International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
NATIONAL CONFERENCE on Developments, Advances & Trends in Engineering Sciences
(NCDATES- 09th & 10th January 2015)

RESEARCH ARTICLE OPEN ACCESS

Flexible and Scalable Access Control set based encryption in Cloud Computing (FSC)

[1]Mr.M.Omkar Sharma, [2]Mr.V.Biksham,
Asst.Prof, CMREngineering College, Hyderabad, India
Assoc.Prof, CMR Engineering College, Hyderabad, India

Abstract
Cloud computing is computing in which large groups of remote servers are networked to allow centralized data storage and online access to computer services or resources. Clouds can be classified as public, private or hybrid. Several schemes employing on cloud for data access. An Atribute-based encryption had been proposed for access control of outsourced 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. In this WE proposed a Flexible and scalable Access control set based encryption(FSC) 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. WE formally prove the security of FSC based on security of the cipher text-policy attribute-based encryption (CP-ABE) scheme .
Index Terms: Fine grained access, Scalability, Ciphertext policy.

INTRODUCTION

Cloud computing is computing in which large groups of remote servers are networked to allow centralized data storage and online access to computer services or resources. Clouds can be classified as public, private or hybrid.

Cloud computing is the result of evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. The cloud aims to cut costs, and helps the users focus on their core business instead of being impeded by IT obstacles.

Private cloud

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally. Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and requires the organization to reevaluate decisions about existing resources. When done right, it can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally capital intensive. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures. They have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from less hands-on management essentially "[lacking] the economic model that makes cloud computing such an intriguing concept".

Public cloud

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free or offered on a pay-per-usage model. Technically there may be little or no difference between public and private cloud architecture, however, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure at their data center and access is generally via the Internet. AWS and Microsoft also offer direct connect services called "AWS Direct Connect" and "Azure Express Route" respectively, such connections require customers to purchase or lease a private connection to a peering point offered by the cloud provider.

Hybrid cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources. Gartner, Inc. defines a hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers. A hybrid cloud service crosses isolation and provider boundaries so that it can't be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service.

Varied use cases for hybrid cloud composition exist. For example, an organization may store sensitive client data in house on a private cloud application, but interconnect that application to a business intelligence application provided on a public cloud as a software service. This example of hybrid cloud extends the capabilities of the enterprise to deliver a specific business service through the addition of externally available public cloud services.

Another example of hybrid cloud is one where IT organizations use public cloud computing resources to meet temporary capacity needs that cannot be met by the private cloud. This capability enables hybrid clouds to employ cloud bursting for scaling across clouds. Cloud bursting is an application deployment model in which an application runs in a private cloud or data center and "bursts" to a public cloud when the demand for computing capacity increases. A primary advantage of cloud bursting and a hybrid cloud model is that an organization only pays for extra compute resources when they are needed. Cloud bursting enables data centers to create an in-house IT infrastructure that supports average workloads, and use cloud resources from public or private clouds, during spikes in processing demands.

Access Control Solutions for Cloud Computing

The Trivial solution describes to protect sensitive data outsourced to third parties is to store encrypted data on servers, while the decryption keys are disclosed to authorize users only. There will be existence of drawbacks in this trivial solution. Such a solution requires an efficient key management mechanism to distribute decryption keys to authorized users, which has been proven to be very difficult. Next, this approach lacks scalability and flexibility; as the strength of authorized users increases, the solution will not be efficient to manage. In case a previously legitimate user needs to be revoked, related data has to be re-encrypted and new keys must be distributed to existing legitimate users again. Last but not least, 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 scalable, flexible, and fine-grained access control solutions. Proposed an access control mechanism based on KP-ABE for cloud computing, together with a re-encryption technique for efficient user revocation. This scheme enables a data owner to delegate most of the computational overhead to cloud servers.

The use of KP-ABE provides fine-grained access control gracefully. Each file is encrypted with a symmetric data encryption key (DEK), which is in turn encrypted by a public key corresponding to a set of attributes in KP-ABE, which is generated according to an access structure. The encrypted data file is stored with the corresponding attributes and the encrypted 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 FCS scheme is that the encrypted 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 cipher text can decrypt the cipher text. For such an application, a better choice is CP-ABE.

Proposed FCS scheme is a 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. However, FCS uses disjunctive normal form policy and assumes all attributes in one conjunctive clause are administrated by the same domain master. Thus the same attribute may be administrated by multiple domain masters according to specific policies, which is difficult to implement in practice. Furthermore, compared with ASBE, this scheme cannot support compound attributes efficiently and does not support multiple value assignments

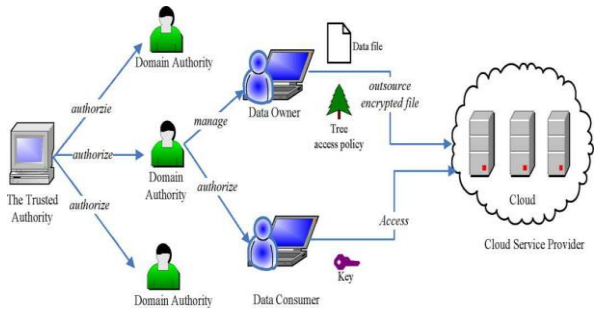

Fig1. System Model

A. 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.

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.

