ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

Cloud Security using EHASBE

Vinisha .C[1], Anju .J[2]

PG Scholar, Computer Science, LBS College of Engineering, Kasaragod, Kerala [1]

Assistant Professor, Computer science, LBS College of Engineering, Kasaragod, Kerala [2]

Abstract: Cloud computing has emerged as one of the most influential paradigm in the IT industry in recent years. Since this new computing technology requires users to entrust their valuable data to cloud providers, there have been increasing security and privacy concerns on outsourced data .Several schemes employing attribute based encryption(ABE) have been proposed for access control of out sourced data in cloud computing; however most of them suffer from inflexibility in implementing complex access control policies. In order to realize scalable flexible and fine grained access control of outsourced data in cloud computing,proposed the use of 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 not only achieves scalability due to its hierarchical structure, but also inherits flexibility and fine grained access control in supporting compound attributes of ASBE. In addition HASBE employs multiple value assignments for access expiration time to deal with user revocation more efficiently than existing schemes. The proposed scheme shows that it is efficient and flexible in dealing with access control for outsourced data in cloud computing with comprehensive experiments. One major problem of existing HASBE schemes is bulky, linearly increasing cipher text. In the CP-ABE schemes used, the size of a cipher text proliferates linearly with respect to the number of included attributes. In this paper, propose a novel PP-CP-ABE construction, named Privacy Preserving Constant-size Cipher text Policy Attribute Based Encryption (PP-CP-ABE), which enforces hidden access policies with wildcards and incurs constant-size conjunctive headers, regardless of the number of attributes. Based on this PP-CP-ABE, provide a new construction named as Privacy Preserving Attribute Based Broadcast Encryption (PP-AB-BE). Compared with existing CP-ABE constructions, PP-CP-ABE significantly reduces the ciphertext size from linear to constant and supports expressive access policies. Thus, PP-CP-ABE can be used in many communication constrained environments.

Keywords: Attribute-based encryption (ABE), privacy-preserving, ciphertext-policy, constant cipher text length, broadcast encryption.

I. INTRODUCTION

Although the great benefits brought by cloud computing paradigm are exciting for IT companies, academic researchers, and potential cloud users, security problems in cloud computing become serious obstacles which, without being appropriately addressed, will prevent cloud computing's extensive applications and usage in the future. One of the prominent security concerns is data security and privacy in cloud computing due to its Internet- based data storage and management. In cloud computing, users have to give up their data to the cloud service provider for storage and business operations, while the cloud service provider is usually a commercial enterprise which cannot be totally trusted. Data represents an extremely important asset for any organization, and enterprise users will face serious consequences if its confidential data is disclosed to their business competitors or the public. Thus, cloud users in the first place want to make sure that their data are kept confidential to outsiders, including the cloud provider and their potential competitors. This is the first data security requirement.

Cipher text Policy Attribute-Based Encryption (CP-ABE) has been a very active research area in recent years which is mainly used for providing data security in cloud. In the construction of CP-ABE, each attribute is a descriptive string and each entity may be tagged with multiple attributes. Many entities may share common attributes, which allows message encryptors to specify a secure data access policy over the shared attributes to reach a group of receivers. A decryptor's attributes need to satisfy the access policy in order to recover the message. These unique features make CP-ABE solutions appealing in many systems, where expressive data access control is required for a large number of users.One major problem of existing CP-ABE schemes is bulky, linearly increasing ciphertext. In this paper, propose a novel PP-CP-ABE construction, named Privacy Preserving Constant-size Ciphertext Policy Attribute Based Encryption (PP-CP-ABE), which enforces hidden access policies with wildcards and incurs constant-size conjunctive headers, regardless of the number of attributes.

