A Dynamic Role-Based Authorization Model in Grid Environment

Weifeng Sun[1], Cheng Guo[2], Peng Zhang[1], Ning Zhang[1], Haotian Wang[1]
[1]School of Software
Dalian University of Technology
116620 Dalian, China
wfsun@dlut.edu.cn, {pengzhang, zhang_ning, wanghaotian}@mail.dlut.edu.cn

[2]Department of Computer Science
National Tsing-Hua University
30013 Hsinchu, Taiwan
guo8016@163.com

Journal of Digital Information Management

ABSTRACT: *In large-scale grid environment, the authorization plays a vital role in access control to resources. For the demand of higher dynamicity, complexity and granularity on grid environment, a novel model named dynamic role-based access control (DRBAC) based on RBAC model is presented. DRBAC introduces conceptions of several objects such as role-graph initial structure, atom role and middle role, and it can automatically adjust the role-graph's structure through dynamically adding new roles or deleting original roles. The authorization mechanism which combines DRBAC model and CAS servers provides an effective method to solve loading and security problems. Theoretical analyses and example demonstrate that it is of high safety and has good time and space complexity when authorizing.*

Categories and Subject Descriptors
D.4.6 [**Security and Protection Access controls**]; **I.2.8** [**Problem Solving, Control Methods, and Search**]; **I.3.6 [Methodology and Techniques]:** Graphics data structures and data types

General Terms
Grid Computing, Access Control, Server Security

Keywords: DRBAC, Dynamic role, CAS server

Received: 12 June 2011, Revised 12 August 2011, Accepted 21 August 2011

1. Introduction

Large-scale, dynamic and cross-domain resource sharing and collaboration is an important way to improve the ability of network applications. This kind of resource sharing and collaboration is not simple file exchanges, but more about direct access to computers, corresponding software, data or other resources. The sharing must be clearly defined and highly controlled by resource providers and resource consumers. Authorization is a user access control mechanism for resources in grid environment. By verifying the identity, ability of the user and context, virtual organization authorizes the user certain rights to access to some specific resources.

Access control is an important part of assuring information security, and is used to present access control strategy of system security. Also, access control makes consistency verification between strategies and provide effective mechanisms for implementation of access control strategy. The main method of implementing access control strategy is authorization. In large-scale grid environment, authorization plays a key role in the process of resource access control. Authorization in grid environment [1, 2] is different from distributed network. Authorization mechanism in grid needs to be more dynamic and flexible and have higher demands on granularity of authorization.

Argonne National Laboratory in United States have developed CAS (Community Authorization Service) [3] authorization model based on Globus project, and European Grid Organization have put forward VOMS (Virtual Organization Membership Service)[4] authorization model based on Datagrid project. In 2006, Anil L.Pereira etc. [5] combined CAS authorization model and RBAC authorization model to solve the problem that resource providers cannot update map files flexibly and synchronously, which reduces the burden of resource providers and improve flexibility and synchronization of authorization model. However, the authorization model brings new problems, [6] such as, the user needs multiple local roles when he is executing a task, then the CAS server has to generate corresponding CAS proxy certificates for every local role, which increases the burden of CAS servers. The maintenance of resource providers for switching between multiple roles and monitoring also consumes many system resources .

This paper proposes a dynamic role-based access control (DRBAC) model based on RBAC model. By assigning to users the common father role of each atom role needed by users or the dynamic-combination-generated father role. The DRBAC model reduces the amount of proxy certificates generated by CAS servers so as to reduce the load. In grid environment, the combined authorization

mechanism between DRBAC model and CAS model has several advantages compared with traditional RBAC and CAS model including the low load of CAS servers, flexibility and convenience of management, and meeting "the principle of least privilege".

The second part of this paper introduces the progress of related work. The third part gives dynamic role-based access control model, and makes some corresponding discussions on the security assurance with CAS model. The forth part we analyze the performance of DRBAC and use an example to prove it. Finally there is the conclusion and future work.

2. Related Work

CAS authorization model provided by Globus uses "push" authorization mode. It introduces a trusted third-party CAS server to take management of the resource access control strategy. If a user need operate or get access to some resources in virtual organizations, he has to send a request to the CAS server to get a permission of a series of operations. The CAS server will verify the user's identity and his request with the descriptions in strategy library to check whether they are consistent or not. If they are consistent, the CAS server will grant a certain permission to the user.

