EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Efficient and Flexible Role-Based Access Control (EF-RBAC) Mechanism for Cloud

Shilpi Harnal[1,*], R.K. Chauhan[1]

[1]Department of Computer Science and Application, Kurukshetra University, Kurukshetra, India

Abstract

The cloud reduces the user's burden to many folds. But cloud providers and cloud users with dynamic relationship, are in distinct security domains. Amongst various challenges with cloud, the crucial one is to detect and protect the user's data from unauthorized accesses. In cloud, users are not legendary by their predefined identities. Instead, they are providing accesses based on their characteristics and attributes. This work is focusing on available access control mechanisms and one that applicable for cloud environment. The paper also proposes an Efficient and Flexible Role-Based Access Control (EF-RBAC) mechanism for the cloud computing environment to achieve confidentiality and security. RBAC limits the accesses for resources within an organization to authorized users only and also guarantees that a user can solely access specific information they are authorized for by the organization policy. The proposed scheme adds flexibility to the RBAC for better cloud user's experience.

Keywords: Cloud Computing, Access control, Role-Based Access Control, Security, DDOS, Multimedia, Information Security.

Received on 08 September 2019, accepted on 01 November 2019, published on 18 November 2019

doi: 10.4108/eai.13-7-2018.161438

1. Introduction

Cloud computing creates several computing resources (computing centers, huge data centers, etc.) to work in a collaborative network system over the internet. Also, a secure, huge and fast network of data storage and computing is supported by the cloud for all kinds of users [1]. The three major cloud service delivery models are Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS).

Although the cloud is providing numerous benefits but the main concern associated with both ends (i.e. consumer and provider) is security [2]. The consumer's concern is about maintaining the secrecy of their private data stored over the cloud servers [28]. However, the cloud provider's concern is that their services and resources remain accessible to authorized persons only. Thus, without proper access control policies, the cloud is always vulnerable to various threats and attacks. Organizations hesitate to adopt cloud for transfer of their multimedia storage and computations to cloud servers as they are not sure that whether they are 100% secure or not. As any access breach to user's personal multimedia data can be a matter of stake for them. In nutshell, the confidentiality and integrity of the user's data stored at the third party public cloud servers must not be compromised in any case [30]. Because of this, an effective mechanism for access control and management can play a crucial role as it is directly linked with the primary required characteristics those are authorization, confidentiality, availability, and integrity.

Cloud providers must ensure the basic functionalities for controlling unauthorized accesses or in other words providing secure access to the services based on the service level agreement (SLA), protecting access of one user's data from other users, providing a consistent state to consumer always, controlling users accesses based on their earlier defined privileges i.e. effectively maintaining and managing access control rules etc.

Traditionally access control mechanisms are of the following three categories:

*Corresponding Author: shilpi13n@gmail.com

- Discretionary access control (DAC): In Discretionary access control (DAC) mechanism the owner of the object is responsible for deciding access rights for other users like in UNIX/LINUX operating system the file's owner decides all the permissions (read/write/execute) for users in same group or for users in other groups. The DAC model is applicable for small applications only, as it incurs the overhead of management for bigger applications having multiple users like in distributed systems.
- Mandatory access control (MAC): For distributed systems, Mandatory access control (MAC) models are more adaptable as they effectively map the users and the resources available. Generally, MAC models are applicable for multi-level security systems where the administrator is responsible for deciding the access rights for users at different security levels. There are different classifications of objects at different levels such as Top Secret, Secret, Classified and Unclassified. Bell LaPadula (that recommends the "no-write-down" and "no-read-up" rules for data access to maintain confidentiality) and the Biba Model (that recommends the "no-write-up", "no-read-down" and "no-execute-up-or-down" rules for maintaining the integrity of data) are exemplified models of such type.
- Role-based access control (RBAC): In RBAC every user is assigned a specific role that supports their job requirements within the organization and access is provided for numerous objects accordingly [33]. The organization's authorities define the permissions and accesses for every role. Likewise, multiple roles can be linked to a single person on a required basis. These kinds of access models are additional, versatile and scalable than the opposite varieties.

The users and cloud providers come under totally different categories of security domains [32]. Conjointly their relationship is dynamic in practicality. That's why RBAC is more suitable for open cloud environments as users are tracked by their roles of access instead of fixed identities [3]. The above two schemes DAC and MAC are not suitable for the cloud environment where all the resource nodes may not be known to each other. For example, a SaaS level user can access the cloud services by numerous means like laptops, notebooks, mobile phones, etc. through the internet. Hence, users are identified on the basis of their characteristics instead of fixed IP addresses. No filtering of packets is possible with traditional firewalls on the basis of fixed IP addresses. Thus any dynamic, as well as cross-domain system can manage access control in cloud computing environment. This work focuses on the requirements of role-based access control method for the cloud along with a brief review of existing work in this area. The further sections intend to propose an efficient and flexible Role-based access control scheme for the dynamic cloud environment and identify future research directions.

2. Requirements of RBAC

In Role-based Access Control (RBAC) model user's roles such as administrator, end user, specialist user or customer etc. are based on various factors such as their designation, authorization, job competency and responsibility [4]. Through RBAC employees can only access the information that they really require for their job to be done [31]. They are inhibited from accessing any other data/information from the system. In addition, the accessible resources can be restricted to certain tasks, such as the ability to read, update or create files [5]. Here, user/subject is a person or system that interacts with the system, access defines the specific type of interaction that allow the flow of information between a user and an object, Access control limits the access of an authorized user, process, program or other systems for an object and role is a job description that defines the assigned responsibilities and authorities for an organization [6] [7].

