International Journal of Advanced Technology in Engineering and Science
Vol. No.5, Issue No. 04, April 2017
www.ijates.com
ijates
ISSN 2348 - 7550

A SURVEY ON HIERARCHICAL ACCESS CONTROL IN CLOUD COMPUTING

Harshwardhan Rathore[1], Prof. P.M. Chouragade[2]

[1]Computer Science and Engg. Dept. Government College of Engineering Amravati, Amravati, (India)
[2]Computer Science and Engg. Dept. Government College of Engineering Amravati, Amravati, (India)

ABSTRACT

Security has always been a big concern when it comes to the field of cloud computing. Granting suitable access rights to various users that access the services and files stored in the cloud has always been a tough job to be done. The paper talks of one such access control policy that is hierarchical access control in clouds. We first introduce the basics of cloud computing and the necessity of access control in clouds. Then we have a glance at all the techniques that have been already used as an attempt to implement hierarchical access control so far and also briefly describe the various key management techniques used in hierarchical access control. Finally we conclude the paper.

Keywords- Access Control; Cloud Computing; Cloud Storage; Hierarchy; Security

I. INTRODUCTION

With the advent of computers a new digital era started in the world. Most of the mechanical computational work which earlier used to be done by man was now being handled by computers. Computers soon became very common and were being used in every possible field. From coffee shops to super markets, hospitals to schools and many other such places computers were employed to make man's task easy. Soon this computer centric world was producing such an enormous amount of data every day that it became impossible for these firms to save this valuable data with themselves. This scenario gave rise to the concept of cloud.

A cloud in simple English is a distant server away from your location owned by a cloud service provider which offers you with additional storage space or computational power on demand on pay per use basis. Firms unable to process or store their data signup with these cloud service providers and use such cloud services to process and store their data.

Cloud service comes up with many added benefits. The user doesn't have to buy or maintain the hardware he uses to store his data. Also the data is remotely accessible at any hour of the day which makes work quite flexible.

II. SECURITY IN CLOUDS

Security has always proved to be a big challenge in the field of cloud computing the prime reason being the fact that your data is being stored with a third party i.e., the cloud service provider. The service provider has full control over your data and can easily breach the system to access it to have an unethical advantage using it.

Encryption has always been the aid to all such security issues in cloud. Using a suitable encryption technique we convert our files to be uploaded to a unreadable format which couldn't be interpreted by the cloud service provider and could only be decrypted by the person who has the decryption key for the decryption of the file. Though efficient, this technique doesn't prove to provide guaranteed security in business institutions and offices where there is a long hierarchy of employees of which the lowest members are the least powerful and only need to access stuff intended for their day to day assignments whereas the ones on the top of the hierarchy are the most powerful and access all the data related to the company's function be it from any department.

III. HIERARCHICAL ACCESS CONTROL IN CLOUDS

In real life scenarios there are many instances where mere encryption isn't sufficient to suffice the amount of security needed in the system. Most of the business institutes maintain a hierarchy in their employees which starts with the Chief Executive Officer (CEO) and ends up to clerks and peons. Now the CEO being at the top needs to take care of all the affairs that are going in the company and hence needs accessibility to all the data regardless of the time it was generated or the department which generated it. On the other hand the members lower down in the chain at the bottom like clerks need only the data of the departments they are associated with and of which only that part of the data that they are going to make use of. For this reason the data should be accessible in a hierarchical form too.

If we encrypt files in the old fashioned way then who so ever with the decryption key may it be the CEO or a clerk can decrypt the file and make use of it. This should not happen with confidential files and hence a hierarchy should be maintained in order to control the accessibility of files. In such cases hierarchical access control comes in handy.[1]

A lot of efforts have been made until now to implement such a system. We can summarize each of them briefly as follows:

3.1 Linear Hierarchies

In a Linear hierarchical system [2]primarily two kinds of entities are found, that is, subjects and objects. Subjects are the entities which access the objects in some or the other way for example users of the system or other programs. On the other hand objects are the entities which are to be accessed for example files, messages or mails. S denotes the set of all subjects whereas O denotes the set of all accessible objects. The set E is the union of set S and set O and is called as the set of all entities.

