J. Cent. South Univ. (2012) 19: 1049−1056
DOI: 10.1007/s11771-012-1108-0

Permission and role automatic assigning of user in role-based access control

HAN Dao-jun(韩道军)[1, 2], ZHUO Han-kui(卓汉逵)[1, 3], XIA Lan-ting(夏兰亭)[1], LI Lei(李磊)[1, 3]

1. Software Research Institute, Sun Yat-sen University, Guangzhou 510275, China;
2. Institute of Data and Knowledge Engineering, Henan University, Kaifeng 475004, China;
3. NetCraft Information Technology (Macau) Co., Ltd., Macau

Abstract: Role mining and setup affect the usage of role-based access control (RBAC). Traditionally, user's role and permission assigning are manipulated by security administrator of system. However, the cost is expensive and the operating process is complex. A new role analyzing method was proposed by generating mappings and using them to provide recommendation for systems. The relation among sets of permissions, roles and users was explored by generating mappings, and the relation between sets of users and attributes was analyzed by means of the concept lattice model, generating a critical mapping between the attribute and permission sets, and making the meaning of the role natural and operational. Thus, a role is determined by permission set and user's attributes. The generated mappings were used to automatically assign permissions and roles to new users. Experimental results show that the proposed algorithm is effective and efficient.

Key words: role-based access control; role; permission assignment; concept lattice

1 Introduction

Access control is the core of computer system security. It protects system resources in a controlled manner by some policies. There are various access control models, such as role-based access control (RBAC) [1], discretionary access control (DAC), mandatory access control (MAC) [2], usage control (UCON) [3], attribute-based access control (ABAC) [4−5], and task-based access control (TBAC) [6]. The RBAC model is popular and applied widely because it is easy to manage and it has realized the logical separation of user and permission. Recently, researchers mainly focused on three aspects of RBAC: extended models of RBAC [7], security analysis of RBAC [8−10] and role analyzing [11−15]. For example, BARKER et al [7] introduced an extended model of status-based access control, extending traditional role, generating status level by some actions and attributes, and making permission assignment flexible; LIU et al [10] focused on the access control system which supports role hierarchy and static mutual exclusion roles using graphplan; SANDHU and COYNE [1] introduced the hierarchy of roles and described the subsume of permission set, COYNE [11] analyzed the role by data mining and set theory.

Since role engineering was introduced by COYNE [11], some researchers proposed various algorithms for role generating, evaluating and optimizing [12−15]. Two traditional methods, bottom-up and top-down, are determined by requirement acquisition. For example, a common method to get roles and permissions for organization chart of systems is suitable to some simple information systems, but limited to complex information systems. The stakeholder is larger for complex information system and the business process is more complicated. It makes permission assignment rely on simple information, e.g., group membership and job function. In practice, assigning permission is related to many factors. If we only focus on user's simple information, it could be difficult to make good decisions. In addition, only with this information, it is hard to distinguish users in the systems. Thus, relying only on role information for assigning permissions is improvable.

Obviously, analyzing the essence and principle of role is of benefit to permission assignment and improves the dynamic of RBAC. Users, as a kind of special entity, could be described by some attributes and values in complex information system, where the value is restricted to domain knowledge. In this work, relations

Foundation item: Project(61003140) supported by the National Natural Science Foundation of China; Project(013/2010/A) supported by Macau Science and Technology Development Fund; Project(10YJC630236) supported by Social Science Foundation for the Youth Scholars of Ministry of Education of China
Received date: 2011−01−12; **Accepted date:** 2011−09−23

Corresponding author: HAN Dao-jun, PhD Candidate; Tel: +86−20−84110658; E-mail: hdj@henu.edu.cn

among user attributes, users, roles and permissions are founded by some mappings, used for automatic permissions assignment of new users. First, two mappings were generated, one between role and permission and the other between role and user. Then, a mapping between user and user attribute is generated using a concept lattice model. The relation between user attribute and permission is investigated according to the fact that users sharing the same or similar attributes could be viewed as a bridge. Those relations are useful information to roles and permissions which can be automatically assigned when new users enter systems.

