Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 5, May -2017

Data Security in Healthcare Domain by using An Efficient File Hierarchy Attribute-Based Encryption Scheme

Dinesh P Kulkarni[1], Dr. Mrs. Sadhana Chidrawar[2]

[1]*M.E. CSE (Student), MPGI, Nanded*
[2]*Dean (Guide), MPGI, Nanded*

Abstract —*In This paper, data security aspect for healthcare domain is proposed by proposing an efficient file hierarchy attribute based encryption method. The multilevel access structure is combined into single level access structure and then subfiles are encrypted with the integrated access structure. The cipher texts of the attributes shared by the files which saves the time cost of encryption.*

Keywords-*cloud computing, Data encryption, Security in healthcare, file hierarchy, attribute based encryption*

I. INTRODUCTION

In healthcare domain, to protect data from leakage, users needs to take care of some security aspects in terms of encryption before sharing the data with other users. Access Control [1], [2] is paramount as it will restrict un-authorized access to the data. With the advancement in the technology and mobile devices, online data sharing has become a new era to share the data over social networking. Meanwhile cloud is one of the best environments to solve the explosive expanding of data sharing. Now a days, attribute-based encryption (ABE) has been widely used since it can keep data privacy and realize fine-grained, one-to-many, and non interactive access control.

Cipher text-policy attribute based encryption (CP-ABE) is one of feasible schemes which has much more flexibility and is more suitable in health care domain.

In cloud computing, owner or authorized administrator accepts the user enrollment and creates some parameters. Cloud manager of cloud servers and provides multiple services for client. Data owner or hospital administrator encrypts and uploads the generated cipher-text to Cloud Service Provider. User downloads and decrypts the interested cipher-text from Cloud Service Provider. The shared files usually have hierarchical structure. That is, a group of files are divided into a number of hierarchy subgroups located at different access levels. If the files in the same hierarchical structure could-based encrypted by an integrated access structure, the storage cost of Ciphertext and time cost of encryption could be saved.

II. IMPLEMENTATION

In this paper we present encryption mechanism by using an access control scheme. The scheme has several benefits which make it especially suitable for delivery of the data. For example, it is extremely scalable by allowing a data owner or cloud administrator or healthcare domain administrator to grant data access privileges based on the data consumers' attributes aspects like age, nationality, gender, pin code rather than an explicit list of user names; and it ensures data security and exclusiveness of access of data by implementing attribute-based encryption. For this purpose, we introduce a File Hierarchy Ciphertext Policy Attribute Based Encryption (FH-CP-ABE) technique. FH-CP-ABE encrypts multilevel access structure within integrated cipher text, so as to enforce flexible attribute-based access control on scalable media. Specifically, this scheme will create a content key which is used to FH-CP-ABE encryption, encrypts data units with the corresponding keys, and then creates Content Key Ciphertext (CKCT). User can decrypt the Content Key Ciphertext by using FH-CPABE decryption into decrypted content key. Then content keys can be decrypted using symmetric decryption algorithm (DES, AES). The scheme offloads computational intensive operations to cloud servers while without compromising user data security.

J. Bethencourt, Amit Sahai, Brent Waters [11], a system for realizing complex access control on encrypted data that we call Ciphertext-Policy Attribute-Based Encryption. By using our techniques encrypted data can be kept confidential even if the storage server is not trusted source; moreover, our methods are more secure against collusion attacks or wanna-cry attacks. Earlier Attribute Based Encryption method used attributes to explore the encrypted data and built sets of rules into user's keys; while in our system attributes are used to explore a user's credentials, and a party encrypting data determines a rule for who can decrypt. Thus, our methods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC).

In addition, we are providing an implementation of this technique and give the result in terms of performance measurements. The system allows for a new type of encrypted access control where user's private keys are specified by a set of attributes and a party encrypting data can specify a policy over these attributes specifying which users are able to

decrypt. The system allows policies to be expressed as any monotonic tree access structure and is resistant to collusion attacks in which an attacker might get multiple private keys. In the future, it would be interesting to consider attribute-based encryption systems with different types of express-ability.

Ciphertext-policy attribute based encryption (CP-ABE) is a promising cryptographic tool, where the encrypt-admin can decide the access structure that will be used to protect the private sensitive data. However, current CP-ABE schemes suffer from the issue of having long or lengthy decryption keys, in which the size is linear to and dependent on the number of attributes. This drawback prevents the use of lightweight devices in practice as storage of the decryption keys of the CP-ABE for us.