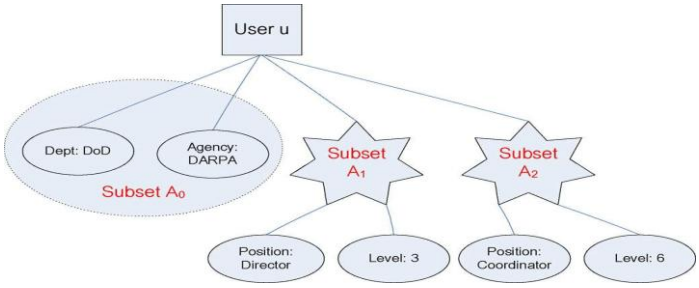

Fig 2. Key structure

Security Model

WE assume that the cloud server in not a trusted in higher level. In the sense that it may interact with data owners/data consumers to harvest file contents stored in the cloud for its own benefit. In the hierarchical structure of the system users given in Fig. 1, each party is associated with a public key and a private key, with the latter being kept secretly by the party. The trusted authority acts as the root of trust and authorizes the top-level domain authorities. A domain authority is trusted by its subordinate domain authorities or users that it administrates, but may try to get the private keys of users outside its domain. Users may try to access data files either within or outside the scope of their access privileges, so data owners/data consumers users may collude with each other to get sensitive files beyond their privileges. In addition, WE assume that communication channels between all parties are secured using standard security protocols, such as SSL.

Development of Hierarchical structure

The FCS scheme represents a hierarchical structure authorized accessing of a file. It describes the hierarchical user grant, data file creation, file access, user revocation, and file deletion.

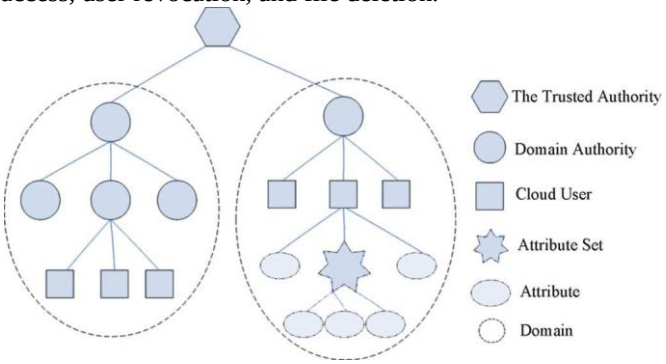

Fig 3. Hierarchical structure

The above figure represents the Hierarchical structure of System users. The Hierarchical structure follows the proposed Hierarchical attribute based scheme.

Proposed Scheme

The proposed FSC scheme seamlessly extends the ASBE scheme to handle the hierarchical structure of system users in Fig.3. 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. FCS describes the main operations as follows: System Setup, Top-Level Domain Authority Grant, New Domain Authority/User Grant, New File Creation, User Revocation, File Access and File Deletion.

Improvement Analysis:

System Setup.

The system is set up with the trusted authority selects a bilinear group and some random numbers. When they 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 computation complexity of Top-Level Domain Authority Grant operation is O (2N+M).

New User/Domain Authority Grant. In this process, 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 where M be the number of sets in a key structure associated with the new domain authority.

File Creation: In this, the data owner needs to encrypt a data file using the symmetric key DEK and then encrypt using FSC. The complexity of encrypting 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 consists of two exponentiations per leaf node in and one exponentiation per translating node in. So the computation complexity of New File Creation is O(2Y+X). Where Y denotes the leaf nodes of key structure and where X denotes the translating nodes of key structure.

User Revocation: In this, the domain authority maintain some state information of user's 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 cipher text components associated with the expiration time attribute. So the computation complexity of this operation is O (1).

File Access: In this, the decrypting operation of encrypted data files. A user first obtains with DEK the Decrypt algorithm and then decrypt data files using. WE will discuss the computation complexity of the algorithm. The cost of decrypting a cipher text 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 which denotes the number of attributes in the key structure, is the attribute set of the data file, is the set of leaf nodes of the access tree or policy tree, and is the set of translating nodes of the policy tree.

B. Implementation

WE published this idea by using a multilevel FCS toolkit based on the toolkit developed for CP-ABE.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 toolkit, our toolkit also provides a number of command line tools as follows:

FCS-setup: Generates a public key and a master key

FCS-keygen: Given and, generates a private key for a key structure. The key structure with depth 1 or 2 is supported.

