RB-GDM: A Role-Based Grid Delegation Model

G Geethakumari geethamaruvada@gmail.com
Research Scholar,DCIS
University of Hyderabad
Hyderabad, 500046, India

Dr Atul Negi atulcs@uohyd.ernet.in
Reader, DCIS
University of Hyderabad
Hyderabad, 500046, India

Dr V N Sastry vnsastry@idrbt.ac.in
Assoc. Professor
IDRBT, Hyderabad
Hyderabad, 500057, India

Abstract

Grid delegation is the procedure by which a valid user endows another user or a program or service with the ability to act on that user's behalf. Delegation is the primary form of authorization in grids. The large and geographically distributed, dynamic, heterogeneous and scalable grid environment poses unique delegation requirements. Presently there are no standard mechanisms to guide grid delegation. As credential delegation has its own limitations in a dynamic grid environment, a new conceptual model is required to effectively formulate the grid delegation requirements. In this paper, we present a framework called Role-Based Grid Delegation Model (RB-GDM) for delegating access rights in grids. The basic unit of delegation in our model is *role*. Derived from the standard RBAC formalisms, this framework explores various approaches for authorization and revocation of delegation.

Keywords: Delegation; access control; authorization; grid computing systems; role based access control

1. INTRODUCTION

Delegation is an important aspect of grid security. In general, delegation is defined as the act whereby an active entity in a system authorizes (delegates) its authority to another entity to execute some functions on its behalf [7]. Delegation in computing systems can take many forms: human to human, human to machine, machine to machine and even machine to human [3]. The consequence of delegation is propagation of access rights. Propagation of access rights in a decentralized collaborative system like the grid presents difficult challenges for traditional access control mechanisms and models.

A delegation activity in a grid environment needs to be monitored and controlled in such a manner so that the resource inside the domain can stay protected [3]. Though various forms of delegation are possible, the most common delegation activities in present day grids are user to user delegation and user to machine delegation. Most of the resource access in grids takes place

through remote authorization and is achieved by user to machine/process delegation. User to machine delegation is the mechanism whereby a user in a distributed environment authorizes a system to access remote resources on his behalf [3]. In some cases the user may delegate the rights to one of the several permissible roles or identities; in order to limit the actions of the process to some subset that user is authorized to. This form of delegation is generally referred to as limited or restricted delegation.

The support of roles is crucial in an open computing environment like the grids, where not all individuals may be registered at a service and requests may arrive from previously unknown parties. In such a context, the ability to invoke a service often depends on the capacity in which a user can operate rather than on who the user actually is. This capacity is characterized as a role. The concept of roles is different from that of groups in the sense that unlike groups, roles carry a dynamic behavior and can be activated and deactivated by users and service providers arbitrarily as required and roles activation is enabled by attaching credentials with a request. Though current grid systems support delegation through impersonation (by issuing a proxy certificate), the revocation of delegation tracing and multi step delegation still remain as the major challenges.

We propose a grid delegation approach built on Role Based Access Control (RBAC) [8]. RBAC has been accepted in the access control design and architecture of security in enterprises as an alternative to traditional discretionary and mandatory access control models. RBAC is designed to suit large-scale enterprise-wide systems and works on the notion that permissions are associated with roles. The users are assigned to appropriate roles and users acquire permissions by being members of roles. Roles can be granted new permissions and permissions can be easily revoked from roles as and when needed. Thus RBAC provides a way to empower individual users and service providers through role-based delegation in a fully distributed environment like grids.

The rest of the paper is organized as follows. In section 2, we present the motivation for our work. In section 3, we propose the role-based delegation framework for delegation and revocation in grids. The different topologies and representation of the components of RB-GDM is discussed in section 4. Section 5 summarizes our work and provides conclusion and future work.

2. MOTIVATION

The core theme behind delegation is that of delegating rights, obligations, privileges and authority. The basic idea of role-based delegation is that users themselves may delegate role authorities (rights) to others to carry out some functions authorized to the former. Though there has been considerable work on various aspects of delegation [10], they cannot meet the grid authorization requirements in the present form. We present a delegation model for Grid resource access based upon roles through which only the rights assigned to roles can be delegated, thereby making it fine-grained. The major contribution of our paper is that unlike the existing identity-based approaches, this framework treats grid delegation as an authorization problem.

