ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, January 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Enhancing Data Security and Data Integrity Using HASBE Scheme in Cloud Environment

Lakshmi.G[1], Nalini.P[2], Vinitha.P[3] Ms.T.Subha M.Tech, (Ph.D)

Final Year, Department of IT, Sri Sai Ram Engineering College, West Tambaram, Chennai -600044, India[1,2,3]

Associate Professor, Department of IT, Sri Sai Ram Engineering College, West Tambaram, Chennai -600044, India[4]

Abstract— Cloud computing is one of the emerging trends in software industry. This technology requires users to delegate their valuable data to cloud providers, thus increasing security and privacy concerns on outsourced data. Attribute-based encryption (ABE) methods have been proposed for access control of outsourced data in cloud. However, they are not efficient in complex access controls of user data. In this paper, we propose hierarchical attribute-set-based encryption (HASBE) by extending cipher text-policy attribute-set-based encryption (ASBE) with a hierarchical structure of users. The proposed scheme achieves scalability due to its hierarchical access structure, and also realizes flexibility and fine-grained access control. HASBE method achieves efficient user revocation than existing methods by assigning multiple values for access expiration time. We prove the security of HASBE based on security of the cipher text-policy attribute-based encryption (CP-ABE) and analyze its performance and computational complexity. In addition to it, we also develop dynamic data operation in the cloud which allows the data owner, who has outsourced the data in cloud, to dynamically perform data operations like modification, deletion and append. We implement our scheme and show that it is both efficient and flexible in dealing with access control for outsourced data in cloud environment.

Keywords— Access control, Data integrity, Data security, Hierarchical access.

I. INTRODUCTION

Cloud is the remote server in which the data is stored in huge amounts which cannot be maintained by the physical server in organizations. Cloud computing is the recent paradigm which is based on distributed, parallel computing. It has been emerged as one of the greatest technology as its benefits are server storage reduction, reduced capital expenditures, more operational efficiency, scalability, flexibility etc. Different service-oriented cloud computing models are on hand, comprehensive of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Many cloud service providers are available E.g., Amazon's EC2, Amazon's S3, and IBM's Blue Cloud are IaaS systems, whereas Google App Engine and Yahoo Pig are representative PaaS systems, and Google's Apps and Salesforce's Customer Relation Management (CRM) System belong to SaaS systems [3]. However the potential users face security problems in cloud computing paradigm. One of the prominent security concerns is data security and privacy in cloud computing due to its Internet-based data storage and management. Cloud service provider is a third party and cannot be totally trusted. Users have to give up their data to the cloud service provider for storage and business operations. Data represents an extremely important asset for any organization and serious consequences will occur if its confidential data is disclosed to the public. Thus, cloud users want to make sure that their data are kept confidential to other public users, together with the cloud provider and their competitors. This is the first data security requirement. Not only the data security and confidentiality, but also the flexibility and scalability in accessing the data are strongly desired in the service-oriented cloud computing model. For example, a health-care information system on a cloud is required to restrict access of protected medical records to eligible doctors and a customer relation management system running on a cloud may allow access of information to high level management executives only. Hence access control of sensitive data is required for security purposes. The schemes proposed to achieve flexible, fine grained access control are applicable only when the data owner and service provider are in same trusted domain. Hence a new access control scheme employing attributed-based encryption is proposed which adopts the key-policy attribute-based encryption (KP-ABE)[1] to enforce fine-grained access of data. However, this scheme falls diminutive of flexibility in attribute management and lacks scalability in dealing with multiple-levels of attribute authorities. In contrast to KP-ABE, cipher text-policy ABE (CP-ABE)[1] turns

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, January 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

out to be well suited for access control due to its expressiveness in describing access control policies.

In this paper, we propose a hierarchical attribute-set-based encryption (HASBE) scheme for access control in cloud computing and also dynamic data operation in cloud data which includes data deletion, modification and append. HASBE extends the cipher text-policy attribute-set-based encryption (CP-ASBE, or ASBE for short) scheme with a hierarchical structure of system or computer users, so as to realize the scalable, flexible and fine-grained access control of the cloud data.