Every big organization with several employees needs to limit their access for various objects to secure their critical data and applications. There are also requirements to limit accesses of third parties, like venders and customers. For the implementation of RBAC, an organization should follow a number of practices such as determining and list the critical resources for which control access is required, identifying the roles having the same access needs based on their work pattern. After this identification, the organization should bind the employees to specific roles and set their access rights.

The following are some requirements analyzed for implementation of Role-based access control mechanism for cloud computing [8]:

- *Authentication:* Authentication is always imperative for every access control systems [9]. Cloud suppliers need some reliable system to attest and authenticate users. In this regard, the access control system helps by keeping track of count, time and location of every access by the user.
- *Trust:* An efficient access control system maintains a trustworthy relationship between the cloud provider and the users. This trusted behavior can greatly influence and attract cloud customers [1].
- *Quality of Service:* Maintenance of the response time and monitoring of computational complexity of system is done by the system, having capability of controlling all the accesses of system and also capable of maintaining Quality of Service (QoS) for various end users. Implementation of control rules for computational complexity is still an arduous task for any access control system. [10]. Access control systems also manage the response time of every access according to organization's requirements [11]. These timely responses help to evaluate the system's performance.
- *Mobility Services and Dynamic Features:* Cloud system is secure, scalable, measurable and dynamic by practice. Along with the flexibility of configurations, it also handles remote and any time access by its users

[12]. Thus it is the responsibility of access management mechanism to handle these mobility features and dynamic necessities of customers for the smooth functioning of system [13].

- *Access policy management:* As the cloud has dynamic behavior, access control policies such as update, insert, delete, export and import need to be flexible enough to adapt itself with changing requirements of the customers. Access control policy should be capable enough to deal with such conflicts [14].
- *Situational and operational awareness:* Other factors such as the functioning of memory, processor, operating system and of other components are also tracked by the access control system, as these operational components can situationally affect the access decisions and performance of the system [4].
- *Sharing and Virtualization of resources:* It is the cloud virtualization that enables the sharing of physical resources. As the demands and requirements of users can change any time, it is the responsibility of the access control system to manage these needs and restrict any unauthorized information to flow [15].
- *Auditing:* Auditing is the indispensable functioning of each cloud. Access control system plays an important role in the auditing of a system by observing the current state of the system, recording any failures or aborts, reporting of any attempt to violate the access rules if any. Another basic functionality and responsibility of the system are to maintain logs of all the transactions performed and changes applied to any objects such as copying, moving, renaming, erasing, etc. [16].
- *Inter-operability and Migration:* Every cloud customer is provided with the in step services on the basis of their usage and demand. Different cloud providers might follow different access management policies. This diversity in access policy can raise problems if any movement from one service provider to another is required or integration is required within two service providers [17]. The authors in [18] have proposed an XML based migration scheme that can help users to easily shift a database schema secured with RBAC to another provider.
- *Applying Privileges:* Ease of assigning, changing and revoking privileges for an object is one of the main aspects of any access control system. The access control system should handle all these requirements for maintaining the usability of a system [12].
- *Verifying, Testing and Updating access control functions:* Verifying, testing and updating access control functions are vital and integral for keeping the system updated and maintain the new security levels as needed from time to time [4]. Thus, timely review and updates of access control functions help to handle future changes and analyze the impacts of activating, deactivating or changing any policies.
- *Delegation of accesses and capabilities:* In cloud environment where users are not recognized by their identities and they are just concerned about the fulfillment of their general tasks, a flexible access control system with dynamic resource management capability plays a vital role in delegation of accesses and capabilities for various users based on their roles in the organization.

3. Related work

Role-based access control (RBAC) scheme, assigns roles to users based on their least privileges and functional requirement to perform a job. Goyal et al., [19] said that Task Role-based access control model (TRBAC) is considered as a viable scheme for the cloud computing environment. According to Bethencourt et al. [20] the traditional access control schemes such as mandatory or discretionary access control models cannot be applicable for an open cloud environment. According to authors, in TRBAC access permissions can be validated dynamically based on the user's role and the task assigned to the user. They have also categorized tasks into two types, one that needs a proper order of execution and other tasks that can be completed in any order. Access permissions are also reassigned dynamically based on the prerequisite of each task and the order of execution of various tasks. Another proposed variant for cloud computing by Yang and Jia [21] is the Attribute-role-based access control (ARBAC) model. For this scheme, data objects are assigned with some attributes and values. To access these objects' attributes, the user has to provide that particular value. Access is provided by the cloud server only once the validation phase is complete. The Ristenpart et al., [22] have extended this by proposing a fine-grained key based ARBAC model with the provision of preserving the privacy of the attribute's values corresponding to an object using the symmetric/private key encryption schemes to protect its privacy. The Shafiq et al., [23] has suggested that certain roles should be fixed and static in some applications, while permissions and users for roles might be assigned dynamically. The Ruj et al., [24] has proposed the involvement of a certified third party for assigning roles to users. They have also proposed to inculcate certain parameters (such as all possible timings and locations of access) to each user's profile to maintain the trust/authentication of users. Thus, the crucial problem of securing outsourced sensitive information of authentic users can only be managed with a trusted access control service [29].

