HABE - Hierarchical Attribute-Based Encryption Scheme in Cloud Environment

R.Rengaraj alias Muralidharan[1], D.Priya Dharshini[2], D.Sangavi [3], K.Vaishali[4], S.B.Vigneshwari[5]
[1, 2, 3, 4, 5] Saranathan College of Engineering,TamilNadu, India

Abstract- *Cloud Computing has been envision as the next production architecture of IT Enterprise. The major problem in cloud computing is secured data sharing. CPABE Cipher-text Policy Attribute-Based Encryption has been a preferred encryption technology in cloud for data security. In this paper, HABE Hierarchy Attribute-Based Encryption scheme is applied for secured data sharing. It provides the hierarchy structure of shared files. These hierarchical files are encrypted and the cipher text components related to attributes could be shared. Therefore, both cipher text storage and time cost of encryption are saved. The proposed scheme is highly secured because the encrypted files are stored in multiple servers. The random key generation process is used for encrypting each and every file in hierarchical structure of shared files. The users can access the data from data owner through cloud provider in real time dynamic cloud environment.*

Keywords- Discrete event simulation, queuing system, size delay function

I. INTRODUCTION

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. In cloud computing, the user needs to encrypt the data before its being shared, in order to prevent the data loss. Data security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, databases and websites. Data security also protects data from corruption. Encryption is an interesting piece of technology that works by scrambling data so it is unreadable by unintended parties. Once encrypted, the message literally becomes a jumbled mess of random characters. But, equipped with the secret code, we can decrypt it and find the original message. The encryption algorithms in data security are Triple DES, AES, Re-encryption, Blowfish, RSA etc.

II. RELATED WORKS

R.Sumathi and E.Kirubakaran[1] proposed a re-encryption technique for providing data security in which data owner encrypts the own file and send the cipher text file to CSP. In turn, CSP re-encrypt the data for providing higher security. In this paper, they loaded the entire file for encryption as well as re-encryption. It is time consuming process.

The methods are theoretically closer to traditional access manage methods such as Role-Based Access Control (RBAC). Attribute-based encryption for fine-grained access control of encrypted data by V. Goyal et al.[3].In an ABE system, a user's keys and cipher texts are label with sets of descriptive attribute and a fastidious key can decrypt a particular cipher texts only if there is a competition between the attribute of the cipher text and the user's key. The cryptosystem of Shay and Waters allowed for decryption when at least k attributes overlap between a cipher text and a private key. Fine-grained two factor access controls for web-based cloud computing services J.K. Liu et al.[4]. In a fine-grained two-factor access control protocol for web-based cloud compute services, using a trivial security device. The device has the following properties:

(1) It can compute some light algorithms, e.g. hash and exponentiation.
(2) It is tamper resistant, K times attribute-based anonymous access control for cloud computing was stated by T.H.Yuen et al.[5]. ABE only deals with authenticated access on encrypted data in cloud storage service. The cloud server may encrypt a chance message using the access policy and asks the user to decrypt it. A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing was explained by K. Liang et al.[6]. To achieve more flexibility on re-encryption, many variants of PRE have been proposed, such as provisional PRE (CPRE), Identity-Based PRE (IBPRE), and Attribute-Based PRE (ABPRE). CPRE allows an encryption connected with a condition to be converted to a new cipher text tagged with a new form. The technologies of IBPRE and ABPRE are rather similar, and a main difference between them is ABPRE enjoys more expressiveness in data sharing.

Heavy computation overhead on the data owner is introduced for key distribution and data management when fine grained data access control is desired and thus do not

scale well. Experts claim that their clouds are 100% secure - but it will not be their head on the block when things go away. It's often stated that cloud computing security is better than most enterprises. The limitations in the contemporary cloud security are data store in cloud but fully not secure, the arbitrary-state ABE scheme to solve the problem of the dynamic membership management, high step by step using secured key.