2 Role description and mapping construction

2.1 Role foundation

For access control purpose, it is much more important to know what user's organizational responsibilities are, rather than who the user is. Thus, RBAC is suitable. In RBAC, role is a result of permission clustering, which could represent the organizational responsibilities of a user. The RBAC has greatly simplified permission management for it implements the logical separation between user and permission by means of conferring or revoking permissions to a role instead of the user. The CORE RBAC model was released by ANSI in 2004 [16], and the main components are shown in Fig. 1.

FERRAIOLO et al [17] pointed out that roles are more stable in a system because an organization's activities or functions seldom change. Afterward, the motivation and the priority of a role setting was introduced [1]. It is found that it is desirable to allow administrators to confer and revoke the membership to users in existing roles without giving these administrators authority to create new roles or change role-permission assignments. Furthermore, conferring a role to a user is simpler than conferring a permission to a user [1]. Here, there are two observations, one is that a role is determined by a permission, and the other is that the utilization of roles makes management simple. The essence of roles needs explanation by some natural and reasonable ways. It is observed that there are other attributes to describe a user besides the name and identity of the user in a complex information system, and traditional methods only select one of the user attributes (e.g. job title, membership and location) for role generation. Selecting a single attribute for the role generation is suitable to some management information systems because users are clearly classified and the management resource is simple. For complex information system, it is difficult to distinguish some users only relying on a single attribute. Thus, some proper permission assignments are difficult. In general, using multiple attributes to generate roles could improve the ability of conferring permissions. Attribute-based access control (ABAC) [4−5] applied uniform description strategy to each element of access control, including subject, operation and object. In ABAC, the permission assignment is easy to manage if the attributes are enough, and it is inconvenient to use in complex information system because of lack of pre-authorization. In this work, the way of role generation is discussed according to the idea of the role extension of SBAC and the virtue of RBAC and ABAC.

Usually, a relational table represents the relation between user and permission, where rows indicate users and columns denote their permissions. An example is given in Table 1. In Table 1, the user set $U=\{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$, the permission set $P=\{p_1, p_2, p_3, p_4, p_5\}$, and the relation set $I=\{(u_1, p_2), (u_1, p_4), \cdots, (u_8, p_5)\}$ are presented. Giving a function f_{u} mapping U to P: $f_u(x)=\{y|(x, y)\in I, x\in U\}$, it is observed that, $f_u(u_1)=f_u(u_5)=\{p_2, p_4, p_5\}$; $f_u(u_2)=f_u(u_6)=\{p_1, p_3, p_5\}$; $f_u(u_3)=f_u(u_7)=\{p_2, p_3, p_5\}$; $f_u(u_4)=f_{\mathrm{u}}(u_8)=\{p_1, p_4, p_5\}$. Then, Table 1 can be simplified as Table 2 if those rows (users) are combined with the same f_u values into one, which is represented by a role r_i in this work.

In Table 2, each role r_i expresses a user subset including the users with the same f_u value, the role set $R=\{r_1, r_2, r_3, r_4\}$, and the following expressions were obtained, i.e., $\mu(r_1)=\{u_1, u_5\}$, $\mu(r_2)=\{u_2, u_6\}$, $\mu(r_3)=\{u_3, u_7\}$, $\mu(r_4)=\{u_4, u_8\}$, $\tau(r_1)=\{p_2, p_4, p_5\}$, $\tau(r_2)=\{p_1, p_3, p_5\}$, $\tau(r_3)=\{p_2, p_3, p_5\}$, $\tau(r_4)=\{p_1, p_4, p_5\}$. Here, the function μ returns the user set of the role r_i, and the function τ returns the permission set of the role r_i.