II. Related Work

We examine the existing access control systems in this section. The traditional method to protect sensitive data outsourced to third parties is to store encrypted information on servers, while the decryption keys are given to authorized users only. However, there are several drawbacks in this solution. It requires an efficient key management mechanism to distribute decryption keys to approved users, which has been proven to be very difficult. And also this approach lacks scalability and flexibility; as the number of authorized users becomes large, the solution will not be efficient. In case a previously legitimate user needs to be revoked of group, relevant data has to be re-encrypted and new keys must be distributed to existing legitimate users again. And also data owners need to be online all the time so as to encrypt or re-encrypt data and distribute keys to authorize users. ABE turns out to be a good technique for realizing hierarchical access control solutions. This scheme enables a data owner to delegate most of the computational overhead to cloud server. In ABE scheme, the encryption of data is done based on attributes. The drawback of this scheme is that it lacks expressibility. In this scheme, cipher texts are not encrypted to one particular user as in traditional public key cryptography. Instead, both cipher texts and decryption keys are associated with a set of attributes or a policy with attributes. A user can able to decrypt a cipher- text only if there is a match between his decryption key and the cipher text. The ABE or attribute based encryption schemes are classified into key-policy attribute-based encryption (KP-ABE) and cipher text-policy attribute-based encryption (CP-ABE) [1], depending how attributes and policy are associated with cipher texts and users' decryption keys.

In a KP-ABE scheme, a cipher text is associated with a set of attributes and a user's decryption key is associated with a monotonic tree access structure. Only if the attributes related to the cipher text satisfy the tree access structure, the user can decrypt the cipher text. In a CP-ABE scheme, the roles of cipher texts and decryption keys are switched; the cipher text is encrypted with a tree access policy chosen by an encryptor or data owner, while the decryption key corresponding to the cipher text is created with respect to a set of attributes. If the set of attributes associated with a decryption key satisfies the tree access policy associated with a given cipher text, then the key can be used to decrypt the cipher text. The problem with the KP-ABE is that the encryptor is not able to decide who can decrypt the encrypted data except choosing descriptive attributes for the data and for some applications like sophisticated broadcast encryption. For such an application, a better choice is CP-ABE. Since users' decryption keys are associated with a set of attributes, CP-ABE or cipher text attribute based encryption is abstractly closer to traditional access control models such as Role-Based Access Control (RBAC). Thus, it is more relevant to apply CP-ABE, instead of KP-ABE, to impose access control of encrypted data. However, basic CP-ABE schemes are far from enough to support access control in modern enterprise solutions, which entail substantial flexibility and efficiency in specifying policies and managing user attributes. In a CP-ABE scheme, decryption keys only support user attributes that are organized logically as a single set, so users can only apply 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 used. ASBE is an extended form of CP-ABE which organizes user attributes into a recursive set structure. For example, the key structure for a student registered in the course is given as

Dept : CS	Role : Grad - Student
Course ID: 101	Role : TA
Course ID: 525	Role : Grad – Student

Here the student in course 101 has role TA and in course 525 he has been assigned Student role. We can see that same attribute can be defined with multiple values. This feature renders ASBE more versatile and flexible in supporting many practical scenarios. Hierarchical attribute-based encryption (HABE) was proposed to achieve fine-grained access control in cloud storage services by combining hierarchical identity-based encryption (HIBE) and Cipher text attribute based encryption. This scheme also supports fine grained access control and fully delegating computation to the cloud service providers. However, HABE uses disjunctive normal form policy and assumes all attributes in one conjunctive clause are administrated by the same domain master. Hence the same attribute may be administrated by multiple domain masters according to specific policies, which is difficult to implement in real time scenarios. And also this scheme cannot support

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, January 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

compound attributes efficiently as that of ASBE and does not support multiple value assignments. Hence ASBE is proved to be the efficient and can also enforce dynamic constraints on combining attributes to gratify a policy, which provides immense flexibility in access control. Using ASBE, we can solve the problem simply by assigning multiple values to the group of attributes in different sets. ASBE can enforce efficient cipher text policy encryption for situations where existing ABE schemes are inefficient. ASBE's capability of assigning multiple values to the same attribute enables it to solve the user revocation problem efficiently, by assigning different expiration times, which is difficult in CP-ABE. The ASBE uses four algorithms.

