# Security for cross tenant access control in Cloud Computing

**Mr Pramod Pillai**

**Abstract— Cloud computing is becoming a de facto standard for most of the technology solution that are emerging now a day. Compute sharing is one the key driving fact behind the popular emergence of cloud solution. In a typical cloud environment various tenant purchase the compute, storage resource using various pricing models and various tenants under a cloud service provider would be sharing the compute and storage resource unlike the legacy the legacy model where-in the tenants owned and maintained the compute and the storage resources. The cloud computing model is easier and cheaper to maintain and hence is emerging as a popular choice among the technology companies. In a cloud computing infrastructure there is no well-defined mechanism to enable sharing of resource between the cloud tenants. Sharing of the resources amount various tenants is not popular due to the security concerns associated with sharing the resource among tenants within a cloud service provider. There are few Solution that try to solve the security problem associated with resource sharing among tenants. Having a trusted mediator between multiple tenants with the authorized is one of the ways to have a secure resource sharing among various tenants. Even few research papers have been written and this thesis attempt to enhance one of the published solutions. Most of the existing research papers explores theoretical way to solve the problem. This paper analyses the security improvements that are possible and the design improvement to simplify the design and enhance the overall security of the system.**

*Index Terms*— **authentication, cloud Security , Cross tenant access control, resource sharing.**

## I. Introduction

Security is one of the biggest concerns in today's technology driven world. Security breach, privacy violation has become every day news. With many critical sectors like healthcare, transport, energy embracing the technological innovation in a big way securing the resources used by these sectors becomes critical. The business needs and the financial aspect is forcing many companies to adapt the cloud infrastructure of their business need. In a cloud infrastructure the infrastructure is owned and maintained by 3rd party, and the other unknown application would be executed in the same physical infrastructure. This is not how the legacy application were run. In legacy environment the software application was run in an independent infrastructure and for all practical purpose the application and hardware were owned by same entity. But the cloud paradigm changes this approach and enables faster and cost-effective deployments. But this approach raised many concerns on the security aspect. It would be a beneficial in terms of economics to share resources with other tenants in the cloud environment. But how to trust the resources that are hosted by the other entities in the same cloud environment. Also, how to share the resource with some unknow entity without risking the security of the host.

## II. BACKGROUND

To address the security concern and to obtain an optimum solution for the resource sharing in public cloud environment many theories and practical solutions were proposed in various papers. The notable few areas are,

- Role based access (RBAC)
- Single Sign-On (SSO) techniques are combined with Security Assertion Markup Language (SAML)
- Mediation service.
- Cross-tenant trust model (CTTM)
- Role based cross-tenant trust model (RB-CTTM)

Role based access (RBAC) is one of the methods that is discussed in many papers [2], [3], which provides granular level of access control. But it has its own limitation since in a cloud environment, either the individual or organization may have more than one tenants and would typically manage separate infrastructure. Therefore, it is very much possible that users do not agree to have a common agreement to manage the access control. RBAC works nicely for the user accounts and not so nicely for some tenants. Nebula Cloud Computing Platform [4] developed by NASA Ames Research Center at Moffett Field, California has integrated RBAC with finer authorization in to private cloud system. But Nebula supports only centralized authority, which is not a feasible solution for the public cloud solution.

Sayler et al in their paper [5] described in details autonomous multi-tenant network security framework known as Jobber. Jobber built a solid inter-tenant network policy solution which can automatically allow optimized communication between trusted tenants while also blocking traffic from untrusted tenants. But some of the security aspect is not demonstrated by the Jobber.

Single Sign-On (SSO) techniques can be combined with Security Assertion Markup Language (SAML) to perform authentication and also simple authorization in cloud environments [6], but fine-grained authorization provided by RBAC solutions is not possible.

The Multi-tenant data architecture by IBM [7] and Microsoft [8] had proposed a resource sharing approach in data-centric cloud using database schema. This solution is applicable to databases solution running on the cloud and not for other shared resources in the cloud.

While these approaches are suitable for specific aspects of cloud computing, there remains need for a general model of cross-tenant access control. Hence it is better to have a trusted entity which can manage the access control of shared resource for all the tenants under a Cloud Service Provider (CSP).

The papers [1], [7], [8] describes the usage of Mediation service to provide the access control in cloud infrastructure. But every solution has its own limitation. Many of the solution considers the resource sharing aspect, but do not consider the full security aspect in the solution. Quratulain et al designed a Mediation service taking security into consideration.