3. GRID DELEGATION FRAMEWORK

In this section, we propose a Grid delegation model called RB-GDM. This model is aimed at supporting role-based delegation and revocation. Our work is built upon the following reference models: RBAC96 model [8], the RBDM0 model [11], RDM2000- A Rule-Based Framework for Role-Based Delegation and Revocation [7], and RB-GACA: A RBAC Based Grid Access Control Architecture [9].

3.1 Basic Elements from Reference Models

The Role Based Access Control Model (RBAC) [8] has the following basic elements: users U, roles R and permissions P. Individual users are assigned roles and roles are assigned certain permissions. According to this model, a *user* is a human being; a *role* is a job function indicative of the responsibility and *permission* indicates the approval to execute some method or perform some action. We generalize the term user to make it represent an individual, an intelligent autonomous agent, or a machine in the Grid environment. We consider the end user of a Grid resource as the basic user or original user. There are two sets of users associated with a role r : as given in [7].

- Original users are those users who are assigned to the role r
- Delegated users are those users who are delegated to the role r

Each constituent local domain of the grid virtual domain may have its own role hierarchies. A role hierarchy is a structure reflecting an organization's lines of authority and responsibility and is represented using a partial ordering operator ≥. For example, if role1≥role2, then role1 inherits permissions of role2. Though role hierarchies in different local domains vary in their semantics level, they are required to enforce the principle of least privilege. The fundamental components of RBAC and RBDM0 which form the basis for our model are depicted in Table 1 and Table 2 respectively.

• U,R,P,S are the sets of users, roles,permissions and sessions respectively
• UA $\subseteq U \times R$ is a many to many user-to-role assignment
• PA $\subseteq P \times R$ is a many-to-many permission to role assignment relation
• RH $\subseteq R \times R$ is a partially ordered hierarchy

TABLE 1: RBAC Components

• UA = UAO ∪ UAD,where • UAO $\subseteq U \times R$ is a many to many original user-to-role assignment relation
• UAD $\subseteq U \times$ R is a many to many delegated user-to-role assignment relation
• Users : $R \rightarrow 2^U$ is a mapping of each role to a set of users Users$(r) = \{u \mid (u, r) \in UA\}$
• Users(r) = Users_$O(r) \cup Users_D(r)$

TABLE 2: RBDM0 Components

3.2 Role Based Delegation

The delegation model RBDM0 suggests only user-to-user delegation, which is the simplest form of delegation. Our model is Grid specific. As mentioned in 3.1, the end-user is considered as the basic user. The end-users are geographically distributed, can be dynamic and can get disconnected from the virtual organization (VO) [4], after the submission of a job. Hence end-user-to-end-user delegation is insufficient and inappropriate in Grids. We need to look at other forms of delegation namely user-to-machine, machine-to-machine, user-to process etc. The basic idea of delegation is *who* is granting or revoking *what* to *whom*. Thus the entities in delegation are the *grantor*, the *grantee* and the *rights*, which are granted. To suit the requirements of Grid environment and to distinguish delegation from direct authorization, we call the grantor of delegation as a *delegator*, the grantee as *delegatee* and the rights that are granted through the roles as *delegated rights*.

There are basically two categories of Grid roles: flat roles and hierarchical roles. For flat roles, no inheritance of permissions between roles is involved. Roles can be delegable or non-delegable. Delegable roles are those roles, a subset of which can further be delegated where as with non-delegable roles, further delegation is not possible. The first challenge in framing a Grid delegation model is to identify various roles in a Grid environment. The subject or active entity in a Grid transaction can be a user (we mean an individual), a process or a machine. Among users, there can be administrators (who possess all the administrative and super user privileges like a domain administrator), basic users (who can submit large scale jobs, access resources etc), developers (whose main objective is to handle the programming and middleware aspects) or guest users [9] (who can only browse the information or submit very small scale jobs).

The components that constitute a Grid delegation mechanism include the delegator, the delegatee, the delegated rights and the associated constraints. We make the following design criteria in RB-GDM.

- A role is considered to be the basic unit of delegation and is a job function.
- The term user in our model represents an end-user, a process or a machine in the broader perspective.
- Delegation between entities in the same role is not allowed since the same roles will have same permissions [3].
- Both single step delegation and multi-step delegation are permissible.
- Delegation can be granted in total or can be partial.
- Each delegating role r has two types of members namely the original members O(*r*) and delegated members D(*r*)*i.*
- Grid users can be a domain administrator, an advanced user, a basic user or a guest user.

