A Security Model Based on User Group and Role Administration

Quandeng Gou, Yihe Liu*
College of Computer Science, Neijiang Normal University, Neijiang 641100, China
liu_yihe@163.com

Abstract—The RBAC (Role-Based Access Control) is an important information security model, and it is also a hot issue in the domain of information security in recent years. A security model based on user group and role administration have been given out in the paper which is according to the practical application of the problem that is the actualization of role administration in our school informationization application system, and from the angle of guarantee the information security. The rules in the new model are reasonable and secure by analysis. It will play a positive role to construct information security model, and also to promote the general role administrative security model.

Keywords- Information Security Model; BLP Model; Biba Model; RBAC Model; Role Administration; User Group

I. INTRODUCTION

Based on the role access control model (Role Based Access Control, referred to as RBAC model) [1] is a research hot issue of system safety management, and it obtained the very good application [2-4] in the application system. But the traditional RBAC model and many improved RBAC model exist many insufficiencies. They do not provide the filtration function for service data, that lead to as long as the user get a service data access permissions by the role, the user can execute corresponding permissions for all examples of this kind of data [5]. The confidentiality is not high [6-7]. At the same time, the information system of large-scale still exist some insufficiencies such as the maintain of authorization is complex, too dependent on the role of management, leading to excessive subdivision of the role in the system of user authority changes frequently, causing the role of redundancy and so on [8]. The documents [8-10] proposed user group concept, used for solving the problems that exist in the traditional RBAC model application. But these are not combined with the security grade problem of subject and object, rarely involve the construction of role management strategies.

A security access control model based on the security grade and the role management of user group is proposed in this paper, with the concept of user group and the idea proposed in literature [4].

II. BASIC CONCEPT AND MODEL

A. Known Concept

Subject and Object: there are a large number of safety
*Corresponding Author

operations in computer system, all of which are called subject. The subject set is represented by S. Anything manipulated by subjects is called object and its set is represented by O.

Classifications: the confidentiality grade of subject s is denoted with T(s); the confidentiality degree of object o is denoted with T(o), the Table 1 showing a specific way to measure them.

Integrity Grade: the integrity grade of subject s is denoted with I(s); the integrity grade of object o is denoted with I(o), the Table 1 showing a specific way to measure them.

Classifications and integrity grade of subject and object are supposed according to “the grade divides standard of the computer information system security protection” of our country (GB 17859-1999) [10].

| Security grade | Security description                  | T(s)/T(o) | I(s)/I(o) |
|----------------|--------------------------------------|-----------|-----------|
| First class    | Customer independence protection class| 1         | 1         |
| Second class   | System audits protection class        | 2         | 2         |
| Third class    | Security mark protection class        | 3         | 3         |
| Four class     | Structurization protection class      | 4         | 4         |
| Fifth class    | Visit verification protection class   | 5         | 5         |

The methods how to define the classifications function and the integrity grade function have no influence on further discussion, only if they are finite real numbers.

DOI: 10.5963/JCEI0102003
User group mentioned in this paper refers to the subject which has some of the same role set, based on the concept of user group, the subject set is denoted with SU. We assume SU={SU(s₁), SU(s₂), ..., }, which element is the user group. The user group of subject s belongs to is recorded as SUᵢ(s) (I = 1, 2, ..., k). S is in the user group of SU₁(s), SU₂(s), ..., SU_k(s). In order to simplify, we sometimes directly recorded as SU(s). Obviously, the elements of the SU may be have public child elements, namely a subject (user) can be in different user groups.

We use a similar representation of the classification and integrity grade of subject, and object, to describe user group classification and integrity grade.

For example, T (SU(s)) is denoted classification of user group SU(s). I(SU(s)) is denoted integrity grade of user group SU(s).

B. Some Basic Properties of the Known Model

1) BLP Model and Biba Model:

According to the basic properties of BLP model[12], the system which is in the safe state, must satisfy: if T(s) ≥ T(o), subjects can read object, and if T(s) ≤ T(o), subjects can write object. According to the basic properties of Biba model[13], the system which is in the safe state, must satisfy: if I(s) ≤ I(o), the subject can read object, and if I(s) ≥ I(o), the subject can write object.