4. Proposed scheme

One of the most required key phases for cloud computing security is Managing and controlling the accesses. As discussed access control helps to maintain authenticity, trust, quality of service (QoS) and quality of experience (QoE) for cloud users. It also helps in auditing and testing of different accesses to the cloud. While considering access control mechanisms for an open cloud

environment, it is proved that as compared to ordinary discretionary access control (DAC) models and mandatory access control (MAC) models, Role-based access control (RBAC) model is more suitable, flexible and scalable. Hence, it is recommended to use RBAC with some extend features for cloud environments. Systems users are dynamically provided with some roles as needed. Accesses/Permissions are applied for the specific roles and the same permissions automatically get applied to all the users of that category. The roles and their respected users are restricted with some constraints such as each user is provided with a limited number of transactions in a day based on their role. If the user's transaction count reaches the predefined threshold value, then the server will automatically stop listening from that user. Some RBAC models also limit the number of users per role. Based on these constraints various models of RBAC are presented below and also table 1 presents summarized information for them [25] [26]:

Table 1. Different RBAC models

Feature	RBAC 0	RBAC 1	RBAC 2	RBAC 3
Role Creation	Yes	Yes	Yes	Yes
User Creation	Yes	Yes	Yes	Yes
Restriction Over User Limit	No	Yes	No	Yes
Restriction Over Transactions of Users	No	No	Yes	Yes

- RBAC 0: It is the simplest model of a role-based access control system in which there is no limit over the number of users per role and number of transactions per user. The roles are directly coupled with certain constraints to be followed strictly.
- RBAC 1: This model is the same as RBAC 0, in which all constraints are directly applicable over the role and there is no limit for the number of users per role. But it added a limit over the number of transactions per user.
- RBAC 2: This model is the same as RBAC 1, by linking specific constraints over the roles. But it added a limit over the number of users per role.
- RBAC 3: In this model, all the roles are bounded with the constraints. However, a count and limit over the transactions is always maintained on role basis and also abound over the total number of users per role is also maintained. This model has performed effectively on the potency ground.

Some primary rules required for RBAC are: A person will get access only if he/she has been allotted a role and users ought to follow only allotted roles and permissions for which they are authorized. In this paper, we have proposed to use Role-Based Access Control (RBAC) for ordinary cloud storage and multimedia cloud storage [27] with some variations to make it more efficient and flexible. We have applied the RBAC 3 model for our work with a limit over the number of transactions per user in a day and a limit over the number of users per role in an organization.

The proposed scheme has added provision of gifting/borrowing transactions to/from one another (within authenticated registered users having the same role only). So that if some users does not need more number of transactions in a day, he/she can borrow or gift his transactions with other users of same role to implement security mechanism against the roles of the system. Role-based access control (RBAC) is a methodology of proscribing network accesses, which supports only the roles within an organization for individual users. Consequently, the number of requests to the server decreases and encompasses a limit, as each user has a count on its access to cloud servers. This leads to improved/reduced response time and decreases the overhead of servers. Additionally, it provides prevention against distributed denial of service (DDoS) attacks. Once the allotted limit exceeds, either the user gets blocked or the server stops listening from that user. The user can request the transaction from another user only when their limit is over. And only an authentic user with the same privileges will be able to gift its remaining transactions to other authentic users having the same privileges.

The general model for the proposed scheme (EF-

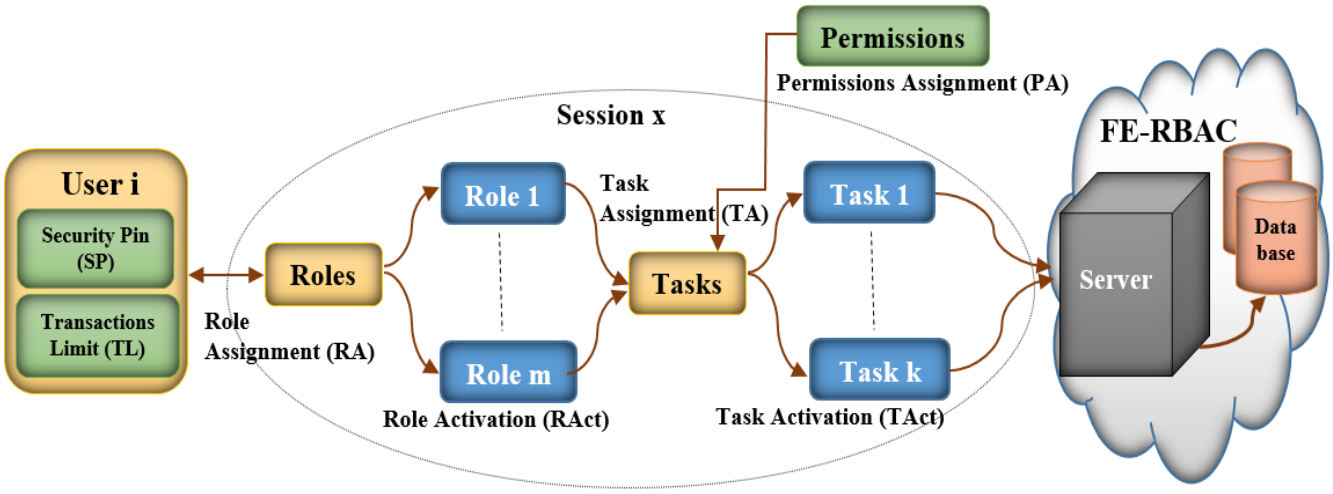

Figure 1. Efficient and Flexible Role-Based Access Control (EF-RBAC)

RBAC) is shown in figure 1 with the following properties:

- Multiple active sessions are possible at a constant time. However, there is just one active session for one active user at a time.
- Each user entered in the organization is assigned with a role (Multiple roles can be allocated to a user similarly) and each role has a limit over the number of transactions that can be used.
- Among the numerous roles outlined for an organization, every role has some predefined rules, attributes, and transaction limits.
- Every role has specific a number of tasks and every task is associated with certain permissions specific to roles.
- Each user manages a personal security PIN number and also keeps the record of his/her transactions used till now.
- Once transactions limit exceeds for a user, if the user 'A' makes a request for transaction from another user 'B' having same role, firstly user 'A' will have to seek permission from user 'B' for his/her secure one time security PIN (Because user 'A' is required to provide security PIN of user 'B' before accessing transaction from user B's account). After the transaction of 'A' is over, the secure PIN is autogenerated again.
- Once this limit exceeds (e.g. of user 'A'), either the user can request a transaction from another user with the same role or the server stops taking further requests from that user's side for that specific role.
- Further if any suspicious behavior is detected like a user is trying to make repetitive attempts after his limit is exceeded or if the user is trying to access a file that he/she is not authorized to access or if the user is trying to make an unauthorized update etc., then the Role-Based access control mechanism will analyze and report the attempt to authorities.