In 2005, Jiageng Li and David Cords put forward a method of extending globus's Metacomputing Directory Service (MDS) [7] to solve the problem of scalable authorization in grid dynamic environment. J.R.Burruss etc. proposed a grid-based ROAM authorization management framework [8] based on VOMS authorization model. However, the model is lack of good scalability and wide applicability. Paper [9] makes an adequate explanation of workflow-based authorization method which gives a corresponding authorization method for event sequence and a solution to "the principle of least privilege".

In 1992, D. Ferraiolo and Kuhn brought forward RBAC model [10]. RBAC model implemented the logical separation between users and permissions by introducing the intermediary of role and simplified the authorization management in a variety of environments. R. Sandhu etc. [11] proposed RBAC96 model and this model gave a general role-based access control framework and four types of conceptual models. On the basis of the original RBAC0, it gave a complete description of the role stratification and restriction. In 2001, D. Ferraiolo and R. Sandhu etc. put forward NIST standard [12] of RBAC model. G.J. Ahn and R. Sandhu brought the definition of role into authorization mechanism, and in paper [13], they gave a role-based constraints language - RCL2000 which is used in authorization constraint. However, in RBAC model, the hierarchy of roles is fixed, which decreases dynamicity and flexibility. Traditional RBAC model is more suitable for access control object and specific-operation system, and it is not enough in the open grid environment.

In order to improve dynamicity and flexibility of RBAC, E. Bertino and James B.D. Joshi etc. successively put forward TRBAC (temporal RBAC)[14] and GTRBAC(generalized temporal RBAC) model [15]. They improved flexibility of the model by time constraint, and used trigger mechanism to activate roles dynamically, thus implemented dynamic assignment of roles and permissions. However, TRBAC model and GTRBAC model had not solved the problem caused by combination of traditional RBAC model and CAS authorization mechanism. Although dynamic activation of roles can improve dynamicity and flexibility of the original RBAC model, multiple local roles are inevitably generated. CAS servers also need to generate multiple CAS proxy certificate correspondingly, which increase the burden of servers.

Under this kind of circumstances, this paper proposed an improved model on the base of the original RBAC, not only to retain the original characteristics of RBAC but also solve a specific problem [16] of too much burden on servers, and the analysis shows that this scheme has better security and better flexibility. Well worthy reference is contributed to the future research on how much permission is given to objects and how to change permission according to dynamic variation of conditions and tasks in the complex grid environment [17].

3. Dynamic Role-based Access Control Model DRBAC

3.1 Basic Concept of DRBAC

In DRBAC, r means roles; P means permission; P(r) means permission of r; r.parents means the father role of r, i.e. the role in upper level of role r; r.children means the child role of r, i.e. the role in lower level of role r. The relationship between them can be expressed as formula 1: the CAS server to get a permission of a series of operations. The CAS server will verify the user's identity and his request with the descriptions in strategy library to check whether they are consistent or not. If they are consistent, the CAS server will grant a certain permission to the user.

$$P(r.children) \subseteq P(r) \subseteq P(r.parents) \qquad (1)$$

Definition 1

Atom Role: r_i denotes atom role, every atom role should be corresponding to specific permission, and their permission is not overlapping. They can create father roles by combination, meanwhile they can have multiple father roles.

Definition 2

Middle Role: Middle roles represent the roles in upper level of atom roles, and they are created by combining atom roles. However, they are not the roles in the top level. Middle role can have multiple levels, and every middle role can have multiple father roles and multiple child roles. In this paper, we use r_m to represent middle roles.

Definition 3

Temporary Role: Because in original role graph there are no roles which are corresponding to permissions required by users, new roles generated by combination of atom roles which are corresponding to these permissions or middle roles are so-called temporary roles.

In the combination process of atom roles, the system will not combine those roles who have permission conflict according to the principle of SoD(separation of duty) [18].

Definition 4

Root Role: Root roles are at the top level of role hierarchy, and they are created by combination of middle roles or atom roles. As a special case, they are also can be created by a single role, witch in this way are not only root roles but also atom roles. Just as atom roles create temporary roles, there exist exclusion relations between roles and roles. Therefore, the system in this paper allows multiple root roles to exist at the same time. In this paper, R represents root role.

The hierarchy of root role, middle role and atom role are shown in Figure 1.

