A SIMPLE SECURE DIRECT KEY ACCESS SCHEME FOR HIERARCHICAL STRUCTURE IN CLOUD COMPUTING

Author 1
Surendra Kumar Pathak
Research Scholar, Department of Mathematical Sciences and Computer Applications,
Bundelkhand University,Kanpur Road
Jhansi,Uttar Pradesh, India.

Author 2
Dr. Alok Kumar Verma
Associate Professor, Department of Mathematical Sciences and Computer Applications ,
Bundelkhand University,Kanpur Road
Jhansi,Uttar Pradesh, India.

Abstract
Cloud computing spread its wings in the market. The Cloud computing is the pay per use model; we are using the resources as per our need. We can scale up and down the resources as per our requirement. In the classical computing user can access the resources as per the access permission given to the user. Identity and access management (IAM) is the important requirement in the cloud computing as well. It adds up the security as well since user can access the resources as per the permission given to him.In a cloud hierarchical system, a cloud class defines the privilege(s) for accessing the resources of the system. In a hierarchy, a cloud class having many users and the data is protected with the help of the key of the class. Customarily, a user related to the more privileged cloud class can access the resources of the lower privileged cloud class[1]. There are two schemes of key assignment scheme-static and dynamic; in this paper we have focused to get the key directly.

Keywords: secure PRF, access control, direct and indirect HKAS.

Introduction

Cloud computing is now adopting gradually both by industry and academia. The main important aspect is data in cloud computing as we are storing our important data on cloud and accessing the data from cloud, so it is also very important that only authorized people can access the data. In the Figure 1, a hierarchy is presented based on an educational institute; in an educational institute a director can access the data of all the department of the Institute. In the next label the Deans/HODs may be there to access the data and in the next level different department are there and having permission to access the data/resources of the same department. It is not possible that the employee of the department can access the data of the director however director can access the data of the entire department since he/she is at the top of hierarchy.

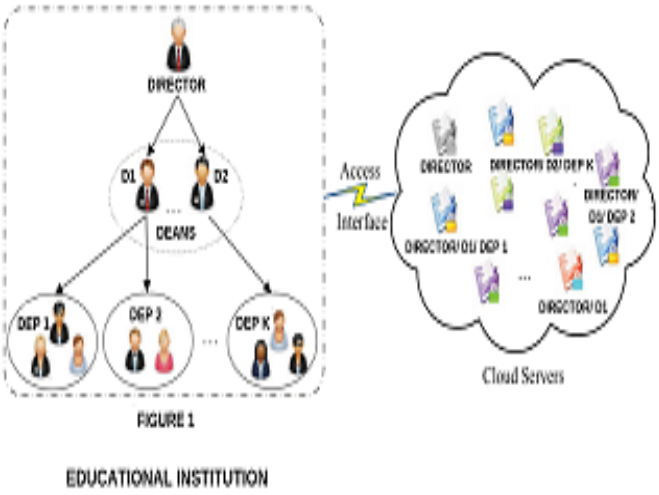

Figure 1: An Educational hierarchical systems interacting with Cloud Server [1]

A no. of hierarchical key assignment strategy (HKAS) proposed for both classical and cloud computing. The HKAS proposed by Akl and Taylor[2] focused that the users can be clubbed into different group based on the accessibility permission of the user. The strategy proposed by Akl and Taylor based on HKA strategy to implement a HAC policy. Subsequently, many researches anticipated methods for the improvement of the performance; they also mentioned underneath policies for dynamic access control or providing separate features[3][4][5]. Akl and Taylor[2] have given the name classes, we have given the name to these classes as cloud class (CC).The CC which is the higher level, has privileged to access the resources of the lower level CC directly but the lower level class cannot access the resources of its ancestor class. However the reverse is not true.

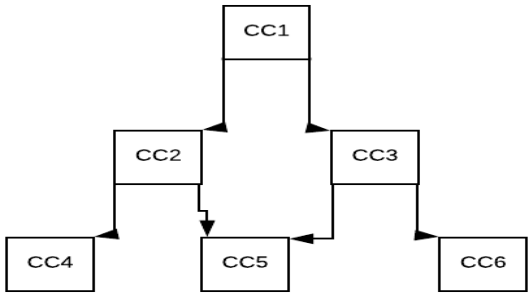

Figure 2: Cloud Hierarchies [6]

A no. of hierarchical key assignment schemes proposed for both classical and cloud computing. Static hierarchy permits only revocation of users whereas a dynamic hierarchy allows addition and deletion of classes and relation between those classes. There are two types of HKAS where direct HKAS directly drives the desired key [6][5][7] whereas the indirect HKAS derives it in two steps first you drive the immediate descendent class's key and then find out the next descendent class[6].The scheme proposed by Tang and et al. [1] used the hierarchical access control and used the pseudorandom function with the keys at the hierarchical level to get the keys of the descendent class.