2.2 Analysis on users and attributes

2.2.1 Equivalence class generation and user description based on attributes

In general, a user is described by some attributes. The common types of the attributes include boolean,

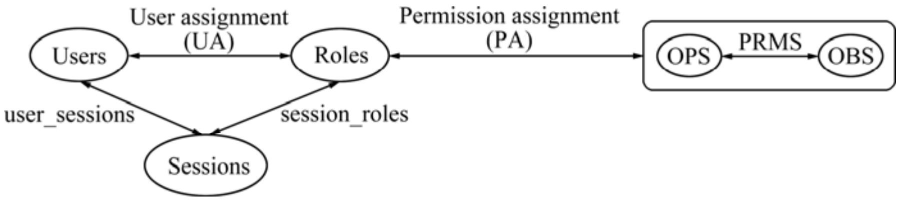

Fig. 1 Model of CORE RBAC

Table 1 Relation between user and permission

User	Permission				
	p_1	p_2	p_3	p_4	p_5
u_1		1		1	1
u_2	1		1		1
u_3		1	1		1
u_4	1			1	1
u_5		1		1	1
u_6	1		1		1
u_7		1	1		1
u_8	1			1	1

Table 2 Relation between role and permission

Role	Permission				
	p_1	p_2	p_3	p_4	p_5
r_1		1		1	1
r_2	1		1		1
r_3		1	1		1
r_4	1			1	1

numeric and symbol, and the ranges of those attributes are determined by the related domain [18]. Given the selected attributes, if users are different, their attribute values are different. Then, the different user sets can be obtained by generating equivalence classes after equivalence relations have been defined.

Use a four-tuple O=(G_1, M_1, V, Q) to denote the users and their attributes, where G_1 indicates the user set, M_1 symbolizes the attribute set, the set V contains all possible attribute value, $Q\subseteq G_1\times M_1\times V$, G_1, M_1 and V are finite, $(g, m, w)\in Q$ and $(g, m, v)\in I\Rightarrow w=v$, where $(g, m, w)\in Q$ means the attribute m of the user g has the value w. Some equivalence classes can be gotten according to the distribution of user attributes and the mappings can be defined as

$$d(m,v)=\{x\mid (x,m,v)\in Q, x\in G_1, m\in M_1, v\in V\}$$

$$k(x,m)=v, (x,m,v)\in Q, v\in V, x\in G_1, m\in M_1$$

$$p(x,M_1')=\bigcap_{i=1}^{|M_1'|}(d(m_i,k(x,m_i)), m_i\in M_1'), x\in G, M_1'\subseteq M$$

$$q(M_1')=\{z\mid z=p(x,M_1'), x\in G\}, M_1'\subseteq M$$

$$h(M_1')=\{y\mid y=q(m_j), m_j\in 2^{M_1'}\}, M_1'\subseteq M$$

Despite of these mappings can generate those appropriate user classes, it is unnatural to get equivalence classes by using them directly. The reason is that inherence of equivalence classes and their equivalence relations are absent. To overcome these shortcomings, the concept lattice model is introduced in this work. Not only could concept lattice express the equivalence class and equivalence relation, but also it denotes the inherence of them by using formal concept, so concept lattice is suitable.

2.2.2 Equivalence class generation by concept lattice

Concept lattice, the core data structure of formal concept analysis theory, is a popular data analysis tool. Every node of concept lattice is a formal concept, which includes two parts, extension and intension. Extension of a formal concept is the object set that belongs to it, and intension is description of concept, that is, common properties of the object set. Also, concept lattice explains the generalization and specialization of concepts simply and vividly using Hasse graph [19].

A formal context is a triple set K=(G_2, M_2, I), where G_2 is a set of objects, M_2 is a set of attributes, I is a relation between G_2 and M_2, and, $I\subseteq G_2\times M_2$. Then, there is a unique partial order set, to generate a lattice structure, to correspond with K. Lattice L is called concept lattice generated by context (G_2, M_2, I), where every node of L is an ordered pair, called formal concept, expressed as (X, Y), $X\in 2^{G_2}$ and X is extension, $Y\in 2^{M_2}$ and Y is intension, where 2^A is power set of a set A. Every ordered pair is complete for I, that is, 1) $X=\{x\in G_2|y\in Y, (x, y)\in I\}$; 2) $Y=\{y\in M_2| x\in X, (x, y)\in I\}$.

