ISSN:0975-9646

Using Attribute-Based Encryption with Advanced Encryption Standard for Secure and Scalable Sharing of Personal Health Records in Cloud

B. Sri Varsha[1], P.S. Suryateja[2]

[1]M.Tech Student, Department of Computer Science and Engineering, Vishnu Institute of Technology, Bhimavaram, Andhra Pradesh, India

[2]Assistant Professor, Department of Computer Science and Engineering, Vishnu Institute of Technology, Bhimavaram, Andhra Pradesh, India

***Abstract*— Personal Health records have emerged as a patient centric service where the health information of a patient is stored on the cloud. It provides an opportunity for the owner to create, manage and share the data on the cloud. But this entails considering the issues of privacy and security. With technological advancement cloud services are being used extensively and there are multiple owners and users of this service. This makes data management complex. This paper focuses on the issues of data access control and security of the health information. It proposes the use of Attribute based Encryption for effective encryption of data; multi authority based ABE to for providing a high degree of privacy. Division of users into multiple domains reduces the complexity of key management. But data and keys on clouds can be intercepted by semi trusted servers. Therefore for high data security Advanced Encryption Standard is used. In addition it is faster on both hardware and software.**

***Keywords*— Personal Health records, Attribute Based Encryption, MA-ABE, Advanced Encryption Standard.**

I. Introduction

Personal Health Records (PHR) emerged as a patient centric service in cloud computing with third party service providers facilitating the process. The developments are dual pronged. On one hand the development of technology promotes efficient management and sharing of data while on the other the privacy and security of data are at a high stake. As the use of cloud computing grows exponentially, the number of users and owners using the PHR service is also increasing immensely. This makes the management of data complex. This paper proposes to discuss how PHRs can be made scalable and secure.

II. Problem Definition

In a multiple owner and multiple user scenarios the privacy of data and its accessibility are very crucial. The data should be encrypted before being placed on the cloud. The owner has to control data access. This being a complex phenomenon, the data access control should be simplified leading to simple key management. Furthermore, with continuously emerging threats the security of the data has to be significantly enhanced.

III. Related work

For encrypting the data initially Public Key Encryption – based schemes were adopted. But it was one to one encryption and moreover it involved high key management overhead. Rather than this one to one method, one to many encryption methods like ABE were proposed. [1] A method for efficient encryption and key management was proposed. It was suggested that the data be encrypted under a set of attributes which enables multiple users to decrypt using the assigned key. The owner can encrypt the data without even knowing the Access Control List. The unique feature of ABE is that it prevents user collusion.

Several authors proposed different types of Attribute based encryption. Narayan etal proposed CP-ABE [2] while Yu et al. (YWRL) suggested key-policy ABE to secure outsourced data in the cloud [3]. Here a single owner can distribute keys containing attribute based access rights to multiple users. They also suggested ways to revoke data. In addition they proposed the concept of a single trust authority (TA) who can manage the data for the owner. But this leads to centralization of authority at one place which is also problematic.

Addressing this problem Chase and Chow put forth the concept of a multiple-authority ABE (CC MA-ABE) wherein user keys are generated by multiple TAs and one part of each key is obtained from each TA. This effectively stalls collusion. But attribute revocation capability still remains a problem. Overcoming these problems this paper proposes a frame work that would make data access control more efficient aiming at confidentiality, data revocability, dynamic data access policy and security.

IV. Proposed Framework

For efficient data access control ABE is used. For efficient management of data and to prevent centralization of authority MA-ABE is used. Using MA-ABE the data is encrypted in the public domain. In the personal domain the owner directly provides the right for data access to the users using KP-ABE. In addition standard servers are not highly secure. Hence to enhance the security of the data on the servers AES algorithm is used. First data is encrypted using AES algorithm and later it is encrypted using ABE.

It is assumed that the data is stored in semi trusted servers. The data is stored on the cloud and it is also considered that there are multiple owners and multiple users. In such a scenario there are certain requirements to be satisfied:

The data has to be confidential. The owner should reserve the right to control the access of data. The owner should decide who shall read which part of the document. As this is a complex phenomenon, the data has to be scalable and the key management has to be simple. The owner should also have the privilege to revoke the data and in times of emergency it should provide break glass access. Above all it should be highly secure. It is the data access control and security that maintain the privacy of a file on the cloud. In short it should be scalable, efficient and usable to both the owners and the users.

