Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.66 – 70

RESEARCH ARTICLE

Secure Hierarchical Temporal Access Control in Cloud Computing

[1] Karthika.RN [2] Vijay Anand.P [3] M.Rajesh Khanna
PG Student Assistant Professor Assistant Professor
Veltech Multitech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai

***Abstract*- Cloud computing is a heavy appropriated registering standard. In spite of the fact that, distributed computing is a persuasive standard, access control is one of the principle security issues in distributed computing. Various methodologies for secure access control of outsourced information in distributed computing oblige figure content arrangement trait based encryption (CP-ABE), key-approach characteristic based encryption (KP-ABE), yet these overarching methodologies break out with unbend ability and they don't help current time. In this paper, the issue of secure access control on cloud is tended to by joining the procedures Hierarchical trait set-based encryption (HASBE) with Temporal Access Control Encryption (TACE). The joined system has the competence to acknowledge adaptable, adaptable, and fine-grained access control of outsourced information in distributed computing by developing figure content strategy quality set-based encryption (ASBE) with a progressive structure of clients and to authorize fleeting access control by utilizing a clock server.**

***Keywords*— Access control; Cloud Computing; Temporal; Re-Encryption; Data security**

I. INTRODUCTION

Cloud is a group of hardware and software interfaces collectively form a computing as a service. Computing is a process of utilizing computer technology to complete a task. Cloud computing is not a technology but it is a computing model. Cloud computing is a computing paradigm in which resources of the computing infrastructure are provided as services over the internet [13]. It is an internet based storage and services.
Cloud computing is "A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction" [4]. "Cloud computing is Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand, like electricity" [5].

Different service-oriented models in cloud computing comprises of the following:

- Platform as a Service (PaaS) e.g., Engine Yard, Cloud Foundry, Open Shift, Yahoo Pig, Jelastic.

- Software as a Service (SaaS) e.g., Sales force's Customer Relation Management (CRM) System, SalesForce.com and Microsoft Office Online.
- Infrastructure as a service (Iaas) e.g., Amazon Elastic Cloud Compute (EC2), Amazon Single Storage Service.

Data security is the chief requirement of Cloud computing. In the cloud, all the users need their data to be kept confidential to cloud provider and their potential competitors. Data confidentiality is not the only security requirement, but flexible, fine-grained access control and temporal access control is also required in the service-oriented cloud computing model. Access control is important when persons seek to secure important, confidential, sensitive information. Access control may be defined as exerting control over a person who can interact with a resource or it needs authority that does controlling.

Data access control is an effective way to assure the data security in the cloud. However, cloud storage service separates the roles of the data owner from the data service provider, and the data owner does not interact with the user directly for providing data access service, which makes the data access control a challenging issue in cloud storage systems. Because the cloud server cannot be fully trusted by data owners, traditional server-based access control methods are no longer applicable to cloud storage systems. To block the untrusted servers from accessing sensitive data, traditional methods usually encrypt the data and only users holding valid keys can access the data. Many access control scheme employing attributed-based encryption is proposed, which adopts the so-called key-policy attribute-based encryption (KP-ABE) to enforce fine-grained access control. However, KP-ABE experience defeat of flexibility in attribute management and lacks scalability in dealing with multiple-levels of attributes authorities. When compared to KP-ABE, cipher text-policy ABE (CP-ABE) turns out to be well suited for access control due to its expressiveness in describing access control policies.

In this paper, we combine the techniques Hierarchical attribute-set-based Encryption (HASBE) and Temporal Access Control Encryption (TACE) to support time range comparisons and re-encryption mechanism. Our scheme has better performance for integer comparison than existing scheme and the hierarchical structure of system users achieve scalable, flexible and fine-gained access control.

The remainder of the paper is organized as follows: Section 2 describes the related work. Section 3 describes the proposed work and Section 4 concludes the paper.

II. ALLIED WORK

In this section, we examine existing access control schemes based on ABE.

A. Key-Policy Attribute–Based Encryption (KP-ABE)

In Key-Policy Attribute-Based Encryption (KP-ABE), each cipher text is labeled by the encrypt or with a set of descriptive attributes. Each private key is associated with an access structure that specifies which type of cipher texts the key can decrypt. The scheme is named as Key-Policy Attribute-Based Encryption, since the access structure is specified in the private key, while the cipher texts are simply labeled with a set of descriptive attributes [4]. An important application of KP-ABE mainly deals with secure forensic analysis. One of the most important needs for electronic forensic analysis is an audit log containing a detailed account of all activity on the system or network to be protected. Such audit logs, however, raise significant security concerns such as a comprehensive audit log would become a prized target for enemy capture. KP-ABE system provides an attractive solution to the audit log problem. Audit log entries could be annotated with attributes such as, for instance, the name of the user, the date and time of the user action, and the type of data modified or accessed by the user action. Then, a forensic analyst charged with some investigation would be issued a secret key associated with a particular access structure which would correspond to the key allowing for a particular kind of encrypted search; such a key, would only open audit log records whose attributes satisfied certain condition [5]. The drawback in this scheme is the person who encrypt exerts no control over who has access to the data she encrypts, except by her choice of descriptive attributes for the data [5]. It has no flexibility in attribute management. It has no scalability in dealing with multiple levels of attribute authorities.