A. *Algorithms Used*

1) *Setup:* The setup algorithm takes input as the implicit security parameter. It provides as outputs, the public key PK and a secret master key MK.

2) *Encrypt (PK,M,A):* Encryption algorithm takes as input the public key PK, a message M, and an access structure A above the universe of attributes. The algorithm encrypts message M and produce a cipher text CT such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message

3) *Key Generation (MK, S):* The key generation algorithm takes as input the master key MK and a set of attributes S that illustrate the key. It provides the output as private key SK.

4) *Decrypt (PK, CT, SK):* The decryption algorithm takes as input the public key PK, a cipher text CT, which consists of an access policy A, and a private key SK, which is a private key for a set S of attributes. If the set S satisfies the access structure A then the algorithm will decrypt the cipher text and return a message M.

The missing part of ASBE is the delegation algorithm, which is used in our proposed scheme to construct the hierarchical structure. The four algorithms of ASBE, discussed above are taken into implementation and also extend ASBE by proposing a new delegation algorithm.

5) *Delegate (SK, ˜ S):* The delegate algorithm takes as input a secret key SK for some set of attributes S and a set S˜ ⊆ S. It output a secret key S˜K for the set of attributes ˜ S.

III. System Model

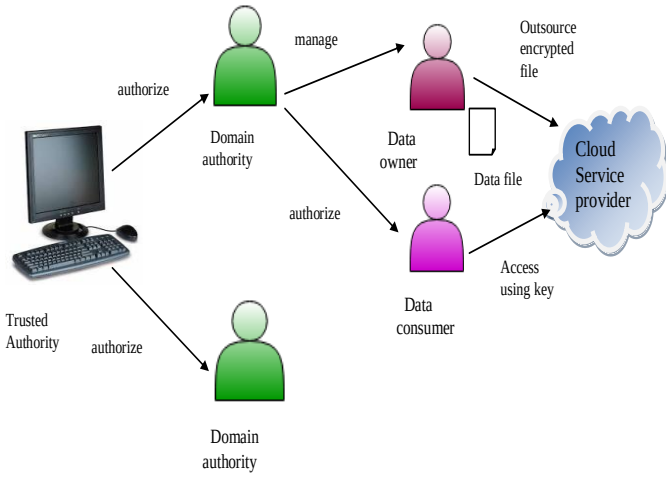

Fig. 1 System model

The System model has been shown. The cloud service provider manages a cloud to provide data storage service. Data owners before uploading encrypt their data files and store them in the cloud for sharing with data consumers or users. If the users need to access the shared data files from the group, they need to download encrypted data files from the cloud and then decrypt them. Each data owner/consumer is managed by a domain authority. In turn, they are managed by their parent domain or the trusted authority. The users, owners of files and each level of domain authorities, along with the trusted authority are structured in a hierarchical manner. The trusted authority is the root authority and responsible for managing top-level domain authorities. Each top-level domain authority is the top-level business, such as an enterprise, whereas each lower-level domain authority corresponds to a lower-level organization, such as an affiliated company in the enterprise. Data owners/consumers are people like employees in an organization. Each domain authority is in charge of managing the domain authorities at the next level or the data owners/consumers in its domain.

In our system, neither data owners nor data consumers need to be always online. They come only when needed, whereas the third party cloud service provider, the trusted authority, and each level of domain authorities are always online. The cloud contains abundant storage capacity and computation power. In addition, only the reading access is given to data users who download the encrypted files. In this model, each party or domain authority is associated with a public key and an undisclosed key, that is kept secretly by the party.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, January 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

The trusted authority is the root of trust and authorizes the top-level domain authorities. The sub- ordinate level authorities trust the next level authority by whom they are administered, but may try to get the private keys of users outside its domain.