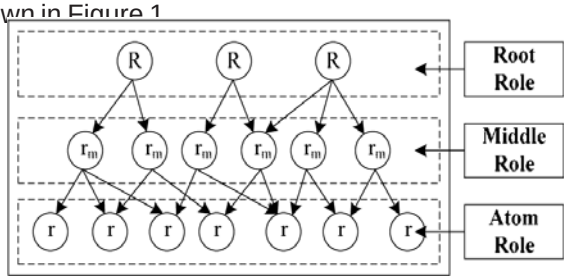


Figure 1. Role hierarchy of the DRBAC role graph

3.2 DRBAC Model

DRBAC model is based on a dynamic graph structure. Its lowest layer has many atom roles, which are the smallest division of a role and have atomic features. One atom role is corresponding to single permission, and all of their permissions are not overlapped. Atom roles are combined to create middle roles or root roles. One atom role can have more than one father roles and middle roles. Root roles are at the end of this role directed graph and have the highest permission. As discussed in the previous part, a system probably has multiple root roles.

When a user need to complete a task, the system will match the user's identity and the task that needs to be done with the corresponding permissions of resource providers, and make them correspond to appropriate atom roles. In this course, the user perhaps needs multiple atom roles. In order to avoid that CAS servers generate corresponding CAS proxy certificate for every role and the resource providers have to maintain too many roles, DRBAC combines multiple atom roles to generate a role dynamically, and assigns this role to the user. Thus CAS servers only need to issue a corresponding proxy certificate to this new role, and at the same time, resource providers reduce the burden of role maintenance.

The detailed processes of DRBAC are:

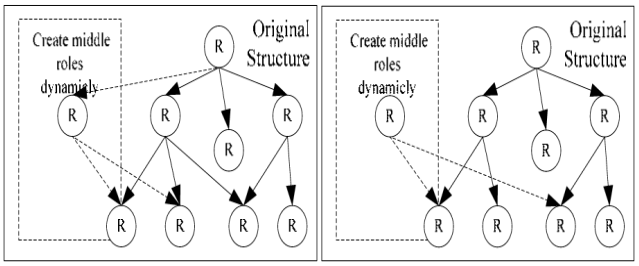
3.2.1 Initial Structure of Role Graph

System administrators develop atom role according to the system and commonly used services at first, then develop middle role and root role according to the relationship between atom roles and possibly existing mutual cooperation, and determine the relationship between them and create role graph. This part is called initial structure or static structure of role graph. It will not change dynamically in dynamic adjustment procedure of role graph. Adjustment of initial role graph needs administrators to complete regularly, because the design of initial role graph will probably be found to be unreasonable in later use.

After a user has proposed its task request, the system makes a judgment on the atom roles or middle roles that are needed by the task at first. The system matches roles according to the initial structure of role graph, and then searches for the common father roles of these roles. If the father roles are found, CAS servers will issue the corresponding CAS proxy certificate to the user.

3.2.2 Dynamic Adjustment of Roles

If initial structure of role graph doesn't include such father roles or their common father roles also include other child roles besides themselves, which is contrary to "the least privilege", we will need to generate a new role or a middle role dynamically, as shown in figure 2.



(a) No common father role (b) Contrary to the least privilege

Figure 2. Dynamic role creation

In the creating process of the new role, we consider two situations. The first situation is creating a temporary role. The characteristic of this kind of role is relatively temporary; the lifetime of the role is determined by the time the user holds this role. When the user complete the given task and doesn't need the permissions of this role any more, this role will delete itself in the role graph automatically. The temporary creation of this kind of role is mainly aim at the characteristic that user's requests for these atom roles is an accidental event and doesn't happen frequently or cyclically. The temporary role is not assigned to corresponding father role. Table1 gives the formal description of the first situation. *RA* (role array) represents request sequence of roles, $Fre(U_i, RA_i)$ represents frequencies of different users' request sequence RA_i for roles, $T_{threshold}$ and n respectively represent the time that threshold of role request set by the system automatically and threshold of times. $active\_add(r_i, r_j, ...r_k, r)$ represents temporal role r created by atom role sequence $(r_i, r_j, ...r_k)$ activation, and $inactive\_delete(r)$ represents delete of temporary role r from role graph. The symbol $\preceq$ represents preference relation between roles levels, $role_1 \preceq role_2$ represents $P(role_1) \subseteq P(role_2)$. $\preceq_A$ represents activation, temporary active created role hierarchy, $\preceq_I$ represents inheritance, relatively steady hierarchy created in dynamic role graph.