4.1. Essential components

For secure access to the multimedia cloud, a number of concepts and essential components are utilized by the EF-RBAC. The efficient and flexible role-based access control model (EF-RBAC) has the following components:

- Us: is a set $\{u_1,\ldots,u_n\}$ of N users
- Ro: is a set $\{r_1,\ldots, r_m\}$ of M roles
- Ts: is a set $\{t_1,\ldots, t_k\}$ of K tasks
- Ss: is a set of current sessions
- Ps: is a set of permissions (Read (R), Write (W), Execute (E) and Delete (D))
- Db: is a set of databases
- Every user $u_i \in$ Us can have multiple roles $\{r_1, r_i,\ldots,r_i\}\subseteq$Ro and each role can have a defined number of tasks $\{t_1, t_i,\ldots,t_j\}\subseteq$Ts. Every task $t_j \in$ Ts is assigned with allotted permissions $\{p_1, p_i,\ldots, pj\}\subseteq$Ps to accomplish the task.
- Most of the relationships in the model are many to many (Us X Ro, Ro X Ts and Ts X Ps) except users to sessions (Us → Ss), as users to sessions is one to one relationship.
- RA: Role Assignment, is many to many mapping between Us and Ro for Users-to-Roles assignment, i.e. RA ⊆ Us X Ro
- TA: Task Assignment, is many to many mapping between Ro and Ts for Roles-to-Tasks assignment, i.e. TA ⊆ Ro X Ts
- PA: Permission Assignment, this is for many to many mapping between Ps and Ts for Permissions-to-Tasks assignment in the model, i.e. PA ⊆ Ts X Ps
- TL: Total Limit, number of transactions that a user can access in a day. After that either the user can request a transaction from another user with the same role by requesting his/her security PIN or the user gets blocked for that particular role for that day.
- SP: User specific auto-generated security PIN. This PIN is used when a user's limit exceeds and he/she makes a request for a transaction from another user as discussed above.
- Ract and Tact: are the sets for currently active roles and tasks respectively.
- Least privilege policy: The proposed scheme follows the least privilege policy that is the policy of granting a user with the only needed roles and permissions to accomplish its task, i.e.:
  $\forall u \in$ Us $\rightarrow\{r_1,\ldots, r_m\}$, $\forall r \in$ Ro $\rightarrow\{t_1,\ldots, t_k\}$,
  $\forall t \in$Ts $\rightarrow\{P_r, P_w, P_x, P_d\}$
  u is assigned with roles which has to perform tasks and tasks need permissions (P_r: read, P_w: write, P_x: execute, P_d: delete) for the user to accomplish the task.
- Delegation of tasks: This can create flexible and versatile functioning for the cloud system. This can be attainable inside users of the same role solely. This means that if a user is busy with another task then the assigned task can be delegated to a different user with a similar role.
  i.e.: $\{u_i, u_j \in$ Us, $t_x \in$ Ts, $r_y \in$ Ro$\}$
- Separation of duties: It states that partitioning of permissions specific to tasks, portioning of tasks specific to roles and partitioning of roles according to users to prevent conflicts.

If Transactions_Used[i] reaches the maximum limit of the day for User[i] i.e. Transactions_Limit[i], then the User[i] can exit or has the option of borrowing transaction from other users having same role/privilege or lower privilege. All the responses returned after the Cloudlets execution by the Virtual Machines includes different parameters like the status of execution (success or failure), CPU time taken, simulation start time, simulation end time, cloudlet id and VM id. Hence, the algorithm i.e. EF-RBAC is providing flexibility and efficiency to the ordinary Role-Based Access Control Model.

4.2. Methodology applied

We have implemented proposed Efficient and Flexible Role-Based Authentication Control (EF-RBAC) over the

cloud using CloudSim (version 3.0) simulator tool to limit the number of accesses a user can have in a day and to add the provision of gifting or borrowing transactions from other users. Cloudsim tool provides an extensible and generalized seamless modeling framework. Cloudsim provides users with cloud-based services and an infrastructural environment to work with. Here all tasks are named as Cloudlets. DataCenter provides all the computational resources and manages all the Virtual Machines (VMs) running over different Hosts. DataCenter Broker acts as an intermediator between the cloud service provider and the users. CloudSim Simulator's general model with DataCenters, Brokers, Cloudlets, VMs, etc. is shown in figure 2.

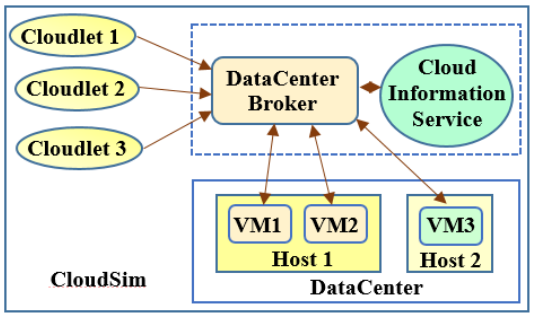

Figure 2. CloudSim Simulator's Framework

We have implemented proposed Efficient and Flexible Role-Based Authentication Control (EF-RBAC) over the cloud using CloudSim (version 3.0) simulator tool to limit the number of accesses a user can have in a day and to add the provision of gifting or borrowing transactions from other users. Cloudsim tool provides an extensible and generalized seamless modeling framework. Cloudsim provides users with cloud-based services and an infrastructural environment to work with. Here all tasks are named as Cloudlets. DataCenter provides all the computational resources and manages all the Virtual Machines (VMs) running over different Hosts. DataCenter Broker acts as an intermediator between the cloud service provider and the users. CloudSim Simulator's general model with DataCenters, Brokers, Cloudlets, VMs, etc. is shown in figure 2.