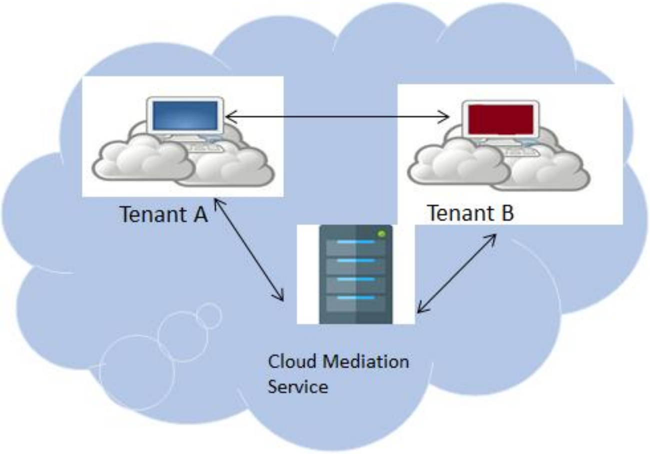

**Figure 1 Cloud Mediation Service Architecture**

Quratulain Alam et al in their paper [1] proposed a concept of cloud resource mediation service (CRMS) which acts a mediator between the various tenants and, the cloud users entrust the data to the cloud service provider (CSP). The architecture diagram of such a system is shown in fig [1]. Here the mediation server, CRMS, acts as a broker between various cloud tenants and the helps to attains the access control over the resources that the tenants share among themselves. The mediation server knows the set of resources that are present in the system, the owner of the resource and the list of tenants with whom the resources can be shared. The solution was not prototyped but proved using simulation models. Also, some of the messaging sequence considered in the paper were not optimal. Also, the resource sharing is done forever or till the originating tenant decides to deactivate the resource sharing but does not account for the cases where the resources has to be shared for a predefined time interval. Sharing the resource for infinite period of time is a risk since the resource is prone to security attacks. As part of this thesis the cross-tenant access is prototyped, and the aforementioned optimization are considered and implemented as part of the prototype.

### A. *Existing Security pit falls.*

Analyzing paper [1], following potential security issues has been identified.

1. There is no specification on how the deactivation of the shared resource by an unknown user is handled. This is security risk since the behavior of un-authorized deactivation is not defined in the system.
2. The paper is not clear on the security of communication channel between various entities.

### B. *Enhancement to the existing design*

The design proposed by [1] is good, but there is a scope of improvement. Following enhancements are proposed to make the design more robust and easier to implement.

1. Adding a timer-based deactivation, will be big leap in the security of the system as well as the enhancing the list of supported functionalities.
2. The message sequence for activation and deactivation of the resource present in paper [1] can be improved. The message sequence can be simplified for ease of implementation without losing the functionality or the security aspect of the message flow.

## III. IMPROVED SOLUTION

### A. *Improved Security*

The deactivation of the resource is to be handled only by the resource which triggered the authorization of the resource sharing. This is easy to relies on the practical system by implementing a secure ID token for every resource that is shared. During the resource sharing a secure ID is created and is shared with the person who is sharing the resource. During the deactivation of the resource the same secure ID has to be presented as part of authorization. Failure to provide the correct secure ID will not result in deactivation of the shared resource. The solution can be further enhanced to include notification to capture such failures.

The second way of resource deactivation if via expiry of activation timer associated with the resource. The timer associated with a shared resource is explained in subsequent section. The timer-based deactivation is system triggered and the security of the system has to ensure that the timer set against the resource is set by the authorized entities alone.

The architecture has a communication channel between the cloud tenants, and between the cloud tenant and the cloud mediation server. This interface has to be secured. The interface can be secured via IP security (IPSec) protocol or using Transport Layer Security (TLS),1.2 version. In the protype implementation this communication channel is secured via TLS1.2. A self-signed certificate is used for this purpose.

### B. *Improved Design*

#### 1) *Timer based Deactivation of shared resource*

Having a timer-based deactivation of resource is a feature in scenarios where the shared resource is to be used for short duration of time and its possible that the resource shared are billed as per the usage. Keeping the shared resource available for infinite amount of time is a security threat too especially when the shared resource is done via public cloud infrastructure. Considering the above factors this thesis proposed to implement the sharing of resource for predefined duration of time. On expiry of pre-defined time the sharing privileges are taken back, and further sharing is possible only via explicit resource sharing invocation.

Message flow associated with enabling timer-based activation of the resource is as shown in fig [2]. The timer-based activation ensures that the resources are not enabled for every there by preventing a potential misuse of resource.

