International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 5, May2014)

Solution for Scalable and Flexible Access Control in Cloud Computing Using HASBE

AYESHA JABEEN
P.G. Student, Department Of Computer Science & Engineering
KBNCE, Gulbarga, Karnataka, India.
Email: ayeshajabeen14@gmail.com

SHAMEEN AKHTAR
Professor, Department Of Computer Science & Engineering
KBNCE, Gulbarga, Karnataka, India.

Abstract: **Our paper discusses on to increase the security of cloud based on Attribute Based Solution concepts and to provide additional security for cloud using Hierarchical Attribute-Set Based Encryption. We propose a hierarchical attribute-set-based encryption (HASBE) scheme for access control in cloud computing. HASBE extends the cipher text-policy attribute- set-based encryption (CP-ASBE, or ASBE for short) scheme with a hierarchical structure of system users, so as to achieve scalable, flexible and fine-grained access control**

***Keywords*- Attribute based encryption, cloud computing, data security.**

I. INTRODUCTION:

CLOUD computing is a new computing paradigm .Is built on virtualization, parallel and distributed com- putting, utility computing, and service-oriented architecture. In the last several years, cloud computing has emerged as one of the most influential paradigms in the IT industry, and has attracted extensive attention from both academia and industry. Cloud computing holds the promise of providing computing. Access of customer information to high-level executives of the company only. In these cases, access control of sensitive data is either required by legislation (e.g., HIPAA) or company regulations. In this paper, we propose a hierarchical attribute-set-based Encryption (HASBE) scheme for access control in cloud Computing. HASBE extends the cipher text-policy at- tribute-set-based encryption (CP-ASBE, or ASBE for short) scheme with a hierarchical structure of system users, so as to achieve scalable, flexible and fine- grained access control. The contribution of the paper is multifold. First, we show how HASBE extends the ASBE algorithm with a hierarchical structure to improve scalability and flexibility while at the same time inherits the feature of fine-grained access control of ASBE. Second, we demonstrate how to implement a full-fledged access control scheme for cloud computing based on HASBE .The scheme provides full support for hierarchical user grant, file creation, file deletion, and user revocation in cloud computing. Third, we formally prove the security of the proposed scheme based on the security of the CP-ABE scheme and analyze its performance in terms of computational overhead. Lastly, we implement HASBE and conduct comprehensive experiments for performance evaluation, and our experiments demonstrate that HASBE has satisfactory performance

II. PURPOSE:

This project describes how to secure the data's in hierarchical level using cloud computing

III. EXISTING SYSTEM:

- Our existing solution applies cryptographic methods by disclosing data decryption keys only to authorize users.
- These solutions inevitably 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.
- On the one hand, the outsourced computation workloads often contain sensitive information, such as the business financial records, proprietary research data, or personally identifiable health information etc.
- On the other hand, the operational details inside the cloud are not transparent enough to customers.

Software update/patches:

Could change security settings, assigning privileges too low, or even more alarmingly too high allowing access to your data by other parties.

Security concerns:

Experts claim that their clouds are 100% secure - but it will not be their head on the block when things go awry. It's often stated that cloud computing security is better than most

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 5, May2014)

enterprises. Also, how do you decide which data to handle in the cloud and which to keep to internal systems once decided keeping it secure could well be a full-time task.

Control:

Control of your data/system by third-party. Data - once in the cloud always in the cloud! Can you be sure that once you delete data from your cloud account will it not exist anymore... ...or will traces remain in the cloud.

IV. PROPOSED SYSTEM:

We propose a hierarchical attribute-set-based encryption (HASBE) scheme for access control in cloud computing.

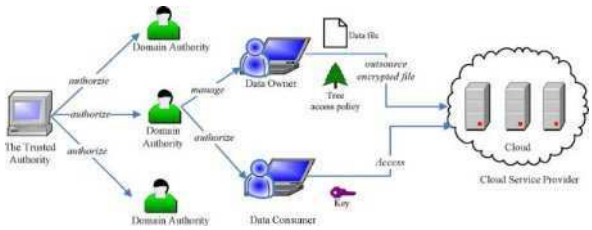

Fig.1. System Architecture

HASBE extends the cipher text-policy attribute- set-based encryption (CP-ASBE, or ASBE for short) scheme with a hierarchical structure of system users, so as to achieve scalable, flexible and fine-grained access control.

