ISSN:0975-9646

Scalable and Flexible Access Control with Secure Data Auditing in Cloud Computing

Devi D[1], Arun P S[2]

Dept.Computer Science & Engg. , Sree Buddha College of Engg.
Alappuzha, Kerala, INDIA

[1]MTech Student, [2]Assistant Professor

***Abstract*— Cloud computing, which enables on demand network access to shared pool of resources is the latest trend in today's IT industry. Among different services provided by cloud, cloud storage service allows the data owners to store and share their data through cloud and thus become free from the burden of storage management. But, since the owners lose physical control over their outsourced data, it arises many privacy and security concerns. A number of attribute based encryption schemes are proposed for providing confidentiality and access control to cloud data storage where the standard encryption schemes face difficulties. Among them, Hierarchical Attribute Set Based Encryption (HASBE) provides scalable, flexible and fine grained access control as well as easy user revocation. It is an extended form of Attribute Set Based Encryption (ASBE) with a hierarchical structure of users. But from the view of integrity and availability, HASBE is not sufficient to provide the data owner with the ability to perform checking against missing or corruption of their outsourced data. So, this paper extends HASBE with privacy preserving public auditing concept which additionally allows owners to securely ensure the integrity of their data in the cloud. We are using homomorphic linear authenticator with random masking technique for this purpose. In this paper, the system under consideration applies to Personal Health Record domain.**

Keywords—Cloud Computing, Access control, Personal Health Record, Homomorphic Linear Authenticator, Random Masking.

I. INTRODUCTION

Cloud computing [2] is defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers) that can be rapidly provisioned and released with minimal management effort or service provider interaction. It has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of advantages. Cloud Computing is transforming the very nature of how businesses use information technology. Infrastructure as a Service (IaaS), Platform as a Service and Software as a Service (SaaS) are the major service oriented cloud computing models. The benefits of cloud computing include reduced cost and capital expendictures, efficient computing, scalability, flexibility and so on.

Cloud storage is an important service of cloud computing which allows data owner to move data from their local computing systems to the cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief from the burden for storage management, universal data access with location independence and avoidance of capital expenditure on hardware, software and personnel maintenances, etc .The data owners are only need to pay for what they actually use. But, moving the data to large data centres which are remotely located, on which data owner does not have any control posses many new security challenges like data security, privacy, availability and data integrity due to its internet-based data storage and management .Also the cloud service provider is usually a commercial enterprise which cannot be totally trusted. Here data confidentiality is not the only security requirement. Flexible and fine grained access control also gets more importance in the case of cloud storage service. For example, a healthcare information system on a cloud is needed to restrict access of protected medical record to eligible doctors.

The traditional method to provide confidentiality to such sensitive data is to encrypt them before uploading to the cloud. If we use the traditional public key infrastructure, each user encrypts his file and stores it in the server, and the decryption key is disclosed only to the particular authorized user. From the view point of confidentiality, this scheme is secure, but this trivial solution requires efficient key management and distribution of keys which is proven to be difficult. These limitations and the need for data sharing lead to the introduction of new access control schemes based on attribute based encryption(ABE).In ABE[3] schemes, cipher texts are not encrypted to one particular user as in traditional public key cryptography. It enables us to handle unknown users also. Different types of ABE schemes provided fine grained access control to data stored in cloud. But, they suffered from certain limitations of scalability, flexibility and user revocation overhead. The limitations of ABE are covered by Hierarchical Attribute Set Based Encryption (HASBE) [1] which is an extension of Attribute Set Based Encryption (ASBE).HASBE achieves scalability due to its hierarchical structure and also inherits fine grained access control and flexibility in supporting compound attributes, of ASBE. Another highlighting feature of HASBE is its easy user revocation method.

However, the fact that data owners no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task. In addition to these access control needs, the owners want to check whether their data is corrupted or it is present in the cloud storage or not. Also, in order to improve the trustiness of cloud such a requirement is essential. But, HASBE does not include such an auditing facility. It is not a practical solution to simply download all the data for its

integrity verification. Since the user's resource capability is constrained and the size of outsourced data is large, it is desirable to give this task to a third party authority.So, the proposed system introducing a third party auditor, to the HASBE system, that checks the integrity of outsourced data on behalf of the data owners.