V. Working of Proposed Frame Work

The key idea is to classify the users into distinct domains –the personal domain and the professional domain. In the Public domain (PUD) the users are medical care providers, pharmacists, insurance agents and so on while friends, relatives and family members constitute the personal domain (PSD).

In the public domain as the number of users is large, Attribute based encryption is used to simplify the key management. The secret key is generated based on certain of attributes. Here the user has to depend on a trusted authority. To solve this problem Multiple Authority ABE is used for data encryption. The authorities are multiple “attribute authorities” (AA), each one controlling a disjoint set of attributes. In PUDs role attributes are defined for the users and the owner can provide fine grained access control to the users.

In the personal domain (PSD) the number of users is less and the owner can provide access privileges to each user. Here KP ABE is used to encrypt the data. In PSD, data access is provided based on data attributes of a file. In this case the key is linear with the number of file categories a user can access.

Using both MAABE and KP ABE provide for a simplified key management and scalability. To provide high security AES algorithm is used.

A. Multiple Authority ABE (MA-ABE)

In this scheme several algorithms are used:

1) ***System Set-up:*** The system defines a common universe of data attributes shared by the PSD. The attributes are role based. And it produces all the system public parameters PK and the master key (MK) for attribute authorities.

2) ***Key Generation:*** The key generation algorithm uses the master key MK and a set of attributes Ur that describe the key, and outputs a secret key SK for user U. SK should contain at least one attribute from every type of attributes governed by AA.

3) ***Encryption:*** The encryption algorithm will encrypt the message M taking the public parameter PK and an access structure A over a set of attributes (Ur) and it will generate the cipher-text CT.

4) ***Decryption:*** The decryption algorithm helps to decrypt the message taking PK and the cipher text CT which was obtained for set of attributes Ur, and a private key SK for Ur. If Ur satisfies the access structure A, then the algorithm will decrypt the cipher text and return a message M.

5) ***Key Distribution:*** This can be done in two ways. First using the algorithm the owner can create the user key using the public key PK, the master key MK, and a user name U. The PK will be used by AA to out put a secret key to the user which can be used for decryption. Alternately the AA gets a request for a secret attribute key and the algorithm is used to verify whether U with public key PK has the required set of attributes. If the condition is satisfied the key is generated.

B. Key Policy ABE (KP-ABE)

In MA the secret key is associated with Attribute Authorities, whereas in KP-ABE secret key is associated with the access structure. At times some of the authorities may be corrupt, then KP ABE is used. It is also used in PSD. KP ABE uses the following algorithms:

1) ***Setup:*** This is a randomized algorithm that takes a security parameter as an only input. It generates the public parameters PK and a master key MK. also called Attribute Key.

2) ***Encryption:*** This is an algorithm that takes as input - an access structure ‘A’ i.e. a set of attributes, and the public parameters PK. In this case the output is cipher text E.

3) ***Key Generation:*** This is a randomized algorithm that takes as input - an access structure A, the master key MK and also the public parameters PK. The output here is a decryption key or Attribute Key AK.

4) ***Decryption:*** This algorithm takes as input - the cipher text E that was encrypted under access structure A, the decryption key AK and the public key PK. The output is message M.

As can be seen encryption of data is done using two techniques, MA ABE which is across different attribute authorities and the other is lateral across the different attributes which are governed by the same KP ABE.

C. *Data Revocation:* Using both MA ABE and KP ABE, an authority can re-encrypt the cipher-texts and update the user's keys which in turn will revoke a user or user's attributes.

D. *Policy Updates:* The owner can update the attributes in the cipher text and the server supports operations like add/ delete/ modify.

E. *Break Glass:* In times of emergency the access rights are given to the Emergency department, the staff get temporary read keys and later the rights can be revoked.

VI. System Architecture

In Fig. 1, initially (1), the PHR owner selects the attributes from the PHR file to give the access rights to the users. In this (2), the owner provides access keys to attribute authorities in case of public domain and to the users in personal domain. (3) here, the PHR file is encrypted symmetrically with advanced encryption standard (AES) and the symmetric data key is again encrypted asymmetrically with attribute based encryption (ABE), according to an access policy over a set of attributes, which specifies with whom the owner is willing to share his/her data and the data owner stores the encrypted data along with encrypted license (which may contain the encrypted data key encrypted using ABE) on the trusted third party (cloud server). In the (4), the users request the data from the cloud server using the access keys. Here (5), the users can access the data from the cloud server after all the key verifications and access policy verifications are satisfied. Here the emergency staff can also access the data using the keys in emergency situations. (6), here PHR owners assign the access keys to the emergency department for the break glass scenario. (7), the access keys are provided to the emergency staff from the emergency department in case of emergency. And finally (8), the access keys assigned to emergency department is taken back by the owner after the break glass scenario, and again assigns new key to the emergency department. The revocation of access keys, access policy rights from the cloud server can also be done by the PHR owners when they required.