The subjects in set S are divided into specific classes where the members of each class hold some specific access permissions and rights. Let us define a relation > such that if class $S_i > S_j$ then the class S_i can access all the data accessible by class S_j. The main perspective of linear hierarchies id that a subject class can have only one child as well as only one parent. If $S_i > S_j$, then S_i is the ancestor class of S_j whereas S_j is said to be the descendent of class S_i.

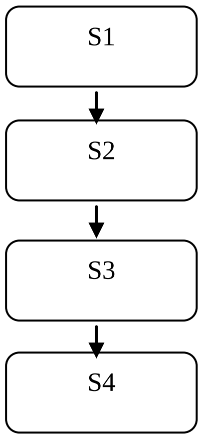

Fig 3.1. Linear Hierarchy

Fig 3.1. illustrates a simple linear hierarchy with four classes. Note that every class has only one child and only one parent which is the required condition for a linear hierarchy. Here, for example, S_1 is the parent of S_2 and an ancestor of S_3 and S_4. Likewise, S_2 is the child of S_1 whereas $S_3$and S_4 are descendents of S_1.

3.2 Tree Hierarchies

Linear hierarchies restricts one node to have only a single parent and a single child. Tree hierarchies on the other hand lets a node have multiple children at the same time. Fig 3.2. illustrates an example of tree hierarchy[3].

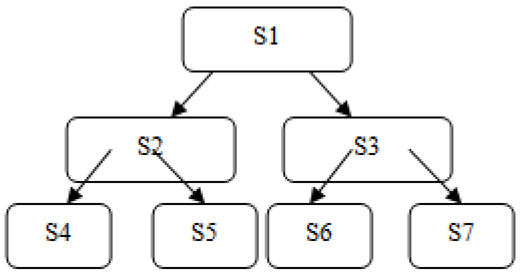

Fig 3.2. Tree Hierarchy

Note here S1 is the parent of both S2 and S3 whereas is the ancestor to all S4, S5, S6 and S7. Similarly S2 and S3 are the children of S1 whereas S4, S5, S6 and S7 are the descendents of S1. Here S1>S2,S3, so S1 can access all the files accessible by both S2 and S3 and their corresponding children and descendents if existed.

3.3 Directed Acyclic Graph Hierarchy

The Directed acyclic graph hierarchy abbreviated as DAG hierarchy lets a class to have many parents and many children at the same time. It means that both the parents of the class will be able to access the files of the child. Figure 3.3. depicts a directed acyclic graph hierarchy.

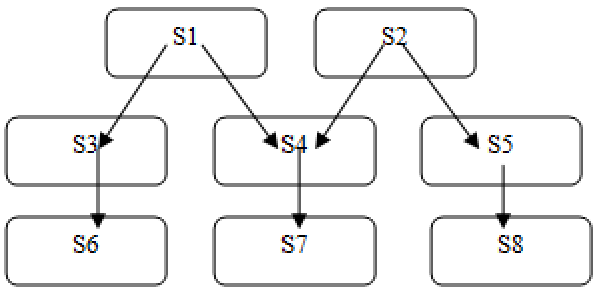

Fig. 3.3. DAG Hierarchy

The above figure simply illustrates a DAG hierarchy with both class S1 and S2 being the parents of class S4 and hence both S1 and S2 can access the resources of the class S4 and it's descendents. Here both S1 and S2 are the ancestors of S7 and S7 hence becomes their descendent.

