore, there must be a central authority who generates the global key and issues secret keys to the user.

III. PROPOSED WORK

A. *Problem statement:*

Our main design goal is to help the data owner achieve fine-grained access control on files stored by Cloud Servers. Specifically, we want to enable the data owner to enforce a unique access structure on each user, which precisely designates the set of files that the user is allowed to access. We also want to prevent Cloud Servers from being able to learn both the data file contents and user access privilege information.

B. *Architecture of the Proposed System:*

The proposed framework for patient-centric, secure and scalable PHR sharing on semi-trusted storage under multi-owner settings. Proposed system's working is based on the below architecture.

Here it consists of cloud server for storage, application server where actually the PHR system resides and user access the system through internet. Here data access members may be from public domain or from personal domain, so both domain data access members can access related data from cloud server through internet.

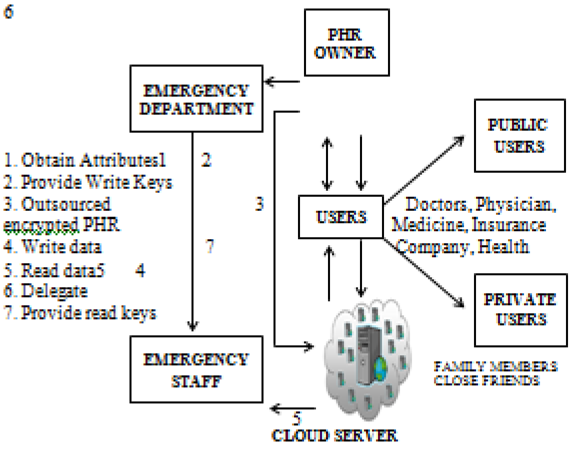

Figure 1 Architecture of patient centric framework

There are multiple sub domains, multiple owners, multiple attribute authority, and multiple users. The main goal of our framework is to provide secure patient-centric PHR access and efficient key management at the same time. The key idea is to divide the system into multiple security domains (namely, public domains (PUDs) and personal domains (PSDs) according to the different users' data access requirements. Users are personally associated with a data owner (such as family members or close friends), and they make accesses to PHRs based on access rights assigned by the owner.

a. ***Multi-Authority ABE:***A Multi-Authority ABE system is comprised of k attribute authorities and one central authority[8].Each attribute authority is also assigned a value, dk. The system uses the following algorithms:

a) ***Set up:*** A random algorithm that is run by the central authority or some other trusted authority. It takes as input the security parameter and outputs a public key, secret key pair for each of the attribute authorities, and also outputs a system public key and master secret key which will be used by the central authority.

b) ***Attribute Key Generation:*** A random algorithm run by an attribute authority. It takes as input the authority's secret key, the authority's value dk, a user's GID, and a set of attributes in the authority's domain and output secret key for the user.

c) ***Central Key Generation:*** A randomized algorithm that is run by the central authority. It takes as input the master secret key and a user's GID and outputs secret key for the user

d) ***Encryption:*** A randomized algorithm runs by a sender. It takes as input a set of attributes for each authority, a message, and the system public key and outputs the cipher text.

e) ***Decryption:*** A deterministic algorithm runs by a user. It takes input a cipher-text, which was encrypted under attribute set and decryption keys for that attribute set. This algorithm outputs a message m.

In cloud Environment PHR Owners need upload data on cloud in such a manner that confidentiality of data and access rights of the data in highest point. Before uploading the file to cloud it should be encrypted and while downloading also it has to decrypted in the application where PHR application is running. This system has three types of users Admin, PHR Owner & Data Access Member. Sample files, attributes types and access policy table of the system are shown below.

b. Sample Files used in this system:

a) Personal File
b) Medical History
c) Current Medical Examination
d) Insurance Details
e) Sensitive Details

c. Attribute Types in this System:

