Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

International Conference on Mobility in Computing- ICMiC13, Organized by Mar Baselios College of Engineering and Technology during December 17-18, 2013 at Trivandrum, Kerala, India, pg.44 – 49

SURVEY ARTICLE

Secure Personal Health Records in Clouds: A Hierarchical Attribute Based Solution

Rojitha Abdulla[1], **Anupriya Vysala**[2]

Department of Computer Science & Engineering, Mar Baselios College of Engineering, Trivandrum, India
[1] rojitha1234@gmail.com, [2] vysala.anu@gmail.com

***Abstract*—PHR or Personal Health Record is the health information that is recorded and maintained by the patient, in the clouds, for global information exchange. Since it is stored in clouds, there is a possibility of sensitive patient data being accessed by unauthorized people for illegal gain. Some open challenges in having PHRs in cloud relate to flexibility, scalability, security and privacy. Even though the technique of encrypting PHRs before outsourcing (for instance, using Attribute Based Encryption) is a common one, there are several issues associated with the current techniques used for encrypting PHRs. This paper proposes a method of encrypting PHRs prior to outsourcing by means of Hierarchical Attribute Set-Based Encryption which achieves the above said challenges along with fine-grained access control to the medical data.**

Keywords- Personal Health Record; Clouds; Security; Flexibility; Attribute Set-Based Encryption; Hierarchical access structure

I. Introduction

A Personal Health Record (PHR) is an electronic record of medical information of a person. It is uploaded and maintained by the patient himself and is stored online based on certain national standards. Only the patient and those authorized by him has access to the stored information. The records are stored on third party servers or clouds like the Microsoft HealthVault, for scalability purposes. The term 'Cloud' basically refers to the internet and 'Cloud Computing' is the term used when applications are uploaded on to the clouds to make it available to the public as a service. Anybody who has an internet connection can access these services on their systems. The clouds or the PHR Service Providers provide the PHR application as a service to the general public while collaborating with various hospitals and "Attribute Authorities". Attribute Authorities are responsible for providing the "attributes" or keys required for authenticating the user.

Outsourcing of data on to the clouds gives rise to many security and privacy concerns. Here the data owners and the service providers are not the same. The Service Providers are third party servers who only provide a platform for the application, which are accessed by the data owners, who require their data to be stored onto the clouds, so as to make it available to different sources. Therefore, there is a chance of the data losing its confidentiality and sometimes its integrity since the data may be accessed by unauthorized individuals in the clouds. This leads to the unavoidable fact that the storing of patient's personal health records on to clouds is a relatively insecure method of data storage. The need for secure data storage and access control mechanisms is evident here.

Various mechanisms have been proposed to deal with such problems. One mechanism, that is suggested, is to have the owner provide each individual, who wishes to access his records, with a password. But this is not a feasible mechanism, because it requires the owners to be continuously online, whenever someone needs access to his records. Another suggested mechanism is to make use of a central authority. However, this leads to a single point of failure, in case of a corrupt authority. Another technique, that is more reliable and has been successfully implemented, is the 'Attribute Based Encryption' (ABE) scheme [10]. Various improvements proposed over the years on the basic ABE algorithm have helped deal with the disadvantages of the earlier versions. In [1], the data users, i.e., those individuals who are authorized by the data owners for data access, are primarily divided into two categories belonging to the private domain and public domain. Private domain represents the relatives and friends of the owner and public domain represents the people in the public sector like the doctors, nurses, medical staff, emergency medical personals etc. The scenario is such that data owners specify different rules of access for people belonging to the two domains. The personal domain is shown to be encrypted by KP-ABE (Key-policy Attribute based Encryption) and the public domain by MA-ABE (Multi-Authority Attribute based Encryption) which is an extension of the Ciphertext Policy ABE.

We propose a unique framework based on the Hierarchical Attribute Set-Based Encryption [2] i.e., the extension of the ASBE (Attribute Set-Based Encryption) [6] technique, for more flexibility and scalability along with fine-grained data access control for health information stored in third party servers with multi-owner, multiple attribute authority scenarios where a recursive access structure is possible unlike in the MA-ABE mechanism. The differentiation of user domain into private and public sector is assumed here also.