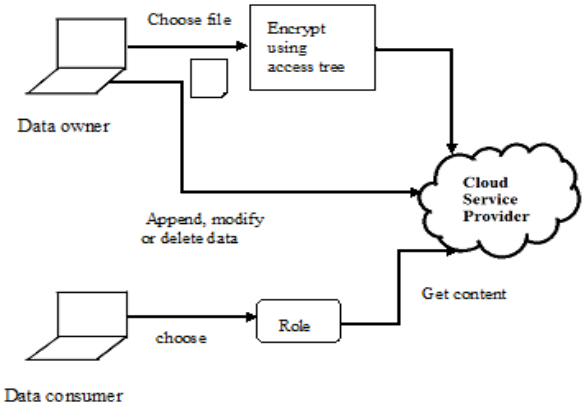

Fig. 2 Complete operations done by our system

In addition to it, we also add dynamic data operation [3] by the data owner which includes the operation like deletion, modification and append [4]. Then the data owner encrypts the file using tree access policy and delegates it to cloud provider. The data consumer downloads the encrypted file and using the access key, for his role can decrypt the data. Only the data that should be shown to that particular user is visible, when he decrypts the data.

IV. Proposed Work

Here we present our system which extends the ASBE scheme with hierarchical structure i.e. HASBE – Hierarchical Attribute Set Based Encryption. In our HASBE scheme [6], a data encryptor specifies an access structure for a cipher text which is referred to as the cipher text policy. Only when the attributes that are specified in the corresponding key structure satisfy those structures, the users with available decryption keys associated with the given attributes, can decrypt the cipher text.

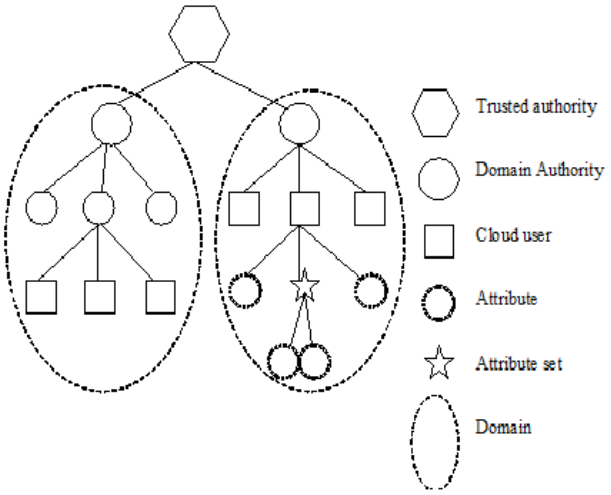

Fig. 3 Hierarchical Structure

We use a recursive set based key structure 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 number of levels of recursions in the recursive set. In our scheme, we use the same tree access structure. In the tree access structure, 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 hierarchical structure is represented as in the diagram. Our system model consists of a trusted authority that is responsible for the entire system and multiple level domain authorities, and also numerous users/consumers that access the files from the cloud. The trusted authority is accountable for generating and distributing system parameters and root master keys and also authorizes the top-level authorities. Those authorities are responsible for delegating keys to subordinate domain authorities at the next level or users in their domain. Hence each user in the system is available with their key structure which specifies the attributes associated with the user's decryption key.

A. Main Operations in HASBE Scheme

1) *Setup:* the trusted authority uses the setup algorithm to create public key PK which is public and master key MK which is secret key.
2) *Top level domain authority grant:* a domain authority is associated with unique ID and attribute set A. If a new top level domain authority wants to join, first it is checked to be valid. If so, then it creates master key for domain authority.
3) *New domain authority or user grant:* If a new user wants to join the system, he is first checked to be valid. If then, the unique master key is created for each user.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, January 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

4) *New file creation:* To protect data stored on the cloud, a data owner encrypts data files and then stores the encrypted data files in it. Each file is encrypted with a data encryption key DEK, that is symmetric, and in turn encrypted with HASBE.

Before providing the file to cloud, the data owner selects the ID for the file, random data encryption key DEK and define the tree access policy T. Then he encrypts the file using DEK with T and uploads to the cloud.