In this research paper, we provide an affirmative answer to the above long standing problem, which will make the CP-ABE very practical. We propose a novel CP-ABE scheme with constant-size decryption keys independent of the number of attributes.[15]

Implementation of FH-CP-ABE:

As shown in Figure 1, the model implementation of data security in cloud computing is given below, which consists of four different entities:

1. authority,
2. CSP,
3. data owner
4. and user.

In this implementation of logic, we assume that data owner has 'k' files with k access levels and M ={m1,..., mk }is shared and hosted from cloud environment which supported cloud computing. Here, m1 is the highest hierarchy and mk is the lowest hierarchy.

If a user are able to decrypt m1,then the user can also decrypt m2,…....., mk .

We are going to see all the terms in detail:

1. **Authority**:
   It is a completely trusted entity and accepts the user enrollment in cloud computing. And it can also execute Setup and KeyGen operations of the proposed scheme.
2. **Cloud Service Provider (CSP):**
   It is a semi-trusted entity in cloud system. It can honestly perform the assigned tasks and return correct results. However, it would like to find out as much sensitive contents as possible. In the proposed system, it provides Ciphertext storage and transmission services.
3. **Data Owner**: It has large data needed to be stored and shared in cloud system. In our scheme, the entity is in charge of defining access structure and executing Encrypt operation. And it uploads Ciphertext to CSP
4. **User**: It wants to access a large number of data in cloud system. The entity first downloads the corresponding Ciphertext. Then it executes Decrypt operation of the proposed scheme.

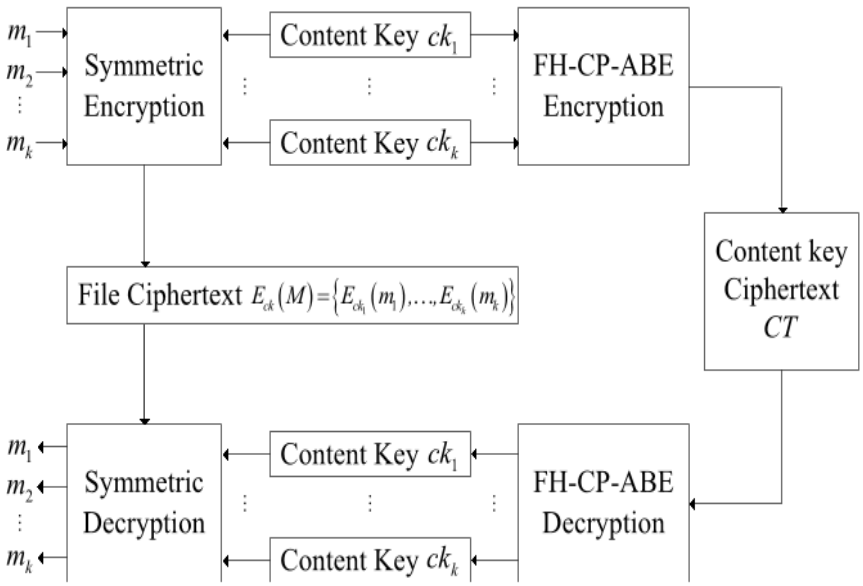

Figure 1

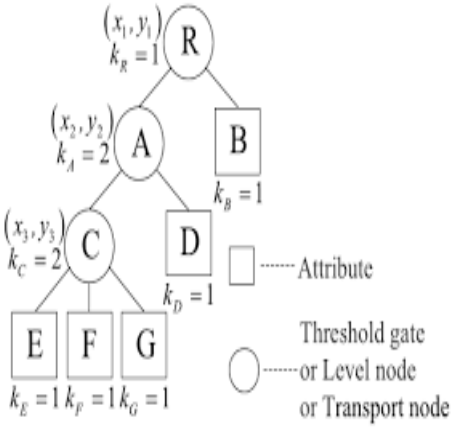

Figure 2

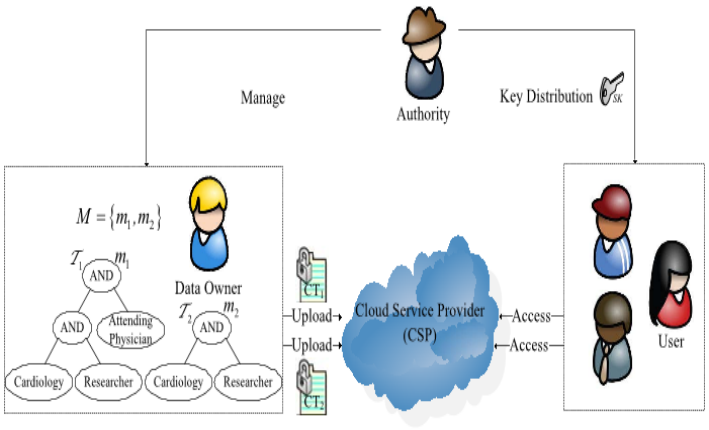

Figure 3

Hierarchical Attribute-Base Encryption (HABE) and Hierarchical Attribute Set Based Encryption (HASBE):-