II. Related Work

Several encryption schemes have been proposed for secure access to the outsourced data. Public Key Encryption schemes were primary mechanisms used for this purpose. But these brought about scalability problems, which in turn paved way for 1-N encryption techniques like the Attribute based encryption (ABE) based schemes. ABE schemes have been proposed in [3], [4], [5] and [9]. Here the common attributes of the various users are used for decrypting the encrypted file. But all of these schemes have a basic disadvantage of using a single trusted central authority. Using a central authority causes bottlenecks and can also lead to the key-escrow problem, in case the central authority is corrupted. Key management and user revocation, while maintaining the scalability of the system, were also considered major problems with these mechanisms. The technique, by which the attribute authorities carry out key updates for those as yet unrevoked users, as in [9] does not lend itself well to scalability.

Yu *et al.* employed KP-ABE (Key-policy ABE) for data encryption. A multi-authority ABE was proposed by Chase and Chow in [7] for a scenario, where multiple attribute authorities were used for encryption and decryption purposes. These two encryption techniques were applied in [1]. Key-policy based encryption was used to encrypt the private sector, whereby the data owners give secret keys to each member of the private domain. Multi-authority based ABE [8] was used for encrypting data for those users in the public sector. There are multiple authorities which provide attributes to the user through different means. The received attributes are used to decrypt the downloaded encrypted file. Here key management overhead is low, user revocation is possible and a patient-centric secure framework is possible. However, multi-authority ABE algorithms are basically ciphertext-policy ABE algorithms; therefore, user attributes are classified as a single group for an authority and only a subset of attributes from this prescribed group can be used to satisfy an access policy. An alternative was suggested by Bobba et al. in [6] where he introduced the cipher text-policy attribute set-based encryption. CP-ASBE or ASBE (Attribute Set-Based Encryption), as it is more commonly called, is a better and extended version of the CP-ABE, whereby user attributes are organized into a recursive set format. The difference between the performance of ABE and ASBE can be illustrated by the following example. A medical officer, who manages two distinct roles in a hospital or any other organization, should not be able to combine the attributes of the two roles together, in order to extract some illegal advantage from one or the other of his roles. This is not implementable by the CP-ABE based schemes, which use techniques similar to Role Based Access Control models (RBAC) [4]. In such an instance, the ASBE provides much better control. It helps in setting dynamic constraints, thus bringing about a halt to the illegal combination of two role attributes, thereby providing much greater access control flexibility. Another alternative is the hierarchical ABE (HABE), which combines the HIBE (Hierarchical Identity Based Encryption) in [9] and the CP-ABE (Ciphertext-Policy Attribute Based Encryption) in [4] and [6]. But this also requires a single authority, which doesn't lend itself well to scalability. It also doesn't support a deeper level of access key structure; as a result, a common attribute in two authorities can lead to difficulties, unlike the case of ASBE, where the problem is solved by assigning different values to similar user attributes that occur in different sets. The ASBE doesn't support the hierarchical structure of multiple levels of authorities. This is where the Hierarchical Attribute Set-Based Encryption scheme (HASBE) [2] lends its advantage, with the inclusion of the delegation algorithm, which helps in key delegation within the respective multiple levels of authorities, starting with a central trusted authority.

III. Proposed System

We propose to utilize the Hierarchical Attribute Set-Based Encryption scheme of [2], which takes into account the hierarchical access policy, that is ever present in hospital settings for the public domain and continue to use the KPABE [11] in the private domain as in [1]. System model consists of numerous owners and users. The whole user domain can be divided into private and public sector as in [1]. The following abbreviations are used in the rest of the paper:

- PK – Public Key
- MK – Master Key
- SK – Secret key
- ID – Identity of user
- A – Access Key Structure
- P – Access Policy
- M – Unencrypted/Decrypted Medical Record
- CT - Encrypted Medical Record
- AA – Attribute Authority