5) *Encrypt:* This algorithm is used to encrypt the data file which is same as that of ASBE.
6) *User Revocation:* The user revocation must be efficiently handled such that the revoked user should not be able to access the files anymore. It can be solved by re-encryption of files such that only the authorized users can view the files. It can be done by two steps i.e. the key update and data re-encryption [8].
7) *File access:* The files can be accessed by the authorized users, where first the cloud provider sends the cipher texts to the users according to their hierarchical structure using decrypt algorithm to obtain DEK. Then using DEK, the files can be decrypted [6].
8) *File modification:* whenever the data owner is online, he can dynamically update the data file in the cloud. With the unique ID of the file, he can update the files.
9) *File Deletion:* Encrypted data files can be deleted only at the request of the data owner. To delete a data file uploaded by the owner, the user sends the unique ID of file and its signature on this ID to the cloud provider. Only after the successful verification of the data owner and the request by the user, the cloud provider deletes the data file from the cloud and it will not be available further to the users.

It uses the four algorithms of ASBE, Setup of system and Key Generation for the data, Encrypt and Decrypt. And also, it uses delegation algorithm which is used to delegate the process to cloud provider [1].

V. Security Proof and Discussion

The security of our scheme is proved directly based on the security of CP-ABE [1]. In CP-ABE and HASBE the cipher texts are associated with access structures and the private keys are identified with attributes. The security model for the CP-ABE scheme is shown which will be taken for the HASBE scheme.

A. Main Security model of CP-ABE

1) *Setup:* The challenger runs the Setup algorithm and gives the public key PK to the adversary.

Phase 1. The adversary makes repeated private keys corresponding to sets of attributes S1.. Sq.

Challenge. The adversary submits two equal length messages M0 and M1. In addition the Adversary gives a challenge access structure A such that none of the sets S1.. Sq1 from Phase 1 satisfy the access construction. The challenger flips an arbitrary coin b, and encrypts Mb under A. The cipher text CT is given to the adversary.

Phase 2. Phase 1 is repeated with the restriction that none of sets of attributes Sq1+1..Sq satisfies the access structure corresponding to the challenge.

Guess. The adversary outputs a guess b′ of b.

Hence the security model requires that the adversary chooses to be challenged on an encryption to an access structure.

B. Merits of HASBE Scheme

Our scheme is compared with the existing systems and the merits are shown as:-

1) *Scalability:* HASBE scheme is the extension of ASBE with a hierarchical structure to effectively delegate the trusted authorities' private attribute key generation operation to lower-level domain authorities by which the workload is shifted to lower-level domain authorities who can provide attribute key generations for end users or consumers. Thus, this hierarchical structure achieves great scalability [7]. In the ABE scheme, the trusted authority performs all of the processes.
2) *Flexibility:* HASBE organizes user attributes into recursive set structure and allows users to impose dynamic constraints on combination of attributes to create the policy.
3) *Fine grained access control:* our scheme easily defines a flexible and fine grained access control where data owner enforces dynamic access policies [7].
4) *Efficient user revocation:* to deal with user revocation, we can update user's key by simply adding a new expiration value to the existing key. We just require an 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 is more efficient and it is also expressive [8].
5) *Dynamic data operation:* For data modification, the data owner and data customers need to be online all the time. And the data owner must verify for the modification or deletion of data file. In our scheme, we can dynamically perform

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, January 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

the data operation, even after encryption of file [5].

VI. IMPLEMENTATION

We are on the process of implementing this scheme with the real time college level data. Here, using this scheme the top trusted authority level is the management or higher authorities of college and lower level domain authorities are the lecturers, professors and the members of the institution and data consumers or the users are the students who are pursuing the education. It is being executed based on HASBE toolkit designed for CP-ABE[2], and experiments are being conducted and analysed with the help of laptop using Pentium IV and hard disk capacity of 40 GB.

We are implementing this system in order to reduce the time required to encrypt or decrypt the file and for key generation.

The time required to setup the system according to the depth of structure is given in Fig. 4, and the time required for the authority grant based on number of attributes and number of subsets is given in the Fig. 5.