In a cloud hierarchy user's key follows the partial order set. Let $Ụ_i$ = {CC1, CC2, CC3, CC4, CC5, CC6…CCn} is a set and ≤ being a binary relation on the set of cloud classes. A partial order set on (Ụ, ≤), CCj<CCi depicts that user in a hierarchy of class CCi is having higher privileged and can access the data/resources in CCj, whereas the reverse is not allowed means lower privileged class cannot access the resources of higher privileged class. The binary relation ≤ satisfies the following three properties of a partial order set-reflexive, anti-symmetric and transitive:

CCi ≤ CCi (Reflexive property)
CCj ≤ CCi and CCi ≤ CCj implies CCi=CCj
(Anti-Symmetric Property)
CCj≤CCk and CCk≤CCi implies CCj≤CCi
(Transitive Property)

A set with a binary relation (Ụ, ≤) is called a poset, is used to assign the keys in hierarchical structure.Figure 2 shows that a class which is placed at upper end in hierarchy having privileged to access the class which is placed at lower end in the hierarchy but the reverse is not allowed. There are situation where a node leave side in hierarchy having higher privileged to access the resources. There should be provision to reallocate the keys so that the keys should be distributed dynamically so that the node which left the system or hierarchy should not be able to access the system.

Methods

In our proposed scheme we introduce a secure pseudorandom function (SecurePRF) along with a key of the node which wants to access the keys of the descendent node.

This paper consists of the description of the security system including cloud and hierarchical key assignment scheme after that we have given the definition of our proposed secure pseudorandom function (SecurePRF).

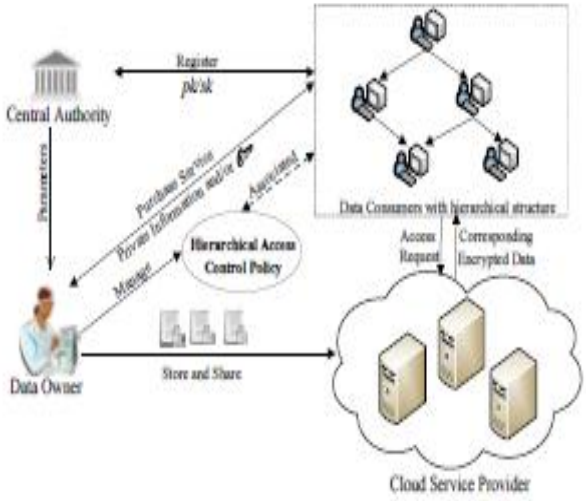

Figure 3: System Model of Direct HKAS [6]

As depicted in the figure a HKAS with connected to cloud computing has the following entities- Data owner, central authority, cloud service provider and data consumer with hierarchies.

It should be noted that data access and controlling policy is maintained by the data owner. The Central authority is used as the main authority which is also a trusted authority. We would like to emphasis here that the security strategy is maintained specifically through data owner and not by the cloud provider. It represents the actual consumer of the data. It keeps pace with data hierarchical structure. The data owner encrypted the shared data is stored on cloud servers with a specific tag ID of the hierarchical node. If a request sends by the data owner to access the CSPs with a hierarchical tag ID, the CSP gets the encrypted data of the user through data owner.

In our method we have neglected the data which is shared and focuses only on hierarchical management of keys. We have suggested that if we are not omitting this case we have to use secure pseudorandom function. We use a poset (Ụ, ≤) to represent the hierarchical construction of the consumer, where Ụ represents the finite set of cloud class and ≤ is the binary relation applied to the set. The term class is used rather than data user or cloud class. A poset can be denoted by the access graph G' = (V, E), where all the vertices in G' coincides with the cloud classes in Ụ' and an edge exists from Ụ' to Ụ" iff Ụ'> Ụ", obviously G ' is an acyclic graph.

Let £ be an acyclic graph of corresponding partial order hierarchies. The problem is now focusing how to assign key keys to each vertex and each vertex should be able to calculate the keys of its descendent vertices. The method to solve this problem is called HKAS is defined as follows:

HKAS for £ is a pair of algorithms (Gen, Der) satisfying the following condition

- Gen $(1^k,G)$ is a probabilistic polynomial time algorithm that takes input as function 1^k and a graph G=(V,E) ε £ and outputs

- pvt : private information for all Viε V
- ki: encryption key for all Viε V
- pub: public information

- Derv (G, Vi,Vj, pvt, pub) is a deterministic polynomial time algorithm that takes input as graph G, two classes Vi , Vjε V, Vi private information pvt and public information pub and outputs the encryption key kj of class Vj if Vj≤Vi or a denial symbol ⊥.