III. TECHNIQUES IN THE PRPOSED SYSTEM

1. MERKLE HASH TREE TECHNIQUE

- In cryptography and computer science a hash tree or merkle tree is a tree in which every non leaf node is labeled with the hash labels or values of its child node.
- Hash tree allow efficient and severe verification of the contents of large data structures. Hash tree can be used to verify any kind of data stored, handled and transferred in between computers.
- It is mainly used to make sure that data blocks received from other peers in peer-peer network are received undamaged and unaltered and also check that the other peers do not lie and send fake blocks
- Cryptographic hash functions such as SHA-2 is used for hashing. If the hash tree only needs protect against unintentional damage, much less secure check sums such as CRC's can be used.
- Top of hash tree is top hash or root hash.

Before downloading a file on P-P networking most of the cases top hash is acquired from a trusted source. The merkle tree is given below:

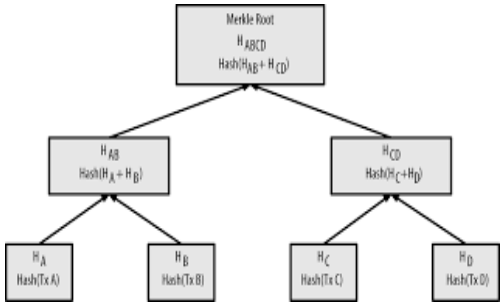

Figure 1. Merkle hash tree

In Fig.1, The top hash or root hash contains the value HABCD , Here we calculate the hash value as Hash(HAB +HCD).The left child of root hash HAB is calculated as (HA + HB).The right child of root hash HCD is calculated as (HC + HD).The left leaf node of HAB is calculated as HA = Hash(TXA).The right leaf node of HAB is calculated as HB =(TXB).The left leaf node of HCD is calculated as HC = Hash(TXC).The right leaf node of HCD is calculated as HD = Hash(TXD).

The cons in this algorithm are, the merkle hash root does not indicate the tree depth, enabling a second preimage attack in which an attacker creates a document other than the original that as same merkle hash root.

2. T- COLORING ALGORITHM

The t-coloring of a graph assigns to each vertex of a graph an integer ("color") such that the absolute value of difference between two colors of adjacent vertices does not belongs to the given set.

- In graph theory, a t-coloring of a graph G=(V,E) given the set T of non-negative integers containing zeros, is a function C:V(G)->N that maps each vertex of G to a positive integer such that (u,w) E(G) =>|C(u) –C(w)| E T.
- V is the vertex and E is the edge of the graph.
- In simple words, the absolute value of difference between two colors of adjacent vertices mesh not belongs to fixed set T.

Pseudo code:

T
T
Here T and T are the adjacent nodes.
(For example,
if T={0} T={0}, adjacent vertices must have different colors;
if T={0,1} T={0,1} , the color of adjacent vertices must differ by at least 2 2.)
The following section elaborates the various modules in the proposed system.
Figure 3 The individual module wise discrete event simulation model is given as below

IV. PROPOSED SYSTEM

Secure data sharing in cloud using HABE is illustrated in fig.2 and fig 3.

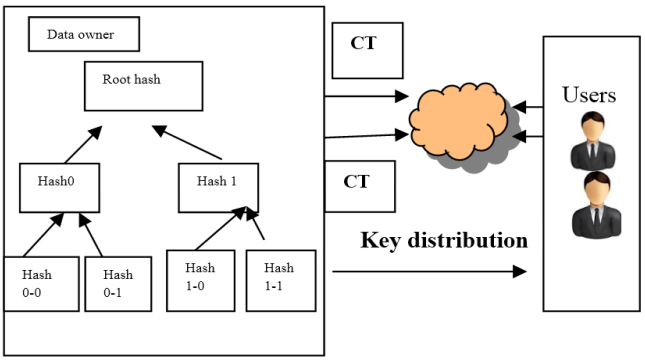

Figure 2. An example of secure data sharing in cloud using HABE.