Before we build a model, it is essential to define the components which can constitute our model. Therefore we present definitions 1-4 (Table 3, Table 4, Table 5 and Table 6 respectively) of RB-GDM. In Table 3, the basic components like grid domain, users, roles, permissions and the user-role, role-permission assignments which are required to build the Role-Based Grid Delegation Model (RB-GDM) are defined. Table 4 defines how the access rights can be delegated through authorization. Table 5 gives definitions for different role permissions like delegable and non-delegable. Table 6 defines what “revocation of delegation” means. Once the components of the model have been defined, we need to arrive at a way in which these components are associated as well as how do they interact to form a delegation mechanism. We achieve this by proposing algorithms for authorization and revocation of delegation. Algorithm 1 gives the method for authorization of delegation as per Definition 2. Algorithm 2, 3 and 4 show the methods for revoking the delegated rights by various means such as grant-dependent, grant-independent or

by time-out as per Definition 4 respectively. Each algorithm has been explained by including the comments as well as examples.

• Let D = $\{D_l\}$, where l = 1 to N be the set of domains of a Grid G • $U^l = \{u^l{}_i\}$, where i = 1 to m_l be the set of users of D_l • $R^l = \{r^l{}_j\}$, where j = 1 to n_l be the set of roles of D_l • $P^l = \{p^l{}_k\}$, where k = 1 to q_l be the set of permissions of R^l • $UA_l \subseteq U^l \times R^l$ is the set of relations of user to role assignment in D_l where $UA_l = UAO_l \cup UAD_l$ • $PA_l \subseteq P^l \times R^l$ is the set of relations of permission to role assignment in D_l • $U^l(r^l{}_j) : R^l \rightarrow 2^{U^l}$ is a function derived from UA_l mapping each role r to a set of users where $U^l(r^l{}_j) = \{u^l{}_i \mid (u^l{}_i, r^l{}_j) \in UA_l\}$ • $P^l(r^l{}_j) : R^l \rightarrow 2^{P^l}$ is a function derived from PA_l, mapping each role r to a set of permissions, where $P^l(r^l{}_j) = \{p^l{}_k \mid (p^l{}_k, r^l{}_j) \in PA_l\}$

TABLE 3: Definition 1 RB-GDM Components

RB-GDM controls role-based delegation by means of predicate can_$delegate \subseteq R^l \times R^{l*}$; between two domains of a Grid G. It is a non-reflexive relation since the user in a role cannot delegate his membership to another user in the same role. The meaning of the above relation is that if $u^l{}_1 \in R^l$ and $u^l{}_2 \in R^{l*}$, then $(r^l{}_1, r^l{}_2) \in$ can_$delegate$, i.e., $(u^l{}_2, r^l{}_1) \in UAD$, provided the constraints are satisfied.

TABLE 4: Definition 2 RB-GDM Authorization of Delegation

Each role can have two types of permissions • Delegable permissions PD_l • Non-delegable permissions PN_l From the RBDM0 model, we extend the following permission components for RB-GDM, where P^l is a set of regular permissions • $PA_l = PDA_l \cup PNA_l$, where PA_l,PDA_l and PNA_l are $\subseteq P^l \times R^l$ which are the many to many permission to role assignment relation, many to many Delegable permission - role assignment relation and the many to many Non Delegable permission - role assignment relation respectively. • $P^l(r^l{}_j) = \{p^l{}_k | (p^l{}_k, r) \in PA_l\}$ Similarly for $PDA^l(r^l{}_j)$ and $PNA^l(r^l{}_j)$ as well

TABLE 5: Definition 3 RB-GDM Role Permissions

There are two possible ways in which revocation can be done. The first approach is by revocation using time out by attaching a time constraint to every assigned delegation [6]. Timeout revocation though self-triggering, is not enough to ensure secured delegation. For tasks, which take much longer periods to complete, this method is not suitable. Tracking the delegator is another issue with this type of delegation and revocation. These issues can be handled by explicit revocation (either grant-dependent or grant independent) [10].

