Securing Data Storage on Cloud by Extending Role based Access Control

Mamoon Rashid
Research Scholar
Department Of Computer Science Engineering
Ramgharia Institute of Engineering and Technology
Phagwara, Punjab, India.

Rishma Chawla
Assistant Professor
Department Of Computer Science Engineering
Ramgharia Institute of Engineering and Technology
Phagwara, Punjab, India.

ABSTRACT
Role-based access control (RBAC) models have generated a great interest in the security community as a powerful and generalized approach to security management and ability to model organizational structure and their capability to reduce administrative expenses. In this paper, we highlight the drawbacks of RBAC models in terms of access control and authorization and later provide a more viable extended-RBAC model, which enhances and extends its powers to make any Cloud Server more secure by adding valuable constraints. Later the Blobs are stored on cloud server which is then accessed by the end users via this Extended RBAC model. We describe a practical implementation of the proposed extended RBAC based architecture and discuss the performance results with its base models. We later show how the users with different premiums can access this architecture in a better way and also how the unknown users for this architecture can be denied the usage of services by adding valuable constraints.

Keywords
Authorization, RBAC, Blobs, Server, Architecture

1. INTRODUCTION
Cloud computing has begun to emerge as a hotspot in both industry and academia; It represents a new business model and computing paradigm, which enables on demand provisioning of computational and storage resources. Economic benefits consist of the main drive for cloud computing due to the fact that cloud computing offers an effective way to reduce capital expenditure and operational expenditure. The definition of cloud computing as per the literature in [1] is ”A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.”

The Cloud Security Alliance has summarized five essential characteristics [2] that illustrate the relation to, and differences from, traditional computing paradigm.

• **On-demand self-service** – A cloud customer may unilaterally obtain computing capabilities, like the usage of various servers and network storage, as on demand, without interacting with the cloud provider.

• **Broad network access** – Services are delivered across the Internet via a standard mechanism that allows customers to access the services through heterogeneous thin or thick client tools (e.g., PCs, mobile phones, and PDAs).

• **Resource pooling** – The cloud provider employs a multitenant model to serve multiple customers by pooling computing resources, which are different physical and virtual resources dynamically assigned or reassigned according to customer demand. Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

• **Rapid elasticity** – Capabilities may be rapidly and elastically provisioned in order to quickly scale out or rapidly released to quickly scale in. From customers’ point of view, the available capabilities should appear to be unlimited and have the ability to be purchased in any quantity at any time.

• **Measured service** – The service purchased by customers can be quantified and measured. For both the provider and customers, resource usage will be monitored, controlled, metered, and reported.

However authorization and access control has always been a fundamental security technique in systems like cloud computing in which multiple users share access to common resources. Authorization is the process of expressing security policies that determine whether a subject (e.g., process, computer, human user, etc.) is allowed to perform an operation (e.g., read, write, execute, delete, search, etc.) on an object (e.g., a tuple in a database, a table, a file, a service, and, more generally, any resource of the system). These policies define the subject's permissions (rights to carry out an operation on an object) in a computer system. Access control is the process of enforcing these policies in order to achieve the desired level of security. Managing and administering the users' privileges is one of the most challenging tasks in access control. Several access control models have been proposed, such as, discretionary and mandatory access control models (DAC and MAC), Clark-Wilson model, Lipner's Integrity model, Chinese wall model, Task based models, and Role Based Access Control models and RBAC has further been extended up to some level. Among these models Role-based access control (RBAC) models have been receiving attention as they provide systematic access control security through a proven and increasingly predominant technology for commercial organizations. One of the main advantages of the RBAC over other access control models is the ease of its security administrations. RBAC models are policy neutral [3]; they can support different authorization policies including mandatory and discretionary through the appropriate role configuration. In spite of the success of the RBAC, researchers have determined that there are still many application security requirements that are not addressed by the existing RBAC models [4]. In the past few years, several RBAC extensions have been proposed to address such security requirements [5, 6, 7, 8, 9, 10, and 11]. Although, these extensions geared and enhanced basic RBAC model, however we find some applications where individually

these extended models fall short. We will uncover the loop-hole in existing extended RBAC model and later will add a mechanism to resolve such an issue.

