International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue (6) NCRTCST-2013, ISSN 2249–071X

National Conference on Recent Trends in Computer Science and Technology (NCRTCST)-2013

Key Structure Based Approach towards Scalable Access Control in Cloud Computing

Challa. Madhavi Latha
Asst. Prof., Department of Computer Science & Engineering
CMRCET, Kandlakoya, Medchal (M), Hyderabad.
Email: saidatta2009@gmail.com

K. L. S. Soujanya
Assoc. Prof., Department of Computer Science & Engineering
CMRCET, Kandlakoya, Medchal (M), Hyderabad.
Email: souj47@gmail.com

***Abstract* – Cloud computing has emerged as one of the most influential paradigms 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. The existing methods for implementing access control policies are not scalable. In this paper we propose a key structure based approach for access control in cloud computing which is scalable. The anticipated scheme not only accomplishes scalability but also acquires adaptability and fine grained access control in supporting compound traits of ASBE due to its hierarchical structure.**

***Keywords* – Key Structure Approach, Attribute Set Based Encryption (ASBE), Cloud Computing, Access Control.**

I. INTRODUCTION

Cloud computing is a style of computing in which dynamically scalable and commonly virtualized resources are provided as a service over the Internet. The cloud service providers allow users to instantiate cloud services on demand and thus purchase precisely the capacity they require when they require based on pay-per-use or subscription-based model. Although cloud computing provides a number of advantages that include economies of scale, dynamic provisioning, increased flexibility and low capital expenditures, it also introduces a range of new security risks. As cloud computing brings with it new deployment and associated adversarial models and vulnerabilities, it is imperative that security takes center stage. To take full advantage of the power of cloud computing, end users need comprehensive security solutions to attain assurance of the cloud's treatment of security issues.

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. As data represents an extremely important asset for any organization the enterprise users will face serious problems if its confidential data is disclosed to their business competitors. Hence cloud users in the first place want to make sure that their data or kept confidential to outsiders including the cloud provider and their potential competitors. This is the first data security requirement.

Cloud computing have many advantages in cost reduction, resource sharing, and time saving for new service deployment. While in a cloud computing system, most data and software that users use reside on the Internet, which bring some new challenges for the system, especially security and privacy. Since each application may use resource from multiple servers. The servers are potentially based at multiple locations and the services provided by the cloud may use different infrastructures across organizations. All these characteristics of cloud computing make it complicated to provide security in cloud computing. To ensure adequate security in cloud computing, various security issues, such as authentication, data confidentiality and integrity, and non-repudiation, all need to be taken into account.

Along with data security flexible access control is strongly desired in the service oriented cloud computing model. Access control is a classic security problem which dates back to the 1960's or early 1970's[9] and various access control models have been proposed. Among them, Bell-La Padula (BLP) [10] and BiBa [11] are two famous security models. Unfortunately these models are only applicable to systems in which data owners and the service providers are with in the same trusted trait. Since data processor and service providers are usually not in the same trusted trait in cloud computing, a new access control scheme employing attributed based encryption [16] is proposed by Yu et al [17], which adopts the so called key policy attribute based encryption to enforce access control. However this scheme falls short of flexibility in attribute management and lacks scalability in dealing with multiple levels of attribute authorities.

In this paper we propose a key structure based approach for access control in cloud computing which is scalable. The rest of the paper is organized as follows. Section II provides Related work. In section III we present our proposed work. In section IV we describe in detail the construction of key structure and show how it is used in access control of data in cloud computing. Lastly we conclude the paper in section V.

II. RELATED WORK

The cloud computing system under consideration consists of five types of parties: a cloud service provider, data possessor, data customers, a number of realm authorities, and a reliable authority. The cloud service provider supervises a cloud to provide data storage

service. Data possessor encrypts their data files and stores them in the cloud for sharing with data customers.