Two mappings f_1: $2^{G_2}\rightarrow 2^{M_2}$ and f_2: $2^{M_2}\rightarrow 2^{G_2}$ are defined for context K, satisfied: $f_1(G_i')=\{m|(x, m)\in I, \forall x\in G_i'\}$, $f_2(M_i')=\{x\mid(x,m)\in I, \forall m\in M_i'\}$. Mappings f_1 and f_2 are called Galois connection of 2^{G_2} and 2^{M_2} For a random $(G_1',M_1')\in 2^{G_2}\times 2^{M_2}$, if $G_1=f_2(M_1')$ and $M_1'=f_1(G_1')$, then (G_1',M_1') is a formal concept of K. and all the formal concepts generated from K are expressed as $C_S(K)$.

According to the definition of mappings f_1 and f_2, each extension of concepts is an indiscernible object set that associates with respective intension of concept. Thus, the total objects of extension are indiscernible, every object has the attributes of intension, and all the object sets, that is, the extension of all concepts, are considered as different definable set systems [20]. All the objects of extension are indiscernible, which is the same as the mapping q functionally. According to the definition of mappings f_1, f_2 and conditions of formal concept generating, a one-one mapping between the extension set and intension set of all the formal concepts is created, i.e., the extension of all concepts form a set E_c, dually, the intension of all concepts form a set I_c, then, a one-one mapping γ is created between the sets E and I as $E_c\rightarrow I_c$.

When using concept lattice model to generate equivalence classes, the first step is converting a four-tuple O to a triple K; then, we generate formal concepts through the concept lattice generating algorithm, and, the equivalence classes can be obtained. The convert **Algorithm 1** is shown as follows.

```
Algorithm 1: ConvertAttributeValueToContext
Input: A four-tuple $O(G_1, M_1, V, Q)$
Output: A triple $K(G_2, M_2, I)$
Step 1: $G_2=G_1$;
Step 2: For all elements of $Q$, if $(g, m, v)\in Q$, then generate $(g', m')\in I$ and $m'\in M_2$, where $g'=g$, and $m'=m\times v$;
Step 3: return $K$.
```

Note that **Step 2** includes $m'=m\times v$, where m is an attribute, v is attribute value, and $m\times v$ still is a new attribute because attribute is not special but a wide concept.

2.3 Mapping construction between attributes and permissions

2.3.1 Analysis on assumptions of mapping construction

From analyzed information, some relations between the attribute sets and the permission sets are found. Before showing those relations in detail, three assumptions are introduced.

Assumption 1: In RBAC, the goal of roles introduced is to reduce the number of managed subjects, the essence of which is to select a role to replace some users to manage, i.e., RBAC makes a lifting about manage object, and it makes manage object change to schemas from instances.

Schema and instance are two important notions. The extension of schema is an instance set, and intension of schema is an attribute set, and all the instances of extension must have total attributes of intension. Instance is a result of schema specialization, which has special attributes besides the attributes of schema's intension, and each instance's special attributes can make a distinguish with other instances. Roles are schemas generated from users whose permissions are the same, and it is a result of clustering and dividing on permission set. Two extremely cases may occur: 1) all users only belong to one role; 2) each user becomes to a different role, i.e., every role has one user. Obviously, it cannot represent the superiority of RBAC when one of extremely cases occurs, so, it is not discussed in this work. Normally, the number of roles is less than the number of users.

Assumption 2: Permission assignment is reasonable and explainable in RBAC when permission delegate is omitted.

In RBAC, the user's background, e.g., job title, membership, and location, are used to permission assignment, and it is not a random but a deliberate behavior. So, permission assignment is reasonable and explainable. Permission delegate is special and arbitrary with outer environment of system, and related to business process. Thus, the permission delegate is not discussed in this work.

Assumption 3: In a complex information system, attribute description information of user is enough.

