ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 3, March 2013

A Novel Role Based Cross Domain Access Control Scheme for Cloud Storage

Punithasurya K , Esther Daniel , Dr. N.A. Vasanthi

Abstract: **Cloud computing is the emerging technology and also it requires strong security when dealing with public cloud data. Cloud access control is one of the security requirements. The basic security requirements for cloud storage are Authentication, authorization and Access control. There are various access control scheme available for cloud storage. To ensure security in cloud, access control is the important prerequisite. Access control gives the authorization rights to the individual users. Access control basically consists of access privileges based on the user requirement. Providing security to the cloud is the major concern. Access control is needed for most of the environment such as peer to peer, grid and cloud. Most of the cloud computing environment uses Role Based Access Control (RBAC). A Novel Role Based Access Control scheme is proposed to enhance the security requirement of cloud storage which is named as secure cross domain access control. Our proposed methodology maintains user's roles, permission and set of user attributes to produce attribute ID for each user. The proposed access control scheme consists of the RBAC, ABAC and DRBAC. This scheme minimizes the time constraints problem and Location constraints problem.**

Keywords: **Access Control, Cloud storage, Security, Time Constraints and Location Constraints.**

I INTRODUCTION

Cloud computing is the most promising technology which provides various resources as services over the internet. Cloud services can be existing pay as you basis. There are various number services available that are SaaS (Software as a Service) , Iaas (Infrastructure as a Service) , PaaS (Platform as a Service). In SaaS service model Cloud users can access application software on demand. They do not need download the package and also there is no need of installing application software on their computer. SaaS model minimizes maintenance and support in the user side. To hold large number of users in cloud, the platform can be virtualized. In IaaS service model cloud provider offers resources from their large pool of cloud centers. It offers processing, storage and networking and other computing resources on demand. It includes software that can run on operating system. Here cloud owners have the control over storage, processing and networking. In PaaS service model cloud provider offers resources such as operating system, programming language environment, web services and database environment and storage environment.

Cloud storage is the online storage where there are virtualized pools of storage environments. These virtualized storage environment is monitored by third party cloud provider. Monitoring as a Service also provided by third party to manage virtualized pools of storage. Cloud provider manages large number of data centers where users' data's can be stored originally. Data centers can be virtualized according to the user requirement. Cloud storage offers more convenient way of storing data off line. Cloud storage has the advantage of storing data online through the internet. The data's that are stored on large pools of data center can be accessed through web enabled desktop or laptop. Giving access to the data stored on the cloud through web is major concern and it enhances security risks. Access control gives the access privileges to the user associated with cloud data. The purpose of ensuring one of the security requirements of cloud storage minimizes the security risks.

Access Control in cloud ensures the users access to the data, resources and checks whether they are authorized cloud users or not. Access control defines the set of policies or procedures which prevents the user access. It also identifies when the unauthorized users are trying to access the cloud environment. As the technology grows the need of

dynamic data storage is much important nowadays. The Grid and cloud environments are enhanced with dynamic online pool of virtualized storage. The relationship between user and resources are being dynamic.

On the whole cloud system provides security by controlling access to the data. The most common access control method is Identity based access control. Access control in cloud depends on the security policies and its procedures. Traditional access control mechanisms are DAC (Discretionary Access Control), and MAC (Mandatory Access Control) finally the most common used access control method is RBAC (Role Based Access Control).

II RELATED WORKS

The traditional access control mechanism is proposed by Ravi S.Sandhu.et al [1] that are DAC (Discretionary access Control) which gives access to the users based on the user identity and authorization. This method defines open policies and it owns its common policies to improve the system performance. In this method each individual object is checked for granting access but this method also has the negative aspect in assurance of flow of information in the system. MAC (Mandatory Access Control) is based on the number of subjects associated with objects and it assigns security levels as Lower and Higher security level. This method has the limitation in modifying security levels to the associated users. RBAC (Role Based Access Control) is the method which consists of roles and responsibilities. It identifies the roles and objects. This method will assign roles to the each user in that cloud environment. Roles can be assigned based on the least privileges and it determines higher roles. RBAC identifies the downside as identifying which role can be associated with which user. ABAC (Attribute based access control) is proposed by Abdul.et.al [2] which considers the users identification, authentication and authorization and also this method describes set of user attributes and its parameter. When dealing with distributed environment it is difficult to manage set of attributes. dRBAC (distributed Role Based Access Control) its solves the problem of giving access control to the distributed system and it determines the problem of time complexity and space complexity. coRBAC (Cloud optimized Role Based Access Control) this method consists of the functionalities of dRBAC and RBAC. This method is proposed by Zhu Tiayani.et.al [3] will improve the certification process.