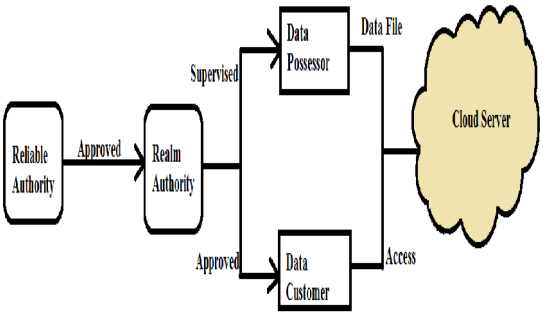

Fig.1. System Model of Cloud Computing.

Data customers download encrypted data files of their interest from the cloud and then decrypt them to access the shared data files. Each data possessor/customer is administrated by a realm authority. A realm authority is managed by its parent realm authority or the reliable authority [20]. Data possessors, data customers, realm authorities, and the reliable authority are organized in a hierarchical manner which is shown in Fig. 1. The reliable authority is the origin authority and accountable for managing top-level realm authorities. Each top-level realm authority corresponds to a top-level organization. Data possessor/customers may communicate to employees in an organization. Each realm authority is accountable for organizing the realm authorities at the next level or the data possessor/customers in its domain. In our scheme, neither data possessors nor data customers will be always online. They come online only when needed, while the cloud service provider, the reliable authority, and realm authorities are always online. The cloud is assumed to have plentiful storage capacity and computation power. In addition, we assume that data customers can access data files for reading only. We imagine that the cloud server provider is entrusted in the sense that it may conspire with hateful users to produce file contents stored in the cloud for its own profit. The reliable authority acts as the origin of faith and authorizes the top-level realm authorities. A realm authority is trusted by its subsidiary realm authorities but may try to get the private keys of users outside its realm. Users may try to access data files either within or outside the range of their access privileges, so hateful users may conspire with each other to get responsive files beyond their privileges.

III. PROPOSED WORK

The proposed HASBE (Hierarchical ASBE) scheme seamlessly extends the ASBE scheme to handle the hierarchical structure of system users in the following figure.

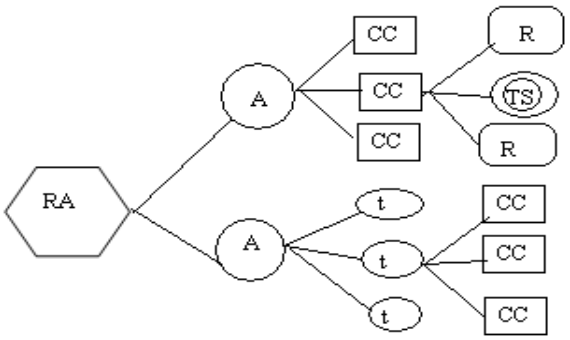

Fig.2. Hierarchical structure of System clients

RA: The Reliable Authority
A: Realm Authority
TS: trait Set
cc: Cloud Clients
t: trait
R: Realm

Recall that our related System model consists of a Reliable authority, multiple realm authorities, and numerous users corresponding to data processor and data customers. The reliable authority is responsible for generating and distributing system parameters and root master keys as well as authorizing the top-level realm authorities. A realm authority is responsible for entrust keys to subordinate realm authorities at the next level or users in its realm. Each user in the system is assigned a key structure which specifies the traits associated with the user's decryption key.

HASBE Architecture:

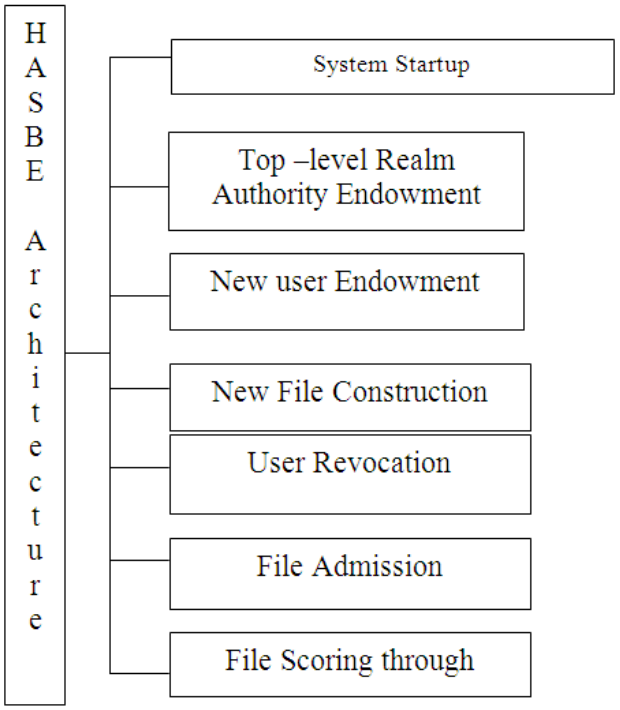

Fig.3. HASBE Architecture of operations

We are now ready to describe the main operations of HASBE: System Startup, Top-level Realm Authority endowment, New user endowment, New File construction, User Revocation, File Admission, and File scoring through.

System Startup: The Reliable authority calls the setup algorithm to create system public parameters PK and master key MK_0. PK will be made public to other parties and MK_0 will be kept secret.

Top level Realm Authority endowment: A realm authority is associated with a unique ID and a recursive trait set T= $\{T_0. T_1, T_2.....T_m\}$ where $T_i=\{ti,1,ti,2,.....ti,n\}$ with ti,j being the j^{th} trait in Ti and ni being the number of traits in Ai. When a new top-level Realm authority , i.e., RA_i, wants to join the system, the Reliable authority will first verify whether it is a valid realm authority. If so, the reliable authority calls Create RA to generate the master key for RA_i, After getting the master key, RA_i can authorize the next level realm authorities or users in its realm.

New User endowment: When a new user, denoted as u, or a new subordinate realm authority, denoted as RA_{i+1}, wants to join the system, the administrating realm authority, denoted as RA_i, will first verify whether the new entity is valid. If true, RA_i assigns the new entity a key structure Ť corresponding to its role and a unique ID.

New File construction: to protect data stored on the cloud, a data processor first encrypts data files and then stores the encrypted data files on the cloud. As in[16], each file is encrypted with a symmetric data encryption key DEK, which is in turn encrypted with HASBE. Before uploading to the cloud, a data file is processed by the data processor as follows:

- Pick a unique ID for this data file.
- Randomly choose a symmetric data encryption key DEK $\leftarrow$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 DEK with T using algorithm Encypt(PK, DEK,T) of HASBE which returns cipher text CT.

Finally the encypted data file is stored on the cloud in format as shown in fig:2

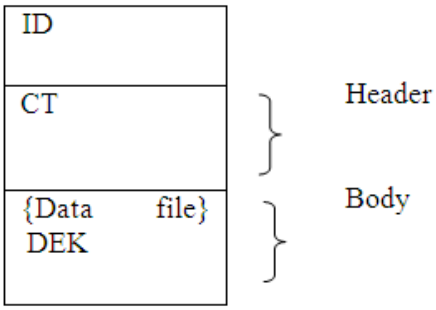

Fig.4. Format of a data file on the cloud

User Revocation: Whenever there is a user to be revoked, the system must make sure the revoked user cannot access the associated data files any more. One way to solve this problem is to re-encrypt all the associated data files used to be accessed by the revoked user, but we must also ensure that the other users who still have access privileges to these data files can access them correctly.

File admission: 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_u) to obtain DEK and the decrypt data files using DEK. Decrypt(CT,SK_u) algorithm is a follows:

Decrypt (CT,SK_u): This algorithm accepts ciphertext CT and user u's key structure as input. The algorithm first calls T(A) to verify whether the key structure A in SK_u satisfies the tree access structure T associated with the CT. the function T(A) is performed recursively. For each node x in T, there is a set S_x of labels returned by $T_x(A)$. if A does not satisfy T, the algorithm returns null; otherwise the algorithm picks one *i* from the set returned by T(A), and calls function DecryptNode(CT,SK_u,t,i) on the root node of T, where t is a node from T.