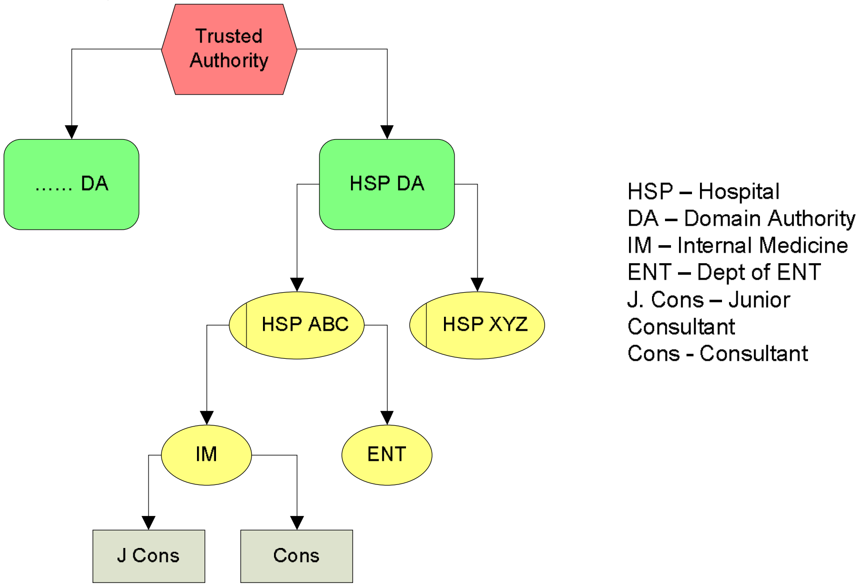

Fig.1. an Example of Hierarchical system of users

A. *Implementation of Private Domain*

In the private domain, KP-ABE algorithm is used. The owner is the trusted authority and provides decryption keys to his friends and relatives. Attributes, based on the inherent properties of a medical record, for example, a medical record category, are utilized by the owner encryption purposes. A user requests the owner for access to the categories of records she needs to access. The owner responds by sending a subset of records applied for. The type and amount of the records allotted by the owner helps in access policy generation, with the help of the KP-ABE algorithm. Here there are no key management problems because the number of users is small and so it is easy for the owner to manage the keys distribution.

B. *Implementation of Public Domain*

1) *Framework of the Public Domain:*

The framework of the proposed public system can be explained as follows. It is assumed that there is a trusted central authority, which controls multiple attribute authorities and provides them with their identity. All authorities may have subordinate authorities or users under their control. Each subordinate authority may have deeper levels of lower level authorities which finally controls the data users. An example framework has been provided in Fig. 1. The research has been carried out under the assumption that central authority can be trusted. Also a user or subordinate level can trust only its higher level authorities. Users are defined on the basis of the roles they play in the community. A user may have more than one role. Attributes are given to the users based on their different roles. Access right to a health record is given to the public users, in accordance with the authorization details based on the requirements of the data owners.

A user in the public domain makes a request for a patient medical record from his organization. The organization takes into account the key structure of the user, compares it with the access policy set by the owner, thereby authenticating him. The different roles of the users are divided into different sets and the attributes of different sets cannot be combined to achieve benefits they must not have access to. An example can be given as follows: Consider a 'Junior Consultant' in the 'Internal Medicine' department of 'Hospital ABC' who also has another role as a temporary 'Consultant' in the 'ENT' department of the same hospital. This can be written as:

{Hospital: ABC,
{Department: Internal Medicine, Role: Junior Consultant}, {Department: ENT, Role: Consultant}}

The attributes of the user, in this case, are:

{Hospital: ABC}
{Departments: Internal Medicine and ENT}
{Roles: Junior Consultant and Consultant}

A data owner may wish his record to be accessed only by a 'Consultant' in 'Internal Medicine' department working in 'Hospital ABC'. So the access policy can be represented as:

{Hospital: ABC, Department: Internal Medicine, Role: Consultant}

If CP-ABE scheme is used, a user could combine all his attributes and group the necessary attributes to arrive at the required access policy, set by the data owner, in this case, {Hospital: ABC, Department: Internal Medicine, Role: Consultant}, which can be used to access patient personal records. This illegal accessing can be prevented using HASBE scheme. In the HASBE technique {Hospital: ABC} will be considered as one set, {Department: Internal Medicine, Role: Junior Consultant} will be taken as another set and {Department: ENT, Role: Consultant} can be taken as a third set. Here {Hospital: ABC} can be taken as a first level set, {Department: Internal Medicine, Role: Junior Consultant} and {Department: ENT, Role: Consultant} can be taken as two sets on the second level. The key structure of the user can therefore be represented as in fig. 2. The key point to be noted is no two sets on the same level can be combined for unauthorized access, thus preserving the privacy and confidentiality of patient records.