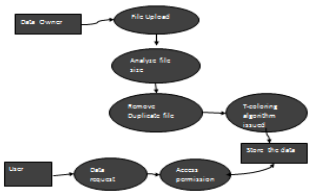

Figure 3. Data flow diagram for the proposed system.

Recently, Attribute-based encryption has been attracted much more attentions since it can keep data privacy and realizes fine grained, one-to-many and non-interactive access control. Cipher text-policy attribute based encryption is one of the feasible schemes used for encrypting general application data.

The steps involved in this system are, first, the cloud owner registers into the system by giving their personal details. They have to create some parameters for authentication purpose in registration process itself. For secure sharing the personal record information, the data owner first register into the system, then using their login credentials they enters into the data sharing system. After that, the medical records have to be loaded. If the medical records contain huge volume of data, the entire size of the file has to be analyzed. Then the data in the files are divided into equal parts using the split-up operation. For split-up here merkle hash tree technique is used. Cryptographic hash functions like MD5 and SHA-2 are used for hashing. Top of the hash tree is top hash. Before downloading a file on peer-peer networking most of the cases top hash is acquired from a trusted source. Using this merkle hash tree technique, the personal health record file is divided in hierarchical structure. These hierarchical files are encrypted with the help of HABE hierarchical attribute based encryption. These encrypted files are loaded into the cloud CSP provides the storage service in cloud. Random key generation process is used for key generation of each and every node in hierarchical tree. The authorized user who needs the data can access the data from data owner trough the cloud provider in real time dynamic cloud environment. In Fig.1 we find the merkle hash tree technique is used in hierarchical form. Here the top of the hash is called top hash or root hash. The child nodes of this merkle hash tree contain contains the spitted part of the medical record with hash values. All the data are stored in hierarchical structure. Each and every node of this hash trees are having different hash values. This hash values are generated with the help of cryptographic hash function generation algorithms.

- The HABE scheme provides encryption process in hierarchical structure. First the entire file is divided in hierarchical structure and then encrypts it with the help of secured encryption technique called hierarchical attribute base encryption HABE.
- Generate separate key for each and every node in hierarchical tree structure with the help of the operation called random key generation. Using this random key generation, we can generate keys individually for all the nodes of split-up file. This will provide higher level of security.
- The key for decryption process by the user who needs the data can get their key directly from the owner. If we use third party for key distribution process there is chance that the third party may be act as a un-trusted one.
- Here for dividing the data in the form of hierarchical structure we are going to use merkle hash tree algorithm. This is one of the best techniques used for generating hierarchical structure of shared files.
- Another major process we are going to do is we can store the split-up files in different servers in cloud environment. For storing like this we use T-coloring algorithm that will find the distance between the two servers. This will give only the partial day when any unauthorized user may tries to access the data by hacking the key.

It should be noticed that the proposed scheme differs from the CP-ABE schemes, which utilize the user layered model. In addition the part of the work is presented in [1]. The work presented in that conference paper is rough and incomplete, where some important aspects haven't been considered. The advantages of the proposed system, data high secure medical report store in cloud, secure data sharing in user, authority person only access data.

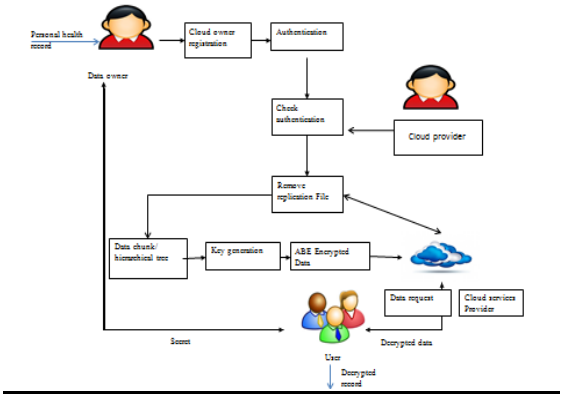

Figure 4. Architecture diagram for HABE

The components involved in the proposed system are in the following section, 1 to 7

4.1 COMPONENTS

4.1 Data Owner Registration and Authentication

Data owner is the person who is going to own the data. Data owner first register themselves by entering their details.