- Low initial capital investment
- Shorter start-up time for new services
- Lower maintenance and operation costs
- Higher utilization through virtualization
- Easier disaster recovery

V. METHODS OF SOLVING PROBLEM:

- Data Owner Module
- Data Consumer Module
- Cloud Server Module
- Attribute based key generation Module
- Domain Authority

Data owner:

The data owner uploads their data in the cloud server. For the security purpose the data owner encrypts the data file and then store in the cloud. The data owner can change the policy over data files by updating the expiration time The Data owner can have capable of manipulating the encrypted data file. And the data owner can set the access privilege to the encrypted data file.

Fig.2. Data Owner module

Data Consumer Module

In this module, the user can only access the data file with the encrypted key if the user has the privilege to access the file. For the user level, all the privileges are given by the Domain authority and the Data. Users are controlled by the Domain Authority only. Users may try to access data files either within or outside the scope of their access privileges, so malicious users may collude with each other to get sensitive files beyond their privileges.

Fig.3. Data Consumer module

Cloud Server Module

The cloud service provider manages a cloud to provide data storage service. Data owners encrypt their data files and store them in the cloud for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them.

Attribute based key generation Module

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 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. The trusted authority calls the algorithm to create system public parameters PK and master key MK. PK will

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 5, May2014)

be made public to other parties and MK will be kept secret. 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, SK) to obtain DEK and then decrypt data files using DEK.

VI. PERFORMANCE ANALYSIS AND IMPLEMENTATION

In this section, we first analyze theoretic computation complexity of the proposed scheme in each operation. Then we implement an HASBE toolkit based on the toolkit developed for CP-ABE [18], and conduct a series of experiments to evaluate performance of our proposed scheme.

A. Performance Analysis

We analyse the computation complexity for each system operation in our scheme as follows.

- **System Setup**. When the system is set up, the trusted authority selects a bilinear group and some random numbers. When **PK** and $\mathbf{MK_0}$ are generated, there will be several exponentiation operations. So the computation complexity of
  System Setup is O(1).
- **Top-Level Domain Authority Grant**.This operation is performed by the trusted authority. The master key of a domain authority is in the form of $\mathbf{MK}_i = (A,D,D_{i,j},D'_{i,j}$ for $a_{i,j} \epsilon A$,Ei for $A_i \epsilon$
  A), where is the key structure associated with a new domain authority, A_i is the set of. Let N be the number of attributes in A, and M be the number of sets in. Then the computation of MK_i consists of two exponentiations for each attribute in , and one exponentiations for every set in . The computation complexity of Top-Level Domain Authority Grant operation is O (2N+M).
- **New User/Domain Authority Grant**. In this operation, a new user or new domain authority is associated with an attribute set, which is the set of that of the upper level domain authority. The main computation overhead of this operation is rerandomizing the key. The computation complexity is O(2N+M) , where N is the number of attributes in the set of the new user or domain authority, and M is the number of sets in A.
- **New File Creation**. In this operation, the data owner needs to encrypt a data file using the symmetric key DEK and then encrypt DEK using HASBE. The complexity of encrypting the data file with DEK depends on the size of the data file and the underlying symmetric key encryption algorithm. Encrypting DEK with a tree access structure T consists of two exponentiations per leaf node in T and one exponentiation per translating node in T. So the computation complexity of New File Creation is O (2|Y|+|X|), where denotes the leaf nodes of and denotes the translating nodes of.
- **User Revocation**. In this operation, a domain authority just maintains some state information of users' keys and assigns new value for expiration time to a user's key when updating it. When re- encrypting data files, the data owner just needs two exponentiations for ciphertext components associated with the attribute. So the computation complexity of this operation is O(1) .
- **File Access**. In this operation, we discuss the decrypting operation of encrypted data files. A user first obtains with the algorithm and then decrypts data files using. We will discuss the computation complexity of the algorithm. The cost of decrypting a ciphertext varies depending on the key used for decryption. Even for a given key, the way to satisfy the associated access tree may be various. The algorithm consists of two pairing operations for every leaf node used to satisfy the tree, one pairing for each translating node on the path from the leaf node used to the root and one exponentiation for each node on the path from the leaf node to the root. So the computation complexity varies depending on the access tree and key structure. It should be noted that the decryption is performed at the data consumers hence, its computation complexity has little impact on the scalability of the overall system.
- **File Deletion**. This operation is executed at the request of a data owner. If the cloud can verify the requestor is the owner of the file, the cloud deletes the data file. So the computation complexity is O (1).

