A New Advance Role Based Access Control Model to Enhance the Transactions at Cloud Server

Meena Rani, Cherry

1Dept. of CSE, Doaba Institute of Engineering & Technology, Kharar, Punjab, India
2Assistant Professor, Doaba Institute of Engineering & Technology, Kharar, Punjab, India

Abstract
In Cloud computing, RBAC enables users to carry out a wide range of authorized tasks by dynamically regulating their actions according to flexible functions, relationships and constraints. This is in contrast to conventional methods of access control, which grant or revoke user access on a rigid, object-by-object basis. In RBAC, roles can be easily created, changed or discontinued as the needs of the enterprise evolve without having to individually update the privileges for every user. In this paper, we have added the concept of GIFT, SWAP and GAIN terms in order to pass the transactions between the users so that each user can do work effectively. This design is implemented in the ASP.NET environment.

Keywords
RBAC, Authentication, Assignment, Roles, GIFT, SWAP, GAIN.

I. Introduction
Cloud computing is a rising area within the field of information technology. Cloud computing is a model for enabling convenient, on-demand network to a shared pool of configurable computing resources e.g. networks, servers, applications and services that can be instantly provisioned and released with minimum management effort or service provider interaction [1]. Cloud computing security sometimes referred as “cloud security” is a sub-domain of computer security, network security and information security [3]. It refers to a broad set of policies, technologies, and controls deployed to provide protection to data, applications and the associated infrastructure of cloud computing. Cloud security is not to be related with security software offerings that are cloud-based. Access control services [4] should be flexible enough to capture dynamic, context or attribute/credential based access requirements and ease enforcement of the principle of least privilege. Such access control services required to integrate privacy protection requirements derived from complex rules. It is important that the access control system deployed in clouds is easily managed and its access distribution is administered efficiently. It is also important to make sure that the cloud delivery models [5] provide access control interfaces for proper interoperability which demands for a policy neutral access control design and enforcement structure that can be used to address cross-domain access control issue. In this paper, RBAC technique has been proposed with addition of some terminologies like GIFT, SWAP and GAIN to solve this security issue. RBAC is method in which permission is associated with roles and users are assigned to appropriate roles. Mandatory Access Control (MAC), Discretionary Access Control (DAC) proved to be challenging for distributed systems and managing the access to resources and system become hard. Role Based Access Control (RBAC) using Reference Ontology presents a [6] RBAC model using role ontology for Multi-Tenancy architecture for specific domain. Ontology transformation algorithms are described to compare the similarities of different ontology. It provides benefit to reduce the complexity of system design and implementation.

RBAC has been shown to be particularly well suited to separation of duties requirements, which ensure that two or more people must be involved in authorizing critical operations.

Fig. 1: RBAC Model

II. Rules for RBAC
Three primary rules are defined for RBAC
1. Role Assignment: A subject can exercise permission only if the subject has selected or been assigned a role.
2. Role Authorization: A subject’s active role must be authorized for the subject. With rule 1 above, this rule ensures that users can take on only roles for which they are authorized.
3. Permission Authorization: A subject can exercise permission only if the permission is authorized for the subject’s active role. With rules 1 and 2, this rule ensures that users can exercise only permissions for which they are permitted.

Fig. 2: Representations of Role Assignments and Permissions

III. Extended version of RBAC Model
Cloud computing is becoming more and more popular nowadays, where data is outsourced into the cloud. Its advantages are obvious: relief of the burden of storage management on data owners, universal data access with independent geographical locations [7] and avoidance of capital spending on hardware, software, personnel maintenance etc. The existing architecture of the RBAC model includes the basic role based access control mechanism & some additional constrains according to which there is limitation over the number of users per role and there is also a limitation over the number of transactions in a day per role. The mechanism is quite tight; it is not flexible and does not exactly fits in the real time environment. Hence our problem becomes creating flexibility in the existing RBAC architecture and for the same purpose two things are required to be merged in the existing architecture.

A. GAIN Transactions
GAIN transaction is a concept in which if the admin feels that a thing is required to be merged in the existing architecture.
administrator can gift some of his transactions to a needier user of a particular role. Introducing this flexibility would make it easier for the admin to run the system as just for few users he would be saved from creating a new category. RBAC enables us to control at both broad and granular levels, what administrators and end-users can do. RBAC also enables you to more closely align the roles you assign users and administrators to the actual roles they hold within our organization. There are mainly two important roles admin and users. Now if user has more number of transactions then to lower the burden, admin gains his/her transaction by adding more number of transactions to his account. It can be done from lower to higher level or from higher to lower level. Admin can perform tasks like exchange of transactions, addition of transactions etc.

B. GIFT Transactions
In an organization admin create the architecture for system and then generates the roles for all users according to their privileges and also make restriction on the number of users per role. Admin also generate users’ role wise where restrictions can be made on number of transactions per day/user/hour etc. This helps to increase the security level. But in GIFT terminology, admin decides to whom he has to gift number of transactions. It is his will whether he wants to give/gift 5/10 or more number of transactions to the users. But this transaction is only one time. It means if first time he gifts 5 transactions per time then after again logging to the system, he can only gift only 5 numbers of transactions at one time. GIFT can also takes place from lower to higher level or from higher to lower level.