2) RBAC Model:

The basic idea of RBAC model[6] is the division of responsibility, which is very similar to an organization. In the RBAC model, users are granted roles, roles are granted permissions, and the permissions are associated with operating. Users get appropriate permissions of the role by the granted role to complete the some operations. We will take the concepts of basic RBAC model in this paper.

III. A SECURITY MODEL BASED ON USER GROUP AND ROLE MANAGEMENT

A. Basic Concept

Security service: for the subject s or the object o, system provides security services (such as authentication, encryption, etc.), which set denoted: s_sa or o_sa, and it is a sub-set of security services set (denoted as: P) in system.

Control granularity: based on the point of safety view, system needs the roughly granularity control or finely granularity control, the roughly granularity set and fine granularity set of object o are denoted respectively by o_rg and o_fg. The whole environment corresponding to the roughly granularity set, fine granularity set are denoted respectively as P_rg, P_fg.

Similar, to the subject definition corresponding granularity control set, use in describing the basic property when the subject as the sponsor of conversation or receiver’s: the subject s roughly granularity set, fine granularity set respectively record are: s_rg, s_fg.

When it cannot carry on the granularity control, the corresponding granularity control set takes for the null set φ.

Below some related symbols are given with basic properties of the RBAC model.

System has a role set denoted R, make R={r₁, r₂, ..., rₙ}. All of roles corresponding permission set is denoted R_P, and we assume that R_P={p₁,p₂,...,pₘ}. A role set which subject s owned is denoted SR(s). A subject set which role r corresponds to is denoted: RS(r). A role set which object o owned is denoted: OR(o). An object set which role r corresponds to is denoted: RO(r). A right set which role r corresponds to is denoted: RP(r). A permission set which s owned based role is denoted SP(s). A permission set which object o is allowed to use and based role denoted: OP(o).

A role set which user group SU(s) owned is denoted SUR(s). A user group which role r corresponds to is denoted: RUS(r). A permission set which user group owned based role is denoted SUP(s). A role set which object o owned user group is denoted: OUR(o). A permission set which object o is allowed to use and which role r of user group corresponds to is denoted: RUP(r).

The access mode that subject s from the security angle consideration visits to object o is denoted with access_mode(s, o). The value range of access_mode(s, o) is a permission set, in which s takes a user, when plays a role, simultaneously considered that the BLP model and Biba model correspondence the safe limit, carries on the operation to o, and it is actually a subset of R_P, which should contain re: read, w: write, c: create; d: delete etc such permission.

In this way, the safety factors of subject s have s, s, T(s), I(s), SR(s), s_sa, s_rg, s_fg, etc. The safety factors of object o have o, T(o), I(o), OR(o), o_sa, o_rg, o_fg, etc.

It is assumed that when the initial conditions does not involve the user group and the rule management, the whole “A Model Based On User Group and Role Administration” is safe, namely, information transmission and other areas are safe.
B. To Introduce Brief User Group and Role Management Module

In the role management module, we set three sub-modules, that are the user and user group management, role management, and other maintenance. In the user management sub-module, it includes user to be added, deleted, modified, and queried, etc. In the role management sub-module, it includes role to be added, deleted, given, recovery, etc.

Following the Section 3.3, the corresponding state transition rules will be given from the security point of view of model description, about the principal part of the user group and role management. But the article does not involve the achievement of specific procedure.

C. Rules of User Group and Role Management Security Model

1) Create a Role:

Creat_role(s, T(s), I(s), SR(s), s_sa, s_rg, s_fg : r, RS(r), RP(r), RO(r), RUS(r), SUP(s), OUR(o), RUP(r))// The subject s increases the role r for the whole system, wherein the subject s has security attributes that are T(s), I(s), SR(s), s_sa, s_rg, s_fg, etc. The rule r corresponds to the subject set for RS(r), to the permissions set for RP(r), and object set RO(r). Wherein the group has security attributes that are RUS(r), SUP(s), OUR(o), RUP(r), etc.

If s_sa $\in$ P or s_rg $\in$ P or s_fg $\in$ P or c $\notin$ SP(s) then go_end// The system does not provide the s security service, or s does not meet the coarse-grained or fine-grained rules of sytsem, or s without creating the role right, end.