To securely introduce an effective TPA, the auditing [10-11] process should bring in no new vulnerabilities towards user data privacy and users should not undergo additional online burden. So the proposed system uses Homomorphic Linear Authenticator with random masking technique. The overall system is going to be applied to online personal health record service.

The rest of the paper is organized as follows. Section II provides an overview on related work. Then we present the proposed method with system model in Section III. In Section IV, we give the implementation details of proposed system. In Section V, we analyze the security of proposed method by comparing with Zhiguo et al.'s scheme. Lastly, we conclude the paper in Section VI.

II. RELATED WORKS

This section reviews the concept of different attribute based encryption schemes. All these schemes are proposed as access control mechanisms to cloud storage.

A. Attribute Based Encryption(ABE)

Attribute based encryption was first introduced for enforced access control through public key cryptography. It used user identities as attributes and these attributes play important role in encryption and decryption. The primary ABE [3] used a threshold policy for access control .In this scheme both the user secret key and the cipher text are associated with a set of attributes. A user is able to decrypt the cipher-text if and only if at least a threshold number of attributes overlap between the cipher-text and user secret key. ABE is implemented for one-to many encryption in which cipher-texts are not necessarily encrypted to one particular user, it may be for more than one number of users. But this threshold semantics are not very expressive to be used for designing more general access control system. This lack of expressiveness limits it's applicability to larger systems.

B. Key Policy ABE(KP-ABE)

The idea behind KP-ABE [4] is that cipher texts are labeled with a set of attributes and the private keys are associated with access structures on these attributes that controls which cipher text a user is able to decrypt. This access structure represents an access policy. The user can decrypt the file only if there is a match between access policies in the key and the attributes in the cipher text. The major disadvantage of this scheme is, since the access policy is built in to the user's private key, the data owner who encrypts the data can't choose who can decrypt the data. He has to trust the key issuer.

C. Ciphertext Policy ABE(CP-ABE)

CP-ABE [5] works in the reverse way of KP-ABE. Cipher texts are associated with access structures which represents access policies. Private Key is associated with a set of user attributes. For decrypting the file, there should be match between access policy in cipher text and attributes in the user's private key. CP-ABE is conceptually closer to traditional access control models such as Role-Based Access Control (RBAC) and hence it is more natural to apply instead of KP-ABE. Here the full control goes to the data owner. CP-ABE has limitations in specifying complex policies and managing user attributes with flexibility. Also it is not much scalable and user revocation is difficult.

D. Hierarchical ABE(HABE)

Hierarchical attribute-based encryption (HABE) is intended to provide high performance, scalability and to avoid bottleneck of attribute authority who issues the key. It is a combination of HIBE (Hierarchical Identity Based Encryption) and CP-ABE. HABE [6] scheme involves hierarchical generation of keys and also adapt the flexibility from CP- ABE.

E. Ciphertext Policy Attribute Set Based Encryption(CP-ASBE)

In CP-ABE scheme, the decryption keys only support user attributes that are organized logically as a single set, so users can only use all possible combinations of attributes in a single set issued in their keys to satisfy policies. To solve this problem, cipher text-policy attribute-set based encryption (CP-ASBE or ASBE for short) is introduced by Bobba, Waters et al [7].It is an extended form of CP-ABE which organizes user attributes into a recursive set structure. It supports compound attributes. Multiple numerical assignments for a given attribute can be supported by placing each assignment in a separate set.

F. Hierarchical Attribute Set Based Encryption(HASBE)

Still scalability is not achieved with ASBE. To achieve scalability along with flexibility and fine grained access control and efficient user revocation, HASBE is proposed. HASBE [1] extends the ASBE algorithm with a hierarchical structure. Different type of parties are arranged in a hierarchy
and keys are delegated through this hierarchical order. HASBE also achieves efficient user revocation (with out requiring complete re-encryption).