The time period of role request is described formally as $P = \sum_{i=1}^{n} O_i.C_i \triangleright x.C_d$ where $C_d, C_1, ..., C_n$ represents different time period, such as year, month, hour etc. O_i represents the specific value. RR(role request) represents user's role request in this paper. Table2 gives an example of periodic description of role request sequence.

In table 2, $<RA_1, r_1>$ {all. Years + {3,7}Months ▷ Months} represents that user's role request sequence is combined as temporal role r_1. Time is since March and July every year, and the period is two months.

The second situation of dynamic role creation is when atom roles are constantly requested by different users or they are used frequently, cyclically by a single user, the system will generate middle roles instead of temporal ones automatically, which will be maintained for a long time in order to be assigned to roles conveniently afterwards. Based on the containing relationship of permission of roles, the middle roles can be appointed to corresponding father roles, forming new hierarchy. Table3 shows the formal description.

Function	Conditions	Rules
$active\_add(r_i, r_j, ... r_k, r)$	$\{P(user\_request) = P(r_i) \cup P(r_j) \cup \cdots \cup P(r_k)\}$ $\left\{ r \in R \middle\| \exists r \left[\begin{array}{l} (r_i.parents = r_j.parents = \cdots = r_k.parents = r) \cap \\ (P(r) = P(r_i) \cup P(r_j) \cup \cdots \cup P(r_k)) \cap \\ (P(r) - (P(r_i) \cup P(r_j) \cup \cdots \cup P(r_k))) = \varnothing \end{array} \right] \right\}$ $O_i.C_i \triangleright x.C_d \le T_{threshold}$ and $Fre(U_i, RA_i) \le n$	$\begin{cases} r_i.parents = r; \\ \cdots \\ r_k.parents = r. \end{cases}$ $r_i, r_j, \cdots, r_k \preceq_A r$
$inactive\_delete(r)$	$T = \sum_{i=1}^{n} O_i.C_i + x.C_d$	$r_i, r_j, \cdots, r_k \preceq_A r$

Table 1. Formal specification of temporal activation role

Role Request	Example of the periodic time expression
RR_1	$< RA_1, r_1 > \{all.Years + \{3,7\}.Months \triangleright 2.Months\}$
RR_2	$< RA_2, r_2 > \{\{2006, 2007\}.\text{Years} + \text{all}.Months + \{1,10\}.\text{Days} \triangleright 2.\text{Days}\}$

Table 2. Example of the periodic expression

Function	Conditions	Rules
$add(r_i, r_j, ... r_k, r)$	$\{P(user\_request) = P(r_i) \cup P(r_j) \cup \cdots \cup P(r_k)\}$ $\left\{ r \in R \middle\| \exists r \left[\begin{array}{l} (r_i.parents = r_j.parents = \cdots = r_k.parents = r) \cap \\ (P(r) = P(r_i) \cup P(r_j) \cup \cdots \cup P(r_k)) \cap \\ (P(r) - (P(r_i) \cup P(r_j) \cup \cdots \cup P(r_k))) = \varnothing \end{array} \right] \right\}$ $O_i.C_i \triangleright x.C_d > T_{threshold}$ and $Fre(U_i, RA_i) > n$	$\begin{cases} r_i.parents = r; \\ \cdots \\ r_k.parents = r; \\ r.parents = r_m \end{cases}$ $r_i, r_j, \cdots, r_k \preceq_I r$
$delete(r)$	$O_i.C_i \triangleright x.C_d \le T_{threshold}$ and $Fre(U_i, RA_i) \le n$	$r_i, r_j, \cdots, r_k \preceq_I r$

Table 3. Formal specification of middle role

3.3 Combination of DRBAC and CAS

In original CAS model, CAS proxy certificate can make user map to local account of resources, and can restrict user's ability and operation of resource nods according to user's identity and tasks needed to be done. CAS authorization model belongs to "push" mode and is passive for users, and doesn't have good access control and initiative. Meanwhile, when new users join in or users' permissions are modified, resource providers also need to update map files in real time.