B. Cipher text-Policy Attribute Based Encryption (CP-ABE)

In CP-ABE schemes attribute policies are associated with data and attributes are associated with keys. Decryption is enabled only those keys which are associated with attributes satisfy the policy associated with the data. The person who encrypt must be able to smartly decide who should or should not have access to the data that she/he encrypts. Thus, our methods are conceptually closer to traditional access control methods such as Role- Based Access Control (RBAC). The user's private key will be associated with an arbitrary number of attributes expressed as strings. On the other hand, when a party encrypts a message, they specify an associated access structure over attributes. A user will only be able to decrypt a cipher text if that user's attributes pass through the cipher text's access structure [5]. CP-ABE [4] users can use all possible combinations of attributes issued in their keys to satisfy policies. This scheme can only support user attributes that are organized logically as a single set. First, this makes it both cumbersome and tedious to capture naturally occurring "compound attributes", i.e., attributes build intuitively from other attributes, and specifying policies using those attributes. The Best and only way to prevent users from combining such attributes in undesirable ways when using current CP-ABE schemes is by appending the attributes as strings. Since the approach has an undesirable consequence, this is a challenging task support policies that involve other combinations of singleton attributes used to build the compound attribute. CP-ABE schemes that support numerical attributes are limited to assigning only one value to any given numerical attribute within a key. But there are many real world systems where multiple numerical value assignments for a given attribute are Common [6]. The capability of assigning multiple values to the same attribute is not applicable. It doesn't solve the user revocation problem efficiently. The disadvantage of KP-ABE was overcome by cipher text-policy attribute-based encryption due to its expressiveness in describing access control policies.

C. *Cipher text-Policy Attribute Set Based Encryption (CP-ASBE)*

The new form of CP-ABE is Cipher text Policy Attribute Set Based Encryption (CP-ASBE) which organizes user attributes into a recursive set based structure and allows users to impose dynamic constraints on how those attributes may be combined to satisfy a policy that can selectively restrict decrypting users to use attributes from within a single set or allow them to combine attributes from multiple sets. Thus, by grouping user attributes into sets such that those belonging to a single set have no restrictions on how they can be combined, CP-ASBE can support compound attributes without sacrificing the flexibility to easily specify policies involving the underlying singleton attributes. Similarly, multiple numerical assignments for a given attribute can be supported by placing each assignment in a separate set. [6].

D. *Hierarchical Attribute-Based Encryption (HABE)*

Hierarchical attribute-based encryption (HABE) model is the combination of Hierarchical Identity-Based Encryption system (HIBE) and a Cipher text Policy-Attribute Based Encryption (CP- ABE) system. HASBE aim is to provide fine-grained access control, full delegation and to share confidential data on cloud servers more efficiently. The HABE scheme eliminates the on-line inquiry for
Authenticated attribute public keys [12] .This scheme
Also includes the drawbacks mentioned in Cipher text- Policy Attribute Based Encryption [6]. The advantages are high performance, fine-grained access control, Scalability, collusion resistant.

E. *Hierarchical Attribute-Set- Based Encryption (HASBE)*

Hierarchical attribute-set-based encryption (HASBE) by extending cipher text-policy attribute-set-based encryption (ASBE) with a hierarchical structure of users. It is scalability due to its hierarchical structure, but also inherits flexibility and fine-grained access control in supporting compound attributes of ASBE. HASBE employs multiple value assignments for access expiration time to deal with user revocation more efficiently than existing schemes. [13] The drawback in this scheme is that it applies cryptographic methods by disclosing data decryption keys only to authorize users. These solutions introduce a heavy computation overhead on the data owner for key distribution and data management when fine grained data access control is desired, and thus do not scale well. The existing HASBE don't support current time, which is essentially an important factor for enforcing temporal access control.