III. PROBLEM STATEMENT

Even though HASBE scheme achieves scalability, flexibility and fine grained access control, it fails to prove the data integrity in the cloud. The data owners are facing a serious risk of corrupting or missing their data because of lack of physical control over their outsourced data. Sometimes the cloud service provider may delete the data which are either not used by client from long-time and which occupies large space in the cloud without the knowledge or permission of data owner.

In general, there is no method called integrity scheme in HASBE to ensure that the data will be remained correctly in the cloud. Hence it is the major drawback of HASBE scheme. In order to overcome this security risk, privacy preserving public auditing concept could be proposed, which integrates data integrity proof with

HASBE scheme. Therefore, it is of critical importance to enable public audit ability for cloud storage so that users can check the integrity of outsourced data and be worry-free.

IV. PROPOSED SYSTEM

The proposed system extends HASBE with a privacy preserving public auditing which integrates data integrity verification ability to HASBE scheme. In this paper, for demonstrating HASBE's functionalities, HASBE scheme applies to Personal Health Record (PHR).

A Personal Health Record (PHR) is an electronic record of an individual's health information .Online PHR service [8-9] allows an individual to create, store, manage and share his personal health data in a centralized way. Since cloud computing provides infinite computing resources and elastic storage, PHR service providers shift the data and applications in order to lower their operational cost.

When storing his PHR data in the cloud, the PHR owner losses control over them and may afraid of the potential privacy exposure. So it is essential to provide security mechanisms based upon the PHR owner's privacy requirements in the cloud. One commonly adopted solution to protect privacy of information is encryption. Basically, PHR data is encrypted before uploading them to the cloud and stores the cipher texts in the cloud. But, this is not the only requirement. As the number of users increases, the system should handle every user without difficulty (scalable) and also it should be able to implement complex access control policies with greater flexibility and should provide fine-grained access control .Existing access control schemes could not achieve all these requirements.So,in order to achieve scalable, flexible and fine grained access control, Hierarchical Attribute Set Based Encryption (HASBE) is proposed.

A. System Model

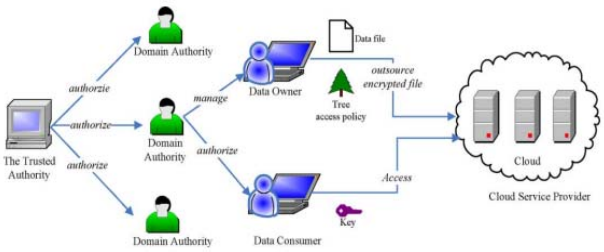

Fig. 1.System Model

As in the fig1.the cloud computing system [1] under consideration consists of five types of parties that form a hierarchy: a cloud service provider, data owners, data consumers, a number of domain authorities and a trusted authority.

In our system Ministry of Health represents trusted authority. There are two domain authorities called National Hospital Association and National Medical Association. Patients and doctors represent data owners and data consumers.

Cloud provides data storage service and is managed by the cloud service provider. Data owners representing patients (and doctors) encrypt their data files and upload them in the cloud for sharing with data consumers. To access the shared data files, data consumers representing doctors (and patients) 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 trusted authority is the root authority and responsible for managing top-level domain authorities.

In this system, data owners and data consumers are need not be always online. They come online only when necessary, while the cloud service provider, the trusted authority, and domain authorities are always online. The cloud is assumed to have abundant storage capacity and computation power. In addition, we assume that data consumers have only reading permission.

B. System Architecture

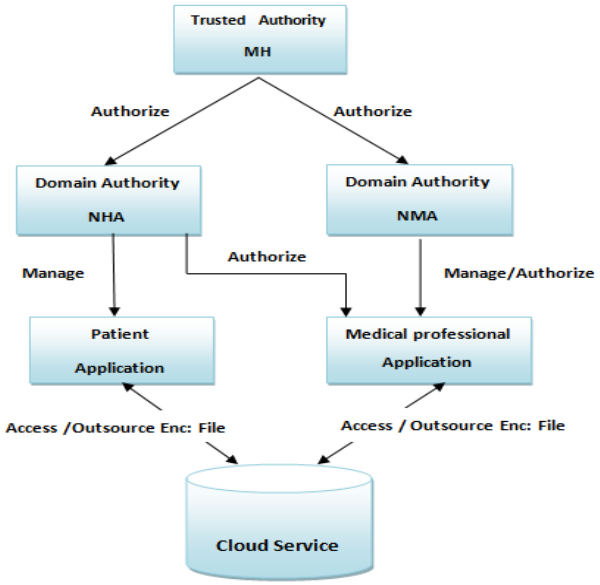

Fig.2.System Architecture

In fig .2, the system users are organized in a hierarchy. Ministry of Health act as the trusted authority that creates and authorizes the domain authorities National Hospital Association (NHA) and National Medical Association (NMA).