Hospital: ABC→ 1st Level
{Department: Internal Medicine, Role: Junior Consultant}, {Department: ENT, Role: Consultant}→2nd Level

Therefore a consultant working on a temporary basis in the required role but in another department of Hospital ABC will not have access to the data owner's record. This idea can be used recursively. It achieves fine-grained data access control by making use of the proposed HASBE technique (described in Sec. C), which is in turn arrived at by extending the ASBE technique (explained in Sec. B).

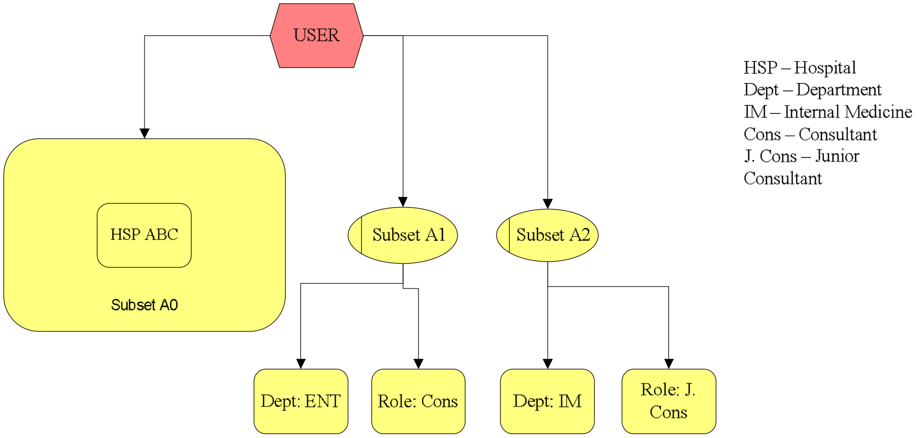

Fig.2. Key Structure of the user in example

2) ASBE scheme:

ASBE can be implemented with the help of the recursive key structure and consists of four algorithms. These can be explained as follows [6]:

- Setup: The depth of the key structure is taken as input and gives as output to the public key PK and the master key MK.
- KeyGen (MK,ID,A): The master key MK, identity of the user ID and access key structure 'A' are taken as input and the secret key 'SK' of the user is derived as output.
- Encrypt (PK, M, and P): Here the public key, the medical record and the access policy are taken as input and cipher text is given as output.
- Decrypt (CT, SK): This algorithm takes as input the encrypted medical record and the secret key and outputs the decrypted medical record.

3) HASBE Scheme:

Since HASBE is an extension of the ASBE, the two schemes are vastly similar except for the delegation algorithm where the trusted authority delegates keys to its subordinates in the hierarchy. The proposed scheme has the following operations [2]:

- Setup (PK, MK): The setup algorithm creates or outputs the public key which is made public and the master key which is kept secret. Here the input is *d* which is considered as the depth of the hierarchy. This function is implemented using bilinear maps.

- Top-Level Attribute Authority Grant: Every attribute authority has an ID and recursive access policy $A = \{ A_0 , A_1 , ….. , A_n \}$. Here $A_i = \{ a_{i,1} , a_{i,2} , ….. , a_{i,m} \}$, $a_{i,t}$ is the t^{th} attribute in A_i and n_i, the number of attributes in A_i. When a new top-level attribute authority wants to enter the system, it is first verified by the central trusted authority. If found valid, it is assigned a master key which is generated by the central authority. Thus the first level of attribute authorities is made valid.