Negative authorization is proposed by Xiaohui.et.al [4] it is needed when considering access control which will identify the faulty user and gives the authorization alert to prevent from the access. This also has the difficulty in identifying which user is said to be undesired user to the particular information. Domain RBAC which gives the resource isolation from the domain which is proposed by Vidya Ranganathan.et.al [5]. There is no proper mechanism to define which roles possibly will come under set of users. Object isolation from the users makes the boundary across the domain. This method will greatly minimize the time constraint problem when managing as separate domain, location also identified based on the domain separator.

III PROBLEM STATEMENT

The problem in implementing access control is location constraints and time constraints. Cloud environment ensures the secure access when considering as individual access method. When the user requesting access to the data from the same cloud environment but if that data is not available in that particular environment means there it identifies the location constraints problem and also there is a problem of getting delay.

There will be the chance of user leaving from that particular cloud environment. To enhance the access control from various domains, access can be established with the extension of Role Based Access Control Model. Generally RBAC is identified by set of users and associated objects to that user. Domain RBAC ensures the isolation of objects. If that Object made visible to public means all the users can view the data from all the domains. To ensure fast access from various domains, it can be achieved with Domain RBAC along with extension of Role Based Access Control.

If the particular data requested by the user is not made visible means user have to request for that particular data to that data owner. Here also problem arises when there is delay in reaching that particular domain. So to overcome this time constraints problem Secure Cross Domain Access is proposed.

IV SECURE CROSS DOMAIN ACCESS CONTROL

Secure cross domain access control is implemented with the combination of Domain RBAC and with the extension of Role Based Access Control. In role based access control there is no proper identification of which information is associated with which user so it is the major problem in identifying separate roles.

In Secure Cross Domain Access Control the flow of system measured as follows those are separation of domains, roles, permission and number of user associated with that domain. Each domain varies with time constraints.

In this Secure Access Control system security of data also enhanced when user uploading data on cloud. Here before uploading data on cloud user have to ensure that their data is encrypted properly with attribute based encryption. Cross domain is achieved with the help of Domain RBAC and Domain separator. Achieving secure access along with data security enhances secure cross domain access. Each domain has its associated users and its associated permissions such as read, write and both.

```
/* For achieving cross domain access */
/* For cross domain access consider set of domains for public sharing */
Domain A= { user n1, user n2, user n3….user nn }
Domain B= { user m1, user m2, user m3…user mn }
Domain C= { user p1, user p2, user p3…. user pn }
Roles = { Domain Manager, Domain Provider, Data owner }
Permissions = { R, W, RW }
For user n2 request resource in Domain A
{ if (user n2 attribute ID matches)
    { Data owner checks user n2 ID; //user n2 attribute ID has been stored in DO1
      Grants Access;
     Else
       Access forbidden; }
Else if (user n2 attribute not matches)
{ Data Owner checks for the user n2 roles & permission
  If (Authorized person in Domain A)
    { Domain A saying that Resource not available; } } }
/* Achieving cross domain access */
For user p1 request resource in Domain B
{ if( user p1's ID==TRUE)
   { user p1's ID available in Domain C;
     Domain C== user p1's ID;
     Domain B checks user ID in Domain Manager;
     Grant access to user p1;}
Else
   { Access forbidden; }}
```