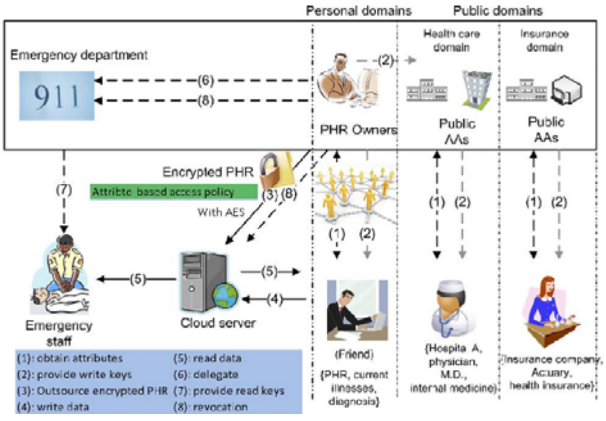

Fig. 1 Architecture

VII. Security

A. Security Issues

Using the dual level encryption technique the privacy of the data is assured. Attacks like eaves dropping, man in the middle problem or denial of service can be resisted. Yet there is a possibility of insider or outsider attacks. The PHRs can be made more secure. Combining this encryption with AES enhances the security of the data. An intruder cannot find the encrypted data easily. It is used for secure transmission of data in encrypted format. In this system AES is used for sending user authentication data in encrypted format.

B. Solutions

The data has to be encrypted. For encryption of data several cryptographic algorithms are used. These algorithms may be symmetric or asymmetric. DES and AES are symmetric while RSA, ECC are asymmetric. The following table shows the performance of these algorithms.

TABLE I

COMPARISON BETWEEN SYMMETRIC AND ASYMMETRIC ALGORITHMS

	DES	AES	RSA	ECC
Factors Contributor	IBM 75	Rijman, Joan	Rivest, Shamir 78	Neal Koblitz,Victor S. Miller
Key Length	56-bits	128,192, and 256	Based on No.of bit in N=p*q	135 bits
Block Size	64-bits	128 bits	Variant	Variant
Security Rate	Not enough	Excellent	Good	Less
Execution Time	Slow	More fast	Slowest	Faster

Among these AES is a strong algorithm.

C. Advanced Encryption Standard (AES)

1) The Advanced Encryption Standard (AES) is an encryption algorithm for securing sensitive data. It has been adopted by the United States government as an Advanced Encryption Standard, a standard algorithm used to encrypt and decrypt sensitive information. AES is a symmetric block cipher with a block size of 128 bits. It allows for three different key lengths which can be 128 bits, 192 bits, or 256 bits; referred to as AES-128, AES-192, and AES-256, respectively. The number of rounds in the encryption process for AES-128 is 10, for AES-192 it is 12, and for AES-256 it is 14.

2) The major loop of AES executes the functions given below:

Functions of Advanced Encryption Standard

- SubBytes()
- ShiftRows()
- MixColumns()
- AddRoundKey()

AES makes use of 10, 12 and 14 rounds. The plain text is transformed into cipher text after repeated transformation rounds in AES. This makes the data secure on the cloud.

3) *Implementation:* AES-128, AES-192, AES-256 (128, 192 and 256 are bits) process the data block in, respectively, 10, 12, or 14 rounds. The transformations are predefined. All the rounds are similar except the last one where the transformation is missed. The rounds operate on two 128 bits i.e., state and round key. Each round from 1 to 10 or 12 or 14 uses a different round key.

The data block is processed as follows:

- The AES encryption routine begins by copying the 16-byte input array into a 4×4 byte matrix named State.
- Input data block also known as state is XORed with the first 128-bits of the cipher key.
- Then the resulting State is serially passed through 10/12/14 rounds.
- The result of the last round is encrypted data.

The process of AES encryption algorithm using 128-bit key, is diagrammatically represented in figure 2.

