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. 2, Issue. 4, April 2013, pg.396 – 401

RESEARCH ARTICLE

An Enhanced HASBE for Cloud Computing Environment

D. Hephzi Rachel[1], S. Prathiba[2]

[1]Department of Computer Science and Engineering, Bharat University, Chennai, India

[2]Department of Computer Science and Engineering, Bharat University, Chennai, India

Abstract— In this era, even sensitive data is stored and shared on the internet using trusted third parties and service providers. Cloud computing is one of the emerging technologies that is being used widely on these days. It makes use of the computing resources such as hardware and software that is delivered over the internet and provides remote services with user's data, software and computation. We propose hierarchical attribute-set-based encryption (HASBE) by extending ciphertext-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 There are many encryption schemes that provide security and access control in clouds. This paper presents an encryption scheme called HASBE that provide security, scalable and flexible fine grained access control and its enhancement, a business model for separating the encryption and decryption service.

Key Terms: - Access control; Security; Cloud computing

I. INTRODUCTION

Today for many organizations they need to store their enormous amount of data. Network storage providers are giving the resources for these organizations on demand. Among these, cloud computing is most cost effective and flexible but it has some security issues. Cloud computing provide accuracy so more data can be centralized into the clouds. Users of this technology are relived from the data storage and maintenance as they entrust their valuable data in to the clouds. The users itself has to secured against the service providers as they should be aware about hacking and leakage of information.

One of the most important security concerns in clouds is the data security and privacy due to internet based data storage and management. For an organization the extremely important asset is the data. If the data is disclosed the enterprise users will face serious issues from their business competitors and the public. All the internet users want to make sure that their data should be confidential to the outsiders and even to the providers. Data confidentiality is always the first security requirement of every internet users. Along with data confidentiality, scalable and flexible access control is also desired by the cloud users.

Traditionally, the sensitive data is encrypted and stored on the servers. The decryption keys are disclosed only to the authorized users. This method has some drawbacks. Efficient key mechanism to distribute the decryption keys is needed for this traditional method. This method does not support scalability and flexibility, because when the number of authorized users increase it is not efficient. So to overcome these issues many schemes are introduced.

This paper focuses on the hierarchical attribute set based encryption schemes and its enhancement that is a business model for separating the encryption and decryption for increasing the efficiency in the access control. Section II presents the literature survey of different encryption schemes and a comparison table and section III concludes with discussions.

II. RELATED WORK

In cloud computing, there are different existing schemes that provide security, data confidentiality and access control. Users need to share sensitive objects with others based on the recipients' ability to satisfy a policy in distributed systems. One of the encryption schemes is Attribute-Based Encryption (ABE) which is a new paradigm where such policies are specified and cryptographically enforced in the encryption algorithm itself. The existing ABE schemes are of two types. They are Key-Policy ABE (KP-ABE) scheme and Ciphertext-Policy ABE (CP-ABE) scheme. In KP-ABE scheme, attribute policies are associated with keys and data is associated with attributes. Only the keys associated with the policy that is satisfied by the attributes associating the data can decrypt the data. In CP-ABE schemes, attribute policies are associated with data and attributes are associated with keys and only those keys that the associated attributes satisfy the policy associated with the data are able to decrypt the data.

A. Key Policy Attribute Based Encryption

Key Policy Attribute Based Encryption scheme is a public key cryptography primitive that is for one-to-many communications. In this, data are associated with attributes for each of which a public key is defined. The one who encrypts the data, i.e., the encryptor associates the set of attributes to the data or message by encrypting it with a public key. Users are assigned with an access structure which is defined as an access tree over the data attributes. The nodes that are interior of the access tree are threshold gates and leaf nodes of the tree are associated with attributes. The secret key of the user is defined to reflect the access structure. So the user is able to decrypt the message that is a ciphertext if and only if the data attributes satisfy the access structure.

In this scheme, a set of attributes is associated with ciphertext and the user's decryption key is associated with a monotonic tree access structure. When the attributes associated with the ciphertext satisfy the tree access structure, then the user can decrypt the ciphertext.

In cloud computing, an access control mechanism based on KP-ABE together with a re-encryption technique is used for efficient user revocation. This scheme enables a data owner to reduce most of the computational overhead to cloud servers. The use of this encryption scheme KP-ABE provides fine-grained access control. Each file or message is encrypted with a symmetric data encryption key (DEK), which is again encrypted by a public key corresponding to a set of attributes in KP-ABE, which is generated corresponding to an access structure. The data file that is encrypted is stored with the corresponding attributes and the encrypted DEK. Only if the corresponding attributes of a file or message 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 to decrypt the file or message.

The first problem with this 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. Furthermore, KP-ABE is not naturally suitable to certain applications. An example of such applications is a type of sophisticated broadcast encryption, where users are described by various attributes and the one whose attributes match a policy associated with a ciphertext can decrypt the ciphertext. KP-ABE scheme supports user secret key accountability.