2. RELATED WORK ON ROLE BASED ACCESS CONTROL MODEL

In recent years, vendors have begun implementing role-based access control (RBAC) features in their database management system, security management, and network operating system products, without general agreement as to what constitutes an appropriate set of RBAC features. Several RBAC models have been proposed [13, 14, 15, 16, 17], without any attempt at standardizing salient RBAC features. To identify RBAC features that exhibit true enterprise value and are practical to implement, the National Institute of Standards and Technology has conducted and sponsored market analysis [18, 19], developed prototype implementations [20], and sponsored external research [21]. However NIST is not alone in recognizing the potential benefits of RBAC technology. RBAC is a technology that is both new and old. The concept of roles has been used in software applications for at least 25 years, but it is only within the past decade that role based access control has emerged as a full-fledged mechanism as mature as traditional mandatory access control (MAC) and discretionary access control (DAC) concepts.

RBAC has emerged as a viable alternative to traditional access control policies, such as DAC and MAC, because it is based on an enterprise's organizational structure. As such, systems, data, and applications administrators and owners can more effectively manage and maintain information resources in a manner consistent with enterprise-wide security policies. RBAC has the further benefit of facilitating systems administration by assigning roles to manage users as opposed to using each individual user's identity to manage users. Although role-based security models have existed for 20 years, their application has until recently been limited. To date, most systems have based access control on the discretion of the owner or administrator of the data as opposed to basing access on organizational or policy needs as is done with RBAC. The explosion of electronic data exchange and interconnection of information systems led to significant productivity gains in the 1990s. However, these same factors have also increased electronic security and integrity concerns. Confidentiality restriction and regulatory requirements have caused organizations to look for improved approaches to manage the types of users that may have access to which data and to which applications. The result is a renewed and growing interest in role-based security models. Several organizations, including NIST, have been working since the early 1990s to define a common standard for RBAC and to spur its implementation by providing research and development support to this emerging technology.

Sandhu et al [12] proposed RBAC 96 which is a family of four constitutes models. In RBAC permissions are associated with roles (the intermediate concept of roles can be seen as collections of permissions), and users are made members of appropriate roles. The notion of role is an enterprise or organizational concept. The definition of role is quoted from Sandhu et al. [12]: A role is a job function or job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role. Permissions are not directly assigned to users; instead they are assigned to roles. RBAC comprise a family of four references models:

RBAC0: contains the core concepts of the Model. It is the minimum requirement for any system that exploits features of RBAC. Users (U), roles (R), and permissions (P) are three sets of entities and the relations between these entities are defined by User-Role Assignment and Permission-Role Assignment [12]. These sets and relations are the main concepts of the RBAC. A user can be member of many roles and each role can have many users. A user can invoke multiple sessions within a session a user can invoke set of roles but each session belongs to only one user. Permission can be assigned to many roles and a role can have many permissions.

RBAC1: adds to RBAC0 a role hierarchy (RH). Role hierarchies are an important concept for structuring roles to represent organization users responsibly and degree of authority.

RBAC2: introduces the concept of constraints. RBAC adds static (not related to sessions) and dynamic (related to sessions) constraints between core concepts [12]. These constraints are considered to be the principle motivation for RBAC because constraints are powerful mechanism to lay out higher-level organizational mechanism [12].Constraints can be applied to User-Role Assignment, Permission-Role Assignment and session.

RBAC3: includes all aspects of RBAC0, RBAC1 and RBAC2 and it is called a unified model of RBAC.

RBAC3 combine RBAC1 and RBAC2 to combine both role hierarchy and constraints. In this model constraints can be applied to the role hierarchy in addition to the constraints in RBAC2.

E-RBAC: E-RBAC model was presented for RBAC 96 model and it filled the role authorization shortage in RBAC96. In E-RBAC users can also direct for authorization. For example, when the authority is authorized after a specific request to access the user's system resources, the system at the same time can judge whether the users own role or whether the user has authority access to the module's functions, as long as one is given the authority between the users and their role, to allow access.