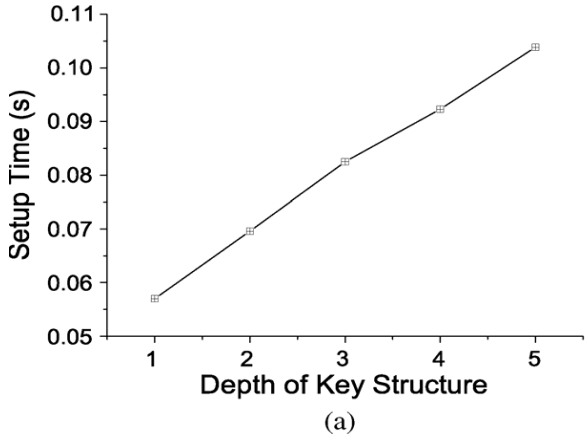

Fig. 4 Time required to setup according to depth

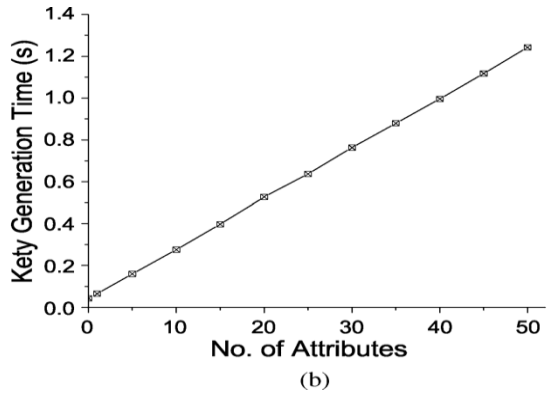

Fig. 5 Time required to setup according to number of attributes

VII. CONCLUSION

In this paper, we introduced the HASBE scheme to achieve scalable, flexible and fine grained access control in cloud computing and also the dynamic data operation by the data owner. The HASBE scheme flawlessly incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE. HASBE shows more benefits than existing schemes. It not only supports compound attributes due to flexible attribute set combinations, but also achieves efficient user revocation by assigning different access expiration times. It is due to the multiple value assignments of attributes. And the security of HASBE scheme is proved based on the security measures of CP-ABE. Also conductive experiments are done in the scheme and proved to be more efficient than existing schemes in access control of outsourced data in cloud platform.

ACKNOWLEDGMENT

We wish to acknowledge our Head of the Department and other professors for helping to develop and prepare this paper and implementation of the same.

REFERENCES

[1] John Bethencourt, Amit Sahai, Brent Waters. "Ciphertext-Policy Attribute-Based Encryption",2007.
[2] J. Bethencourt, A. Sahai, and B. Waters. "The cpabe toolkit." http://acsc.csl.sri.com/cpabe/.
[3] Kartik Sharma, Gitesh Kumar, Parul Saluja, Prashant Dalal, Kapil Narwal. "A Secure Method of Dynamic Data Operation in cloud computing, International Journal of advancements in Research & Technology", Volume 2, Issue 5, May-2013.
[4] Ayad F. Barsoum and M. Anwar Hasan. Department of Electrical and Computer Engineering, University of Waterloo, Ontario, Canada., "Enabling Data Dynamic and Indirect Mutual Trust for Cloud Computing Storage Systems.", 2011.
[5] Dinesh.C, P.G Scholar, Computer Science and Engineering, Mailam Engineering College, Mailam, Tamilnadu. "Data Integrity and Dynamic Storage Way in Cloud Computing.", October 2011.
[6] Zhiguo Wan, Jun'e Liu, and Robert H. Deng, *Senior Member, IEEE.* "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.
[7] Shucheng Yu, Cong Wang, Kui Ren, Wenjing Lou, " Achieving secure, scalable and fine-grained data access control in cloud computing", INFOCOM, 2010 Proceedings IEEE.
[8] Zhiqian Xu ; Martin, K.M. , "Dynamic User revocation and Key Refreshing for Attribute based encryption in cloud storage",Trustcom; 2012 IEEE 11th International Conference on.
[9] Google AppEngine. http://appengine.google.com