

# Research on access control in grid environment

Wei Rong

Department of Information Technology

Hubei Police University, Wuhan, China

Email: oxfordwr@126.com

**Abstract**—Role-Based Access Control, for short RBAC, is a security technology that ensures system resources can not be accessed by non-authorized users. It can be used in network security and works well for grid security as well. Based on the research of RBAC model at present, the paper proposes a role-based access control model in grid environment. The model still discusses the issues of role authorization rules, permission conflict, and global authorization and so on. At the same time, the related formal description is given. The future work is to realize the model and implement the related system in specific project.

**Index Terms** –grid, access control, RBAC, role authorization, permission conflict

## I. INTRODUCTION

Computational Grid environment is a wide distributed system. It's another strategic infrastructure which can provide high performance computing and information service after the second Internet. And it will do it for every scientific and industry domain in the world. Since computational grid environment equals to a virtual supercomputer, and it has super work power, how to ensure the system security and the effective control of resource becomes very important.

Access control plays an important role in grid system. In computer system security standard, the trusted computer theory attributes the security protection to access control. In trusted computer model, when one subject wants to access some object, it must do it under the control of access privilege so that only authorized users can access system resources.

In grid system, access control includes many different management domains. In each management domain, the resources can be distributed at different hosts with different operation system. In grid security, the most important thing is to ensure that only legal users can be authorized to access grid resources, meanwhile, all unauthorized users can't access grid resources. This function is mainly implemented by access control.

In the paper, we propose an access control model. It can deal with global role and local authorization and how

to eliminate the conflict between them. The novelty of our work lies in the fact that our model can assort with the role mapping among management domains.

## II. ACCESS CONTROL IN GRID ENVIRONMENT

### *A Global roles and local roles*

When a grid system is taken into account, access control will be more complex. Because a grid includes many management domains, and each domain maybe distributes in the distributed network, so grid access control policy will be implemented in global management and local autonomy. The trouble lies in that the grid access control policy allots different access permission and range to various global users in every local area. In order to handle it, we import RBAC policy to grid security and propose a role permission distribution model.

In role-based grid access control system, subject can be grid user, service or user role. User will be assigned to corresponding roles according to his permission and duty. While a user acts as a certain role, he has to be restricted by his access permission. In order to improve the efficiency, we can use the method of role inheritance, which reflects the relationship between role permission and role duty. We use  $(R, < R)$  to express role inheritance relationship.  $\forall r1 \in R, r2 \in R, r1 < R r2 \Rightarrow r1$  is the child role of  $r2$ , of course,  $r2$  is the sire role of  $r1$ .

Global role reflects the global permission in grid system, but practical data operations take place in local management database. So, global permission can work only when it is mapped to local management database. That's to say, a global role must have a corresponding role in local management domain.

There are two methods to define global roles: from-bottom-to-top and from-top-to-bottom. The from-bottom-to-top method defines global roles with

permission according to the local roles that exist in local management domain, as to ensure the local autonomy. The from-top-to-bottom method is that system defines all the global roles one time, and then creates local roles related with global roles. The method ensures role global control. Compared with the two methods, the former greatly holds the local management independency, while the latter works well in practical application.

### B Authorization Rules (AR)

Authorization rules decide the judgment of access permission from subject to object. AR consists of Explicit Authorization Rule and Implicit Authorization Rule, i.e.,  $AR = EAR \cup IAR$ . EAR can be got from global authorization command between subjects, while IAR comes from the authorization transmission between subject and object.

#### 1) Explicit Authorization Rule (EAR)

EAR is the authorization rule that stems from the direct authorization naming or authorization operation transmission. EAR can be described as follows:

$EAR(S, O, Action, Kind\ of\ right, G)$

The meaning is to allow or disallow subject (S) to perform specific operation (Action) on object (O). G is the authorization initiator or grantor. EAR supports two types of AR: positive and negative. When “Kind of right” is “+”, it’s positive authorization, surely, “-” stands up for negative authorization. Positive authorization allows such an operation while negative authorization disallows. All EAR can be expressed as:

$$EAR \subseteq S \times O \times A \times RK \times G \quad (1)$$

In the expression, S is authorization subject, G is authorization grantor, O is authorization object, and A is authorization action such as select, update, insert etc, RK is authorization type, i.e., authorization kind of right,  $RK = \{+, -\}$ .

#### 2) Implicit Authorization Rule (IAR)

IAR generates from the transmission of EAR. IAR includes:

**Rule 1:** A role’s positive authorization implies that all sire roles of this role are authorized the same permission. That is:

$$\forall ar \in EAR, RK(ar) = '+' \Rightarrow \forall r >_R S(ar),$$

$$\exists (r, O(ar), A(ar), -, G(ar)) \in IAR$$

**Rule 2:** A role’s negative authorization implies that all child roles of this role are disallowed the same permission. That is:

$$\forall ar \in EAR, RK(ar) = '-' \Rightarrow \forall r <_R S(ar),$$

$$\exists (r, O(ar), A(ar), -, G(ar)) \in IAR$$

System can ensure that each sire role owns greater permission than the related child role through inheritance policy.