3. BLOB STORAGE ON CLOUD SERVER

Blob storage [22] is a service for storing large amounts of unstructured data that can be accessed from anywhere in the world via HTTP or HTTPS. A single blob can be hundreds of gigabytes in size, and a single storage account can contain up to 100TB of blobs. Common uses of Blob storage include:

- Serving images or documents directly to a browser
- Storing files for distributed access
- Streaming video and audio
- Performing secure backup and disaster recovery

The Blob service contains the following components:

Storage Account: All access to Cloud Storage is done through a storage account. This is the highest level of the namespace for accessing blobs. An account can contain an unlimited number of containers, as long as their total size is under 100TB.

Container: A container provides a grouping of a set of blobs. All blobs must be in a container. An account can contain an unlimited number of containers. A container can store an unlimited number of blobs.

Blob: A file of any type and size. There are two types of blobs that can be stored in Cloud Storage: block and page blobs. Most files are block blobs. A single block blob can be up to 200GB in size. Page blobs, another blob type, can be up to 1TB in size, and are more efficient when ranges of bytes in a file are modified frequently.

4. PROBLEM FORMULATION

Although current RBAC models restrict a user to access the resources if it is not assigned as a member to any particular role of the architecture, however the RBAC nowhere defines that how many users could be assigned to each role. This limitation largely affects the architecture not only in terms of its security but also when one accesses the resources on a shared server where it affects the network bandwidth also. RBAC also allows users to access resources based on the roles. All users are made members of roles and permissions are also associated with roles. Later when any user wants to access any resource, its access depends on the constraints imposed on that particular role and it is imposed on all other users of that role also. This limitation makes architecture quite ordinary as there is no such mechanism to provide different level of accesses to different users under one particular role. Also RBAC does not impose any transaction limit for users to access available resources under specified roles. This limitation leads to insecurity of the system where resources can be accessed at free will even if login of any existing user is hacked or uncovered. Keeping all these flaws of RBAC into consideration, there is need of an architectural system which will address all these issues and hence will result as a more powerful RBAC model.

5. PROPOSED ERBAC MODEL

This Extended RBAC Model is proposed to address aforesaid problems so as to enhance its robustness. Firstly in this model during the creation of any new role, the constraint is imposed on role to decide how many users can access this role later. Here an organization can make use of this feature to provide membership of role to only such users which are beneficial for the organization. Secondly another constraint is imposed on all the normal users who can access resources later so that they can access the organizational resources on a limited basis for some specific time intervals. Although it looks that it will result in inconvenience to users as they can make limited transactions and thus cannot access the resources at free will. However it is one of the strong dimensions where we can make RBAC more powerful. This is because imposing transactions limits will reduce the chances of minimizing loss of resources of any particular user if cannot be saved and eliminated in totality. However here organization can assign the limit based on the usual usage of users. Next whenever any user is made member of any particular role, then at that time we are providing an option there to rate this user based on his membership status. This feature will largely help an organization to give different accesses to users irrespective of belonging to the same role. This decision is to be taken on the basis of premium membership to be attained by the user. Here by using this feature all normal users can access resources only to that extent to which that particular role will allow them. However the premium users will get access more than to role limit and will be decided on the basis of their premium membership. Finally in this proposed system we are planning to use the Blob service of resources so as use it service as on Cloud. Later we will impose all the added features on these blobs so as to check its integrity and robustness to evolve as an enhanced model. All these features are summed up in the following flow based diagram:

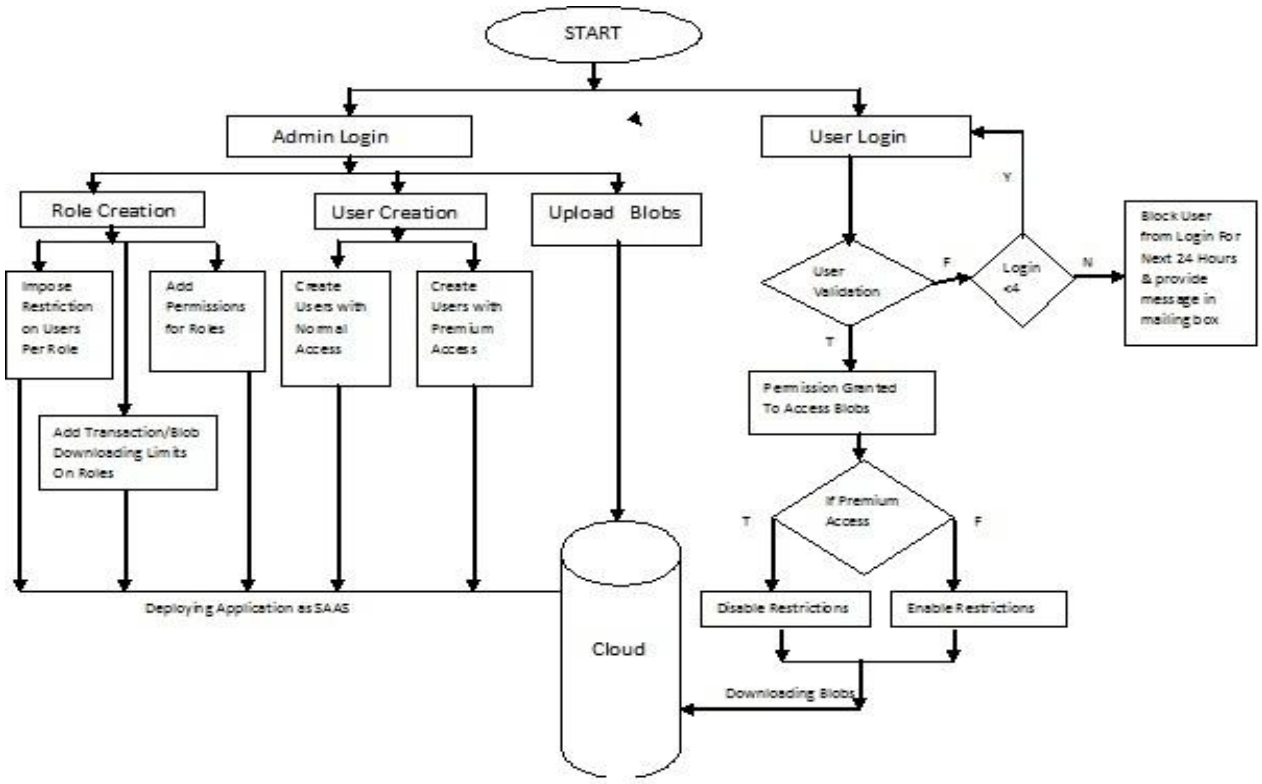

Fig.1 Design of Extended RBAC Architecture

6. IMPLEMENTATION

We have implemented the above architecture of the Extended RBAC. The system is implemented in MVC Framework with C# Scripting Language. Later this implemented web service is hosted on Microsoft's Azure cloud platform where the cloud use SQL database for its main storage of table contents. Next we uploaded the data on this third party Public Cloud via our implemented Extended RBAC interface for whose storage we created a storage account on Microsoft Azure Platform. Finally we allowed our users to make access of this already stored data on Microsoft Azure cloud via this Extended RBAC model.

We have performed our experiments on a machine with Intel(R) Core TM i5 @ 2.50 GHz processor, 4 GB of RAM and Microsoft Windows 7 Ultimate 64 bit Operating System.