The revocation procedure can be through any of the following ways • Remove one or more pieces of permissions from delegation • Revoke delegation role owned by a fixed delegable role • Remove one or more pieces of permissions from a fixed delegable role to its regular (original) role

TABLE 6: Definition 4 Revocation of Delegation

4. RB-GDM INTERCONNECTION FRAMEWORK

We propose three approaches, one for intra-domain access control and the remaining two for inter-domain access control. These interconnection networks are realized based on the definitions (1-4) given in section 3.2. The interactions between the components of the RB-GDM are governed by the algorithms (1-4). The example scenarios are given in section 5. The basic procedure to authorize and revoke delegation is the same for all the three interconnection networks except that RB-GDM components will be organized in different forms in each case.

4.1 Intra Domain Role Based Delegation

It is essentially similar to an enterprise wide role delegation where the entities within a domain delegate tasks to others using a hierarchical relationship. The appropriate association would be

that of the superior entity delegating to the subordinate. Figure 1 illustrates this framework. The RB-GDM core model that supports intra-domain role-based delegation is shown in Figure 2.

4.2 Inter- Domain Role Based Delegation

This method can be realized in two ways; by treating any two domains as equals or peers in the association or by considering a master-slave relationship.

4.2.1 Peer-to-Peer Role Based Delegation

This approach is suitable when the transactions take place between two domains of flat association (for example, between two investment banking domains). This type of delegation is based on the mapping between a role in the delegator domain to its corresponding equivalent role in the delegatee domain, which ensures that role delegation is made to a peer role. Figure 3 represents a peer-to-peer role delegation. The RB-GDM for peer-to-peer inter-domain role-based delegation is as in Figure 4.