FCS-keydel: Given and 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.

FCS-keyup: Given, the private key, the new attribute and the subset, generates a new private key which contains the new attribute.

FCS-enc: Given, encrypts a file under an access tree policy specified in a policy language.

FCS-dec: Given a private key, decrypts a file.

FCS-rec: Given, a private key and an encrypted file, re-encrypt the file. Note that the private key should be able to decrypt the encrypted file.

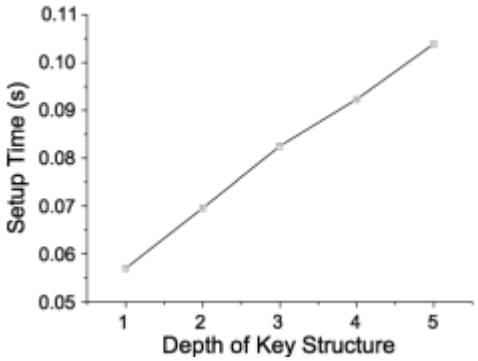

Fig. 4.(a) Key Structure

The above fig 4 shows the time required to setup the system for a different depth of key structure. Our scheme can be extended to support any depth of key structure. The cost of this operation increases linearly with the key structure depth, and the setup can
be completed in constant time for a given depth. Except for this
experiment, all other operations are tested with the key structure
depth of 2.

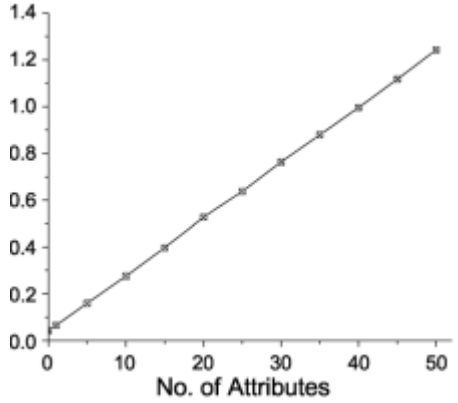

Fig 4(b).No of attributes

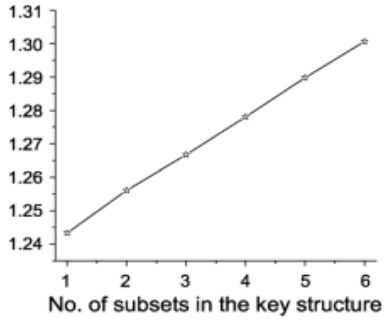

Fig 4(c). No of subsets in the key structure

Top-Level Domain Authority Grant is performed with the command line tool. The cost is determined by the number of subsets and attributes in the key structure. When there is only one subset in the key structure, the cost grows linearly with the number of attributes as Fig. 4(b) shows. While the number of attributes in the key structure is fixed to be 50, the cost also increases linearly with the number of subsets as shown in Fig.4(c)

CONCLUSION

In this paper, we introduced the FCS scheme for realizing scalable, flexible, and fine-grained access control in cloud computing. The FCS scheme describes incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE. Finally, we implemented the proposed scheme, and conducted experiments on performance analysis and evaluation, which got efficiency and better over existing schemes.

REFERENCES:

[1]. R. Bobba, H. Khurana, and M. Prabhakaran, "Attribute-sets: A practically motivated enhancement to attribute-based encryption," inProc ESORICS, Saint Malo, France, 2009.

[2]. R. Martin, "IBM brings cloud computing to earth with massive new data centers,"InformationWeekAug. 2008 [Online].Available:http://www.informationweek.com/news/hardware/data_centers/209901523.

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

[4]. J. Bell, Hosting Enterprise Data in the Cloud—Part 9: Investment Value Zetta, Tech. Rep., 2010.

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

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

[7]. J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," inProc. ACM Conf. Computer and Communications Security (CCS), Alexandria, VA, 2005.

[8]. V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attibute-based encryption for fine-grained access control of encrypted data," inProc. ACM Conf. Computer and Communications Security (ACM CCS), Alexandria, VA, 2006.

Mr.M.OmkarSharma, working as a Assistant Professor in CMR Engineering College, Hyderabad, Telangana, india. I am having 4 years of teaching experience and 2 years of industrial experience and I was completed my M.Tech and B.Tech in Computer Science and Engineering in 2012 and 2008 respectively from JNTUH , Hyderabad, Telangana, India. I presented many papers in international and national conferences held at various places. I am interested in doing research in computer networking and also interested to study in various upcoming technologies cloud computing, mobile computing etc

Mr.V.Biksham, working as Assoc. Prof. at CMR Engineering college, currently he is pursuing Ph.D.from JNTU Hyderabad and done M.Tech(SE) from JNTU, Hyderabad. He is having total 10 years of teaching and 1 year of industry experience and had published 4 papers in National and 2 papers in International journals.His areas of interest are Computer Networks, Software Engineering and Information Security.