Fig 1: Flow of Cross Domain Access Method

```
/* For Adding Data Security */
/* Adding ATTRIBUTE KEY GENERATION to make user identity secure */
For user m1's ID attribute
{ generate user m1 ID;
  Set of users m1's attribute -> Domain B;
  Attribute checks in Domain manager;
Now,
   Generate Random Value[unique] attribute = i;
   Generate Random value [ user ] = r;
   Secret key= (i+r).user m1's ID;
 }
/* Encrypting user data along with ID */
 Encrypt ( user ID. (i+r)+data)
/* Decrypting user data along with ID */
Decrypt ( user ID. (i+r)+data)
```

Fig 2: Ensuring Security in Cross Domain

Roles can be domain provider, domain manager and data owner. When user n1 request resource from domain A, It will grant the access based on the role and permission defined on that user n1.

Suppose if user n2 request resource from Domain B, here the scenario works where the domain B checks the identity of user n2 in Domain A. if it's authorized user, it will grant the access immediately else it will deny the access. Here Domain B checks the user identity in Domain A else it will check in Domain Manager. The Domain manager consists of all the users' identity along with attributes to verify the identity of each user in each domain. Based on the user ID it will verify the identity. Along with user ID and generated secret value the data can be encrypted in the user side. So the user data will be saved in encrypted format. The generated secret key is unique to each user.

This scheme achieves the secure cross domain access control which will ensure both cross domain access and security of user data stored in the cloud storage environment. This method also minimizes the time constraints problem. There won't be delay in receiving user data. Whenever user requests for the data from another domain it will grant the access once it verified the user identity. Here User Identity is important to ensure cross domain access. Security of the data also ensured using the user ID. This ID generation based on the Attribute Based Access Control.

V EXPERIMENT RESULTS

The results can be obtained through achieving secure cross domain access control. Basically Google Cloud Storage uses the Role Based Access Control Model and most cloud environments use the same [13]. To achieve secure cross domain access we have to consider same cloud provider from the different domains.

We have taken the basic parameters as time, location and availability to prove the secure cross domain access. Time constraints problem compares the user requested time and received time and location constraints compares the user from requested location and data origin. Availability verifies when the user request for the particular data from different domain, it should displays the all the available data along with the access permission to access the data in the accurate period of time of data.

This can be achieved using real time cloud environment google cloud storage to ensure role based access control which defines the Access Control List [13]. This Access control List defines the set of users and associated access to that user. ACL and roles and permissions also define the role based access control. Domain Identifier used to define the domain RBAC.

Using gsutil command line tool [13] we can verify the RBAC access method using the access control list.