The National Medical Association manages and authorizes the Medical Professionals and National Hospital Association manages the patients and authorizes the Medical Professionals. Here both Medical Professionals and patients can outsource and access the encrypted files from the cloud.

In addition to this there exists a Third party Authority to perform auditing of cloud data storage.

C. Auditing

To effectively support public audit ability without retrieve the data blocks themselves, the Homomorphic Linear Authenticator [HLA] technique is used. To achieve

privacy-preserving public auditing, the homomorphic linear authenticator is uniquely integrated with random masking technique.

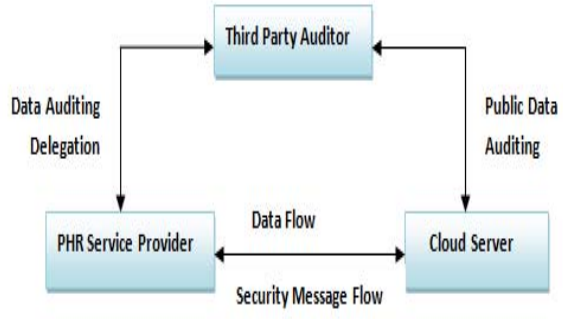

Fig.3.Auditing Architecture

In Fig.3, the cloud data storage service involves three different entities: the cloud user (PHR Service Provider), who has large amount of data files to be stored in the cloud; the cloud server (CS), which is managed by the cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources; the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. It is of critical importance for users to ensure that their data are being correctly stored and maintained. Here cloud users resort to TPA for ensuring the storage integrity of their outsourced data. The TPA interact with the cloud server in order to check
the integrity of files uploaded in the cloud and provides the information to the PHR Service Provider that the integrity is retained or not.

D. HASBE Scheme

HASBE [1] seamlessly extends the ASBE scheme to handle the hierarchical structure of system users in the fig.4.

Recall the system model in fig.1. The trusted authority is responsible for generating and distributing system parameters and root master keys as well as authorizing the top-level domain authorities. A domain authority is responsible for delegating keys to subordinate domain

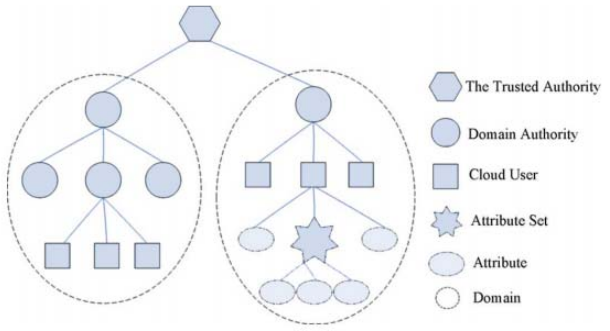

Fig.4.Hierarchical Structure of System Users

authorities at the next level or users in its domain. Each user in the system is assigned a key structure which specifies the attributes associated with the user's decryption key.

E. Key Structure

A recursive set based key structure [1] is used in this scheme, where each element of the set is either a set or an element corresponding to an attribute. The depth of the key structure is the level of recursions in the recursive set, similar to definition of depth for a tree. For a key structure with depth 2, members of the set at depth 1 can either be attribute elements or sets but members of a set at depth 2 may only be attribute elements. Consider the example shown in Fig 4, where,

{Dept: Cardiology, Specialty: Nuclear Cardiology
{Hospital: Hospital A, Level: 3},
{Hospital: Hospital B, Level: 6}}

is a key structure of depth 2. It represents the attributes of a person who is working in Hospital A with Level 3 and working in Hospital B with Level 6 in specialty Nuclear Cardiology of the Dept: Cardiology.