4.1.2 Check Authentication

Check authentication phase is used for checks whether only authorized owner is accessing our system or not.

4.1.3 Remove Replication

The replicas are removed with the help of the keyword given by their owner.

4.1.4 Data Chunking

File size loaded single file is divided into several chunks. Finally spitted data are stored in different locations by taking some default name by itself.

4.1.5 Key Generation

Key generation process is carried out with the help of random key generation operation.

4.1.6 ABE Encrypted data

After dividing the file into hierarchical structure, each and every chunk is encrypted using attribute based encryption scheme.

4.1.7 Decrypted Data

Now the data owner has to verify whether the user requesting for data is authorized one or not. If he is the authorized user the key for decrypting the data is directly send by the owner to the user.

The performance of the proposed system is measured using the performance metrics such as storage overhead communication cost and computation efficiency. The storage overhead is one of the most significant issues of the access control scheme in cloud storage systems. In our scheme, besides the storage of attributes, each sub server also needs to store a public key and a secret key for each user in the system.

The user is assigned to data owner from the Provider. Each user can freely get the cipher texts from the server. To decrypt a cipher text, each user may submit their secret keys issued by data owner together with its global public key to the server and ask it to generate decryption token for some cipher text receiving the decryption token, the user can decrypt the cipher text by using its global secret key.