File scoring through: Encrypted data files can be deleted only at the request of the data processor. To delete an encrypted data file, the data processor sends the file's unique ID and its signature on this ID to the cloud. Only upon successful verification of the data processor and the request, the cloud deletes the data file.

IV. KEY STRUCTURE

We use a recursive set based key structure as in [19] where each element of the set is either a set or an element corresponding to a trait. The depth of the key structure is the level of recursions in the recursive set, similar to definition of depth for a tree. For a key structure with depth2, members of the set at depth 1 can either be trait elements or sets but members of a set at depth 2 may only be trait elements.

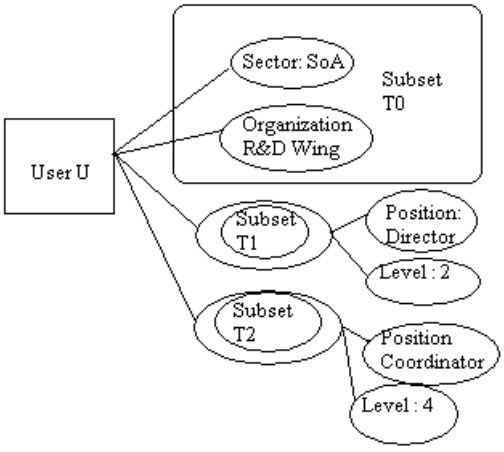

Fig.5. Example Key structure.

Consider the example shown in Fig 5, where {sector: SoA, organization: R&D Wing, {Position : Director, Level : 2}, {Position: Coordinator, Level:4}} is a key structure of depth 2. It represents the traits of a person who is both a director of level 2 for a unit and a coordinator of level 4 for another unit in the Research and Development Wing of the Sector of Agriculture(SoA).

Discussion

In this discussion, we compare our model with the existing scheme on security features of access control for cloud computing. These are Scalability, Flexibility, Effieient user Revocation, Expressiveness.

- *Scalability:* we extend ASBE with a hierarchical structure to effectively entrust the reliable authority's private trait key generation operation to lower-level realm authorities. By this, the work consignment of the reliable root authority is reallocated to lower-level realm authorities, which can provide trait key generations for end users. Thus this hierarchical structure accomplishes great scalability.
- *Flexibility:* Flexibility is achieved by organizing user traits into a recursive set structure and allows users to impose dynamic constraints on how those traits may be combined to gratify a policy.
- *Efficient user Revocation:* To transaction with user revocation in cloud computing, we append an expiration- time trait to each user's key and utilize multiple value assignments for this trait. So we can update user's key by simply accumulation a new expiration value to the existing key.
- *Expressiveness:* In our model a user's key is associated with a set of traits, so it is conceptually closer to traditional access control methods.

V. CONCLUSION

In this paper we introduced key structure scheme for HASBE for realizing, scalable access control in cloud computing. Our structure incorporates a tree structure of System users by applying entrust algorithm to ASBE. We implemented the proposed scheme and conducted comprehensive performance analysis and evaluation which showed its efficiency and advantages over existing schemes.

ACKNOWLEDGEMENT

The authors would like to thank the organizers of NCRTCST for their encouragement and 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/

[3] Amazon Web Services (AWS) [Online]. Available: https://s3.amazonaws.com/

[4] R. Martin, "IBM brings cloud computing to earth with massive new data centers," *InformationWeek* Aug. 2008 [Online]. Available: ttp://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.

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

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

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

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

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

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

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

[14] 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.

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

[17] 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.

[18] J. Bethencourt, A. Sahai, and B. Waters, "Cipher text-policy attributebased encryption," in *Proc. IEEE Symp. Security and Privacy*, Oakland, CA, 2007.

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

[20] A. Sahai and B. Waters, "Fuzzy identity based encryption," in *Proc. Acvances in Cryptology—Eurocrypt*, 2005, vol. 3494, LNCS, pp. 457–473.