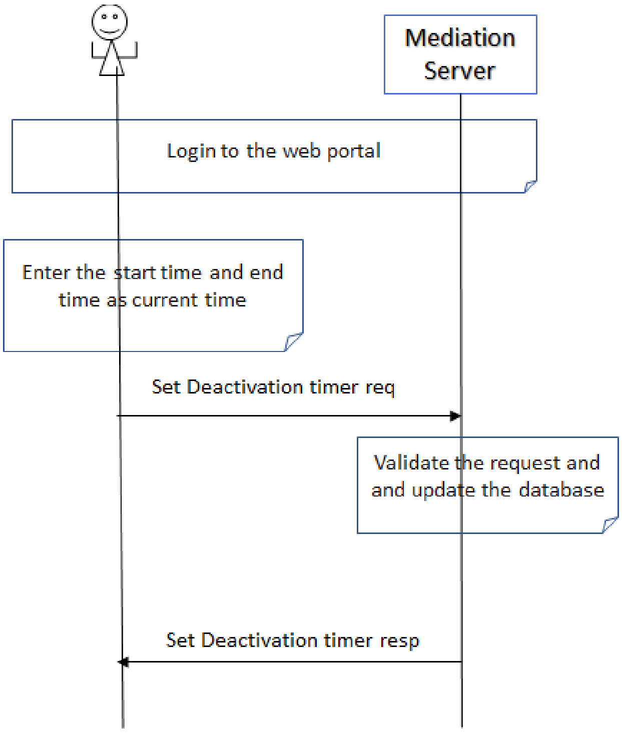
Mediation Server
Login to the web portal
Enter the start time and end time as current time
Set Deactivation timer req
Validate the request and and update the database
Set Deactivation timer resp

Figure 2 Enabling timer for activated shared resources

The details of the message exchange between various entities for enabling the timer on an activated resource is as explained below.

1) The user logs on to the web portal.
2) The user selects the menu to set the timer associated the shared resource
3) User sets the start time and end time indicating the time duration for which the resource is to be shared and clicks the submit button.
4) The web client sends the shared resource name, start time and end time details to the web server running on the mediation server.
5) The mediation server update the with the time start time and end time details.

### 2) Optimization of Activation and Deactivation message flow

The message sequence for activation and deactivation of the resource present in paper [1] can be improved further. The message sequence can be simplified for ease of implementation without losing the functionality or the security aspect of the message flow. In the original work by [1] the activation of the resource involves the participation of requesting tenant , Cloud Resource Mediation Service , the tenant which owns the resource. This is not optimal, and this paper proposes an optimized solution where in the activation of resource requires the participation of the resource sharing tenant and the mediation server alone. This optimization removes many message flows and thereby simplifies the resource sharing implementation without compromising the overall functionality or the security.

### a) Optimization of Activation Message

The activation is the process of making the shared resource available to other users. The policy details selected during the activation process determines the group of users with whom the resource is shared. The user is expected to login to the mediation server web portal to perform the activation operation. The unique security key is made available to the user as part of account creation. The account creation is not detailed in this paper since it is not relevant in the context of this paper. The prototype is done by using the file as a shared resource.

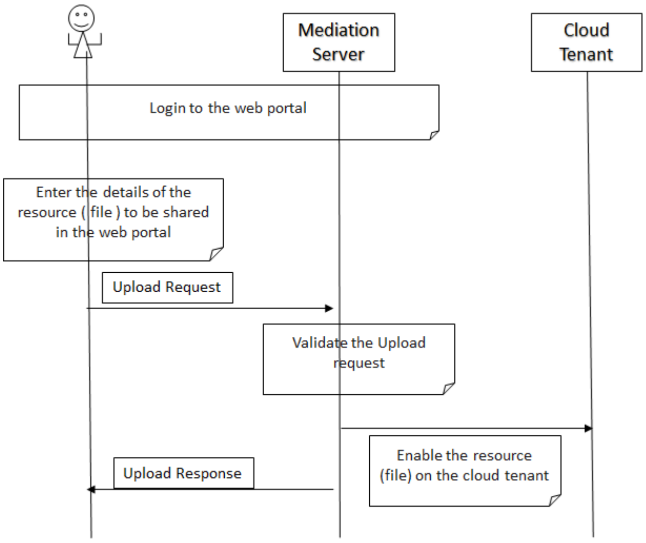
Mediation Server
Cloud Tenant
Login to the web portal
Enter the details of the resource ( file ) to be shared in the web portal
Upload Request
Validate the Upload request
Enable the resource (file) on the cloud tenant
Upload Response

Figure 3 Activation of resource

Message flow associated with the Activation of the shared resource is as shown in fig [3] and the corresponding explanation is detailed below.

1) User logs into the mediation server web portal using username, password and the unique security key.
2) User Selects
   a. The file to be shared
   b. File access policy.
   c. And enters the file access key, which is used to encrypt the file.

3) The web portal sends the details of the user name, file name, file access key, file access policy details to the mediation server.
4) Mediation server on receiving the activation request fetches the name of the file, file access policy, file access key and the username.
5) Mediation server uses the file access key to encrypt the file using AES encryption.
6) Mediation server uses the username to identify the associated tenant.
7) Based on the tenant type mediation server uses appropriate APIs to upload the file to the cloud file server.
8) Mediation server stores the values to identify the file and its ownership into the internal database to identify the file.