R=R $\cup$ \{r\}

\(\forall r' \in R, R'=R_P \cup RP(r)\)// Refresh the permission set of rule r corresponding to

\(\forall s' \in RS(r), SR(s')=SR(s') \cup \{r\}\)// Refresh the permission set of the subject owning the rule r

\(\forall o' \in RO(r), OR(o')=OR(o') \cup \{r\}\)// Refresh the permission set of the object owning rule r

\(\forall o' \in RO(r), PUP(o')=PUP(o') \cup POP(r)\)// Refresh the permission set of the rule corresponding to rule r, and this rule is the user group which object o owned.

2) Delete a Role:

Delete_role(s, T(s), I(s), SR(s), s_sa, s_rg, s_fg : r, RS(r), RP(r), RO(r), RUS(r), SUP(s), OUR(o), RUP(r))// The subject s deletes the role r. The subject s has security attributes that are T(s), I(s), SR(s), s_sa, s_rg, s_fg. The rule r corresponds to the subject set for RS(r), to the permissions set for RP(r), and object set RO(r). Wherein the group has security attributes that are RUS(r), SUP(s), OUR(o), RUP(r), etc.

- DOI: 10.5963/JCEI0102003
If s_sa ∉ P or s_rg ∉ P_rg or s_fg ∉ P_fg or d ∉ SP(s) then go_end// If s without corresponding security attributes, to the end 
∀ s’: RS(r), SR(s')=SR(s')\{-r\} //Refresh the rule set of the subject owned 
∀ o’: RO(o), OR(o')=OR(o')\{-r\} //Refresh the rule set of the object owned 
∀ SU(s')∈ RUS(r), SUP(s')=SUP(s')\{r\} //Refresh the permission set of the user groups that owned rule r 
∀ SU(s')∈ RUS(r), SUR(s')=SUR(s')\{-r\} //Refresh the rule set of the user group that has owned rule r 
∀ o’ ∈ RO(o), PUP(o')=PUP(o')\{r\} // Refresh the permission set corresponding to, the user group rule which object o has owned 
∀ o’ ∈ RO(o), OUR(o')=OUR(o')\{-r\} //Refresh the rule set of the user group that object o has owned 
R=R\{-r\} //Refresh the permission set of system 
∀ r’ ∈ R, R_P=∪ RP(r') //Refresh the permission set of rule r 
∀ s’ ∈ S, SP(s')=∪ RP(r') \{r'\} //Refresh the permission set of subject 
∀ o’ ∈ O, OP(o')=∪ RP(r') \{r'\} //Refresh the permission set of object 
RS(r)=RO(r)=RP(r)= RUS(r)= RUP(r)=Φ 
∀ s ∈ S, ∀ o ∈ O, access_model(s,o )= SP(s) \cap OP(o) //The following discussion is the same with that discussion of Section 3.3.1, here omitted 

3) Assign Permissions to Roles: 
Append_purview(s, T(s), I(s), SR(s), s_sa, s_rg, s_fg: r, RS(r), RP(r), RO(r), p), RUS(r), SUP(s), OUR(o), RUP(r)//The subject s increases powers p for rule r. The subject s has security attributes that are T(s), I(s), SR(s), s_sa, s_rg, s_fg. The original rule r corresponding to subject set, permissions set and object set is respectively RS(r), RP(r) and RO(r). Wherein the group has security attributes that are RUS(r), SUP(s), OUR(o), RUP(r), etc. 
If s_sa ∉ P or s_rg ∉ P_rg or s_fg ∉ P_fg or d ∉ SP(s) then go_end// If s without corresponding security attributes, to the end 
RP(r)=RP(r) ∪ \{p\} //Refresh the permission set of rule r 
∀ r’ ∈ R, R_P=∪ RP(r') //Refresh the permission set of system 
∀ s’ ∈ RS(r), SP(s')=SP(s') ∪ RP(r') //Refresh the permission set of the rule r which the subject has owned 
∀ o’ ∈ RO(r), OP(o')=OP(o') ∪ RP(r') //Refresh the permission set of the rule r which the object has owned 
∀ SU(s')∈ RUS(r), SUP(s')=SUP(s') ∪ RP(r') //Refresh the permission set of the user groups that has owned rule r 
∀ o’ ∈ RO(o), PUP(o')=PUP(o') ∪ POP(r')// Refresh the permission set of the rule corresponding to, and this rule is of the user group which object o owned 
∀ s ∈ S, ∀ o ∈ O, access_model(s,o )= SP(s) \cap OP(o) //The following discussion is the same with that discussion of Section 3.3.1, here omitted 