With the combination of CAS model and RBAC model, these two situations can be improved effectively. With users mapping to CAS global roles and global roles mapping to local roles of resource providers, resource providers can restrict users' access to resources by setting their own role access control list, which meets

grid security mechanism in maximum, that is, in the premise of not breaking local security mechanism. Also, global roles in VO layer can control users' permission and access control granularity, which can decrease corresponding burden for resource providers. When users' access control information changes or new users join in, we just need to adjust CAS servers instead of modifying role access control lists of the resource providers in real time. However, when a user needs multiple roles in a task, CAS servers have to create multiple CAS proxy certificates for the user, which will not only incease the burden of CAS servers, but also increase the burden of maintenance of multiple roles by resource providers.

DRBAC model proposed in this paper can complete grid authorization effectively combined with CAS servers, and in the meantime, it enhances the safety and reduce the difficulty of maintaining roles by resource providers. The definite data flow is shown in figure 3.

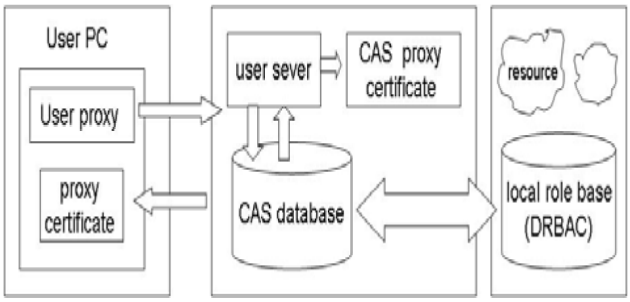


Figure 3. Role hierarchy of the DRBAC role graph

3.4 Security Analysis of DRBAC

3.4.1 The Principle of Least Privilege

"The principle of least privilege" is one of the most basic principles in system security and the essential privilege of every subject (user or process) in grid when completing some kind of operation. That is to say, user will not be given too much permission. The DRBAC model creates atom roles by dividing traditional RBAC roles, which makes role assignment granularity smaller to reach a state of undividable permission, and then becomes more beneficial for the implementation of "the principle of least privilege". The mechanism of creating temporal roles and middle roles dynamically in DRBAC model makes that other atom roles will not be assigned to user agent, therefore this model can meet "the principle of least privilege" well.

3.4.2 Security Issues of Roles

In real systems, the use of some roles has an order. For example, if a user corresponds to role A at first, and then corresponds to role B, there exist no security risks in this situation. However, if the user corresponds to role B, and then corresponds to role A, there exist potential security risks in this situation. There are also some roles that are time limited or mutually exclusive. For example, a user can't own the role of referee and the role of athlete at the same time. Therefore, in the creation process of initial role graph and the combination process of dynamic roles, these child roles can't be classified as the same role. That is to say, if in the process of user request for permission, there exists exclusion or collaborative cheating between roles, which leads to the result that these roles can't be owned by the same user, then this paper still assigned these roles to user agents instead of combining them, and CAS servers issue multiple corresponding CAS proxy certificates. Resource providers also use corresponding work-flow-based RBAC [19] access control model to monitor these roles owned by users, so as to make sure that users won't use these roles simultaneously or cheat together.

4. Performance Analysis and Example Declaration

4.1 Analysis of Time Complexity

In DRBAC, time complexity of role searching and dynamic creation is a key problem for resource providers. Resource providers need to maintain system role library. When users request for roles, resource providers search for the common father role of different atom roles and create new middle roles dynamically when necessary. In the maintenance process, resource providers also need to delete some of middle roles. For resource providers, this paper considers their load problem comes from the aspect of time complexity. Time complexity of the best case of role matching in the model is $O(1)$, and the worst case of role matching in the model is $O(n)$, where n is the number of atom roles.

In RBAC users must request a certificate from the source providers. Such cause a huge cost for the network itself. While in DRBAC, the network latency is greatly reduced due to the reduction of the direct requests from the source providers. Thus makes the DRBAC has a better resistance to the mutative net work than the RBAC.

4.2 Analysis of Space Complexity

In the worst case, system will generate a corresponding temporal role or middle role every time when users request a role array. As a result, role space expands. This case is possible to happen during the primary system time period. Users' temporal request roles all have certain time limits, and once exceeded, the temporal roles will be deleted automatically.In a stable system, users' requests for some certain role arrays must be frequent events, while requests for some other role arrays are accidental events. With increasing of time period, role arrays that are frequently requested for will turn into stable middle roles and the original middle roles that are not frequently used will be deleted initiatively by the administrator when updating the system. Thus, after many time periods, the size of role space in the system will converge on that of middle roles and a certain number of dynamic temporal roles. Then, as some temporal roles are deleted and new ones are created, the size of role space waves in a certain convergence domain.