This scheme Hierarchical attribute-based encryption (HABE) is derived by Wang et al [9]. It is designed to achieve fine grained access control in cloud storage services. It is a combination of HIBE and CP-ABE. In the HABE scheme, there are multiple keys with different usages. Therefore, we first provide a summary of the most relevant keys in table available in appendix to serve as a quick reference.

Then, HABE scheme is defined by presenting randomized polynomial time algorithms as follows:

1. ***Setup*** (K) ➔ (params,MK0):
   The RM takes a sufficiently large security parameter K as input, and outputs system parameters params and root master key MK0.
2. ***CreateDM*** (params,MKi, PKi+1) ➔ (MKi+1):
   Whether the RM or the DM generates master keys for the DMs directly under it using params and its master key.
3. ***CreateUser*** (params,MKi, PKu, PKa) ➔(SKi,u, SKi,u,a):
   The DM first checks whether U is eligible for a, which is administered by itself. If so, it generates a user identity secret key and a user attribute secret key for U, using params and its master key; otherwise, it outputs "NULL".
4. ***Encrypt*** (params; f ;A; {PKa|a E A})➔(CT): A user takes a file f, a DNF access control policy A, and public keys of all attributes in A, as inputs, and outputs a Ciphertext CT.
5. ***Decrypt***(params, CT, SKi,u, {SKi,u,a|aECCj}➔(f):A user,whose attributes satisfy the j-th conjunctive clause CCj, takes params, the ciphertext, the user identity secret key, and the user attribute secret keys on all attributes in CCj, as inputs, to recover the plaintext.

However, HABE uses disjunctive normal form policy. It assumes all attributes in one conjunctive clause those are administrated by the same domain master. Thus the same attribute may be administrated by multiple domain masters according to specific policies, which is most complicated to implement in practice. This scheme has issues with multiple values assignments. HASBE scheme is proposed and implemented by *Zhiguo Wan et al [10]*.

The cloud computing system consists of five types of parties: a cloud service provider, data owners, data consumers, a number of domain authorities, and a trusted authority. The cloud service provider manages a cloud and provides data storage service. Data owners encrypt their data files and store them in the cloud for sharing with data consumers.

To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them. Each data owner/consumer is administrated by a domain authority. A domain authority is managed by its parent domain authority or the trusted authority. The HASBE scheme extends the ASBE scheme to handle the hierarchical structure of system as shown in figure-4.

The trusted authority is responsible for managing top-level domain authorities. It is root level authority. For example, for an IT enterprise, employees are kept in the lowest domain level and above that there is department and above that there is top level of domain we call it as a trusted domain.

It generates and distributes system parameters and also rootmaster keys. And it authorizes the top-level domain authorities. A domain authority delegates the keys to its next level sub-domain authorities. Each user in the system is assigned a key structure.

Key specifies the attributes associated with the user's decryption key. *Zhiguo Wan et al* [10] given a HASBE scheme for scalable, flexible, and finegrained access control in cloud computing.

The HASBE scheme consists of hierarchical structure of system users by using a delegation algorithm to CP-ASBE. HASBE supports compound attributes due to *flexible attribute set combinations* as well as achieves efficient user revocation because of attributes assigned multiple values. Thus, it provides more scalable, flexible and fine grained access control for cloud computing.