- New Attribute Authority/User Grant: If a new user or subordinate attribute authority (AA_{i+1}) wants to enter a system, it is first verified by the next higher level attribute authority (AA_i) or the top level AA whichever is appropriate. If found valid, it is assigned an ID and a position in the hierarchy according to its role by the AA_i. Thus if the new entry is a user, secret key of the particular user is given as an output along with his access policy. In the case of a new subordinate attribute authority, its master key and the access policy are the outputs. Access policy of a user or AA is always a subset of its higher up AA's.

- New Record Creation: Each record before outsourcing onto the clouds is first assigned an ID and a symmetric encryption key, with the help of which the initial medical record encryption is implemented, after which the access policy for the above said record is defined, and is used for the next level encryption using the Encrypt function, thereby leading to the creation of the encrypted record.

- Encrypt (PK, M, P) – The public key, record and access policy is taken as input and returns the encrypted record.

- User Revocation: Here an expiration time is added to the access policy of each user on entry to the system. The user can only access the particular medical record provided the current time is less than or equal to the expiration time. The expiration time can be extended, if necessary.

- File Access – User downloads the encrypted record from the cloud and decrypts it using the decryption key obtained from the Decrypt function.

- Decrypt (CT, SK) – This function takes as input the encrypted medical record and secret key of user and returns the decryption key.

- File Deletion – Deletion can be done only by the record owner. The record ID and owner signature is send to cloud which deletes the record after verification of the owner sent data.

User revocation, record access and deletion are other operations along with Encrypt and Decrypt functions.

IV. Conclusion

In this paper, we have researched HASBE as a viable option, in case of Public domain, for the encryption of PHR files before outsourcing onto third-party servers. The research has been carried out in order to achieve flexibility along with scalability for the specified application. Here a hierarchical structure of users is considered. Multiple levels of authorities are therefore present and fine-grained access control can be made possible through the key delegation algorithm which allows for keys to be delegated from one level to its subordinate. Efficient user revocation can also be achieved. As proposed in [1], KP-ABE is used for PHR file encryption in the Private domain.

References

[1] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, *"Scalable and secure sharing of Personal Health Records in Cloud Computing using Attribute Based Encryption."* in IEEE transactions on parallel and distributed systems Vol 24, pages 131-143, Jan 2013.

[2] Zhiguo Wan, Jun'e Liu, Robert H. Deng, *"HASBE: A Hierarchical Attribute-Based Solution for flexible and scalable access control in cloud computing"* in IEEE transactions on Information forensics and security, Vol. 7, no. 2, April 2012.

[3] X. Liang, R. Lu, X. Lin and X. S. Shen, *"Patient self-controllable access policy on phi in ehealthcare systems,"* in AHIC 2010,2010.

[4] J. Bethencourt, A. Sahai and B. Waters, *"Ciphertext-policy attribute based encryption,"* in IEEE S&P '07, 2007, pp 321-324.

[5] J. A. Akinyele et al., *"Self-protecting electronic medical records using attribute based encryption,"* Cryptography ePrint Archive, Report 2010/565, 2010, http://eprint.iacr.org/.

[6] R. Bobba, H. Khurana and M. Prabhakaran, *"Attribute-sets: a practically motivated enhancement to attribute-based encryption,"* in Proc. ESORICS, Saint Malo, France, 2009.

[7] M. Chase and S. S. Chow, *"Improving privacy and security in multi-authority attribute-based encryption,"* in CCS '09, 2009, pp.121–130.

[8] Melissa Chase *"Multi-authority Attribute based Encryption,"* Computer Science Department Brown University Providence, RI 02912.

[9] A. Boldyreva, V. Goyal, and V. Kumar, *"Identity-based encryption with efficient revocation,"* in ACM CCS, ser. CCS □08, 2008, pp.417–426.'

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, *"Attribute-based encryption for fine-grained access control of encrypted data,"* in CCS '06, 2006, pp. 89–98.

[11] S. Yu, C. Wang, K. Ren, and W. Lou, *"Achieving secure, scalable, and fine-grained data access control in cloud computing,"* in IEEE INFOCOM'10, 2010.