4) Delete the Permission of a Role: 
Delete_purview(s, T(s), I(s), SR(s), s_sa, s_rg, s_fg: r, RS(r), RP(r), RO(r), p), RUS(r), SUP(s), OUR(o), RUP(r) // The subject s will delete the permission p of the role r. The subject s has security attributes that are T(s), I(s), SR(s), s_sa, s_rg, s_fg. The original rule r corresponding to subject set, permissions set and object set is respectively RS(r), RP(r) and RO(r). Wherein the group has security attributes that are RUS(r), SUP(s), OUR(o), RUP(r), etc. 
If s_sa ∉ P or s_rg ∉ P_rg or s_fg ∉ P_fg or d ∉ SP(s) then go_end// If s without corresponding security attributes, to the end 
RP(r)=RP(r)\{-p\} //The following discussion is the same with that discussion of Section 3.3.1, here omitted 

DOI: 10.5963/JCEI0102003
5) **Give a Rule to a Subject:**

Append_rule(s, T(s), I(s), SR(s), s'_{sa}, s'_{rg}, s'_{fg}, s': r, RS(r), RP(r), RO(r)) // The s add rule r for the subject s'. The subject s has security attributes that are T(s), I(s), SR(s), s'_{sa}, s'_{rg}, s'_{fg}. The rule r corresponding to subject set, permissions set and object set is respectively RS(r), RP(r) wherein the group have security attributes that are RUS(r), SUP(s), OUR(o), RUP(r), etc.

If s'_{sa} \not\subset P or s'_{rg} \not\subset P_{rg} or s'_{fg} \not\subset P_{fg} or c \not\in SP(s) then go_end // If s without corresponding security attributes, to the end

\[
RS(r) = RS(r) \cup \{s'\} // Add the subject s' in subject set of rule r corresponding to
\]

\[
SR(s') = SR(s') \cup \{r\} // Refresh the rule set of the subject s' owned
\]

\[
\forall r' \in SR(s'), SP(s') = SP(s') \cup RP(r') // Refresh the permission set of subject
\]

\[
\forall s \in S, \forall o \in O, access_model(s, o) = SP(s) \cap OP(o) // The following discussion is the same with that discussion of Section 3.3.1, here omitted
\]

6) **Create a User Group:**

Creat_User Group (SU(s), T(SU(s)), I(SU(s)), SUR(s), s'_{sa}, s'_{rg}, s'_{fg}; r, RUS(r), RUP(r), RUO(r)) // The subject s' increases the user group SU(s) for the whole system, wherein the subject s has security attributes that are SU(s), T(SU(s)), I(SU(s)), SUR(s), r, RUS(r), RUP(r), RUO(r).

If s'_{sa} \not\subset P or s'_{rg} \not\subset P_{rg} or s'_{fg} \not\subset P_{fg} or c \not\in SP(s) then go_end // If s without corresponding security attributes, to the end

\[
SU = SU \cup SU(s)
\]

\[
\forall SU(s') \in RUS(r), SUR(s') = SUR(s') \cup \{r\} // Refresh the rule set of the user groups that has owned rule r
\]

\[
\forall SU(s') \in RUS(r), SUP(s') = SUP(s') \cup RP(r) // Refresh the permission set of the user groups that has owned rule r
\]

\[
\forall o' \in RO(r), OUP(o') = OUP(o') \cup OUR(o) // Refresh the rule set of the user groups which object o owned
\]

\[
\forall o' \in RO(r), PUP(o') = PUP(o') \cup POP(r) // Refresh the permission set of the rule corresponding to, and this rule is of the user group which object o owned
\]

The following discussion is the same with that discussion of Section 3.3.1, here omitted

Due to the limited space, in addition to the basic rules, other management rules don’t describe all here, readers can continue by imitation the methods above, also can go on through reference documentation [4].