a) Friends
b) Hospitals
c) Insurance
d) Emergency

IV. SYSTEM FLOW DIAGRAM

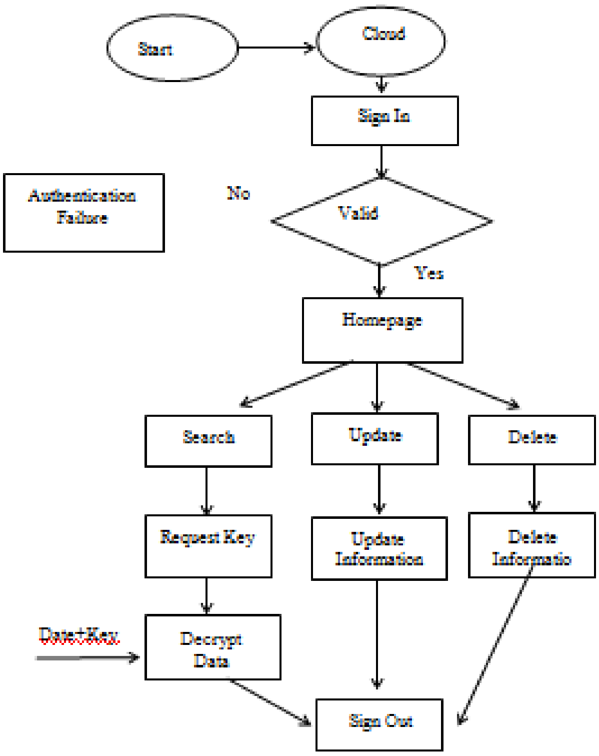

Figure 2 System Flow Diagram

V. SECURITY ISSUES

A. Data confidentiality:

Unauthorized users who do not possess enough attributes satisfying the access policy or do not have proper key access privileges should be prevented from decrypting a PHR document, even under user collusion. Fine-grained access control should be enforced, meaning different users are authorized to read different set of documents.

B. Write access control:

We shall prevent the unauthorized contributors to gain write-access to owners PHRs, while the legitimate contributors should access the server with accountability.

C. Dynamic policy updates:

The data access policies should be flexible, dynamic changes to the predefined policies shall be allowed, especially the PHRs should be accessible under emergency scenario.

D. On-demand revocation:

Whenever a user's attribute is no longer valid, the user should not be able to access future PHR files using that attribute.

This is usually called attribute revocation, and the corresponding security property isforward secrecy. There is also user revocation, where all of a user's access privileges are revoked.

E. Scalability, efficiency, and usability

The PHR system should support users from both the personal domain and public domains. Since the set of users from the public domain may be large in size and unpredictable, the system should be highly scalable, in terms of complexity in key management, communication, computation and storage. Additionally, the owners' efforts in managing users and keys should be minimized to enjoy usability.

VI. ADVANTAGES

A. Security:

Without the user providing secret key no one can access the user's profile. Only the members of the personal and public domain can access the record, even the members cannot get the whole access of writing or reading the record.It is up-to the owner's wish of providing read or write access to the users.The data's are highly secured by using ABE, as the information is encrypted before outsourcing it to others. To decrypt the information we need a secret key.

B. Storage:

The whole information is stored in the server. The requested attributes are encrypted and are then stored in the cloud server. For the purpose of memory allocation the records are divided into attributes which saves memory space. The encrypted data is stored in the cloud server for the purpose of better output.

C. Portability:

The users or the members of the PUD or PSD can access the information from anywhere and anytime as the encrypted data's are stored in the cloud server. It reduces the cost for accessing the information as it can be accessed from anywhere and anytime.

VII. CONCLUSION

In this paper, our main goal is to provide a secure service in the medical field. For this secure service. we proposed the separation of users as personal and public users which makes the key management and security an easier task. We provide this service to a third party server to reduce the management risk of the PHR owner. We use multi authority-ABE (MA-ABE) for providing encryption of each attributes for the purpose of secured data access control. As the users are large in number we use fine grained data access control which means different types of users can access with different types of attributes.

VIII. REFERENCES

[1]. Scalable and secure sharing of Personal Health Records in Cloud Computing using Attributebased encryption- Ming Li Member, IEEE, Shucheng Yu, Member, IEEE, Yao Zheng, Student Member, IEEE, KuiRen, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE

[2]. H. L¨ohr, A.-R.Sadeghi, and M. Winandy, “Securing the e-health cloud,” in Proceedings of the 1st ACM International Health Informatics Symposium, ser. IHI " 10, 2010, pp. 220–229.

[3]. M.Li,S.Yu,K.Ren, and W.Lou, ”Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings,” in SecureComm’10,Sept.2010,pp. 89-106.

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

[5]. L. Ibraimi, M. Asim, and M. Petkovic, “Secure Management of Personal Health Records by Applying Attribute-Based Encryption,” technical report, Univ. of Twente, 2009.

[6]. Yu, C. Wang, K. Ren, and W. Lou, “Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing,” Proc. IEEE INFOCOM ’10, 2010.

[7]. S. M¨ uller, S. Katzenbeisser, and C. Eckert, “Distributed attribute- based encryption,” in Proceedings: Information Security and Cryptology-ICISC’08 (P. J. Lee and J. H. Cheon, eds.), vol. 5461 of Lecture Notes in Computer Science, (Seoul, Korea), pp. 20–36, Springer, December 3-5 2008.

[8]. M. Chase and S. S. Chow, “Improving privacy and security in multi-authority attribute-based encryption,” in CCS ’09, 2009, pp.121–130.