DOI 10.17148/IJARCCE

II. RELATED WORK

A. Attribute-Based Encryption

Attribute Based Encryption (ABE) was first proposed as a fuzzy version of IBE. ABE schemes are classified into key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE), depending how attributes and policy are associated with ciphertexts and users' decryption keys. In a KP-ABE scheme a ciphertext 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 associated with the ciphertext satisfy the tree access structure, can the user decrypt the ciphertext. In a CP-ABE scheme the roles of ciphertexts and decryption keys are switched; the ciphertext is encrypted with a tree access policy chosen by an encryptor, while the corresponding decryption key is created with respect to a set of attributes . In a CP-ABE scheme, 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, Bobba introduced ciphertext-policy attribute-set-based encryption (CP-ASBE or ASBE for short). ASBE is an extended form of CP-ABE which organizes user attributes into a recursive set structure. ASBE can enforce dynamic constraints on combining attributes to satisfy a policy, which provides great flexibility in access control. Later Herranz proposed a more general construction of CP-ABE with constant ciphertext independently. Their proposed scheme achieves constant ciphertext with any monotonic threshold data access policy . To protect the privacy of the access policy, KSWscheme ,NYO scheme , RC scheme and YRL1 scheme were proposed, where the encryptor-specified access policy is hidden.

B. Access control solution for cloud computing

The traditional method to protect sensitive data outsourced to third parties is to store encrypted data on servers, while the decryption. ABE turns out to be a good technique for realizing scalable,flexible, and fine-grained access control solutions. Keys are disclosed to authorized users only. Yu proposed an access control mechanism based on KP-ABE for cloud computing, together with a re-encryption technique for efficient user revocation. The encrypted data file is stored with the corresponding attributes and the encrypted data encryption key (DEK). If the associated attributes of a file stored in the cloud satisfy the access structure of a user's key, then the user is able to decrypt the encrypted (DEK), which is used in turn to decrypt the file. The first problem with Yu et al.'s scheme is that the encryptor is not able to decide who can decrypt the encrypted data except choosing descriptive attributes for the data, and has no choice but to trust the key issuer. Wang proposed hierarchical attribute-based encryption (HABE) to achieve fine-grained access control in cloud storage services by combining hierarchical identity-based encryption (HIBE) and CP-ABE. This scheme also supports fine-grained access control and fully delegating computation to the cloud providers. One major problem of existing CP-ABE schemes is bulky, linearly increasing ciphertext which can be solve with the help of propose scheme that is EHASBE

III.SYSTEM MODEL

Cloud computing system under consideration consists of five types of parties: a cloud service provider, data owners, data consumers, a number of domain authorities, and a trusted authority 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.

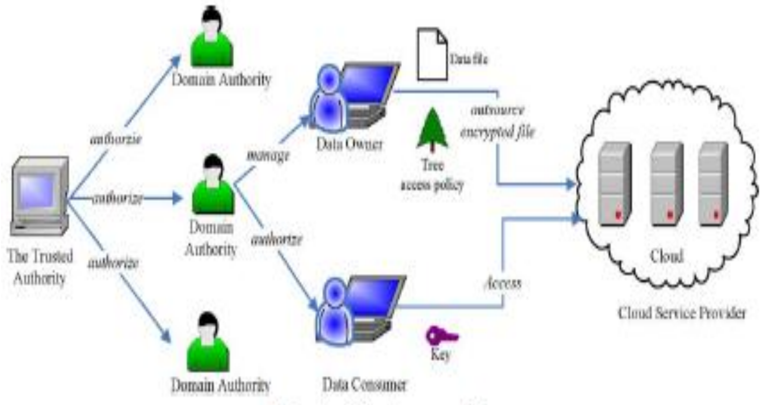


Fig. 1. System model.

To access the shared data files, data consumers 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. Data owners, data consumers, domain authorities, and the trusted authority are organized in a hierarchical manner as shown below. The trusted authority is the root authority and responsible for managing top-level domain authorities.

IV. EHASBE MODEL

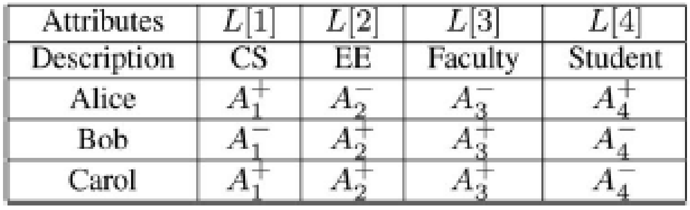
In this section, describe how to use attributes to form a data access policy, Then we present the bilinear map, which is the building block of ABE schemes.