B. Cipher Text Policy Attribute Based Encryption

In several distributed systems a user should only be able to access data if a user possesses a certain set of credentials or attributes. The only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. For realizing complex access control on encrypted data Ciphertext-Policy Attribute-Based Encryption can be used. By using this technique encrypted data can be kept confidential even if the storage server is untrusted; moreover, our methods are secure against collusion attacks. Previous Attribute-Based Encryption systems used attributes to describe the encrypted data and built policies into user's keys; while in our system attributes are used to describe a user's credentials, and a party encrypting data determines a policy for who can decrypt. Thus, our methods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC).

In ciphertext-policy attribute-based encryption (CP-ABE), depends how attributes and policy are associated with cipher texts and users' decryption keys. In a CP-ABE scheme, a ciphertext is associated with a monotonic tree access structure and a user's decryption key is associated with set of attributes. In this scheme, the roles of cipher texts and decryption keys are switched as that in KP-ABE) 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. As long as the set of attributes associated with a decryption key satisfies the tree access policy associated with a given ciphertext, the key can be used to decrypt the ciphertext. Since users' decryption keys are associated with a set of attributes, CP-ABE is conceptually closer to traditional access control models such as Role-Based Access Control (RBAC). Thus, it is more natural to apply CP-ABE, instead of KP-ABE, to enforce access control of encrypted data.

However, basic CP-ABE schemes are far from enough to support access control in modern enterprise environments, which require considerable 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 use all possible combinations of attributes in a single set issued in their keys to satisfy policies.

C. Cipher Text Policy Attribute Set Based Encryption

Ciphertext Policy Attribute Set Based Encryption (CP-ASBE)- a new form of CP-ABE - which, unlike existing CP-ABE schemes that represent user attributes as a monolithic set in keys, 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.

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, ciphertext-policy attribute-set-based encryption (CP-ASBE or ASBE for short) is introduced. ASBE is an extended form of CP-ABE which organizes user attributes into a recursive set structure.

Specifically CP-ASBE allows, 1) user attributes to be organized into a recursive family of sets and 2) policies 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. While restricting users to use attributes from a single set during decryption can be thought of as a regular CP-ABE scheme, the challenge in constructing a CP-ASBE scheme is in selectively allowing users to combine attributes from multiple sets within a given key while still preventing collusion, i.e., preventing users from combining attributes from multiple keys.

III. SYSTEM MODEL

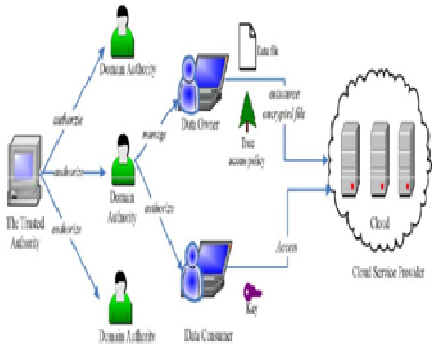

Fig 1: System Model

As depicted in Fig. 1, the 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. 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 in Fig. 1.

The trusted authority is the root authority and responsible for managing top-level domain authorities. Each top-level domain authority corresponds to a top-level organization, such as a federated enterprise, while each lower-level domain authority corresponds to a lower-level organization, such as an affiliated company in a federated enterprise. Data owners/consumers may correspond to employees in an organization. Each domain authority is responsible for 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 will 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 can access data files for reading only.

A.HASBE Scheme

The proposed HASBE seamlessly extends the ASBE scheme to handle the hierarchical structure of system users in the figure below:

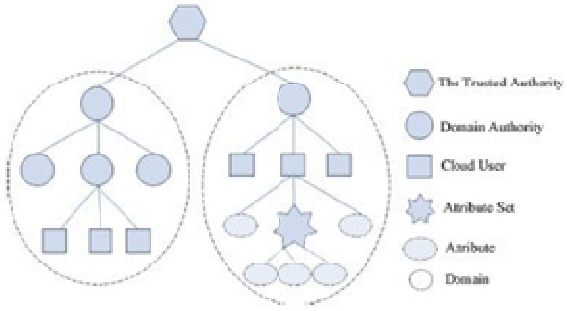

Fig.2: Hierarchical structure of system users

Recall that our system model consists of a trusted authority, multiple domain authorities, and numerous users corresponding to data owners and data consumers. 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 main operations of HASBE: 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 and master key MKo will be made public to other parties and MKo 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 , wants to join the system, the trusted authority will first verify whether it is a valid domain authority. If so, the trusted authority calls to generate the master key MKo for DA . After getting the master key, DA 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 (CK,SKu) to obtain and then decrypt data files using DEK.

B.HASBE Features

The proposed scheme HASBE on security features in implementing access control for cloud computing.

Scalability: We extend ASBE with a hierarchical structure to effectively delegate the trusted authority's private attribute key generation operation to lower-level domain authorities. By doing so, the workload of the trusted root authority is shifted to lower-level domain authorities, which can provide attribute key generations for end users. Thus, this hierarchical structure achieves great scalability. Only has one authority to deal with key generation, which is not scalable for large-scale cloud computing applications.