Attribute description information of user includes attributes and corresponding range. In this work, we only focus on classifying attribute which could represent a schema of users. Given an application system with attribute number of m, if the range of every attribute is v_i, $0<i\le m$, where $v_i\in N$ and $v_i>1$, then maximal of equivalence classes is $S_1=\prod_{i=1}^{m} v_i$. Furthermore, the number of user classifying is determined by S_1. This assumption ensures that the custom user set can be gotten under different filter conditions. On the contrary, if user's attribute description information is not enough, then permission assignment may be affected because some users are indiscernible.

Among these assumptions, **Assumption 3** is practical, and it could be implemented by **Algorithm 2**.

```
Algorithm 2: Isuserattributeenough
Input: $U_x$, $C_S(K_1)$//$U_x$ is user set generated from roles; $C_S(K_1)$ is generated from user attribute context $K_1$
Output: True/False
Begin
(1)   For all $u_x\in U_x$
(2)       Flag=False;
(3)       For all $N_i\in C_S(K_1)$
(4)           If Extension($N_i$)=$u_x$ then Flag=true
(5)                                              break;
(6)           EndIf
(7)       EndFor
(8)       If the flag is false then return "False" EndIf
(9)   EndFor
(10)  return True;
End
```

In summary, the upper assumptions are necessary conditions of our researches.

2.3.2 Mapping construction between attribute and permission sets

According to content analyzed, given a complex information system, and there is an equivalent relation between users and permission sets, a mapping between them can be generated, and some user equivalence classes are created by attribute set in user attribute table. Obviously, attribute description and permission assignment are related to user closely because the user is an important entity. Thus, the information of attribute description and permission assignment of user are analyzed and a dependency relation, that is, a key mapping between attribute and permission sets is generated. The following steps are the generating process of some mappings.

Step 1: Generating a one-one mapping δ between role and user sets on the basis of U_A, i.e., δ: $R\rightarrow U_x$, where $U_x\subset 2^U$.

Step 2: Generating a one-one mapping φ between role and permission sets according to P_A, i.e., φ: $R\rightarrow P_x$, where $P_x \subset 2^P$.

Step 3: Generating a one-one mapping γ between intension and extension of formal concepts after a concept lattice is generated according to user and attribute context, i.e., γ: $A_c\rightarrow U_y$, where $U_y \subset 2^U$, $A_c \subset 2^A$.

Step 4: Generating a mapping λ between attribute and permission sets when **Algorithm 2** returns "True", i.e., λ: $U_x\rightarrow U_y$.

Because γ is a one-one mapping, inverse function of γ is easy to create, that is, $A_c=\gamma^{-1}(U_y)=\gamma^{-1}(\lambda(U_x))=\gamma^{-1}(\lambda(\delta(R))=\gamma^{-1}(\lambda(\delta(\varphi^{-1}(P_x)))$.

Step 5: Let $\rho=\gamma^{-1}\circ\lambda\circ\delta\circ\varphi^{-1}$, then $A_c=\rho(P_x)$. For all $p_x\in P_x$, add $\rho(p_x)$ to set A_{c1}, and construct a new set U_z, where $U_z=\gamma^{-1}(A_{c1})$, $A_{c1}\subset A_c$, and $U_z\subset U_y$. Thus, a one-one mapping ρ' is generated between P_x and A_{c1}, where ρ': $A_{c1}\rightarrow P_x$.

When the mapping ρ' is generated, permission set can be gotten according to attribute set. The total description of domain and range about mappings γ, λ, δ and φ are shown in Fig. 2. Mappings ρ' and φ are valuable, and it can get the role and permission sets by attribute set of users. Thus, it can automatically assign permissions and roles when new user logs in system.

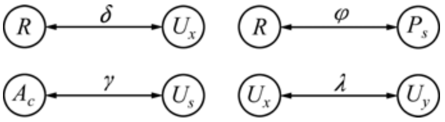

Fig. 2 Total description of domain and range about some mappings

3 Experimental analysis

Select access control of administrative examination and approval business process fragment as experiment domain, then analyze the related data and generate the mapping between attribute and permission sets. The following is experimental explanation.