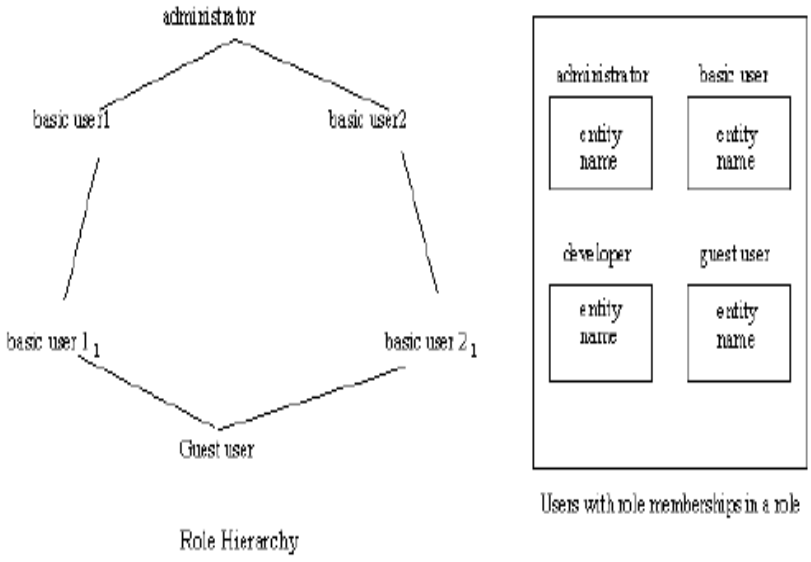

FIGURE 1: Intra-Domain Role Delegation Framework

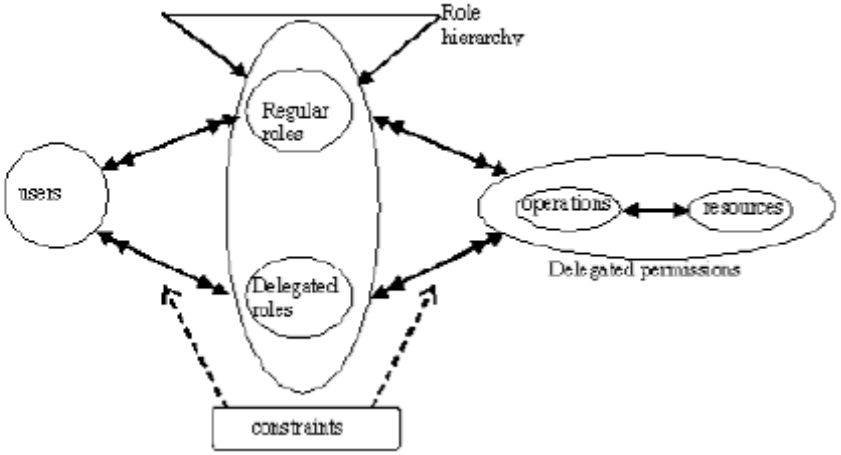

FIGURE 2: RB-GDM Core Model

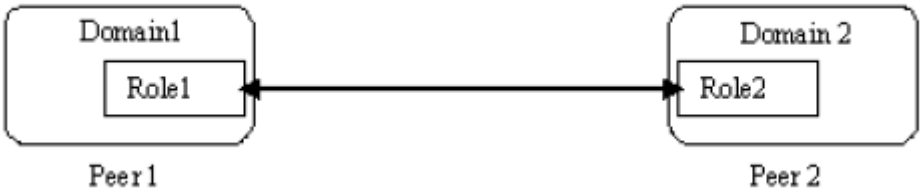

FIGURE 3: Peer to Peer Interconnection Framework for Inter-Domain Role-Based Delegation

Role-based authorization of delegation is shown through the following algorithm1. It shows as to how the permission is associated with the delegated role and how delegation is granted or denied.

Algorithm 1 Role-Based Authorization of Delegation

Input:
[1] A delegation request $(u^l{}_i, r^l{}_j)$
[2] UA_l
[3] PA_l

Output:
True, if the delegation is granted
False, if delegation is denied

Begin

Step 1: $U^l(r^l{}_j) \leftarrow$ get(UA_l); /* from Definition 2, returns the user-role assignment set for the domain*/ UA_l is such that $UA_l = UAO_l \cup UAD_l$

Step 2: for $U^l(r^l{}_j) = \{u^l{}_i | (u^l{}_i, r^l{}_j) \in UA_l\}$

if $P^l(r^l{}_j) = \{p^l{}_k | (p^l{}_k, r^l{}_j) \in PDA_l\}$ /* returns true if the permission is associated with the delegated role and delegation is granted, otherwise the condition returns false and delegation is denied. This decision is guided by the policies of the role-permission relationship set of the given domain*/

if $((u^l{}_i, r^l{}_j) \in can\_delegate)$

Step 3:return true;

else

Step 4:return false;

End

/*Example : Let the delegatee user be in a role ''basic user". It sends a request for delegation along with the role and access specifications. The delegator for example, an administrator maps the user-role relations and the role-permission relations and grants or rejects delegation accordingly.*/

Revocation of delegation is as important and crucial as authorization of delegation. The revocation procedure is done as represented in the algorithm 2.

Algorithm 2 Revocation of Delegation-Grant Dependent

Input:
[1] UA_l
Output:
True, if the delegation is revoked
False, if delegation is not revoked
Begin
if($(u^l{}_i, r^l{}_j) \in can\_revoke$)
Step 1:return true;
else
Step 2:return false;
End

```
/* Example : Let the delegatee user be a process
and the delegator be an administrator. The
execution of the algorithm results in revocation
of delegated rights by the delegator (in this
case the administrator) himself.*/
```

Figure 4 shows the framework for peer-to-peer inter-domain Role-Based Grid Delegation Model.

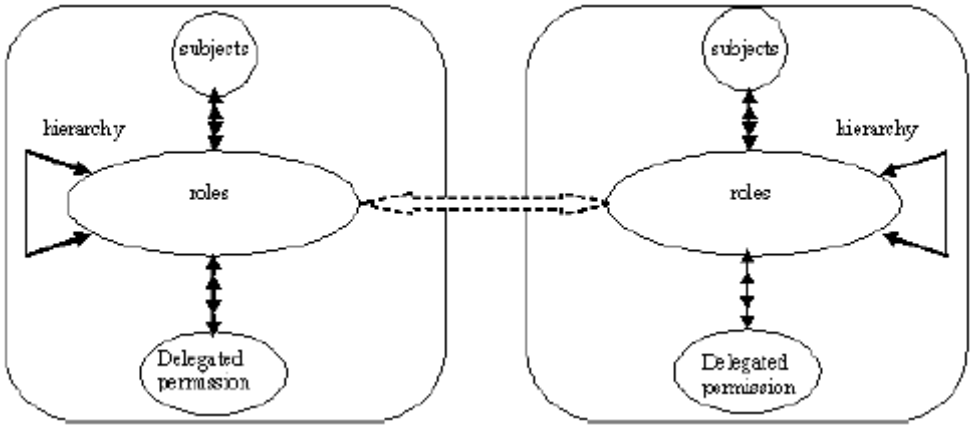

FIGURE 4: Peer-to-Peer Inter-Domain RB-GDM

4.2.2 Master/Slave Role Based Delegation

This approach helps to grant role delegation between two domains, which have a master/slave kind of association between them. It implies that the roles of one domain act as superiors to the roles of the second domain. An example of such an arrangement is the association between a domain representing a central bank and a subsidiary bank domain (hierarchy between two different domains). Typical hierarchical domain relationship can also exist in a Grid between a virtual organization (VO) or virtual domain and the member domains of the VO. Figure 5 shows the hierarchy interconnection framework for inter-domain role delegation.