4.3 Example Declaration

In figure 4, the roles encircled by dotted lines are temporal roles. The example below simplifies comparison among

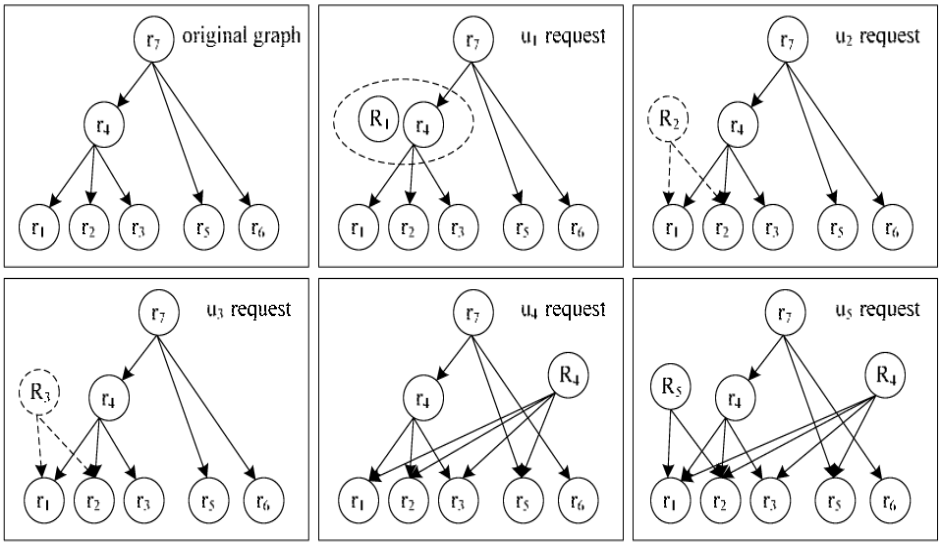


Figure 4. A graph of a DRBAC application

conditions such as frequency of role request, time threshold assigned to the role, etc.

As shown in figure 4, user 1 u_1 's request sequence <r_1, r_2 ,r_3 > is corresponding to role r_4 in the original role graph. So thanks to the design of role graph, no other new roles are needed. While u_2 ,u_3 are requesting for role array <r_1, r_2 > , the system generates temporal roles R_2 and R_3 because request frequency and request time are below the demand of threshold. When user u_5 requests for <r_1, r_2 > again, the system generates a middle role R_5 as a result of higher request frequency of these two roles than the threshold set by the system. Also for the same reason, the system generates the middle role R_4 consequently when user u_4 requests for <r_1, r_2, r_3, r_4, r_5>.When authorizing, the CAS server just needs to grant certificates to the temporal roles, middle roles and corresponded existing father roles instead of every atom role that is requested for.

In this example of the application with five users, the number of CAS proxy certificates is decreased about by half through dynamic creation. The optimized effect is notable. Though there are additional two temporal role spaces and two static role spaces created, after a long time, the role spaces will converge on a stead size because less and less new roles need to be created. The burden of CAS servers will be effectively reduced in order that the extra time and space cost will be ignored. Later adjustments to role graph will make it more suitable to the vast user group and more secure and convenient for resource providers to maintain roles.

The above example well illustrates the process of role combination, and further analysis explains that DRBAC is able to avoid producing CAS proxy certificates and the management of time threshold can well control the use for roles and the maintenance of role space. Meanwhile, due to the introduction of conception of atom role, the role granularity becomes smaller,then through dynamic combination, making authorization process meet "the principle of least privilege".

5. Conclusion

This paper puts forward a dynamic role-based authorization model, which combines DRBAC access control model and CAS model.This model creates temporal roles or middle roles to reduce role numbers assigned to users by dynamic combination of atom roles so as to decrease the number of CAS proxy certificates, preventing high load on CAS servers.