F. *Temporal Attribute Based Encryption (TABE)*
Cloud based data sharing service involving three different entities data owner, cloud server, and some data users (e.g., computers, mobile devices, or general equipment's). In order to implement temporal access control, a data owner has a collection of data to be outsourced to cloud servers in an encrypted form of Encrypt. At all times the data owner can assign a private key SKL to data users by using GenKey. To access these data, the data users download data from cloud servers and then decrypt them by utilizing Decrypt. AND/OR operations and basic fine-grained access control are not within the scope of this technique.

G. *Temporal Access Control Encryption (TACE)*
A cloud-based data storage service involving three different entities, data owner, cloud server, and many data users (e.g., computers, mobile devices, or general equipment's). In addition, in order to implement temporal access control, we require a clock server designed to always provide exactly the same current time by communicating with each other [13]. This scheme explores temporal attributes in specifying and enforcing the data owner's policy and the data user's privileges in cloud-based environments.

III. PROPOSED WORK

The proposed work combines HASBE with TACE. It incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE for realizing scalable, flexible, and fine-grained access control in cloud computing. HASBE not only supports compound attributes due to flexible attribute set combinations, but also achieves efficient user revocation because of multiple value assignments.

A. *Operations*
The main operations of proposed system are setup, top-Level Domain Authority Grant, New File Creation, File Access, and File Deletion.

1) *System Setup:*
To create and master key MK0 the Setup algorithm is called by the trusted authority. System public parameters PK will be made public to other parties. Master key MK0 will be kept secret. The key structure depth is denoted by d. The Setup algorithm outputs the public key and master key.
Setup (d=2) (PK, MK0)

2) *Top-Level Domain Authority Grant:*
A domain authority is associated with a unique ID and a recursive attribute set A={ A0, A1, …….., Am}, where Ai= {ai,1, ai,2,….., a i, n} with a i, j being the jth attribute in Ai and ni being the number of attributes in Ai. When an new top-level domain authority, i.e., DAi, wants to join the system, the trusted authority will first verify whether it is a valid domain authority. If he is valid, the trusted authority calls CreateDA to generate the master key for DAi. After getting the master key, DAi can authorize the next level domain authorities or users in its domain.

3) *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. A data file is processed by the data owner as follows: Pick a unique ID for this data file. Randomly choose a symmetric data encryption key DEK R K, where K is the key space, and encrypt the data file using DEK. Define a tree access structure T for the file and encrypt with T using algorithm Encrypt(PK,M,T) which returns cipher text CT. Finally, the encrypted data file is stored on the cloud.

4) *File Access:*
When a user sends request for data files stored on the cloud, the cloud sends the corresponding ciphertexts to the user. The user decrypts them by first calling Decrypt (CT,SKu) to obtain DEK and then decrypt data files using DEK.

5) *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. After successful verification of the data owner and the request, the cloud deletes the data file.

6) *Clock Server:*
In addition, in order to implement temporal access control, we require a clock server designed to always provide exactly the same current time by communicating with each other. First, the data owner makes use of a temporal access policy *P* to encrypt data before store it to clouds. Second, once receiving an access request from a user, the cloud service checks whether corresponding temporal constraints can be satisfied in *P* with respect to the current time *tc*, then employs a re-encryption method to convert the encrypted data into another cipher text *Ctc* that embed current time *tc* and sent it the user. Finally, the authorized user can use her/his private key *SK* with access privilege *L* to decrypt *Ctc* .

IV. CONCLUSION AND FUTUREWORK

This paper contains several encryption schemes for secure sharing of outsourced data in cloud server. It can be applied to achieve scalable, flexible, security, privacy, data confidentiality and fine-grained access control of outsourced data in cloud computing. The study concludes that the Hierarchical attribute-set-based encryption combined with temporal access control encryption with clock server is the advanced encryption scheme for outsourcing data in the cloud service provider which supports current time. On the other hand the techniques and strategies of encryption in cloud computing have to be improved with its distinct characteristics in mind. There is more scope for future research in the field of secure data sharing in the cloud.

REFERENCES

1.V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attibute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM Conf. Computer and Communications Security (ACM CCS)*, Alexandria, VA, 2006.
2.S. Yu, C. Wang, K. Ren, and W. Lou, "Achiving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM 2010*, 2010, pp. 534–542.
3. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in *Proc. IEEE Symp. Security and Privacy*, Oakland, CA, 2007.
4. R. Bobba, H. Khurana, and M. Prabhakaran, "Attribute-sets: A practically motivated enhancement to attribute-based encryption," in *Proc. ESORICS*, Saint Malo, France, 2009.
5. A. Sahai and B. Waters, "Fuzzy identity based encryption," in *Proc. Acvances in Cryptology—Eurocrypt*, 2005, vol. 3494, LNCS, pp. 457–473.
6. 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.