### *b) Optimized Deactivation Message Flow*

Deactivation is a process of removing the shared resource from the shared access list. Post deactivation the resources are not available for user to access it. The deactivation is achieved by setting the deactivation timer to current time. So this would reuse the implementation done to set the timer value of the activated resource.

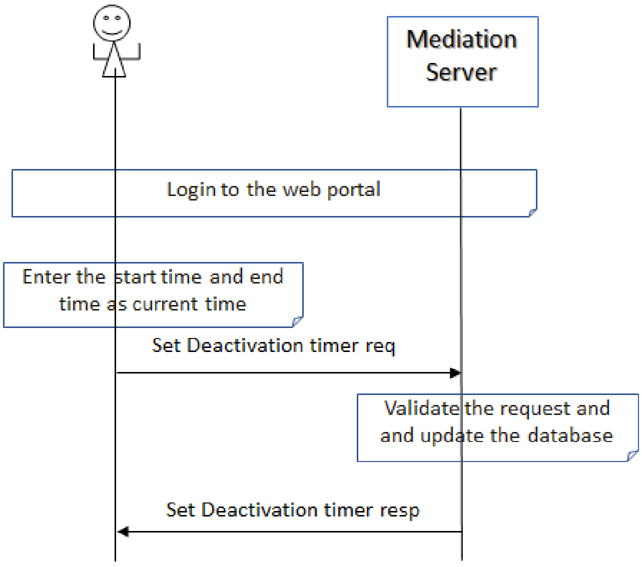
Mediation Server
Login to the web portal
Enter the start time and end time as current time
Set Deactivation timer req
Validate the request and and update the database
Set Deactivation timer resp

Figure 4 Deactivation of shared resource

The message flow associated with the deactivation is as shown in fig [4] and the detailed explanation is documented below.

1) The user logs on to the web portal.
2) The user selects the menu to set the timer associated the shared resource
3) User sets the start time and end time as the current time.
4) The web client sends the shared resource name, start time and end time details (equal to current time) to the web server running on the mediation server.
5) The mediation server updates the database with the time start time and end time details (as current time).
6) The resource is no longer active since the activation timer has elapsed

### *c) Security Design improvements*

Further to the improvement identified above, additional security improvements were made as part of the thesis in the following areas

1. Security key during account creation.
2. Key to access the shared resources

As part of account creation procedure, a security key is generated and shared only with the person who created the account via a secure email access. This security key is necessary to perform subsequent login to the system. This brings in place sort of 2-way authentication mechanism while logging into the system.

When the resource is shared by a user a unique key is generated against the shared resource. This key is shared with the set of users who can access the shared resources. This key is required while accessing the shared resource and is passed by the requesting tenant to the mediation server. This approach enhances the security of the shared resources and helps to prevent the security breaches.

## IV. Conclusion

Security aspect should be considered with great importance while designing any system. Even if the system has a well-built mathematically proven principles incorporated in the design, it is imperative to analyze the system design considering the security aspect. The security threat modelling should be applied while designing the system to find the security weakness in the system and appropriate solution should be applied where ever applicable.

## Acknowledgment

I thank my professor and guide Dr P Venkataratnam for the encouragement and constant support to pursue this research.

## REFERENCES

[1] Quratulain. Alam, S. U. R. Malik, A. Akhunzada, K. K. R. Choo, S. Tabbasum, M. Alam, "A Cross Tenant Access Control (CTAC) Model for Cloud Computing: Formal Specification and Verification", IEEE Transactions on Information Forensics and Security, vol. 12, no. 6, 2017, pp. 1259–1268.
[2] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed NIST standard for role-based access control. ACM Trans. on Information and System Security (TISSEC), 4(3):224–274, Aug. 2001.
[3] J. McKenty. Nebula's implementation of role based access control (RBAC).
[4] http://nebula.nasa.gov/blog/2010/06/03/nebulas-implementation-role-based-access-control-rbac/,2010.
[5] A. Sayler, E. Keller, and D. Grunwald, "Jobber: Automating inter-tenant trust in the cloud," presented at the 5th USENIX Workshop Hot Topics Cloud Comput., Jun. 2013.
[6] Single Sign-On with SAML on force.com http://wiki.developerforce.com/page/Single Sign-On with SAML on Force.com, 2013.
[7] R. F. Chong. Designing a database for multi-tenancy on the cloud. https://www.ibm.com/developerworks/data/library/techarticle/dm201dbdesigncloud/index.html
[8] F. Chong, G. Carraro, and R. Wolter. Multi-tenant data architecture. http://ramblingsofraju.com/wp-content/uploads/2016/08/Multi-Tenant-Data-Architecture.pdf

**Mr Pramod Pillai**
MTech in Digital Electronics
Visvesvaraya Technological University -
Extension Center UTL Technologies Ltd
Bangalore , Karnataka , India