C. SWAP Transactions
It is of permanent nature. A new user create new account but according to restriction on number of users per role only limited number of users can create their account and because of this malicious attacks will be lesser. Already existing users can login and get access only if they are valid user and if number of transaction is less or more among users then they can SWAP their transactions in order to balance the transactions. Admin can also take the access of the SWAPPING transactions. It can only be done at lower level unlike GIFT and GAIN. [11]

IV. Implementation Framework

A. New Features for Smart & Efficient RBAC
1. To add the constraint of GIFT to make the current scheme more Efficient & flexible.
2. To add the constraint of SWAP to make the current RBAC more Efficient & flexible.
3. To add the constraint of GAIN to make user free and make the changes according to requirement by the administrator.

Many RBAC mechanisms have been proposed earlier and they had tried to enhance the security in their own way and had added so many new features in RBAC like Limit over the number of user per role [12], Limit over the number of transaction per day or per hour & Keep backup data for restoration. But they were not enough to secure the system from unauthorized users. So, the new research work will add a mechanism named GIFT transaction which will provide security at the system as well as at the server & having limited number of transactions the users were not able to exchange the transactions when it was required and a particular user was in need of them. So to overcome this problem the two more parameters named gain & Swap which are helpful in utilizing the transactions usefully according to the need [13].

The new mechanism is proposed for better security of data and three more parameters are added for better transferability of transactions for proper & proficient use. The parameters named Gift, Gain & Swap are used to regulate the transfer of transactions according to the requirement & availability. This work represents the working of smart & efficient RBAC [14] which consists of two types of modules. The modules are administrator and User. The following diagram shows the working of new extended RBAC.

B. Algorithm of GIFT and SWAP:
1. Start
2. Initialise database = d
3. Initialise ids = id
4. If id = d
5. Proceed
6. If id = d = TRUE
7. Do either SWAP or GIFT.
8. SWAP:
   Transaction of users 1 = Transaction to user 2
9. GIFT:
   If transaction of user < admin, then
   Do gift
   Pass transactions to user.
   Limit no. of transaction.
10. Systems= start
11. If valid, then do step 7.
12. Repeat Steps.

![Fig. 3: Block Diagram of Enhanced and Backup Storage Key Ontology at Cloud Server](image-url)
C. Flowchart of GIFT and SWAP

D. Algorithm of GAIN

1. Start
2. Initialise database = d
3. Initialise system = s
4. Enter admin id = id
5. do
6. Check al no. Of transactions. = t1, t2, t3......tn.
7. Do
8. Check no. Of users = u1, u2, u3......un.
9. If
10. User n transaction > user (n-1) transaction
11. Validate transaction
12. If transaction= TRUE
13. Do
14. Gain transactions from user (n)
15. Success.

E. Flowchart of GAIN:

V. Results and Discussion

This proposed model is compared with the current RBAC and showing the results in Fig.1 & Fig. 2 and it concludes that this new smart & efficient RBAC architecture having better results. It means security level can be enhanced or improved with the help of this new architecture. This new architecture increase the security by using RBAC mechanism. Three new features are added into this architecture and that is GIFT, GAIN & SWAP which makes the current RBAC more efficient & fast.

Table 1: Performance Table of Two Different RBAC Approaches

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Previous RBAC Performance Percentage</th>
<th>Current RBAC Performance Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Based Constrains</td>
<td>100 %</td>
<td>100 %</td>
</tr>
<tr>
<td>Adding Constrains</td>
<td>80 %</td>
<td>88 %</td>
</tr>
<tr>
<td>Restriction to Transaction</td>
<td>80 %</td>
<td>89 %</td>
</tr>
<tr>
<td>Backup Policy</td>
<td>70 %</td>
<td>89 %</td>
</tr>
<tr>
<td>Accuracy of Data</td>
<td>72 %</td>
<td>96 %</td>
</tr>
</tbody>
</table>
another aspect of RBAC that distinguishes it from roles and permissions are connected only to roles not directly to users in a group-based mechanism with both users and the groups to which they belong. The ability to tie permissions directly to users in a group-based mechanism can be regarded as a “loophole” that makes it difficult to control access. Within an organization, roles are relatively stable, while users and permissions are both numerous and may change rapidly. There is a superficial similarity between RBAC roles and traditional groups. As normally implemented, a group is a collection of users, rather than a collection of permissions and permissions can be associated with both users and the groups to which they belong. The ability to tie permissions directly to users in a group-based mechanism can be regarded as a “loophole” that makes it difficult to control user-permission relationships. RBAC requires all access through roles and permissions are connected only to roles not directly to users. Another aspect of RBAC that distinguishes it from traditional group mechanisms is the concept of a session which allows activation of a subset of roles assigned to a user. Core RBAC [16] includes those systems with a robust group mechanism that supports the construction of a many-to-many relation among users and permissions. The challenges of cloud computing require the design of authorization and access control schemes. In this thesis, we classify the design goals of authorization and access control schemes. In this thesis, we classify the design goals of authorization and access control schemes and define a system model that fulfills the design goals under which a distributed scheme is proposed. Then, we analyze the security of our scheme by checking their fulfillment of our design goals. Finally, we evaluate the efficiency of our scheme theoretically and experimentally. The results show that our scheme is quite improved because it can reduce the burden on the authorization server by separating the authorization process from the access control process and releasing the authorization server from the access control process. The future scope lies in the diffusion of the RBAC technology with the static and dynamic analysis of SQL parse trees in order to maximise the advantages of each method and to ensure that high prevention of attacks is provided in case the any of the method fails. One method to gain unauthorized access to data is to insert a tautology into the query. In SQL, if the where clause of a SELECT or UPDATE statement is disjointed with a tautology, then every row in the database table is included in the result set.

**References**