#### 3) Permission Conflict and its elimination

A role’s permission can be endowed with positive or negative authorization; what’s more, it can get from the inheritance of explicit authorization or implicit authorization. Because of different authorization methods, a role’s privileges will easily conflict. Then, a conflict mechanism is required to handle such a problem.

Definition: Permission Conflict is that two authorization clauses have same subject, same object and same action, yet different kind of right. It is expressed as  $conflict(ar1, ar2)$ . That is :  $\forall ar1, ar2 \in AR, conflict(ar1, ar2) \Rightarrow S(ar1)=S(ar2) \wedge O(ar1)=O(ar2) \wedge A(ar1)=A(ar2) \wedge RK(ar1) \neq RK(ar2)$

When permission conflict takes place, system will take some measures to eliminate it in order to know the final privileges subject should have to object. One effective rule is to endow all the privileges with different PRI so that system will hold the permission with high PRI and eliminate low permission.  $(AR, <)$  is used to show the permission PRI relationship. Assume two permission  $ar1$  and  $ar2$ , if  $ar1 < ar2$ , then  $ar1$ ’s PRI is higher than  $ar2$ , so  $ar2$  is kept while  $ar1$  will be lost. There are two rules:

**Rule 1:** If two authorization records conflict, the one of Explicit Authorization has higher PRI than the one of Implicit Authorization. That is :  $\forall ar1, ar2 \in AR, (conflict(ar1, ar2) \wedge ar1 \in IAR \wedge ar2 \in EAR) \Rightarrow ar1 < ar2$

**Rule 2:** If two authorization records both are Explicit Authorization or Implicit Authorization, and they conflict, the one of negative authorization has higher PRI than the one of positive authorization. That is :  $\forall ar1, ar2 \in AR, ((ar1 \in EAR \wedge ar2 \in EAR) \vee (ar1 \in$

$$IAR \wedge ar2 \in IAR) \wedge \text{conflict}(ar1, ar2) \wedge (RK(ar1) = '+' \wedge RK(ar2) = '-' ) \Rightarrow ar1 \prec ar2$$

The two rules show that EA has higher PRI than IA; negative authorization PRI is higher than positive authorization.

#### 4) Global Authorization and Local Authorization

When a grid system is taken into consideration, the keystone of access control is to transform grid global authorization to local authorization in local autonomy and to keep global authorization up with local database. When a global authorization happens, system will decompose the related subject and object to be local subject and object. Meanwhile, the decomposed result will be passed to local management area database. Then, a series of corresponding local authorizations between the decomposed subject and object generate. A global authorization consists of many local authorizations, which are not the same. Obviously, the success of global authorization depends on the success of all the local authorizations. There are some problems, for example, some local authorizations may succeed while others fail.

**Global Authorization Rule:** A global authorization succeeds only if all the decomposed local authorizations succeed. Use “global\_ar” to denote a global authorization while “localize (ar)” to be decomposing global authorization, succeed (ar) to be authorization success. That is:

$$\forall \text{globe\_ar} \in AR, \text{succeed}(\text{globe\_ar}) \Leftarrow \forall ar \in \text{localize}(\text{globe\_ar}), \text{succeed}(ar)$$

The transform between global and local authorization effectively prevents permission conflict. If local management domain changes the permission of local role, system will update the corresponding global role according to authorization rules and conflict rules.

**Local Authorization Rule:** Local authorization succeeds only when successfully updating all the corresponding global role authorization. Use “local\_ar” to denote a local authorization, global (ar) denotes all the related global authorization, succeed (ar) denotes an authorization success. Then:  $\forall \text{local\_ar} \in AR, \text{succeed}(\text{local\_ar}) \Leftarrow \forall ar \in \text{global}(\text{local\_ar}), \text{succeed}(ar)$

#### C Access Control Model in Grid Environment

In a grid system with many local management domains, global users and local users are in two different

levels. Global users are managed by global grid administrator while local users, local autonomy system. It's the role that relates the global user with local user. Global grid system has great flexibility to manage global users and make global security policy without affecting local database security control to local users. Fig.1 is a relationship between global users and local users.