IV. KEY MANAGEMENT TECHNIQUES

Key management has always been critical part of hierarchical access control. Keys are managed in mainly three ways that are centralized, decentralized and distributed approach. In the centralized approach [4,5,6,7,8] a central authority is responsible for creating the various service groups and assignment and management of the keys. In the decentralized approach [9,10,11,12,13] a set of managers are assigned the job of key assignment and management. It is called decentralized as no centralized authority does the whole task of key assignment. The distributed approach [14,15,16,17,18] is very similar to the decentralized approach except that all the managers agree to each key assigned to various classes.

How a parent or an ancestor could derive the key of a child or a descendent has always been a major issue. There are two approaches which have been proposed this issue namely dependent key approach and independent key approach. We will describe both these approaches in brief in this section.

4.1 Dependent Keys Approach

Every object needs a secret key for encryption. In order to decrypt this file we need the same key to decrypt the file which was used while it's encryption. Dependent key approach doesn't force the user to provide the exact key which was used while encryption, rather allows the user to generate the key with the help of his own key combined with some public parameter in order to decrypt the encrypted object. This approach also emphasizes more on the relationships between the various security classes. The dependent key approach can be further divided into two categories namely direct dependent key approach and indirect dependent key approach.

The indirect key approach forces a class to compute all the intermediate keys in between itself and the descendent whose resources the class is trying to access. In other words class S_i will have to compute all the intermediate keys of various classes between itself and class S_j in order to generate the key of class S_j to access it's resources. Sandhu [19,20], Gudes [21], Yang and Li [22], He et al. [23], Wang et al. [24], and Gawdan et al. [25] proposed indirect schemes that are based on one-way functions. A one way function is used to compute the key from its parent's key in such schemes.

Contrary to the indirect key approach, the direct key approach [26,27] lets a class to directly generate the key of its descendent with its own key added with some public parameters without computing the keys of the intermediate classes falling between itself and its ancestor whose resources are needed to be accessed. In other words a class S_i can directly compute the key of its descendent S_j without computing any intermediate keys corresponding to the intermediate classes falling between them.

4.2 Independent Keys Approach

The independent keys approach [28] emphasizes more on the relationship between the various users and their accessible objects rather than the relationship between various security classes. The users who can access the same set of resources and possessing the same access rights are collected into a single resource group. Each resource group is given a key to minimize the number of global keys and the approach deals with the interaction between these service groups and their corresponding respective keys.

V. CONCLUSION

The above schemes if applied to cloud computing could bring up a totally new way cloud could be accessed and could also increase the security. The hierarchical techniques namely linear, tree and DAG increase in functionality, flexibility and complexity in ascending order. Talking of key management, the dependent key approach could minimize the total number of global keys but turns out to be more computation hungry. On the other hand the independent key approach could be simple to deploy but may generate a large number of service groups and hence increasing the number of global keys.

REFERENCES

[1] Shaohua Tang, Xiaoyu Li, Xinyi Huang, Yang Xiang, Lingling Xu, Achieving Simple, Secure and Efficient Hierarchical Access Control in Cloud Computing, IEEE Transactions on Computers (Volume: 65, Issue: 7)

[2] Hani Ragab Hassen, Hatem Bettahar, Abdelmadjid Bouabdallah, Yacine Challal. An e_cient key management scheme for content access control for linear hierarchies. Computer Networks,Elsevier, 2012, 56 (8), pp.2107-2118. <hal-00690007>

[3] H. R. Hassen, A. Bouabdallah, H. Bettahar, Y. Challal, Key management for content access control in a hierarchy, Comput. Netw. 51 (11) (2007)

[4] C. K. Wong, M. Gouda, S. S. Lam, Secure group communications using key graphs, IEEE/ACM Transactions on Networking 8(1) (2000) 16-30.

[5] H. Harney, C. Muckenhirn, RFC 2093 - Group Key Management Protocol(GKMP) Architecture (July 1997).

D. Balenson, D. McGrew, A. Sherman, Key Management for Large Dy- namic Groups: One-Way Function Trees and Amortized Initialization, draft-balenson-groupkeymgmt-oft-00.txt, internet-Draft (February 1999).