**D. Safety Analysis of the Rules**

From 3.3 each rule’s definitions can know, the rule definition has strictly used the RBAC model basic property. Each rule has quoted the classic information security model BLP model and security properties of Biba the model. According to the above, we thought that condition switching rules described which is based on the user group and role management is safe. According to finite-state machine principle [14, 15], when question initial condition is security, the model which obtained based on these condition switching rules is safe, reasonable.

**IV. CONCLUSION**

In this paper, some common rules based on user group and role management are given, and that using basic properties of the BLP model, Biba model and RBAC model. After the analysis of these rules, we believe that they are reasonable, safe, and thus the model based on these states transfer rules are safe and reasonable. This paper provides an effective means of modeling method for information security model of security described and demonstrated, through the formal description and verification of the user group, role management model, and it will help to promote the theoretical research of information security.

The method described in the paper has been referred to in our school informatization construction, and obtained the preliminary application.

**ACKNOWLEDGEMENT**

This work is supported by Sichuan province science and technology department the soft science project 2013ZR0089. Sichuan province department of education natural science point item 13ZA0003. Sichuan province 2011 and 2012 academic and technical leader training funded projects ([2011] No. 441 and [2012] No 319).
REFERENCES

[1] Sandhu RS, Samarati P. Access control: principles and practice[J]. IEEE communications. 1994, 32(9): 40-48.
[2] Ouyang Ming Xing, Zhang Huazhe. MIS system in a large network of role-based rights management [J]. Computer engineering and Applications, 2003, 39 (4): 142-144(in Chinese).
[3] Zeng Junfang, warm courage, Yang one even. E-government system based on role jurisdiction management study [J]. Computer engineering and Applications, 2004, 40 (22):156-160 (in Chinese).
[4] LIU Yihe, SHEN Changxiang. A Security Model Based On Role Administration [J]. Computer applied research, 2007, 24 (5): 119-121(in Chinese).
[5] Xiong Zhihui, etc. for the two-dimensional role of information systems security access control model [J]. Computer Engineering and Scienc, 2008, 30(9): 1-3, 17 (in Chinese).
[6] Kandala S, Sandhu R S. Secure Role Based Workflow Models[C] // Proc of the 15th Annual Working Conf on Database and Application Security, 2002 : 45-58.
[7] Massacci F, Mylopoulos J, Zannone N. Minimal Disclosure in Hierarchical Hippocratic Databases with Delegation [C] // Proc of ESORICS’ 05, 2005 : 438-454.
[8] HUANG Guangfang. The RBAC Model Based on User Group & Free — gained Authority Journal of Zhanjiang Normal College, 2009, 26(3): 1098-1100 (in Chinese).
[9] Xing Han fa, etc. based on the expansion of the role and user group access control model [J]. Computer applied research, 2009, 26(3): 1098-1100 (in Chinese).
[10] MU Xiao—dong etc. Application Research of User Group in Role Based Access Control Model [J]. Fire Control& Command Control, 2012, 37(8): 170-172 (in Chinese).
[11] Computer information system security classification criteria [S], People’s Republic of china: GB 17859-1999, 1999.9.
[12] Bell D E, Lapadula LJ. Secure compter system[R]. USA: mathematical foundation. MTR-2527, Mitrecorp, Bedford, MA, 1973.
[13] Biba, K. Integrity Considerations for Secure Computing Systems[R]. USA: Mitre Report MTR-3153, Mitre Corporation, Bedford, MA, 1975.
[14] Li Jun, Sun Yufang. Computer security and security model [J]. Computer Research and Development, 1996, 33(2):312-320(in Chinese).
[15] Liu Yihe. A formal description of B / S model information security system [J]. Computer Science, 2004, 31(9A):217-219(in Chinese).

Quandeng Gou: Master, lecturer, Research interests: Information Security.  
*Yihe Liu (corresponding author): Ph.D., Professor, Research interests: Information Security.