The following algorithm 3 gives the procedure through which grant-independent delegation is revoked.

Algorithm 3 Revocation of Delegation-Grant Independent

Input:
[1] UA_l
Output:
True, if the delegation is revoked
False, if delegation is not revoked
Begin
if $((u^l{}_i, r^l{}_j) \in can\_revoke^*)$
Step 1: return true;
/*where $u^l{}_i$ belongs to the original user set UAO_l*/ **else**
Step 2:return false;
End
/* Example : Let us consider a user $\{u^l{}_i\}$ in a role ``basic user" and the delegator be another basic user who is granted rights by a site administrator. When this algorithm is invoked, it matches the user (delegatee) with its corresponding role i.e., *guest* and the delegated rights are revoked by the administrator and not by the delegator.*/

Figure 5 shows the hierarchy topology of inter-domain role delegation in the proposed model.

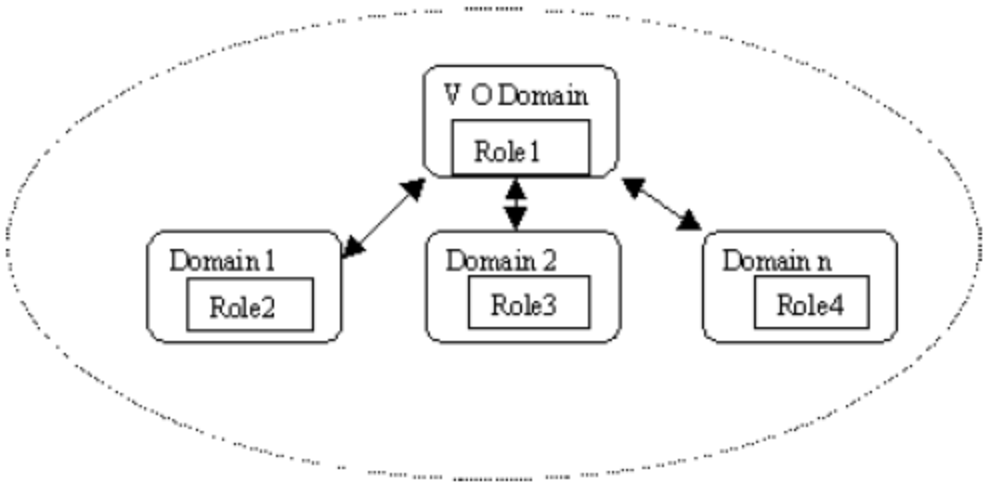

FIGURE 5: Hierarchy Topology of Inter-Domain Role Delegation

Algorithm 4 gives the method for delegation revocation based on timeout concept.

Algorithm 4 Revocation of Delegation-Using Timeout

Input:
[1] UA_l
[2] t (Duration of delegation)
Output:
True, if the delegation is revoked
False, if delegation is not revoked
Begin
if ($(u^l{}_i, r^l{}_j) \in can\_revoke(t)$
Step 1: return true;
/*t is the duration after which the delegation is revoked*/ **else**
Step 2:return false;
End
/* Example : Consider a user $\{u^l{}_i\}$ in a role ``guest". When this algorithm is invoked, it matches the user with its corresponding role i.e., *guest* and if the duration t has expired, the delegated rights are revoked.*/

Figure 6 shows the Master/Slave framework of the Inter-Domain Role-Based Grid Delegation Model.