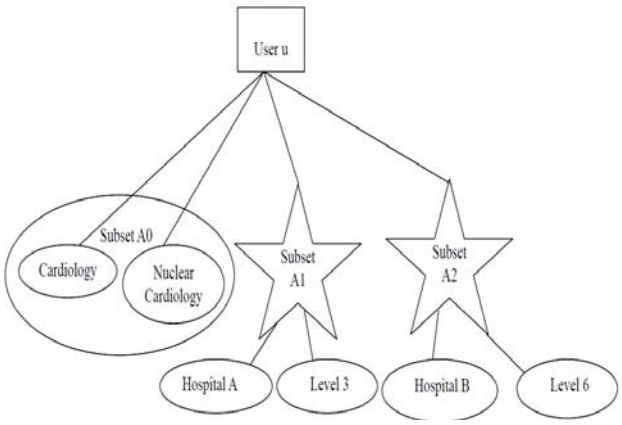

Fig.5.Key Structure

The key structure defines unique labels for sets in it. For key structures of depth 2, just an index of the sets at depth 2 is sufficient to uniquely identify the sets. Thus if there are m sets
at depth 2 then a unique index i where $1 \leq i \leq m$ is assigned to each set. The set at depth 1 is referred to as set 0. Using this
convention, a key structure of depth 2 can be represented as $A=\{A_0,A_1,\ldots,A_m\}$, where A_0 is the set at depth 1 while A_i is the ith set at depth 2, for $1 \leq i \leq m$. In the key structure in Fig .5, {Dept: Cardiology, Specialty: Nuclear Cardiology} corresponds to A_0, and {Hospital: Hospital A, Level: 3} and {Hospital: Hospital B, Level: 5} correspond to A_1 and A_2, respectively. The key structure of user and master key is combined to generate the secret key.

F. Access Structure

Access structure ensures that whether the user have the rights to access/download the file or not. This kind of access structure is given to the file by the data owner.

In the tree access structure [1], leaf nodes are attributes and non leaf nodes are threshold gates. Each non leaf node is defined by its children and a threshold value. The threshold values for "AND" should satisfy n of n attributes and those of "OR" should satisfy 1 of n attributes. An example of the access tree structure is shown in, where the threshold values for "AND" and "OR" are 2 and 1, respectively.

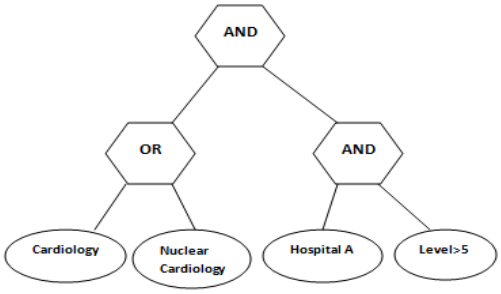

Fig.6. Access Structure

The above access structure demands that only a person who is
working in Hospital A under cardiology or nuclear cardiology with level greater than 5 can access the data files protected by the access policy. In CP-ABE schemes, a person who has private keys corresponding to attributes on the key structure shown in Fig.5 would be able to access the data files, which compromises the security of the access policy in Fig. 6. Such problems are effectively prevented using attribute-set-based encryption which forbids combining attributes across multiple sets.

G. Key Updation for Revocation

Data owner can add an attribute like expiration-time to user's key structure which indicates the time until which the key is considered to be valid. Once the time expires then the key will be considered as invalid and user will no longer have the file access rights. To perform this key expiration-time operation, access structure associated with data files must include check on expiration-time attribute as a numerical comparison. For e.g.: assuming a user 'U' has a key with expiration-time 'X' and a data file whose access policy is associated with expiration time 'Y', then user 'U' can decrypt the data file only when X>=Y. In this method, user's key can be updated without entire key regenerating and redistributing at the end of expiration time. On the other hand, the data owner can change the policy over data files by updating the value of expiration time attribute associated with the leaf node in the access tree and so it is only need to recompute the secret key component associated with expiration time attribute. Rest of the key remains unchanged and thus it reduces computational overhead in the revocation process.