For the proposed mechanism if the user is trying to execute a transaction within his/her specified limit and role then the task or Cloudlet is submitted to the DataCenter Broker. The broker then assigns the Cloudlet to an available Virtual Machine (VM) at a Host of DataCenter for execution. The step wise algorithm for the procedure discussed above is defined as follows:

1. Start
2. Create M Roles
3. Create N Users
4. Limits the transactions (T) for each Role
5. Assign Role to users
6. Set Transactions_Limit for each user
7. Initially set Transactions_Used = 0 for each user
8. Start Simulation by submitting array of VMs to DataCenter Broker
9. Repeat While (User i has some task to execute)

```
{
If (Transactions_Used[i] < Transactions_Limit[i] AND
        User[i] has task to execute)
{
      -Submit Cloudlet to Broker
      -Broker assigns Cloudlet to available VM
      -VM executes the Cloudlet and send response
      -Set:
      Transactions_Used[i] = Transactions_Used[i]+1
[End of If]
}
Else If (Transactions_Used[i] = Transactions_Limit[i]
        AND User[i] has task to execute)
{
      Notify the User about the Limit Exceed
      If (User wants to request Transaction from j User)
      {
          -Make a request for security PIN to user j
          -Submit the security_PIN[j] and Cloudlet to
          Broker
          -Broker assigns Cloudlet to available VM
          -VM executes the Cloudlet and send response
          -Set:
         Transactions_Used[j] =Transactions_Used[j]+1
          -Auto-generate security PIN[j] for user j
      } [End of If]
} [End of If]

Else If (Transactions_Used[i] = Transactions_Limit[i]
        AND User[i] has no task to execute)
{
        -Notify the User about the Limit Exceed
} [End of If]
        -Notify the server about users limit, where
        N=Total number of users
} [End of While]
```

10. End of Simulation

The flow/activity chart given in Figure 3 depicts the workflow and sequence diagram given in Figure 4 depicts the sequence of events for the above discussed algorithm. Thus it is clear from the algorithm that Simulation starts by initializing DataCenter, Broker, Hosts and Virtual Machines (VMs). Transactions_Used[i] array stores the number of transactions executed by user 'i' and Transactions_Limit[i] array stores the maximum number of transactions assigned to user 'i'. The simulation runs until any User 'i' is having some tasks or Cloudlet to execute. Transactions_Used[i] is incremented by one each time User[i] executes a Cloudlet.

If Transactions_Used[i] reaches the maximum limit of the day for User[i] i.e. Transactions_Limit[i], then the User[i] can exit or has the option of borrowing transaction from other users having same role/privilege or lower privilege. All the responses returned after the Cloudlets execution by the Virtual Machines includes different

parameters like the status of execution (success or failure), CPU time taken, simulation start time, simulation end time, cloudlet id and VM id. Hence, the algorithm i.e. EF-RBAC is providing flexibility and efficiency to the ordinary Role-Based Access Control Model.

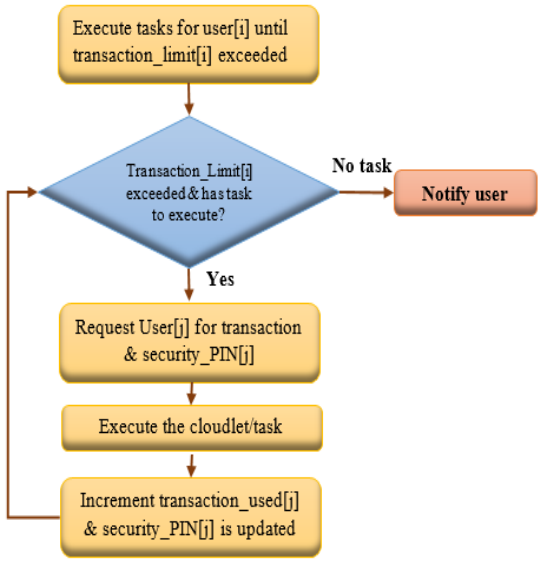

Figure 3. EF-RBAC Algorithm's working with Flow

5. Case study: Netflix cloud management system

A case study can describe the actual experience of users/organizations with a cloud server. Here, we are considering a qualitative scenario related to multimedia content management for the Netflix cloud server for a better understanding of the above proposed model. Netflix is an American company for providing and producing media streaming services as per subscriber's monthly subscription packages worldwide. Their rental plans vary on the basis of the number of screens it offers, the device to be used for watching such as smart phone, laptop, television, tablet and also on the basis of the quality of video required i.e. HD or non-HD. The lowest plan offers streaming over a single device but the highest plan offers a maximum of four streaming. These plans are very expensive and not much flexible to suit everyone's needs.

5.1. Challenges with ordinary systems

There are a number of challenges or issues related to service provisioning, access or security are also associated with Netflix. Such as:

- Country wise stuff provided to users. This means Indian viewers can't view other country's shows and vice versa.
- It is not possible for a user to subscribe for selected shows/series or for limited hours per day subscription, to reduce the costing.
- The main issue is the sharing of Netflix passwords. According to court, any such instance is illegal and violation of rules. But as they are offering premium plans with four parallel HD streaming, users are into this plan by sharing the password of a single account. This can compromise the private details of a user.

