Role-Based Multi-Tenancy Access Control Scheme For Cloud

Ms. LakshmiDevi K.T.
MTech. (CSE) Scholar, laks.chinnu@gmail.com
Dept. of PG Studies, Regional Center, VTU, Mysore-570019

ABSTRACT
Today Cloud Computing is very famous technology to share the number of resources. Security and Privacy are the big issues in the Cloud Computing. Multi-tenancy is one of the important issues among the core technologies of cloud computing applications. Many tenants can access the different applications and computing resources in the same cloud server, whereas concurrent use by many users on a database or application will lead to large data volume, time consuming and security issues. Under these circumstances, it is particularly important to separate application and data for conflicts avoidance to enhance the system and data security. This paper emphasizes on the use of identity management and Role-Based Access Control (RBAC) under the multi-tenant cloud environment, so that each user is assigned one to many roles while each role is assigned to different privileges. Such privilege assignment can easily and effectively manage the access rights of users, thereby maintaining application independence and data isolation or protection while improving process performance and cloud security.

Keywords
CloudComputing; Multi-Tenancy; RBAC; RB-MTAC.

1. INTRODUCTION
To provide the security and privacy are the big challenges in Cloud Computing. A security architecture framework should be established with consideration of processes enterprise authentication and authorization, confidentiality, access control, integrity, nonrepudiation, security management, etc. In the cloud, due to multi-tenancy architecture (MTA), data from multiple clients are stored and managed by the same software[2,3]. When the software makes mistakes, potentially millions of clients may access private data of other clients. Furthermore, data stored in a cloud may be available to cloud administrators and they may access or modify data for their own benefits. The MTA has increased the security risk due to the sharing of software, data and data schemas by multiple tenants. The cloud providers are responsible for ensuring that one customer cannot break into another customer’s data and applications.

Access control is a fundamental aspect of information security that is directly tied to the primary characteristics such as confidentiality, integrity and availability[4]. Cloud computing service providers should provide the following basic functionalities from the perspective of access control: (i) Control access to the service features of the cloud based on the specified policies and the level of service purchased by the customer. (ii) Control access to a consumer’s data from other consumers in multi-tenant environments. (iii) Control access to both regular user functions and privileged administrative functions. (iv) Maintain accurate access control policy and up to date user profile information. Various access control models are in use, including the most common Mandatory Access Control (MAC), Discretionary Access Control (DAC) and Role Based Access Control (RBAC). All these models are known as identity based access control models. In all these access control models, user and resources are identified by unique names. Identification may be done directly or through roles assigned to the subjects. Now a day’s distributed systems are developed rapidly that provide virtual organization with autonomous systems.

This paper emphasizes on the use of identity management and Role-Based Access Control (RBAC) under the multi-tenant cloud environment, so that each user is assigned one to many roles while each role is assigned to different privileges. Such privilege assignment can easily and effectively manage the access rights of users, thereby maintaining application independence and data isolation or protection while improving process performance and cloud security. This paper takes into consideration the different perspectives of roles, since different users can be assigned to different role sets, which corresponds to different privileges[1]. Moreover, the different privileges allow access to different Web sites and different data in order to improve security and efficiency. Using the characteristics of different user authority could produce environmental isolation and data isolation. As distinguished from the Discretionary Access Control and Mandatory Access Control, the RBAC offers more flexibility in the granting of privileges, making it easier to manage. Users are assigned to a role and granted privileges that correspond to that role directly. Using the role-based access control simplifies the management of privileges, since the addition, deletion, query, and modification of privileges are applied to the “role” instead of the individual user.

Based on the above reasons, this paper proposes Role-Based Multi-Tenancy Access Control called RB-MTAC that integrates a set of identity management and RBAC with consideration of multi-tenant and multi-user cloud environment. The RB-MTAC method can easily assign the functions or resource with access privilege to users, in order to enhance processing performance, quality of service (QoS), and security as well as privacy on the cloud. Hence, the purposes of RB-MTAC scheme in cloud computing are follows.

1) RB-MTAC combines the identity management and role-based access control of a multi-tenancy environment in cloud computing, which is effective and simple to manage privileges that protect the security of application systems and data privacy.
2) The method provides good identity management and access control to privileges to block a non-tenant user accessing through identity management, while access control prevent tenant users without specific privilege from viewing and accessing specific applications and database.
3) The method also offers a mechanism that manages user privileges using role-based viewpoints, so administrators only need to modify role privileges to easily change user privileges, so that would reduce potential errors from constant modifications.
4) The application independence and data isolation of different tenants. Under the cloud environment, the systems

Department of Computer Science & Engineering, PESITM, Shivamogga.
and data of different tenants could be stored in the same place and that can prevent other tenants using the data and system of that tenant either intentionally or unintentionally.

2. LITERATURE REVIEW

"Alexander Lenk, Markus Klems, Jens Nimis, Stefan Tai and Thomas Sandholm[5]" "What’s Inside the Cloud? An Architectural Map of the Cloud Landscape" says that The Cloud computing stack aims at facilitating communication about different Cloud technologies and services, including placing more complex offerings such as Google App and which in turn may be provided as Cloud services themselves, however, requires a good understanding of the numerous emerging Cloud computing technologies as well as of already available services solutions offered in the open Cloud market.

"Ravi Sandhu[6]" "Future Directions in Role-Based Access Control Models" says that the reader that research on RBAC models has just begun and much interesting and challenging work remains to be done. The RBAC arena is intrinsically dominated by practical considerations and offers an opportunity for good theoretical research to be translated into practical impact on products and practice.