V. IMPLEMENTATION

This section discusses some main operations [1] of the system: System Setup, Top-Level Domain Authority Grant, New Domain Authority/User Grant, New File Creation, User Revocation, File Access, and File Deletion.
System Setup: The trusted authority calls the SETUP algorithm to create system public parameters PK which will be made public to other parties and master key MK_0 which will be kept secret.
Top-Level Domain Authority Grant: A domain authority is associated with a unique ID and a recursive attribute set .When a new top-level domain authority, i.e.DA_i , wants to join the system, the trusted authority will first verify whether it is a valid domain authority. If so, the trusted authority generates the master key MK_i for DA .After getting the master key, DA_i can authorize the next level domain authorities or users in its domain.
New File Creation: To protect data stored on the cloud, a data owner first encrypts data files and then stores the encrypted data files on the cloud. Each file is encrypted with a symmetric data encryption key DEK, which is in turn encrypted with HASBE.
File Deletion: Encrypted data files can be deleted only at the request of the data owner. To delete an encrypted data file, the data owner sends the file's unique ID and its signature on this ID to the cloud. Only upon successful verification of the data owner and the request, the cloud deletes the data file.
File Access: When a user sends request for data files stored on the cloud, the cloud sends the corresponding cipher texts to the user. The user decrypts them by first calling DECRYPT (CT, SKu) to obtain DEK and then decrypt data files using DEK.
The above desirable operations are implemented by using six algorithms: Setup, CreateDA, CreateUser, KeyUpdate, Encrypt, Decrypt and Auditing involves four algorithms KeyGen, SigGen, GenProof, VerifyProof.
Setup (d): Here d is the depth of key structure. Algorithm takes a depth parameter d as input. It outputs a public key PK and master secret key MK_0. In our scheme, we describe for key structures of depth 2, and it can be extended to any depth d.
CreateDA (PK, MK_0, A): Take as input the public key PK, master secret key MK_0 and a key structure 'A'. It outputs a master key MK_i for top-level domain authority DA_i .
CreateUser (MK_i, DA_{i+1}, Ã): Take as input the master key MK_i of domain authority DA_i , the identity of new Domain Authority DA_{i+1} and a key structure Ã of DA_i .It outputs a master key MK_{i+1} for new Domain Authority.
CreateUser (MKi, U, Ã):Take as input the master key MK_i of domain authority DA_i , the identity of user u and a key structure Ã. It outputs a secret key SK_u for user u.
KeyUpdate (MK, old_value, new_value) : This is done by the domain authority. It takes as the input the master secret key, old attribute value, updates the secret key by updating old _value with the new_value.
Encrypt (PK,M,T):Take as input the public key PK, a message M, and an access tree T. It outputs a cipher text CT.
Decrypt (CT,SK_u): Take as input a cipher text CT and a secret key SK_u for user u. It outputs a message m. If the key structure A associated with the secret key satisfies the

access tree T, associated with the cipher text CT, then m is the original correct message M. Otherwise, m is null.

A public auditing scheme consists of four algorithms KeyGen, SigGen, GenProof, VerifyProof which are used in two different phases.

1. Setup Phase

KeyGen : is a key generation algorithm that is run by the user to setup the scheme. It generates public and secret parameters.

SigGen : is used by the user to generate verification metadata, which may consist of digital signatures. It also computes file tags. User sends verification metadata to the server and delete them from local storage.

2. Audit Phase

Up on getting data auditing delegation, TPA retrieves the file tag and verifies the signature. If it is false it quits. Otherwise TPA generates challenge message and is given to the cloud server.

GenProof :is run by the cloud server to generate a proof of data storage correctness in response to the challenge, while VerifyProof is run by the TPA to audit the proof.

VI. SECURITY ANALYSIS

In this section, we compare our scheme with security feature of the one proposed in [1].