Let us first assume that the administrator of this Extended RBAC makes successful login. Now s/he will be redirected to the main interface of this Extended RBAC where s/he can now go for the creation of roles and then impose permissions on these created roles. Here at this time the administrator can decide how many users can be allowed to use this particular role in future and also the download limit is defined for these users. After then s/he can create users who can be made members for already created roles based on their accessing attributes. While creating the users, the administrator will decide that particular role to which this user is made a member and also the nature of accessing data is decided over here i.e. whether the permissions and downloading limit constraints are to be imposed on this user or not which will be decided on the premium membership to be purchased by this user. Here the administrator will generate the passwords for users as well by virtue of which users can make their login in future for accessing this architecture. Next the administrator will upload the data content which is to be stored in storage account already created on Windows Azure Cloud. Now authentic users can make login to this architecture to access the stored data on Microsoft Azure Cloud Storage account. Here if anyone(as a unknown user) will try to access this web interface based on the guess work then s/he can be warned in three attempts to go for authentic details and then in fourth attempt s/he can be blocked for next 24 hours. This is the added security mechanism in our architecture which will enhance the security mechanism for accessing our implemented architecture. Once making successful login the users can access and download the stored content on Cloud based on their permissions and restrictions imposed on them which were initially decided in admin interfaces of the architecture. Some of the snapshots of our architecture are shown below so as to understand how above mentioned featured are to be executed.

This is the main interface of our architecture where our users and administrator will make login:

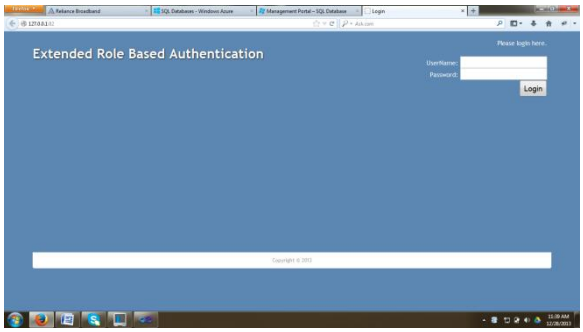

Fig.2 Administrator Login Interface

Here is an interface where administrator can create roles:

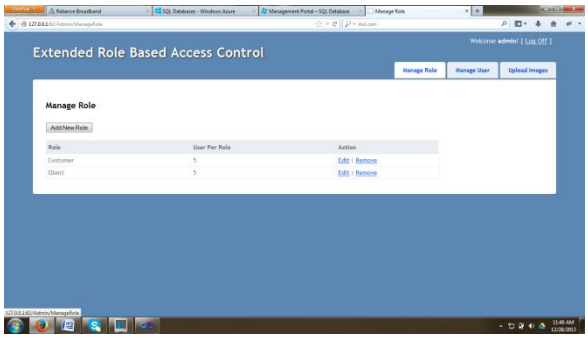

Fig.3 Role and User Creation Interface

This is an interface where administrator can decide how many users are allowed to act as memebers for this particular role, to decide permission and also the downloading limit of data for these users:

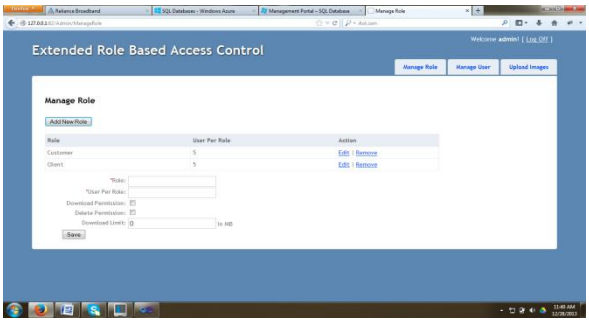

Fig.4 Role Creation Interface

The following is an interface where users are to be created and assigned membership in terms of roles and to decide the nature of access based on their premium membership.Also it adds feature of activeness which can be enabled or disable based on the user behavoiur so as to allow him to access or to get blocked from using this architecture:

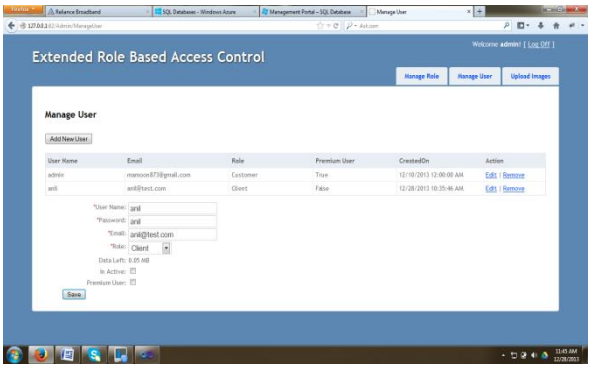

Fig. 5 User Creation Interface

The following is an interface where the administrator can upload the necessary data which is to be stored on Microsoft Azure Cloud:

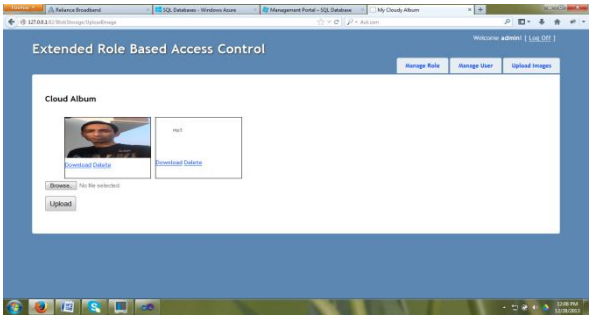

Fig. 6 Content Uploading Interface

This is an interface where the any unknown user tries to access this Extended RBAC architecture based on the guess work:

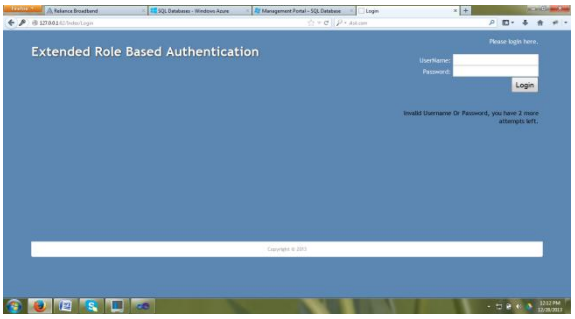

Fig. 7 Interace validating Unknown Users

This is an interface where the user makes successful login and can use the data services until his limits credits will get expired:

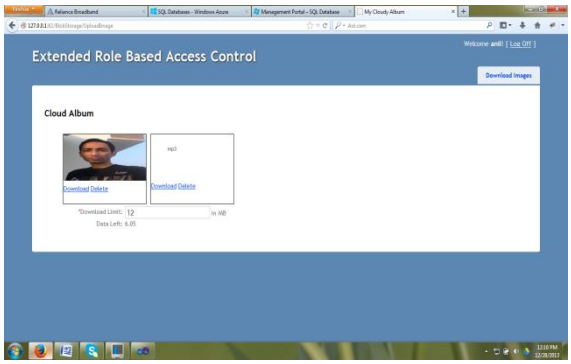

Fig. 8 User Accessing Data Interface

This is an interface where the user tries to access the storage data service after reaching the download limit:

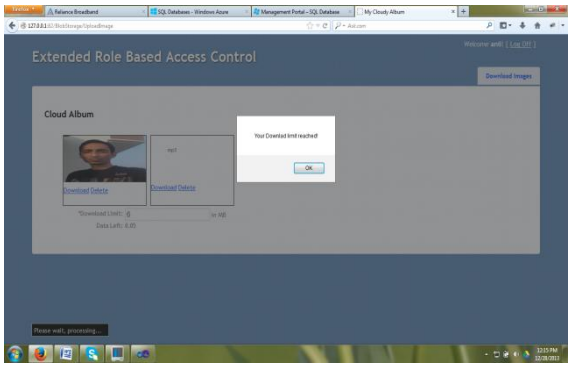

Fig. 9 User with al Downloaded Limit Interface

7. RESULTS AND COMPARATIVE ANALYSIS WITH OTHER RBAC MODELS

In this implemented Extended RBAC Model, we have successfully added the features of imposing user limits on roles, downloading limits for every user under some role and varying access limits for users to access the data based on their premium memberships. For test results we have created two roles in this system architecture named as 'Customer' and 'Client' and later added the user count of 5 and 3 for these roles respectively. The permissions imposed on these roles are delete and download. For test purposes we added the data limit of 5MB and 10 MB respectively for these roles. Later during the creation of users for these roles of 'Customer' and 'Client', I added admin and 'anil' as users belonging to these roles respectively. Here during the creation of user memberships for these roles, I kept 'admin' as premium one and 'anil' as normal user. Next for test case I logged in this system as admin and downloaded data beyond the limit of the role of 5 MB under which admin is assigned membership. In next attempt I logged in as 'anil' and downloaded data only up to 10 MB as it is the limit of Client role under which 'anil' is assigned as member. This is the premium membership feature of this extended architecture. For testing the user limit on roles I created 3 users for 'Client' role while logging as admin and after then and after then I tried to create 4 users under Client role but the system responded with message box showing user limit has reached. This feature completely minimized the chances of signing in for unknown users.