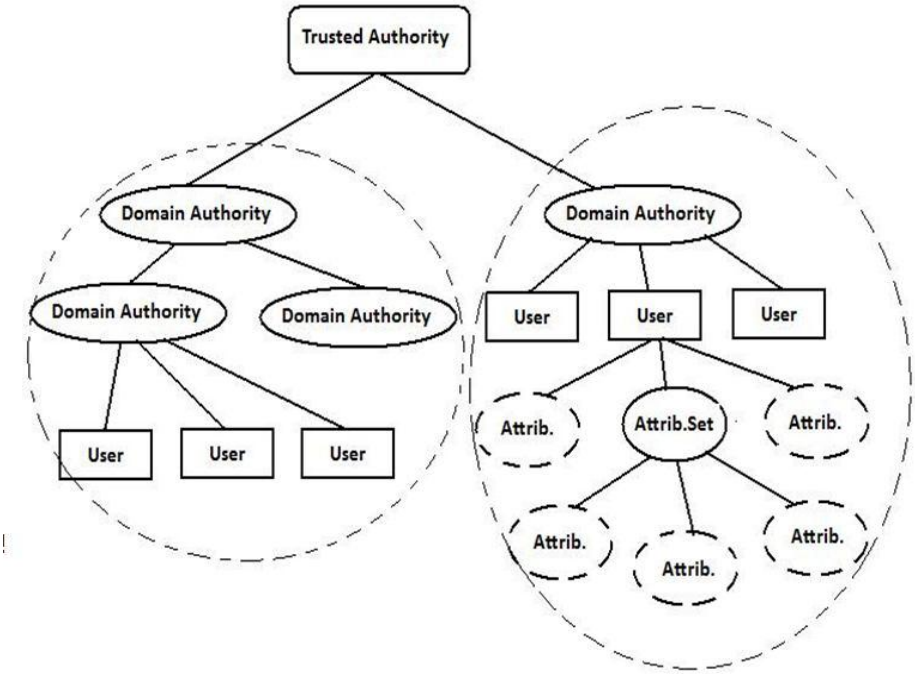

Figure 4

III. CONLUSION

In this Paper, we have proposed data security solutions in Cloud environment for Health care domain,We proposed a variant of CP-ABE to efficiently share the hierarchical files in cloud computing. The hierarchical files are encrypted with an integrated access structure and the Ciphertext components related to attributes could be shared by the files. Therefore, both Ciphertext storage and time cost of encryption are saved. The proposed scheme has an advantage that users can decrypt all authorization files by computing secret key once. Thus, the time cost of decryption is also saved if the user needs to decrypt multiple files. Moreover, the proposed scheme is proved to be secure under DBDH assumption

IV. REFERENCES

[1]. D.Boneh and M. Franklin. "Identity-Based Encryption from the Weil Pairing." *In Proc. of CRYPTO'01, Santa Barbara, California, USA, 2001.*
[2]. J.Bethencourt, A. Sahai, and B. Waters. "Ciphertext-Policy Attribute- Based Encryption." In Proc. of SP'07, Washington, DC, USA, 2007.
[3]. A.Sahai and B. Waters. "Fuzzy Identity-Based Encryption." In Proc. Of EUROCRYPT'05, Aarhus,Denmark, 2005.
[4]. V. Goyal, O. Pandey, A. Sahai, and B. Waters. "Attribute-Based Encryption for Fine-grained Access Control of Encrypted Data". In Proc. of CCS'06, Alexandria, Virginia, USA, 2006.
[5]. R.Ostrovsky, A. Sahai, and B. Waters. "Attribute-based encryption with non-monotonic access structures". In Proc. of CCS'06, New York, NY, 2007.
[6]. Zhibin Zhou, Dijiang Huang" On Efficient Ciphertext-Policy Attribute Based Encryption and Broadcast Encryption"
[7]. Rakesh Bobba, Himanshu Khurana and Manoj Prabhakaran, "Attribute- Sets: A Practically Motivated Enhancement to Attribute-Based Encryption", July 27, 2009
[8]. J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-policy attribute based encryption," *IEEE Symp. Security and Privacy*, Oakland, CA, 2007.
[9]. G.Wang, Q. Liu, and J.Wu, "Hierachical attibute-based encryption for fine-grained access control in cloud storage services," in *Proc. ACM Conf. Computer and Communications Security (ACM CCS)*, Chicago, IL, 2010.
[10]. Zhiguo Wan, Jun'e Liu, and Robert H. Deng, "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing" IEEE Transactions On Information Forensics And Security, Vol. 7, No. 2, April 2012

APPENDIX

TABLE 1: COMPARISON REVIEW OF VARIOUS ABE TECHNIQUES

Parameters v/s ABE-Technique	Fine-grained access control	Efficiency	Computational Overhead
KP-ABE	Low, High if there is re-encryption technique	Average High for broadcast type system	Most of computational overheads
EKP-ABE	Better Access control than that of KP-ABE	Higher than KP-ABE, allows constant cipher text only	Reduces computational overheads
CP-ABE	Average Realization of complex Access Control	Average Not efficient for modern enterprise environments	Average computational overheads
CP-ASBE	Better Access Control than that of CP-ABE	Better than CP-ABE as there is Less collusion attacks	Lower than CP-ABE computational overheads
HIBE	Lower than CP-ASBE	Better, Lower as compared to ABE schemes	Most computational overheads
HABE	Good Access control	Flexible and scalable	Some of overhead
HASBE	Better Access control	Most efficient and flexible	Less overhead than others