- Scalability: compared with scheme in [1] our scheme also achieves scalability, by shifting the authority rights to sub ordinates and data owner which decreases the workload of root authority.
- Flexibility: compared with [1] scheme our scheme also achieves flexibility, by allowing the user attributes to organized in a recursive set structure.our scheme supports the compound attributes and multiple numerical assignments for a given attribute conveniently.
- Fine-grained access: compared with [1] scheme our scheme also achieves fine grained access control, by allowing data owner to define expressive and flexible access policy for data files.
- Efficient User Revocation: compared with [1] ,this scheme also achieves efficient user revocation. To deal with user revocation in cloud computing, we add an attribute to each user's key and employ multiple value assignments for this attribute. So we can update user's key by simply adding a new expiration value to the existing key. we just require a domain authority to maintain some state information of the user keys and avoid the need to generate and distribute new keys on a frequent basis, which makes this scheme more efficient than existing schemes.
- Expressiveness: compared with [1] scheme, our scheme provides expressiveness, where user keys are associated with attributes rather than access policy.
- Data Integrity: compared with [1] our scheme additionally provides the data integrity proof, by allowing the owners to ensure the integrity of outsourced data periodically with the help of a TPA, without introducing new security vulnerabilities.

VII. CONCLUSION

In this paper, we proposed the privacy preserving public auditing concept for HASBE scheme, to overcome the drawback of, absence of integrity assurance method in HASBE. Even though HASBE scheme achieves scalability, flexibility, and fine grained access control, it fails to provide data integrity in the cloud. Since, the data owner has no physical control over his outsourced data, such an auditing is necessary to prevent cloud service provider from hiding data loss or corruption information from the owner. Audit result from TPA would also be beneficial for the cloud service providers to improve their cloud based service platform, and users can give their data to the cloud and be worry free about the data integrity. The proposed system preserves all advantages of HASBE and also adds an additional quality of integrity proof to this system.

REFERENCES

[1] Zhiguo Wan, Jun'e Liu, and Robert H. Deng, *Senior Member, IEEE,"* HASBE: A Hierarchical Attribute set-Based Solution for Flexible and Scalable Access Control in Cloud Computing", ieee transactions on information forensics and security, vol. 7, no. 2, april 2012

[2] Kangchan Lee," Security Threats in Cloud Computing Environments", International Journal of Security and Its Applications ,Vol. 6, No. 4, October, 2012.

[3] Cheng-Chi Lee1, Pei-Shan Chung2, and Min-Shiang *Hwang ,"*A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments", *International Journal of Network Security, Vol.15, No.4, PP.231-240, July 2013*

[4] Vipul Goyal *Omkant Pandeyy Amit Sahaiz Brent Waters,"* Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data"

[5] John Bethencourt, Amit Sahai, Brent Waters **"**Ciphertext-Policy Attribute-Based Encryption ", in Proc. IEEE Symp. Security and Privacy, Oakland, CA, 2007.

[6] Guojun Wanga, Qin Liu a,b, Jie Wub, Minyi Guo, Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers, www.elsevier.com locate / /cose

[7] Rakesh Bobba, Himanshu Khurana and Manoj Prabhakaran," Attribute-Sets: A Practically Motivated Enhancement to Attribute-Based Encryption" University of Illinois at Urbana-Champaign, July 27, 2009

[8] Ming Li, Shucheng Yu, ,Yao Zheng, Kui Ren, and Wenjing Lou," Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption" in IEEE Transactions On Parallel And Distributed Systems ,2012

[9] Chunxia Leng1, Huiqun Yu, Jingming Wang, Jianhua Huang," Securing Personal Health Records in Clouds by Enforcing Sticky Policies" in TELKOMNIKA, Vol. 11, No. 4, April 2013, pp. 2200 ~ 2208 e-ISSN: 2087-278X.

[10] Cong Wang, Qian Wang, Kui Ren, Wenjing Lou (2010), "Privacy Preserving Public Auditing for Data Storage Security in Cloud Computing".

[11] Jachak K. B., Korde S. K., Ghorpade P. P. and Gagare G. J.,"Homomorphic Authentication with Random MaskingTechnique Ensuring Privacy & Security in CloudComputing", Bioinfo Security Informatics, vol. 2, no. 2,pp. 49-52, ISSN. 2249-9423, 12 April 2012