In comparison to other RBAC models, our Extended RBAC architecture added the power to the existing RBAC models in terms of authentication and access control. In RBAC0 only the concept of roles, users and permissions were used. In RBAC1, all features of RBAC0 were used with added implementation of Role hierarchy. In RBAC2, all the features of RBAC1 were implemented with the concept of constraints. In RBAC all features of RBAC's were implanted together and turned to universal model. In comparison to these RBAC models first I enhanced every existing feature of RBAC and implemented it to impart better results. Then later I added the features of imposing user limits on roles, downloading limits for every user under some role and varying access limits for users to access the data based on their premium memberships. This helped the RBAC to turn as much enhance model. This comparison study of our extended RBAC architecture is compared with other existing models to highlight the impact of this work in terms of following table structure:

Feature/ Model	RBAC 0	RBAC 1	RBAC 2	RBAC 3	OUR ERBAC
Permissions on Roles	✓	✓	✓	✓	✓
Role Hierarchy	—	✓	✓	✓	✓
Role Constraints	—	—	✓	✓	✓
Limiting the users per Role	—	—	—	✓	✓
Downloading Limit on Users	—	—	—	—	✓
Varied Role Access based on Premium Membership	—	—	—	—	✓

Table 1 Results Achieved In comparison to other

Base Models

The experimental study of this running system proved this architecture to be better in terms of Constraints and the performance comparison of every activity to be shown as well in terms of the following Column chart:

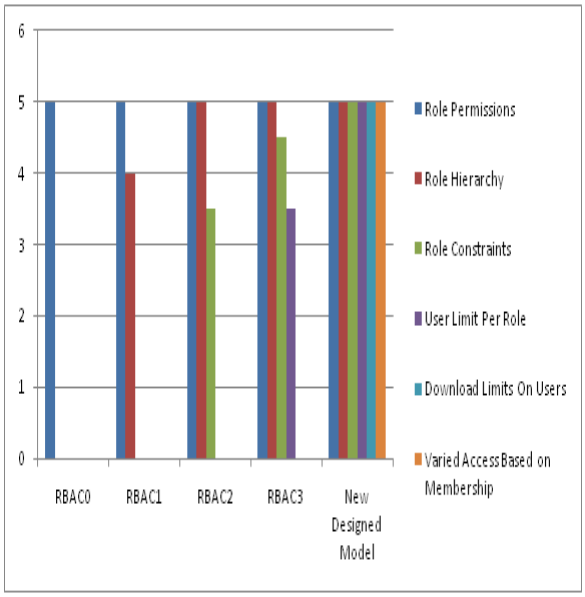

Fig 10 Performance comparisons in various models

Clearly this extended new model shows improvement in all elements of Permissions assigned to roles, Role Hierarchy, Constraints imposed on roles, limits applied on users per role, data downloading limits on users and nature of varying data accesses of users based on their membership.

8. CONCLUSION

This proposed model has outlined a sketch for new RBAC which addresses the security features for any multi-centric application. We have investigated the state-of-art of the access control models. In particular, we have investigated the current RBAC extensions as they are most influential authorization models in the security community. This proposed model, however, showed that, all or most of the existing RBAC extensions are not suitable for specifying security requirements of that application. Although this study might not be exhaustive, however we believe that this model will provide another picture of the RBAC will result into much enhanced and more powerful model than other existing models. The investigations in this study open the doors to the software stake holders to identify and realize the added features which can make their organizational architecture more robust than existing RBAC models.

9. REFERENCES

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud computing and grid computing 360-degree compared,” Grid Computing Environments Workshop, 2008. GCE’08, 2009, pp. 1-10.