"David F. Ferraiolo and D. Richard Kuhn[7]" "Role-Based Access Controls" says that in many organizations in industry and civilian government, the end users do not "own" the information for which they are allowed access. For these organizations, the corporation or agency is the actual "owner" of system objects, and discretionary access control may not be appropriate. This paper proposed a definition of the requirements and access control rules for RBAC proposed in this paper could be used as the basis for a common definition of access controls based on user roles.

3. ROLE-BASED ACCESS CONTROL

RBAC model provides a powerful way to satisfy the access control needs. An access control policy is a statement which specifies the rules about how to setup the process for granting or denying authorizations to the users. The concept of role is central to RBAC. As defined by the standard, "a role is a permission to perform an operation on an object—that is, an action, function, or task that a user can invoke. The term object can refer either to information containers (such as files, directories, or database tables) or resources (such as printers, network drivers, or computers). A session is a mapping between a user and a set of assigned roles, with one-to-many user-session assignments and many-to-many session-role assignments. RBAC allows a user to activate multiple roles simultaneously in a single session, although it is not necessary to activate all roles. Sessions implement the core RBAC model, which supports many-to-many user-permission assignments job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role. RBAC’s fundamental rationale is that a role is an intermediate element between users and permissions. An RBAC implementation directly assigns users to roles (many-to-many assignments) and permissions to roles (many-to-many assignments), and thus indirectly assigns users to permissions. According to the standard, “the permissions available to the user are the permissions assigned to the roles that are currently active across all the user’s sessions.” A user is normally considered to be a human being, but it could also be a process, machine, or network. Permissions are an also part of the core RBAC model are two functions to review the set of users assigned to a given role and the set of roles assigned to a given user.

![Figure 1. RBAC Basic Models](image-url)
• roles : \( S \rightarrow 2^R \), a function mapping each session \( s_i \) to a set of roles \( \text{roles}(s_i) \subseteq \{ r | (\text{user}(s_i), r) \in \text{UA} \} \) (which can change with time) and session \( s_i \) has the permissions \( \text{U} r \in \text{roles}(s_i) \) \( \{ p | (p, r) \in \text{PA} \} \)

3.1 Three primary rules defined for RBAC
1. Role assignment: A person can exercise a permission only if the person has selected or been assigned a Role.
2. Role authorization: A person’s active role must be authorized for the person. This rule ensures that users can take on only roles for which they are authorized.
3. Permission authorization: A person can exercise permission only if the permission is authorized for the person’s active role. With rule1 and rule2, this rule ensures that users can exercise only permission for which they are authorized.

3.2 Multi-Tenancy
Multi-tenancy refers to a principle in software architecture where a single instance of the software runs on a server, serving multiple client organizations (tenants). Multi-tenancy contrasts with multi-instance architectures where separate software instances (or hardware systems) operate on behalf of different client organizations. With a multitenant architecture, a software application is designed to virtually partition its data and configuration, and each client organization works with a customized virtual application.

4. PROPOSED SCHEME
An ideal RB-MTAC method, which supports identity management and access control privilege is proposed with the design of RB-MTAC prototype scheme. Hence, this study offers identity management and role-based access control under the MTA environment, which combines the concept of multiple tenants and the characteristics of role-based access control, collectively known as the RBMTAC as depicted in Figure 2. In Figure 2, every tenant or user needs to log in with an account number and password while the system authenticates the user account through identity management. If the users are valid, the role assignment will capture the role corresponding to the user from the RB-MTAC database and assign the role and access rights, which belong to the user. The user can carry out activities, access functions and computing resources in the system through RB-MTAC.

The RB-MTAC can execute the identity management first when users log in to the application or system, then the identity is confirmed with the account number, password and other relevant authentication credentials. The user account will then enter the Role Tables to capture the role sets of the user and integrate the functional privilege within role sets through function-based access control, generating the Access Control List (ACL) for users in addition to providing encryption and privacy security services. In RBAC each user is assigned one to many roles while each role is assigned to different privileges. The illustrated examples are shown in Figures 3. There are a number of users. Each user assigned to one or more roles. Each role owning the respective function and the corresponding rules: Role A only owns the privilege to query system X and query database Z while Role B owns the privilege to query and modify Document Y and insert or modify Database Z. Roles can add or modify System X, insert or delete Document Y and delete data from Database Z.
5. RESULTS AND DISCUSSION
A RBAC model is used to reduce the complexity of system design and implementation. Compare the possible effect on the performance and the differences between the RB-MTAC method and UBAC (User-Based Access Control) system. The RBMTAC’s system management is by roles while UBAC is by individual users. Determine whether the use of different access will affect the system. The user-based systems can manage all users’ privileges in the ACL, but in the RB-MTAC, all roles’ privileges are defined by the administrator and therefore the number of roles are always smaller than the number of users. Hence, the RB-MTAC will enhance the throughput and response time, and enforce the application’s isolation and data security and privacy.

6. CONCLUSION
Access control decisions are often based on the roles individual users take on as part of an organization. RBAC can be quite effectively used by a number of organizations in order to maintain the security and stability of the organizations. The main contribution of this paper is to provide a set of privileges and the identity management scheme for corporations in cloud computing environment. The RB-MTAC scheme will obtain better processing performance and higher security than other conventional user accessing control methods.

7. ACKNOWLEDGEMENT
I would like to thank the Department of Computer Science and Engineering, Vishvesvaraya Technological University, Mysore for providing support to conduct this research work.

8. REFERENCES

[6]. Ravi Sandhu “Future Directions in Role-Based Access Control Models”.
[7]. David F. Ferraiolo and D. Richard Kuhn “Role-Based Access Controls”.