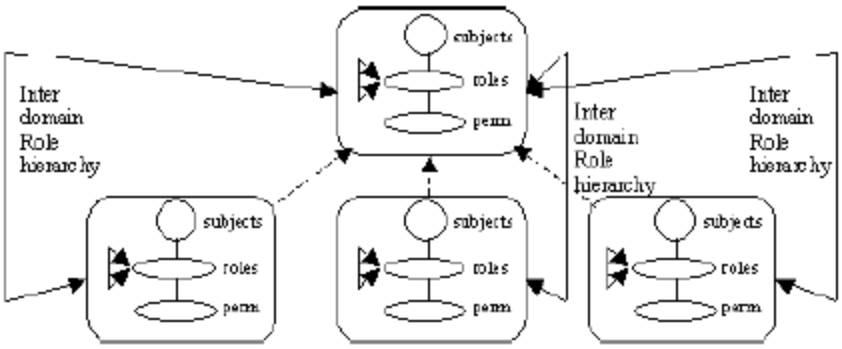

FIGURE 6: Master/Slave Inter-Domain RB-GDM

5. CONSLUSION & FUTURE WORK

The RB-GDM based on the standard RBAC and RBDM models provides a flexible delegation decision-making approach. RB-GDM models various grid delegation requirements like dynamic delegation, partial or restricted delegation and coarse-grained/fine-grained delegation. This framework is an innovative approach towards using role as the criterion for delegation in grids.

Instead of relying on just the identity credentials, our approach uses a role as the delegation authorization credential. By categorizing the grid users into various roles, we achieve granular delegation of access rights. We provide three different inter connection frameworks to suit the

delegation requirements in the intra-domain and inter-domain grid setup. Our model supports both peer to peer as well as hierarchical role relationships. Delegation based on role hierarchies ensures restricted delegated access rights to the resources.

Our model is the first attempt towards conceptualizing grid delegation requirements. We also intend to develop a delegation policy management framework using XACML for policy representation, as it ensures a platform independent and generic approach to policy expressions concerned with the grid delegation entities. We plan to take up the issue of role mapping between various domains in our future work.

Acknowledgement: The first author wishes to express her sincere gratitude to IDRBT (Institute for Development and Research in Banking Technology), Hyderabad, A.P, India for their encouragement and also for providing research grant.

6. REFERENCES

1. Calvelli C. and Varadharajan V, "*An Analysis of Some Delegation Protocols for Distributed Systems*", Proceedings of the Computer Security Foundations Workshop, June 16-18, 2002.

2. David F. Snelling ,Sven van den Berghe and Vivian Qian Li, "*Explicit Trust Delegation: Security for Dynamic Grids*", FUJITSU Sci. Tech. Journal 40(2), 282-294, 2004.

3. Ezedin Barka and Ravi Sandhu, "*A Role-Based Delegation Model and Some Extensions*", Proceedings of the 23rd National Information Systems Security Conference, October 2000.

4. Foster.I, Kesselman.C and Tuecke.S,"*The Anatomy of the Grid: Enabling Scalable Virtual Organizations*", International Journal of Supercomputer Applications, 15(3), 2001.

5. Gasser. M. and McDermott E., "*An Architecture for Practical Delegation in a Distributed System*", Proceedings of the IEEE Symposium on Security and Privacy, 1990.

6. Lalana Kagal, Tim Finin and Yun Peng, "*A Delegation Based Model for Distributed Trust*", Proceedings of the Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents, International Joint Conferences on Artificial Intelligence, Aug 2001.

7. Longhua Zhang, Gail-Joon Ahn and Bei-Tseng Chu, "*A Rule-Based Framework for Role-Based Delegation and Revocation*", ACM Transactions on Information and System Security, 6(3), 404-441, 2003.

8. Sandhu.R, Ferraiolo.D and Kuhn.R, "*The NIST Model for Role Based Access Control: Towards a Unified Standard*", Proceedings of the 5th ACM Workshop on Role Based Access Control, July 26-27, 2000.

9. Weizhong Qiang, Hai Jin, Xuanhua Shi, Deqing Zou, and Hao Zhang, "*RB-GACA: A RBAC Based Grid Access Control Architecture*", International Journal of Grid and Utility Computing, 1(1), 61-70, 2005.

10. Xinwen Zhang, Sejong Oh and Ravi Sandhu, "*PBDM: A Flexible Delegation Model in RBAC*", Proceedings of the Symposium on Access Control Models and Technologies, June 2-3, 2003, Italy, pp 149-157.

11. Ezedin Barka and Ravi Sandhu, "*Role-Based Delegation Model/ Hierarchical Roles (RBDM1)*", Proceedings of the 20th Annual Computer Security Applications Conference, December 6-10, 2004, Tucson, Arizona, USA.