V. SCREENSHOTS

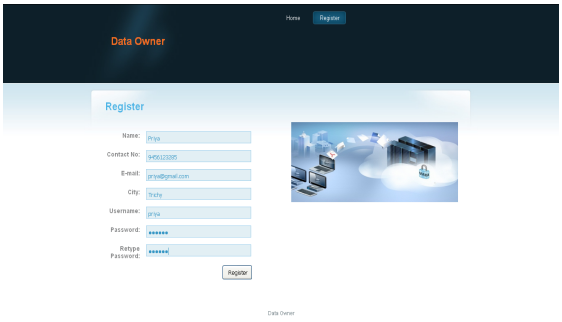

Figure 5. Data owner registration

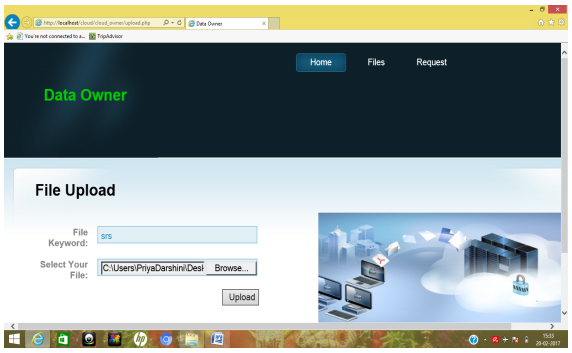

Figure 6. File uploading

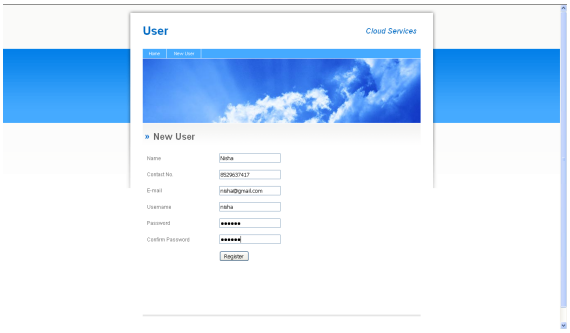

Figure 7. User Registration

Figure 8. User's request

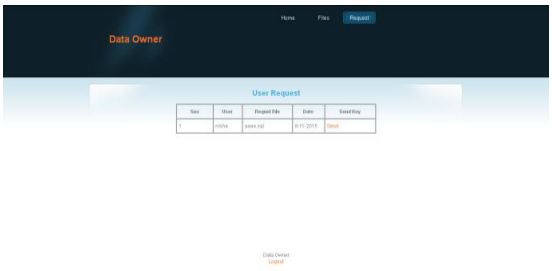

Figure 9. Data owner approval

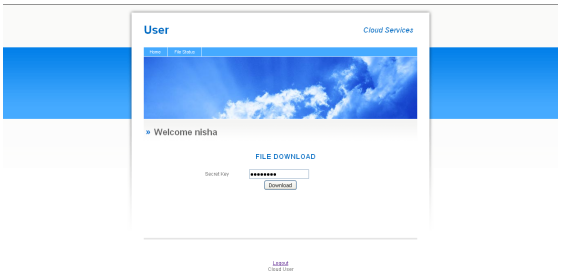

Figure 10. User downloads the file with the key given by the data owner.

VI. CONCLUSION

In this proposed system, the security of the system is highly increased due to merkle hash tree and T-coloring algorithms. We are using HABE for encryption that is hierarchical attribute based encryption. This will provide higher efficiency, both cipher text storage and time cost of encryption are saved. Here subdivision of data into layered form, if a particular data from a node is taken, it doesn't reveal the entire information. The key for decryption process is directly given to the user by the data owner. Hence, only the authorized user can access the data.

REFERENCES

[1] R.Sumathi and E.Kirubakaran, "SCEHSS: Secured Cloud-based Electronic Health record Strategy Scheme with Re-Encryption at Cloud Service Provider", International journal of Computer and Communication Engineering, vol. 2,no. 2, pp. 162-166, March 2013.

[2] C. Chu, W. T. Zhu, J. Han, J. K. Liu, J. Xu, and J. Zhou, "Security concerns in popular cloud storage services," IEEE Pervasive Computing, vol. 12, no. 4, pp. 50–57, October-December 2013.

[3] T. Jiang, X. Chen, J. Li, D. S. Wong, J. Ma, and J. K. Liu, "TIMER: secure and reliable cloud storage against data re-outsourcing," Proceedings of the 10th International Conference on Information Security Practice and Experience, vol. 8434, pp. 346–358, May 2014.

[4] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud based revocable identity-based proxy re-encryption scheme for public clouds data sharing," Computer Security in ESORICS 2014, vol. 8712, pp. 257–272, September 2014.

[5] T. H. Yuen, Y. Zhang, S. Yiu, and J. K. Liu, "Identity-based encryption with post-challenge auxiliary inputs for secure cloud applications and sensor networks," Computer Security in ESORICS 2014, vol. 8712, pp.130–147, September 2014.

[6] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, T. V. X.Phuong, and Q. Xie, "A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing," IEEE Transactions on Information Forensics and Security, vol. 9, no. 10, pp. 1667–1680,October 2014.

[7] T. H. Yuen, J. K. Liu, M. H. Au, X. Huang, W. Susilo, and J.Zhou,"ktimes attribute-based anonymous access control for cloud computing,"IEEE Transactions on Computers, vol. 64, no. 9, pp. 2595–2608,September 2015.

[8] J. K. Liu, M. H. Au, X. Huang, R. Lu, and J. Li, "Fine-

grained two factor access control for web-based cloud computing services," IEEE Transactions on Information Forensics and Security, vol. 11, no. 3, pp.484–497, March 2016.

[9] Waters, "Fuzzy identity-based encryption," Advances in Cryptology–EUROCRYPT, pp. 457–473, May 2005.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," Proceedings of the 13th ACM conference on Computer and communications security, pp. 89–98, October 2006.

[11] W. Zhu, J. Yu, T. Wang, P. Zhang, and W. Xie, "Efficient attribute-based encryption from R-LWE," Chinese Journal of Electronics, vol. 23, no. 4,pp. 778–782, October 2014.