Let $P=\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\}$, where every permission is explained as follows. Permission p_1 means expendable material purchase application, p_2 means equipment purchase application, p_3 means large equipment purchase application, p_4 means retirement application, p_5 means small approval, p_6 means large processing, p_7 means cash payment, p_8 means check payment, and p_9 means transfer of public. A role set $R=\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\}$. A user set $U=\{u_1, u_2, \cdots, u_{20}\}$. Attribute set and range are Department (technology department, comprehensive department, finance department); Level (primary, middle-level, high-level); Post (department head, associate department head).

Table 3 gives the relation between roles and permissions.

Table 3 Relation between role and permission

Role	Permission								
	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9
r_1	1			1					
r_2	1	1		1					
r_3	1	1	1	1					
r_4				1	1				
r_5				1	1	1			
r_6							1		
r_7							1	1	
r_8							1	1	1

Then, $\varphi(r_1)=\{p_1, p_4\}$, $\varphi(r_2)=\{p_1, p_2, p_4\}$, $\varphi(r_3)=\{p_1, p_2, p_3, p_4\}$, $\varphi(r_4)=\{p_4, p_5\}$, $\varphi(r_5)=\{p_4, p_5, p_6\}$, $\varphi(r_6)=\{p_7\}$, $\varphi(r_7)=\{p_7, p_8\}$, $\varphi(r_8)=\{p_7, p_8, p_9\}$. $P_x=\{\{p_1, p_4\}, \{p_1, p_2, p_4\}, \{p_1, p_2, p_3, p_4\}, \{p_4, p_5\}, \{p_4, p_5, p_6\}, \{p_7\}, \{p_7, p_8\}, \{p_7, p_8, p_9\}\}$.

Then, according to the requirement of system, the mapping results are obtained, i.e. $\delta(r_1)=\{u_1, u_4, u_5, u_7\}$, $\delta(r_2)=\{u_2, u_3\}$, $\delta(r_3)=\{u_6\}$, $\delta(r_4)=\{u_8, u_9, u_{10}, u_{11}, u_{12}\}$, $\delta(r_5)=\{u_{12}\}$, $\delta(r_6)=\{u_{13}, u_{14}, u_{15}, u_{16}\}$, $\delta(r_7)=\{u_{17}, u_{18}, u_{19}, u_{20}\}$, $\delta(r_8)=\{u_{18}\}$. $U_x=\{\{u_1, u_4, u_5, u_7\}, \{u_2, u_3\}, \{u_6\}, \{u_8, u_9, u_{10}, u_{11}, u_{12}\}, \{u_{12}\}, \{u_{13}, u_{14}, u_{15}, u_{16}\}, \{u_{17}, u_{18}, u_{19}, u_{20}\}, \{u_{18}\}\}$.

Table 4 indicates the relation of users and attributes.

Applying **Algorithm 1** to Table 4, a context C_1 is obtained as given by Table 5. Moreover, a concept lattice generated from C_1 is shown in Fig. 3.

Extension and intension of each formal concept in $C_S(C_1)$ are explained as following. 1(u_1–u_{20}, $\varnothing$), 2($u_1u_2u_3u_4u_5u_6u_7$, a), 3($u_8u_9u_{10}u_{11}u_{12}$, b), 4($u_{13}u_{14}u_{15}u_{16}u_{17}u_{18}u_{19}u_{20}$, c), 5($u_1u_4u_5u_7u_8u_9u_{13}u_{14}u_{15}u_{16}$, d), 6($u_2u_3u_{10}u_{11}$, e), 7($u_6u_{12}u_{17}u_{18}u_{19}u_{20}$, f), 8($u_1u_4u_5u_7$, ad), 9($u_{17}u_{18}u_{19}u_{20}$, cf), 10($u_{13}u_{14}u_{15}u_{16}$, cd), 11($u_{10}u_{11}$, be), 12(u_8u_9, bd), 13(u_2u_3, ae), 14($u_6u_{12}u_{18}$, fg), 15(u_6, afg), 16(u_{12}, bfg), 17(u_{18}, cfg), 18(u_{19}, cfh), 19($\varnothing$, $abcdefgh$). According to mapping γ, the mapping results are, $\gamma(a)=\{u_1, u_2, u_3, u_4, u_5, u_6, u_7\}$, $\gamma(b)=\{u_8, u_9, u_{10}, u_{11}, u_{12}\}$, $\gamma(c)=\{u_{13}, u_{14}, u_{15}, u_{16}, u_{17}, u_{18}, u_{19}, u_{20}\}$, $\cdots$, $\gamma(cfh)=\{u_{19}\}$. So, we can get $A_c=\{\{a\}, \{b\}, \cdots, \{cfh\}\}$, and $U_y=\{\{u_1, u_2, u_3, u_4, u_5, u_6, u_7\}, \{u_8, u_9, u_{10}, u_{11}, u_{12}\}, \cdots, \{u_{19}\}\}$. The mapping λ is generated between U_x and U_y, where $\lambda(x)=x$, that is, for a $x\in U_x$, the same element in U_y is found out.