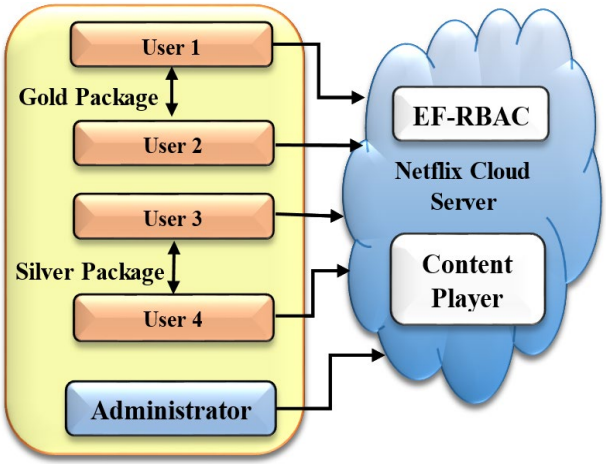

Figure 5. Netflix Cloud Management System

5.2. Case Study: Proposed Solution for Netflix

The challenges discussed in the last section are related to the current access policies of Netflix and can be easily managed through the proposed solution (EF-RBAC) as shown in Figure 5. Here, we are considering only two entities in the system for the sake of simplicity, those are Administrator and Users/Subscribers. Description:

- Roles defined are: User and Administrator.
- Packages defined: Gold and Silver. Multiple packages can also be added.
- Packages are defined flexibly according to the user's requirements such as the number of hours/videos per day or number of shows/series subscribed.
- Assume users with the Gold package can view ten videos and users with the silver package can view five videos per day. This can greatly help to reduce/limit the server's load and also in reducing the subscription cost.
- As stated in EF-RBAC, if a user's video limit for a day is exhausted, they can borrow it from another user if required, by simply sharing a Secure PIN. Otherwise, they are blocked for that day. So, the users are not required to share their account's password to please anyone.
- Suppose User 3 got her limit exceeded for a day, then she can request a transaction from User 4. If User 4 agrees, she will give her secure PIN to User 3. The secure PIN at User 4 will be auto generated after usage

Table 2. EF-RBAC against ordinary methods of access

S. No	Comparison Property	DAC	MAC	RBAC	TRBAC	ARBAC	EF-RBAC
1	Least privilege principle	N	N	Y	Y	N	Y
2	Delegation of duties	Y	N	N	N	N	Y
3	Separation of duties	N	N	Y	Y	N	Y
4	Auditing	Y	Y	Y	Y	Y	Y
5	Configurational flexibility	N	N	Y	Y	N	Y
6	Situational and Operational analysis	N	N	N	Y	Y	Y
7	Handling of heterogeneity	N	N	N	N	N	Y
8	Scalability	N	N	Y	Y	N	Y
9	Limiting the accesses	N	N	N	N	N	Y
10	Flexibility to request transaction from other user	N	N	N	N	N	Y

of the previous one. After that User 3 can view video from her own account.

- Thus, the user borrowing transaction, need not to login with another user's ID.
- Thus, the EF-RBAC scheme results in limited and predictable load for the server and also reducing rates of packages as per the user's requirements. This also helps to protect one's credentials, as for sharing a transaction users need not to share their login details.

6. Analysis, benefits & limitations of proposed scheme

6.1. Analysis

The EF-RBAC scheme is providing flexible, efficient and dynamic functionality like modification of permissions assigned for various tasks, updating of specific tasks for particular roles and also updating of roles specific to various users can be done as per the requirements of currently active tasks of the organization. Furthermore, by imposing a limit over the number of transactions specific to each user and role, this scheme is providing an upper bound for the server's load. Also, it is flexible enough by providing transaction requesting facility among users with the same role if one's limit exceeds.

If the organization has N number of users and T number of transactions, the worst case complexity of the algorithm for looping through all the N users for the execution of T tasks will be O(N*T). Thus, a server needs to handle a fixed number of tasks daily, unlike ordinary schemes with no limit over the server load. Also, table 2 is presenting the functionality analysis of EF-RBAC against ordinary methods of access control over various parameters.

6.2. Benefits

For the proposed scheme users are assigned with roles based on the least required privileges policy for an object. Every access is tracked and any practice of unauthorized access is also captured. In nutshell, we can say that this scheme is scalable, dynamic and support active and passive workflow in the system. Apart from these, the proposed mechanism can also provide the following benefits:

- *Protection against Distributed Denial of Service Attacks (DDoS):* With limited accesses scheme this will also provide protection against distributed denial of service (DDoS) attacks. As no attacker can barr the services with limited accesses.
- *Decreased threats on the server:* It will minimize the risk of information access by the intruders by limiting the accesses for users. Also, it reduces the chance of information misuse by even authentic users, as only required information is available to them under the least privilege policy.
- *Reduces the cost of organization:* As the model is scalable enough, the cost of management, operations, and maintenance also varies according to services availed with time.
- *Improved server response time & operational efficiency:* It reduces server workload by limiting the per day accesses as per users/roles, unlike ordinary access control methods. This leads to higher operational potency and better response time for the servers.
- *Separation of duties and auditing*: This model reduces conflicts by separating each permission for tasks, tasks specific to roles and roles for each user. The auditing process gets simplified by this approach.
- *Delegation of tasks:* This policy leads to easy auditing and visibility of tasks for administration. Thus if any user is overloaded with tasks, then the administration has the choice to delegate his/her duties to different users.
- *Improved security as it follows the least privilege principle:* As data is a valuable asset nowadays, this scheme is surely enhancing the security from suspicious attempts from internal and external users. So it is decreasing the risk of data leakage and breaches by intruders.
- *Limited network usage if organization have numerous employees:* No matter the number of employees an organization hires, the network usage will always be limited. The network cost and server cost will always be measurable with the proposed scheme.