[6] A. Penrig, D. Song, D. Tygar, Elk, a new protocol for e_cient large-group key distribution, in: Security and Privacy, 2001. S P 2001. Proceedings. 2001 IEEE Symposium on, 2001, pp. 247 -262. doi:10.1109/SECPRI.2001.924302.

[7] C. K. Wong, S. S. Lam, Keystone: A group key management service, in: International Conference on Telecommunications, ICT, 2000.

[8] A. Ballardie, RFC 1949 - Scalable Multicast Key Distribution (1996).

[9] T. Hardjono, B. Cain, I. Monga, Intra-domain Group Key Management for Multicast Security (September 2000).

[10] S. Mittra, Iolus: A framework for scalable secure multicasting, 1997, pp. 277-288.

[11] L. R. Dondeti, S. Mukherjee, A. Samal, Scalable secure one-to-many group communication using dual encryption, Computer Communications 23(17)(2000) 1681-1701.

[12] S. Rafaeli, D. Hutchison, Hydra: a decentralised group key management, in:Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002.WET ICE 2002. Proceedings. Eleventh IEEE International Workshops on, 2002, pp. 62 - 67. doi:10.1109/ENABL.2002.1029990.

[13] M. Steiner, G. Tsudik, M. Waidner, Di_e-Hellman key distribution ex- tended to group communication, 3rd ACM Conference on Computer and Communications Security (1996) 31-37.

[14] Y. Kim, A. Perrig, G. Tsudik, Simple and fault-tolerant Key Agreement for Dynamic Collaborative groups, 7th ACM Conference on Computer and Communications Security (2000) 235-244.

[15] C. Becker, U. Wille, Communication complexity of group key distribution, ACM Press, 1998.

[16] O. Rodeh, K. P. Birman, D. Dolev, Optimized group rekey for group communication systems, in: In Proceedings of ISOC Network and Distributed Systems Security Symposium, 2000.

[17] C. Boyd, On key agreement and conference key agreement, Information Security and Privacy: Australasian Conference LNCS(1270) (1997) 294-302.

[18] R.Sandhu, Cryptographic implementation of a tree hierarchy for access control, Information Processing Letters 27, no. 2 (1988) 95-98.

[19] R. Sandhu, On some cryptographic solutions for access control in a tree hierarchy, IEEE (1987) 405-410.

[20] E. Gudes, The design of a cryptography based secure _le system, IEEE Transactions on Software Engineering. SE-6, 5 (1980) 411-420.

[21] C. Yang, C. Li, Access control in a hierarchy using one-way functions, Elseveir Computers & Security 23 (2004) 659-664.

[22] Z. hua He, Y.-S. Li, Dynamic key management in a user hierarchy, in: Anti-counterfeiting, Security and Identi_cation, 2008. ASID 2008. 2nd International Conference on, 2008, pp. 298-300. doi:10.1109/IWASID.2008.4688404.

[23] Z. Wang, X. Du, Y. Sun, G. Xu, Group key management based on in- formation ow policy for multi-privileged groups, in: Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on, 2010, pp. 485 {489. doi:10.1109/ICITIS.2010.5689555.

[24] I. Gawdan, C.-O. Chow, T. Zia, Q. Sarhan, A novel secure key management module for hierarchical clustering wireless sensor networks, in: Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on, 2011, pp. 312 {316. doi:10.1109/CIMSim.2011.63.

[25] I. Ray, I. Ray, N. Narasimhamurthi, A cryptographic solution to implement access control in a hierarchy and more, SACMAT'02 (2002) 65-73.

[26] X. Zou, B. Ramamurthy, S. S. Magliveras, Chinese remainder theorem based hierarchical access control for secure group communication, ICICS '01: Proceedings of the Third International Conference on Information and Communications Security (2001) 381-385.

[27] C. K. Wong, M. Gouda, S. Lam, Secure group communications using keygraphs, Networking, IEEE/ACM Transactions on 8 (1) (2000) 16 -30. doi:10.1109/90.836475.