Flexibility: HASBE organizes user attributes into a recursive set structure and allows users to impose dynamic constraints on how those attributes may be combined to satisfy a policy. So HASBE can support compound attributes and multiple numerical assignments for a given attribute conveniently.

Fine-grained access control: Based on HASBE, our scheme can easily achieve fine-grained access control. A data owner can define and enforce expressive and flexible access policy for data files as the scheme.

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 our scheme more efficient than existing schemes.

Expressiveness: In HASBE, a user's key is associated with a set of attributes, so HASBE is conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC). Thus, it is more natural to apply HASBE, instead of KP-ABE, to enforce access control.

IV. ENHANCEMENT

Under the business model proposed in this study, the data storage cloud system provider is authorized to store the user's encrypted data, but does not have access to the Decryption Key. Thus, the storage system can only retrieve encrypted user data, but is unable to decrypt it. The cloud computing system responsible for encrypting user data has authority over all encryption keys required for data encryption but, given that the encryption provider does not store the user's data, internal mismanagement of the decryption keys still poses no risk of unauthorized disclosure of the user's data.

This study proposes a Business Model for Cloud Computing Based on a Separate Encryption and Decryption Service. In this business model, Encryption/Decryption as a Service and Storage as a Service (SaaS) are not provided by a single operator. In addition, the SaaS provider may not store unencrypted user data and, once the provider of Encryption/Decryption as a Service has finished encrypting the user data and handed it off to an application (e.g. a CRM system), the encryption/decryption system must delete all encrypted and decrypted user data.

The concept of dividing authority is often applied in business management. For example, responsibility for a company's finances is divided between the accountant and cashier. In business operations, the accountant is responsible for keeping accounts, while the cashier is responsible for making payments. By keeping these two functions separate, the company can prevent the accountant from falsifying accounts and embezzling corporate funds. Official documents frequently need to be stamped with two seals (i.e., the corporate seal and the legal

representative's seal), thus preventing a staff member from abusing his position to issue fake documents, and these seals are normally entrusted to two different people. These examples of the division of authority are designed to avoid a concentration of power which could raise operational risks.

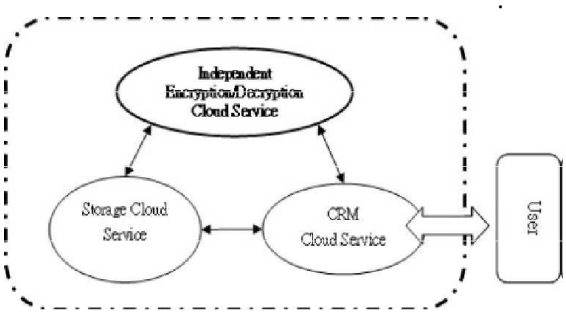

Fig 3: Business model concept integrating separate cloud services for data encryption/decryption, CRM and storage

To illustrate the concept of our proposed business model, Fig. 3 presents an example in which the user uses separate cloud services for CRM, storage and encryption/decryption. According to the user's needs, CRM Cloud Services could be swapped for other function-specific application services (e.g., ERP Cloud Services, Account Software Cloud Services, Investment Portfolio Selection and Financial Operations Cloud Services).

V. CONCLUSION

This paper proposes the hierarchical encryption scheme and its enhancement which is concentrated in efficient access control. The HASBE scheme seamlessly incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE. Finally it is concluded that, HASBE scheme concluded the realization of scalable, flexible, and fine-grained access control in cloud computing.

REFERENCES

[1] Zhiguo Wan, Jun'e Liu, and Robert H. Deng, "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

[2] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-policy attribute based encryption," IEEE Symp. Security and Privacy, Oakland, CA, 2007.

[3] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute Based Encryption for Fine-Grained Access Conrol of Encrypted Data", ACM conference on Computer and Communications Security (ACM CCS), 2006.

[4] M. Pirretti, P. Traynor, P. McDaniel and B. Waters, "Secure Atrribute-Based Systems", ACM conference on Computer and Communications Security (ACM CCS), 2006.

[5] A. Kapadia, P. Tsang and S. Smith, "Attribute-based publishing with hidden credentials and hidden policies", NDSS, 2007.

[6] G.Wang, Q. Liu, and J.Wu, "Hierachical attibute-based encryption for fine-grained access control in cloud storage services," ACM Conf. Computer and Communications Security (ACM CCS), Chicago, IL, 2010

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

[8] Dan Boneh, Xavier Boyen, Eu-Jin Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext", Advances in Cryptology—EUROCRYPT 2005, volume 3493.

[9] Nuttapong Attrapadung, Benoit Libert, and Elie de Panafieu, "Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts", 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011.

[10] Neena Antony, A. Alfred Raja Melvin, "A Survey on Encryption Schemes in the Clouds for Access Control" International Journal of Computer Science and Management Research, 2012

[11] Jang Hwang and Hung-Kai Chuang, Yi-Chang Hsu and Chien-Hsing Wu, "A Business Model for Cloud Computing Based on a Separate Encryption and Decryption Service".