[2] Cloud Security Alliance (CSA). “Security Guidance for Critical Areas of Focus in Cloud Computing V2.1,” (Released December 17, 2009).http://www.cloudsecurityalliance.org/guidance/csaguide.v2.1.pdf

[3] R. Sandhu. \Role hierarchies and constraints for latice-based access controls." In E. Bertino, H. Kurth, G. Martella, and E Monotolivo Eds. LNCS 1146, Proceedings of the European Symposium on Research in Computer Security 1996, Rome, Italy.

[4] E. Bertino, P. A. Bonati, and E. Ferrari, \TRBAC: A temporal role-based access control model," ACM Transactions on Information and System Security, 4(3):191-233, 2001.

[5] K. Devdatta and T. Anand, \Context-aware role-based access control in pervasive computing systems," In Proceedings of the 13th ACM Symposium on Access Control Models and Technologies, Estes Park, CO, 2008

[6] Z. Xinwen, O. Sejong, and S. Ravi, \PBDM: a exible delegation model in RBAC," In Proceedings of the 8th ACM Symposium on Access Control Models and Technologies, Como, Italy, 2003.

[7] S. Chakraborty and I. Ray, \TrustBAC: integrating trust relationships into the RBAC model for access control in open systems," In Proceedings of the 11th ACM Symposium on Access Control Models and Technologies. Lake Tahoe, CA, 2006.

[8] E. Bertino, P. A. Bonati, and E. Ferrari, \TRBAC: A temporal role-based access control model," ACM Transactions on Information and System Security, 4(3):191-233, 2001.

[9] H. Shen and F. Hong, \A context-aware role-based access control model for web services," In Proceedings of the IEEE International Conference on e-Business Engineering, Beijing, China 2005.

[10] I. Ray and M. Toahchoodee, \A spatio temporal role based access control model," In Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Redondo Beach, CA, 2007.

[11] I. Ray, M. Kumar, and L. Yu, \ LRBAC: A location-aware role-based access control model," In Proceedings of the 2nd International Conference on Information Systems Security, Kolkata, India, 2006.

[12] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, \Role-Based access control models," IEEE Computer, 29(2):38-47, 1996.

[13] D. Ferraiolo and R. Kuhn. Role-Based Access Control. In Proc. of the NIST-NSA Nat. (USA) Comp. Security Conf., pp 554-563, 1992

[14] M. Nyanchama and S. Osborn. Access rights administration in role-based security systems. In J. Biskup, M. Morgenstern, and C. E. Landwehr, editors, Database Security, VIII: Status and Prospects, pages 37-56. North-Holland, 1994.

[15] D. Ferraiolo, J. Cugini, and R. Kuhn. Role-based access control: Features and motivations. In Proc. of

the Annual Computer Security Applications Conf., IEEE Press, 1995.

[16] L. Giuri and P. Iglio. A formal model for role based access control with constraints. In proc. of the Computer Security Foundations Workshop, pp. 136-145. IEEE Press, 1996.

[17] R Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-based access control models. IEEE Computer, 29(2), February 1996.

[18] D. Ferraiolo, D. Gilbert, and N. Lynch. An examination of federal and commercial access control policy needs. In Proc. of the NIST-NSA Nat. (USA) Comp. Security Conf., pp 107-116, 1993

[19] C. Smith, E. Coyne, C. Youman and S. Ganta. Market analysis report: NIST small business innovative research (SBIR) grant: role based access control: phase 2. A marketing survey of civil federal government organizations to determine the need for role-based access control security product, SETA Corp., July 1996.

[20] D. Ferraiolo, J. Barkley, and R. Kuhn. A role-based access control model and reference implementation within a corporate internet. ACM Transactions on Information and System Security, 2(1), 1999.

[21] H. Feinstein. Final report: NIST small business innovative research (SBIR) grant: role based access control: phase 2. SETA Corp., October 1996.

[22] Microsoft Azure Documentation support onhttp://www.windowsazure.com/enus/develop/net/how-to-guides/blob-storage/