- *Compliance enhancing:* As most of the costs are countable, it gives the ability to easily verify all the policies and activation compliances.

well as reducing the server's overhead. A case study for the Netflix Cloud server is also discussed followed by the benefits and limitations of the proposed EF-RBAC scheme. For further improvement, some good authentication mechanisms can also be applied within this

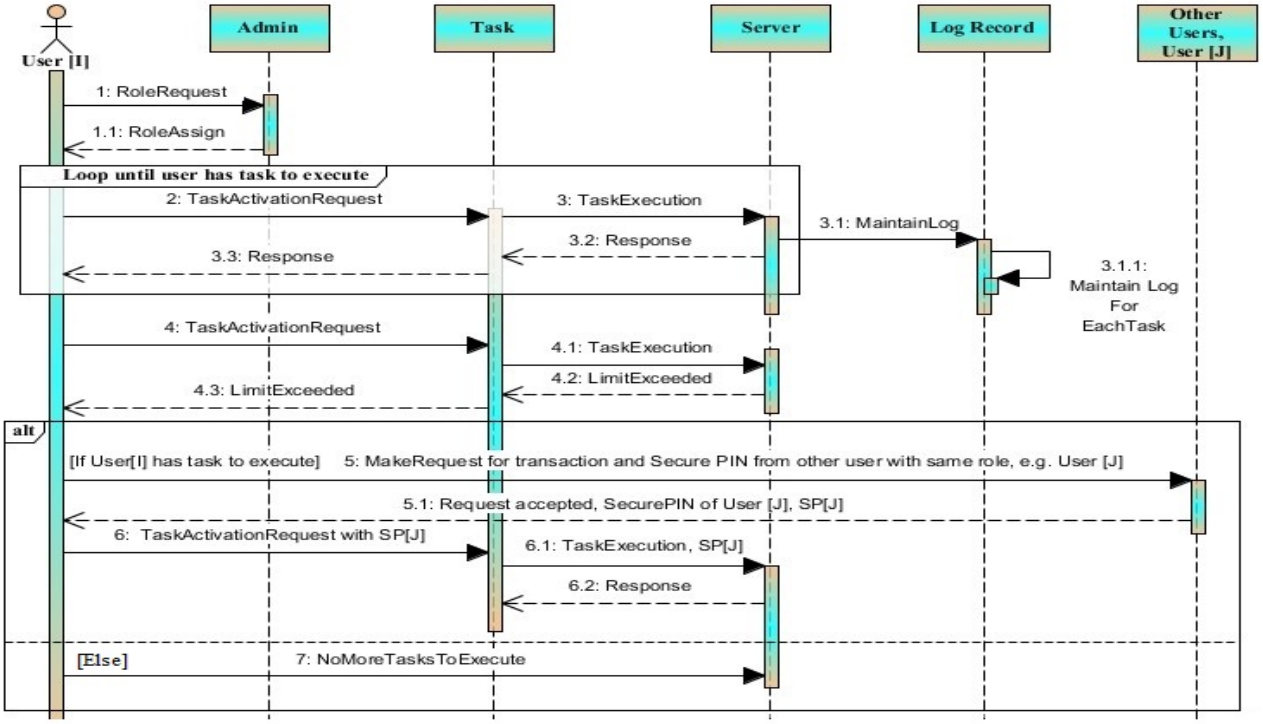

Figure 4. Sequence of events for EF-RBAC Algorithm's

6.3. Limitations of the proposed scheme

Sometimes if the roles of users are changing dynamically, it may create confusion regarding which user is associated with which privileges. Roles are assigned on the basis of the Least privilege principle, but still, if roles are changed frequently then some confusion may arise.

7. Conclusion and future scope

There are several access control mechanisms available but for the open cloud computing environment, but only a few access control schemes are applicable. The best and most applicable approach for such environments seems to be Role-based access control (RBAC). The scalability, measurability and flexibility properties of this model make it most useful amongst users. This work has implemented an efficient and flexible RBAC mechanism with some variations over ordinary RBAC. The implementation is done with the CloudSim (version 3.0) simulator and the steps are discussed in the form of an algorithm in paper. The proposed scheme is implemented for real applications can conspicuously give hopeful results as it is limiting the intruder's accesses, providing protection against DDoS attacks, improving security as model.

References

[1] Wang, W., Han, J., Song, M. and Wang, X. (2011). The Design of a Trust and Role Based Access Control Model in Cloud Computing. *In proceedings of the 6th International Conference on Pervasive Computing and Applications, IEEE*, ICPCA-2011, pp. 330-334.

[2] Harnal, S. and Chauhan, R.K. (2019). Hybrid Cryptography based E2EE for Integrity & Confidentiality in Multimedia Cloud Computing. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), Scopus*, 8(10), pp. 918-924.

[3] Sandhu, R.S., Coyne, E.J., Feinsteink, H.L. and Youman C.E. (1996). Role-Based Access Control Models. *IEEE Computer*, 29(2), pp. 38-47.

[4] Younis, A., Kashif, K. and Merabti, M. (2014). An access control model for cloud computing. *Journal of information security and applications, Science Direct*, Elsevier Science Inc., 19(1), pp. 45-60.

[5] Rouse, M. (2018). Role-based access control (RBAC). *The Essential Guide [Internet]*. https://searchsecurity.techtarget.com/ definition/role-based-access-control-RBAC.

[6] Blaze, M., FeigenBaum, J. and Keromytis, A.D. (1998). Keynote: Trust management for public-key infrastructures. *Cambridge Security Protocols 1998, LNCS*, Berlin: Springer, Heidelberg, 1550, pp. 59-63.

[7] Alattab, B.S., Fadewar, H.S. (2014). Security Issues and Challenges in Cloud Computing. *International Journal of Emerging Science and Engineering (IJESE)*. ISSN: 2319–6378. 2(7), pp. 22-26.