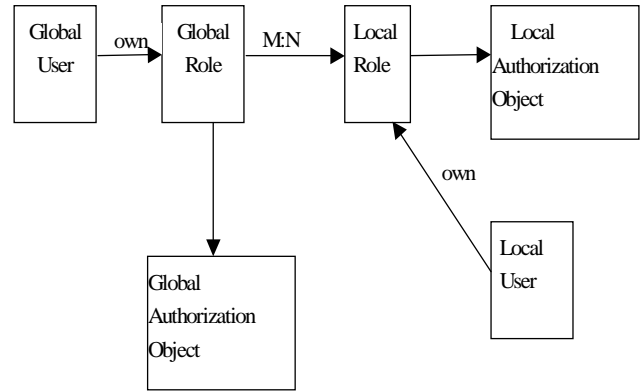


Figure 1. Global User—Local Relationship

In order to keep the autonomy of every local management domain, grid access control model is generously designed with the method of from-top-to-bottom. First, global roles are made out according to practical work. Then, role permission authorization requirements are submitted to every local management domain administrator. At last, local management domain administrator transforms the global roles to local roles and specific local users in the light of such requirements. How to relate local role with global user is up to local management domain administrator, which makes local management autonomous. Compared to traditional management method that global system specifies the relationship of global users and local users, the method has some advantages as follows:

- In a grid system, there are a lot of local management domains and these domains are often geographically distributed. In order to associate global users with local users, global system has to clear about some local users of every local management domain as well as their corresponding permissions. It's hard and difficult for global system. When introducing role and using the method of from-top-to-bottom, it's easy for global system to realize.

- Local users are managed by local domain administrator so that their privileges can be created, deleted at will. If global system relates global user with local users, it's impossible to find the modification of local users or local users' privileges so that a lot of global operations will fail to perform.

From the above, grid system should let local management domain administrator realize the relationship between global and local transform as to implement a free privileges distribution in grid environment. Global system only cares about global roles. It asks local management domain to endow global roles with some specific permissions. Local management domain administrator relates the global with local roles according to the requirements. Table 1 is such an example.

TABLE 1 ROLE CORRESPONDING TABLE

	LR1	LR2	...	LRn
GR1	√	√		
GR2		√		√
...				
GRn				

A common grid system access control model is designed and shown in fig.2. In the model, global system defines some global roles and gives the permission requirement information. After getting the requirements, local management domain administrator sets some corresponding local roles and relates them with global roles. Local management domain administrator will relate the local roles with local users. This is the working theory of our grid system access control model.

### III. CONCLUSION

Access control is an important approach to grid system, so it's meaningful to introduce access control to enforce grid system security. This paper analyses three popular access control policies, and proposes authorization rules and conflict rules according to access control characteristic. At the same time, a new access control model is given to adjust to privilege distribution

policies with many local management domain systems. Our future work is to handle security policies negotiation among various management domains.

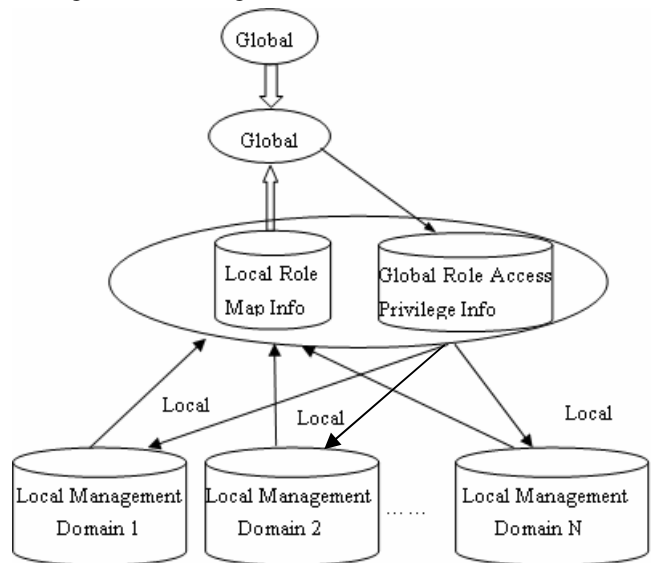


Figure 2. Grid System Access Control Model

### REFERENCES

- [1] Hong Fan, He Xubin, and Xu Zhiyong. "Role-Based Access Control". *Mini-Micro Systems*, Vol. 21, pp. 198-200, February 2000.
- [2] Nagaraj, S.V., "Access control in distributed object systems: problems with access control lists". *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2001. Proceedings. Tenth IEEE International Workshops on 20-22*, pp. 163-164, June 2001.
- [3] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman. "Protection in operating systems". *Communications of the ACM*, vol. 19, pp. 461-471, August 1976.
- [4] Moyer MJ, Ahamad M, "Generalized Role-based Access Control". *21th IEEE International Conference on Distributed Computing, Sydney, Australia, 2001*, pp. 67-68.
- [5] Johnston W, Mudumbai S, and Thompson M, "Authorization and Attribute Certificates for Widely Distributed Access Control". *15th IEEE International Conference on Data Engineering, San Diego, CA, 1998*, pp. 158-161.