B. Implementation

We have implemented a multilevel HASBE toolkit based on the cpabe toolkit (http://acsc.csl.sri.com/cpabe/) developed for CP-ABE [18] which uses the Pairing-Based Cryptography library (http://crypto.stanford.edu/pbc/). Then comprehensive experiments are conducted on a laptop with dual core
2.10-GHz CPU and 2-GB RAM, running Ubuntu 10.04. We make an analysis on the experimental data and give the statistical data. Similar to the cpape toolkit, our toolkit also provides a number of command line tools as follows:

- **hasbe-setup**: Generates a public key **PK** and a master key $\mathbf{MK_0}$.
- **hasbe-keygen**: Given **PK** and $\mathbf{MK_0}$, generates a private key for a key structure. The key structure with depth 1 or 2 is supported.
- **hasbe-keydel**: Given **PK** and $\mathbf{MK}_i$ of DA , delegates some parts of DA 's private keys to a new

user or DA in its domain. The delegated key is equivalent to generating private keys by the root authority.

- **hasbe-keyup**: Given **PK**, the private key, the new attribute and the subset, generates a new private key which contains the new attribute.
- **hasbe-enc**: Given **PK**, encrypts a file under an access tree policy specified in a policy language.
- **hasbe-dec**: Given a private key, decrypts a file.
- **hasbe-rec**: Given **PK** , a private key and an encrypted file, re-encrypt the file. Note that the private key should be able to decrypt the encrypted file.

VII. CONCLUSION

In this paper, we introduced the HASBE scheme for realizing scalable, flexible, and fine-grained access control in cloud computing. The HASBE scheme seamlessly incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE. HASBE not only supports compound attributes due to flexible attribute set combinations, but also achieves efficient user revocation because of multiple value assignments of attributes. We formally proved the security of HASBE based on the security of CP-ABE by Bettencourt et al.. Finally, we implemented the proposed scheme, and conducted comprehensive performance analysis and evaluation, which showed its efficiency and advantages over existing schemes.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES:

[1]. R. Buyya, C. ShinYeo, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Comput. Syst., vol. 25,pp. 599–616, 2009.

[2]. Amazon Elastic Compute Cloud (Amazon EC2) [Online]. Available: http://aws.amazon.com/ec2/ Amazon Web Services (AWS) [Online]. Available: https://s3.amazonaws.com/

[3]. R. Martin, "IBM brings cloud computing to earth with massive new data centers," InformationWeek Aug. 2008 [Online]. Available:http://www.informationweek.com/news/hardware/data_centers/209901523 Google App Engine [Online]. Available: http://code.google.com/appengine/

[4]. K. Barlow and J. Lane, "Like technology from an advanced alien culture: Google apps for education at ASU," in Proc. ACM SIGUCCS User Services Conf., Orlando, FL, 2007.

[5]. B. Barbara, "Salesforce.com: Raising the level of networking," Inf. Today, vol. 27, pp. 45–45, 2010.

[6]. J. Bell, Hosting EnterpriseData in the Cloud—Part 9: InvestmentValue Zetta, Tech. Rep., 2010.

[7]. A. Ross, "Technical perspective: A chilly sense of security," Commun. ACM, vol. 52, pp. 90–90, 2009.

[8]. D. E. Bell and L. J. LaPadula, Secure Computer Systems: Unified Exposition and Multics Interpretation The MITRE Corporation, Tech. Rep., 1976.

[9]. K. J. Biba, Integrity Considerations for Secure Computer Sytems The MITRE Corporation, Tech. Rep., 1977.

[10]. H. Harney, A. Colgrove, and P. D. McDaniel, "Principles of policy in secure groups," in Proc. NDSS, San Diego, CA, 2001.

[11]. P. D. McDaniel and A. Prakash, "Methods and limitations of security policy reconciliation," in Proc. IEEE Symp. Security and Privacy, Berkeley, CA, 2002.

[12]. T. Yu and M. Winslett, "A unified scheme for resource protection in automated trust negotiation," in Proc. IEEE Symp. Security and Privacy, Berkeley, CA, 2003.

[13]. J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in Proc. ACM Conf. Computer and Communications Security (CCS),

[14]. Alexandria, VA, 2005. [16] 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.