We represent (pvt, k,pub) as the output of **Gen** (1^k,G), where pvt and k represents private information and key.There are different cases may arise when considering dynamic key management- either entering or deleting from the system. If a new class is entered into the hierarchy, suppose two classes already are there Vi and Vj and we are inserting a new class Vk between Vi≤Vj then the inserted hierarchy would be Vi≤Vk≤Vj.

Suppose we want to remove a class from the existing system, if m+ 1 classes are there in the system with a hierarchical structure. Let Vr is a class which we want to remove from the system. In this condition data owner requires to update all the encryption keys belonging to the descendent classes of Vr.

Results

Various schemes have been analyzed viz. Atallah et al., Lin et al., A.D. Sentis et al., E. Freire et al. and Tang etal. And the results have been tabularized and shown in the table 1. We would like to emphasis that where Freire et al. and Tang et al. has taken security assumption as PRF but we are focusing on Secure PRF. In this scheme size of a portion in in Secure PRF is S, number of cloud classes is m.

Table 1: Comparison of different strategy

Strategy	Secret Information	Public information	Key derivation	Nature of Dynamism	Security Type used	Security Idea
M.J. Atallah et al. [3]	S	S|E| + |V|	$(C_{DT-SE}+T_{PRF})l$	Exist	KI	CPA Secure + PRF
Y.L. Lin et al. [8]	S	$(3\|V\|+\sum k_i+4)S$	$C_H+C_F+C_{DT-ECC}$	Exist	N/A	CPA - Secure-OWHF
A. D. Santis et al. [9]	S	$(\|E\| + 2\|V\|)\rho$	$(l+2)C_{DT-SE}$	Exist	KI	CPA - secure
E.S.V. Freire et al.[10]	S	2S	$(l+1)T_{PRF}$	NO	S-KI	PRF
S. Tang et al.[1]	4S	$(\|V\|^2+1)$	2M+2A	Exist	S-KI	PRF
Our Strategy	Our proposal to use Secure PRF instead of PRF					Secure-PRF

Notations used in Table 1:

|E|: denotes no of edges in the access graph G;

|V|: denotes no of classes in the access graph G;

l: path length between class V_i and V_j when class Vi wants to derive the encryption key of class V_j;

ρ: denotes the size of cipher text in a symmetric key encryption strategy;

C_{DT-SE}: denotes decryption time of a symmetric key encryption strategy;

T_{PRF}: time of calculating the PRF;

C_{DT-ECC}: denotes decryption time of an Elliptic Curve based public key encryption scheme;

Conclusion

We have seen that there are a number of HKAS algorithms available and each has proposed the idea about their security assumption, key derivation and the nature of dynamic hierarchy. Pseudorandom function when used with the hierarchy of users in cloud computing produced HKAS security. We projected a novel hierarchical method based on secure pseudorandom function .In Cryptography no function is perfectly secure but secure pseudorandom function can enhance the performance of the HKAS. The future work can be done in the direction to reduce the public information and perfectly use the secure PRF in HKAS.

References

[1] S. Tang, X. Li, X. Huang, Y. Xiang, and L. Xu, "Achieving Simple, Secure and Efficient Hierarchical Access Control in Cloud Computing," *IEEE Trans. Comput.*, vol. 65, no. 7, pp. 2325–2331, 2016.

[2] J. Crampton, "On Key Assignment for Hierarchical Access Control," 2006.

[3] M. J. Atallah, K. B. Frikken, and M. Blanton, "Dynamic and Efficient Key Management for Access Hierarchies," 2005.

[4] A. R. Nimje, P. V. T. Gaikwad, and P. H. N. Datir, "Attribute-Based Encryption Techniques in Cloud Computing Security : An Overview," vol. 4, pp. 419–423, 2013.

[5] Y. F. Chung, H. H. Lee, F. Lai, and T. S. Chen, "Access control in user hierarchy based on elliptic curve cryptosystem," *Inf. Sci. (Ny).*, 2008.

[6] Y. Chen, C. Chu, W. Tzeng, and J. Zhou, "CloudHKA : A Cryptographic Approach for

Hierarchical Access Control in Cloud Computing ,” vol. 3.

[7] M. Abinaya and T. Sivakumar, “Secure Key Management Scheme for Dynamic Hierarchical Access Control Based on ECC,” vol. 5, no. V, pp. 1076–1080, 2017.

[8] Y. L. Lin and C. L. Hsu, “Secure key management scheme for dynamic hierarchical access control based on ECC,” *J. Syst. Softw.*, vol. 84, no. 4, pp. 679–685, 2011.

[9] A. De Santis, A. L. Ferrara, and B. Masucci, “Efficient provably-secure hierarchical key assignment schemes,” *Theor. Comput. Sci.*, vol. 412, no. 41, pp. 5684–5699, 2011.

[10] E. S. V. Freire, K. G. Paterson, and B. Poettering, “Simple, efficient and strongly KI-secure hierarchical key assignment schemes,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7779 LNCS, pp. 101–114, 2013.