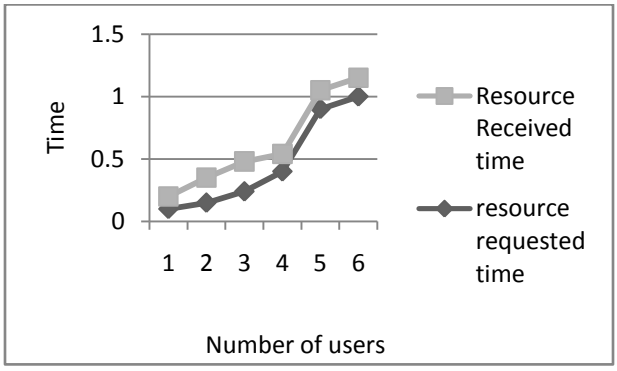

Fig.3: Time showing user requested vs received

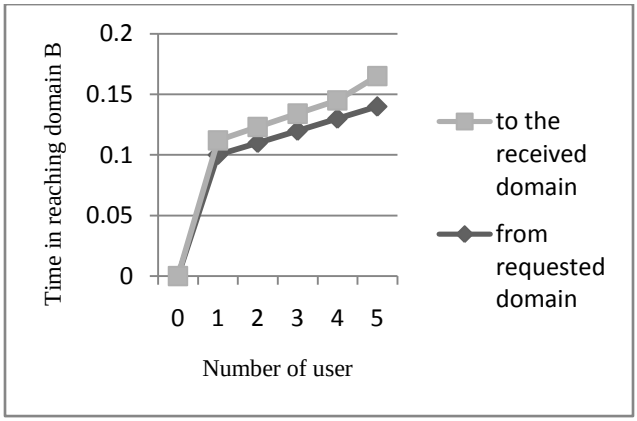

Fig 4: Achieving Access between two Domains

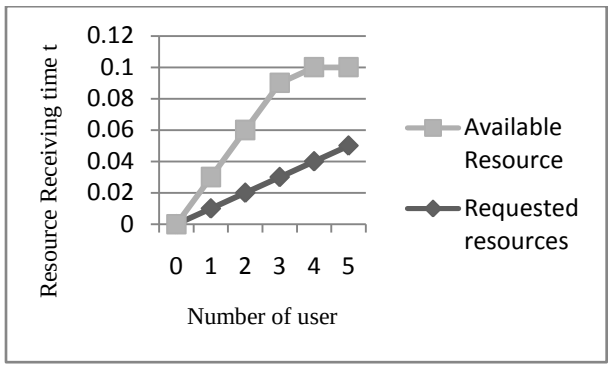

Fig 5: Resource Availability from different Domains

In Fig 3 the ratio between user requested resource time and received resource from various domains. In this graph it calculates the time that it achieves between requested and received time and also total time it takes for both. In Fig 4 it shows that it achieves cross domain access within certain time period where it takes minimum time to reach the next domain and also it defines the time taken to verify the identity in next domain. Here it ensures the negative authorization to identify the faulty user.

In Fig 5 it mentions the time it has taken to display the available resources in the next domain. Here it identifies the resources that are available in the next domain. When the user request to view the resource available in the next domain means it has to first verify the identity then it displays available resource.

VI CONCLUSION

In this paper we have discussed on various access control methodologies and proposed new access control scheme which ensures security along with cross domain access. This method helps to minimize the time constraints and location constraints problem and this will combine the domain identification along with role based access control. Domain Identification ensures three identities such user ID, data, roles and permission associated with that user. The proposed method defines all the roles and permissions and also the time that user logged on. Here this method greatly improves the time efficiency and location identification.

REFERENCES

[1] Ravi S. Sandhu and Pierangela Samarati "Access Control: Principles and Practice" IEEE Communications Magazine, September 1994.

[2] Abdul Raouf Khan " Access control in cloud computing Environment" ARPN Journal of Engineering and Applied Science,vol 7, No.5 May 2012.

[3] Zhu Tiayni, Liu Weidong, Song jiaxing "An Efficient role based access control system for cloud computing" 2011 11th IEEE International Conference on Computer and Information Technology.

[4] Xiaohui Li, Jingsha he, Ting Zhang "Negative Authorization in Access Control for Cloud Computing" International Journal of security and its Applications. Vol. 6, No.2 April 2012.

[5] Vidya Ranganathan, Guha P Venkataraman "Object Isolation for cloud Domain RBAC" IEEE Conference 2011.

[6] Ramadan Abdunabi, " Extensions to the Role Based Access Control Model for Newer Computing paradigms", Oct 26,2010

[7] Lorenzo Cirio, Isabel F. Cruz, Roberto Tamassia, "A Role and Attribute based Access Control System using Semantic Web Technologies".

[8] Mariana Raykova, Hang Zhao, Steven M. Bellovin, "Privacy Enhanced Access Control for Outsourced Data Sharing".

[9] Yingjie Xia, Li Kuang and Mingzhe Zhu "A Hierarchical Access Control Scheme in Cloud using HHECC" Information Technology Journal 9 (8): 1598-1606 , 2010

[10] Hazen A.Weber "Role Based Access Control: The NIST solution" San Institute of Info Reading Room, October 3 ,2008.

[11] Joon S.Park, Gail-Joon Ahn, Ravi Sandhu "Role-based Access control on the web using LDAP"

[12] Yingjie Xia, Li Kuang and Mingzhe Zhu "A Hierarchical Access Control Scheme in Cloud using HHECC" Information Technology Journal 9 (8): 1598-1606 , 2010

[13] https://developers.google.com/storage/docs/accesscontrol, https://developers.google.com/storage/docs/cross-origin.