[8] Alattab, B.S. (2015). Role-Based Access Control's Framework for Cloud Computing. *In Conference: 103rd Indian Science Congress*, At Mysore University.

[9] Choudhury, A.J., Kumar, P., Sain, M., Lim, H. and Jae-Lee, H. (2011). A strong user authentication framework for cloud computing. *In: proceedings of 2011 IEEE Asia-Pacific Services Computing Conference*, pp. 110-115.

[10] Hu, V.C., Kuhn, D.R. and Ferraiolo, D.F. (2006). The computational complexity of enforceability validation for generic access control rules. *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 1.

[11] Hu, V.C. and Scarfone, K. (2012). Guidelines for access control system evaluation metrics. *National Institute of Standards and Technology,* NISTIR 7874.

[12] Ferraiolo, D.F., Barkley, J. and Kuhn, D.R. (1999). A role-based access control model and reference implementation within a corporate intranet. *ACM Transaction on information and System Security (TISSEC),* 2(1), pp. 34-64.

[13] Jin, X., Krishnan, R. and Sandhu, R. (2012). A unified attribute-based access control model covering DAC, MAC and RBAC. *In Proceedings of IFIP Annual Conference on Data and Applications Security and Privacy*, Lecture Notes in Computer Science, SpringerLink, 7371, pp. 41-55.

[14] Keromytis, A.D. and Smith, J.M. (2007). Requirements for scalable access control and security management architectures. *ACM Transactions on Internet Technology (TOIT)*, 7(2).

[15] Almutairi, A., Sarfraz, M., Basalamah, S. and Aref, W.G. (2012). A distributed access control architecture for cloud computing. *IEEE Software*, 29(2), pp. 36-44.

[16] Crago, S., Dunn, K., Eads, P., Hochstein, L., Kang, D.L. and Kang, M. et al. (2011). Heterogeneous cloud computing. *In IEEE International Conference on Cluster Computing (CLUSTER)*, Austin, TX, USA.

[17] Patil, V., Mei, A. and Mancini, L.V. (2007). Addressing interoperability issues in access control models. *In ASIACCS 2007, Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pp. 389-391.

[18] Kaur, G. and Singh, S. (2013). Implementing XML-based Role and Schema Migration Scheme for Clouds. *International Journal of Engineering and Technology (IJET),* 5, pp. 220-225.

[19] Goyal, V., Pandey, O., Sahai, A. and Waters, B. (2006). Attribute-based Encryption for Fine-Grained Access Control of Encrypted Data. *Proceedings of the 13th ACM Conference on Computer and Communications Security,* pp. 89-98.

[20] Bethencourt, J., Sahai, A. and Waters, B. (2007). Cipher text-Policy Attribute-Based Encryption. *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 321-334.

[21] Yang, K. and Jia, X. (2012). Attribute-based Access Control for Multi-Authority Systems in Cloud Storage. *Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems*, pp. 536-545.

[22] Ristenpart, T., Tromer, E., Shacham, H. and Savage, S. (2009). Hey, You, Get off my Cloud: Exploring Information Leakage in Third-Party Compute Clouds. *Proceedings of the 16th ACM Conference on Computer and Communications Security*, Chicago, USA, pp 199-212.

[23] Shafiq, B., Joshi, J.B.D., Bertino, E. and Ghafoor, A. (2005). Secure Interoperation in a Multi-domain Environment Employing RBAC Policies. *IEEE Transactions on Knowledge and Data Engineering,* 17(11), pp. 1557-1577.

[24] Ruj, S., Nayak, A. and Stojmenovic, I. (2011). DACC: Distributed Access Control in Clouds. *Proceedings of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications.* pp. 91-98.

[25] Zhu, Y., Hu, H., Ahn, G.J., Huang, D. and Wang, S. (2012). Towards Temporal Access Control in Cloud Computing. *2012 Proceedings IEEE INFOCOM*, IEEE, USA, 2012.

[26] Kaur, M., Wadhwa, P. (2014). An Analytical Review of Role Based Access Control Over Cloud Computing. *International Journal of Latest Scientific Research and Technology*. pp. 15-18.

[27] Harnal, S. and Chauhan, R.K. (2016). Multimedia Support from Cloud Computing: A Review. *In International Conference on Microcom-2016, IEEE*, NIT, Durgapur.

[28] Chenthara S., Ahmed, K., Wang, H. and Whittaker, F. (2019). Security and Privacy-Preserving Challenges of e-Health Solutions in Cloud Computing. *IEEE Access, IEEE*, 7, pp. 74361 – 74382.

[29] Wang, H., Yi, X., Bertino, E. and Sun, L. (2016). Protecting outsourced data in cloud computing through access management. *Concurrency and Computation: Practice and Experience, Wiley Online Library,* Special issue paper, 28 (3), pp. 600-615.

[30] Kabir, E., Mahmood, A., Wang, H. and Mustafa, A. (2015). Microaggregation Sorting Framework for K-Anonymity Statistical Disclosure Control in Cloud Computing. *IEEE Transactions on Cloud Computing, IEEE,* pp. 1-1.

[31] Wang, H. and Zhang, R., (2013). Role-based access control to outsourced data in cloud computing. *Proceedings of the Twenty-Fourth Australasian Database Conference*. 137.

[32] Wang, H., Jiang, X. and Kambourakis, G. (2015). Special issue on Security, Privacy and Trust in network-based Big Data. *Information Sciences*, *Elsevier Science Inc.,* 318(C), pp. 48-50.

[33] Wang, H., Cao, J. and Zhang, Y. (2005). A flexible payment scheme and its role-based access control. *IEEE Transactions on Knowledge and Engineering, IEEE,* 17(3), pp. 425-436.