Security analysis shows that this model can meet "the principle of least privilege" well, and implement part-of-permission authorization in traditional RBAC model. Also the reasonable initial design of structure of role graph and later adjustment guarantee the security of the model. With the help of monitoring by resource providers, the risk of collaborative cheating is reduced to a minimum. Moreover, the minimized granularity contributes a lot to the authorization in grid environment due to its cross-domain resources and high complexity.

Meantime, we can't overlook the extra cost that DRBAC brings about. Creating new roles also speeds execution time and takes up system space, especially in the beginning period of the long-lasting system. However, in a large and stable system, compared to the advantages of this model, the extra cost seems too trivial.

6. Acknowledgment

This work is supported in part by Natural Science Foundation of China under Grant No.61103233, Nature Science Foundation of China-Japan Science and Technology Agency under grant Project No.51021140004, partially supported by Nature Science Foundation of China under grant No.90715037, 61070181 and 60903153.

Reference

[1] Stell, A.J., Innot, S R.O., Watt, J.P. (2005). Comparison *of advanced authorization infrastructures for grid computing. In:*Proceedings of 19th International Symposium on High Performance Computing Systems and Applications (HPCS'05*)*,Guelph, Ontario, Canada, 195-201.

[2] Cornwall, L.A., Jensen, J., Kelsey et al, D.P. (2004). Authentication and authorization mechanisms for multi-domain grid environments. *Journal of Grid Computing*, 2 (4) 301-311.

[3] Pearlman, L., Welch, V., Foster et al, I. (2002). A community authorization service for group collaboration. *In:* Proceedings of IEEE 3rd Intl. Workshop on policies for distributed systems and networks, Monterey, California, 50-59.

[4] Alfier, R., Cecchini, R., Ciaschini et al, V. (2003). VOMS, an authorization system for virtual organizations. *In:* Proceedings of the 1st European Across Grids Conference, Santiago de Compostela, Spain, 33-40.

[5] Pereira, A.L., Muppavarapu, V., Chung, S.M. (2006). Role-based access control for grid database services using the community authorization service. *IEEE Transactions on Dependable and Secure Computing*, 3 (2) 156-166.

[6] Xiao-jun Zhu, Shi-qin Lv, Xue-li Yu, Guang-Ping Zuo. Dynamic Authorization of Grid Based on Trust Mechanism, *In:*IPTC '10 Proceedings of the 2010 International Symposium on Intelligence Information Processing and Trusted Computing.

[7] Li, J.G., Cords, D. (2005). A scalable authorization approach for the globus grid system. *Future Generation Computer Systems*, 21(2) 291-301.

[8] Burruss, J.R., Fredian, T.W., Thompson, M.R. (2006). ROAM: An authorization manager for grids, *Journal of Grid Computing*, 4 (4) 413-423.

[9] Kim, S.H., Kim, K.H., Kim et al, J. (2004). Workflow-based authorization service in the grid, *Journal of Grid Computing*, 2(1) 43-55.

[10] Ferraiolo, D., Kuhn, R. (1992). Role-based access control. *In:* Proceedings of 15th National Computer Security Conference, Gaithersburg, Md., 554-563.

[11] Sandhu, R., Coyne, E.J., Feinstein et al, H.L. (1996). Role-based access control models. *IEEE Computer*, 29 (2) 38-47.

[12] Ferraiolo et al, D.F. (2001). Proposed NIST standard for rolebased access control. *ACM Transactions on Information and System Security*, 4 (3) 224-274.

[13] Ahn, G.J.,Sandhu, R. (2000). Role-based authorization constraints specification, *ACM Trans. Information and System Security*, 3 (4) 207-226.

[16] Lason, Martin. (2010).Role-based access control in software services: theory and practice. *In:* Proceedings of 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2010, 465-470.

[17] Gouglidis, Antonios., Mavridis, Ioannis. (2011). Role-Based Secure Inter-operation and Resource Usage Management in Mobile Grid Systems. *Lecture Notes in Computer Science*, 29-36.

[18] Gligor, V.D., Gavrila, S.I., Ferraiolo, D. (1998). On the formal definition of Separation-of-Duty policies and their composition.*In:* Proceedings of 1998 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, Calif., 1998, 172-183.

[19] Bouchahda, Ahlem. (2010). RBAC+: Dynamic access control for RBAC-administered web-based databases.*In:* Proceedings of 4th International Conference on Emerging Security Information,Systems and Technologies, SECURWARE 2010, 135-140.