A. Attributes, Policy and Anonymity

Let $U=\{A_i\}_{I\ \varepsilon\ [1,k],}$ be the Universe of attributes in the system.Each A_i has three values:$\{A_i^+,A_i^-,A_i^{\ *}\}$. When a user u joins the system, u is tagged with an attribute list defined as follows:

Definition 1: A user's attribute list is defined as ,$L=\{L[i]_{i\varepsilon[1,k]}\}$
where$L[i]$ Ɛ $\{A_i^{\ +},A_i^{\ -}\}$ and k is the number of attributes in the universe.
$A_i^{\ +}$ denotes the user has A_i; $A_i^{\ -}$denotes the user does not have A_i or A_i is not a proper attribute of this user. For example, suppose $U=\{A_1=CS,A_2=EE,A_3=faculty,A_4=student\}$. Alice is a student in CS department; Bob is a faculty in EE department; Carol is a faculty holding a joint position in EE and CS department. Their attribute lists are illustrated in Table I.

TABLE 1
Attribute Examples



Attributes	$L[1]$	$L[2]$	$L[3]$	$L[4]$
Description	CS	EE	Faculty	Student
Alice	A_1^+	A_2^-	A_3^-	A_4^+
Bob	A_1^-	A_2^+	A_3^+	A_4^-
Carol	A_1^+	A_2^+	A_3^+	A_4^-

As the actual data access policy is hidden in the ciphertext header, effective measures are required to avoid ambiguity. In other words, when a decryptor receives a ciphertext header without knowing the access policy, he/she should NOT try a large number of access policies when performing decryption. To this end, we adopt a AND-gate policy construction so that each decryptor only needs to try once on each ciphertext header.
The hidden AND-gate access policy is defined as follows:

Definition 2: Let $W=\{W[i]_{i\varepsilon[1,k]}\}$be an AND-gate access policy, where W[i] Ɛ{Ai+,Ai-,Ai*} . We use the notation $L \models W$ to denote that the attribute list L[i] of a user satisfies W , as:

$$L \models W \Leftrightarrow W \subset L \bigcup \{A_i^*\}_{i\in[1,k]}.$$

Ai^+ or Ai^- requires the exact same attribute in the user's attribute list. As for Ai*, it denotes a wildcard value, which means the policy does not care about the value of attribute. Effectively, each user with either $Ai^{\ +}$ or Ai^- fulfills Ai* automatically.

V. EHASBE CONSTRUCTION

It consist of four fundamental algorithms

A. Setup(1^λ,k)algorithms:

The Setup algorithm takes input of the security parameter 1^λ and the number of attributes in the system k.It returns public key PK and master key MK . The public key is used for encryption while the master key is used for private key generation.

B. KeyGen(PK,MK,L)algorithm

The KeyGen algorithm takes the public key PK, the master key MK and the user's attribute list Las input. It outputs the private key SK of the user.

C. Encrypt(PK,W,M)

The Encrypt algorithm takes the public key PK, the specified access policy W and the message M as input.The lgorithm outputs ciphertext CT, such that only a user with attribute list satisfying the access policy can decrypt the message. The ciphertext also associates the anonymized access policy ~W.

D. Decrypt(PK,SK,CT)

The Decrypt algorithm decrypts the ciphertext CT when the user's attribute list satisfies the access policy. It takes the public key PK , the private key SK of the user and the ciphertext CT , which only includes the anonymized access policy ~W as input

VI. CONCLUSION

By doing this project found out a better way to protect the data of the users in an efficient way. The main advantage project EHASBE brought up is its simplicity of user interface and the complexity of the security that is provided. The domain authority or the administrator cannot alter or readout the confidential data that is encrypted by the user. And since a secret key is provided the user can ensure that only a trusted system can download the data that is uploaded. Also all types of data can be stored and encrypted. Since the encrypted data are stored in cloud, its security value increases. So i conclude hereby that this project

REFERENCES

[1] RZhiguo Wan, Jun'e Liu, and Robert H. Deng "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing

[2] R. Buyya, C. ShinYeo, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing

[3] Amazon Elastic Compute Cloud (Amazon EC2) [Online]. Available:http://aws.amazon.com/ec2/

[4] R. Martin, "IBM brings cloud computing to earth with massive new Data centres," InformationWeek Aug. 2008 [Online]. Available: http:// www.informationweek.com/news/hardware/data_centers/209901523

[5] Google App Engine [Online]. Available: http://code.google.com/appengine/

[6] 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.

DOI 10.17148/IJARCCE