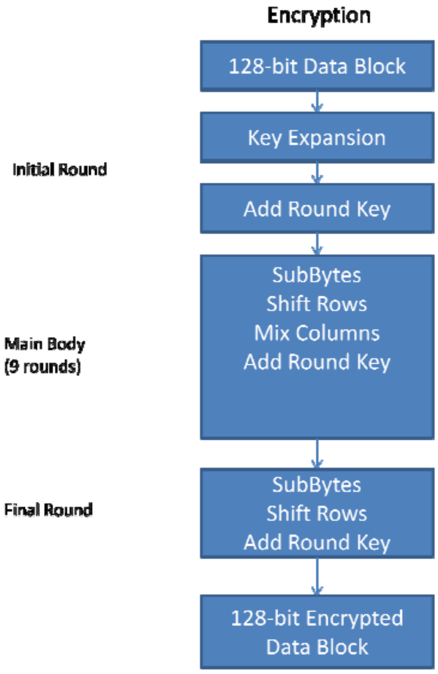

Fig. 2. Process of AES

4) *Algorithm:*

a) Key Expansion: Using the key schedule of Rijndael, round keys are derived from the cipher key.

b) Initial Round - AddRoundKey: Then using bitwise XOR each byte of the state is combined with the round key.

c) Rounds

i) SubBytes*:* This is a non-linear substitution step where each byte is swaped with another according to a lookup table.

ii) ShiftRows*:* In this transposition step each row of the state is shifted cyclically a certain number of steps.

iii) MixColumns*:* A mixing operation which operates on the columns of the state, combining the four bytes in each column.

iv) AddRoundKey

d) Final Round (no Mix Columns)

v) SubBytes

vi) ShiftRows

vii) AddRoundKey

5) In encryption AES is more suitable. ABE is considered to be more expensive. So the data is not directly encrypted using ABE. Generally symmetric key is used for encrypting bulk of the data and asymmetric key like ABE is suitable for encrypting short key value. First data is encrypted using AES with 128 bits keys and the AES keys are again encrypted/ decrypted using ABE and are sent together with ciphertext.

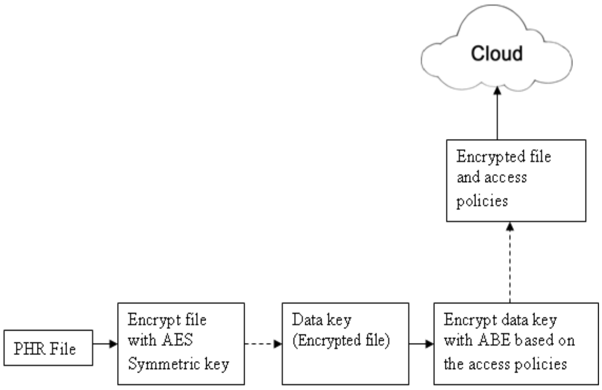
Cloud
Encrypted file and access policies
PHR File
Encrypt file with AES Symmetric key
Data key (Encrypted file)
Encrypt data key with ABE based on the access policies

Fig. 3. Encrypting data with AES and ABE

Figure 3, shows encryption of data using both AES and ABE.

The PHR file is first encrypted symmetrically using AES algorithm key and then, that key called data key is again encrypted asymmetrically with ABE algorithm according to the access policies over a set of attributes that specifies with whom the owner is able to share her/his data. Then the encrypted data and the encrypted data key are stored on the cloud server.

VIII. CONCLUSION

The paper discusses secure and scalable sharing of data in a multiple owner, multiple user, multiple authority scenario. ABE is used to encrypt the owner's data and MA ABE is used to manage efficient and on-demand user revocation, dynamic policy changes. Security is ensured through AES.

REFERENCES

[1] Vipul Goyal, Omkant Pandey, Amit Sahai and Brent Waters, Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data *ACM CCS (2006)*

[2] S. Narayan, M. Gagne´, and R. Safavi-Naini, "*Privacy Preserving EHR System Using Attribute-Based Infrastructure*," Proc. ACM Cloud Computing Security Workshop (CCSW '10), pp. 47-52, 2010.

[3] S. 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.

[4] M. Chase and S.S. Chow, "*Improving Privacy and Security in Multi-Authority Attribute-Based Encryption*," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 121-130, 2009.

[5] R.Sinduja and G.Sumathi. "*Improving Cloud Security by Enhanced HASBE using Hybrid Encryption Scheme*". Compusoft 2 May 2013