For all $a_c\in A_{c1}$, we could get the permission set by mapping ρ', and the results are indicated as follows. $\rho'(\{a, d\})=\{p_1, p_4\}$, $\rho'(\{a, e\})=\{p_1, p_2, p_4\}$, $\rho'(\{a, f, g\})=\{p_1, p_2, p_3, p_4\}$, $\rho'(\{\mathrm{b}\})=\{p_4, p_5\}$, $\rho'(\{b, f, g\})=\{p_4, p_5, p_6\}$, $\rho'(\{c, d\})=\{p_7\}$, $\rho'(\{c, f\})=\{p_7, p_8\}$, $\rho'(\{c, f, g\})=\{p_7, p_8, p_9\}$.

When mapping ρ' is generated, then the relation between permission and attribute sets are found out, and those relation are valuable, i.e., when a new user u_x enters, we can get attributes and values of u_x, permission

Table 4 Users and attributes

User	Department	Level	Post	User	Department	Level	Post
u_1	Technology department	Primary		u_{11}	Comprehensive department	Middle-level	
u_2	Technology department	Middle-level		u_{12}	Comprehensive department	High-level	Department head
u_3	Technology department	Middle-level		u_{13}	Comprehensive department	Primary	
u_4	Technology department	Primary		u_{14}	Finance department	Primary	
u_5	Technology department	Primary		u_{15}	Finance department	Primary	
u_6	Technology department	High-lever	Department head	u_{16}	Finance department	Primary	
u_7	Technology department	Primary		u_{17}	Finance department	High-level	
u_8	Comprehensive department	Primary		u_{18}	Finance department	High-level	Department head
u_9	Comprehensive department	Primary		u_{19}	Finance department	High-level	Associate department head
u_{10}	Comprehensive department	Middle-level		u_{20}	Finance department	High-level	

Table 5 Context C_1 of users and attributes

User	Technology department (*a*)	Comprehensive department (*b*)	Finance department (*c*)	Primary (*d*)	Middle-lever (*e*)	High-lever (*f*)	Department head (*g*)	Associate department head (*h*)
u_1	1			1				
u_2	1				1			
u_3	1				1			
u_4	1			1				
u_5	1			1				
u_6	1					1	1	
u_7	1			1				
u_8		1		1				
u_9		1		1				
u_{10}		1			1			
u_{11}		1			1			
u_{12}		1				1	1	
u_{13}			1	1				
u_{14}			1	1				
u_{15}			1	1				
u_{16}			1	1				
u_{17}			1			1		
u_{18}			1			1	1	
u_{19}			1			1		1
u_{20}			1			1		

assigned and role assigned automatically according to mappings ρ' and φ. For example, when a user u_x enters, the attributes and values are {Department= “comprehensive department”, Level=“middle-level”}. According to mappings ρ' and φ, the permission set of user u_x is $\{p_4, p_5\}$, and the role of user u_x is r_4. So, the roles and permissions could be automatically assigned to a user by corresponding attributes and values.

5 Related work

All the entities of access control are described by attributes in ABAC [4−5], and each attribute has special attribute authority. Thus, ABAC can conguarously treat

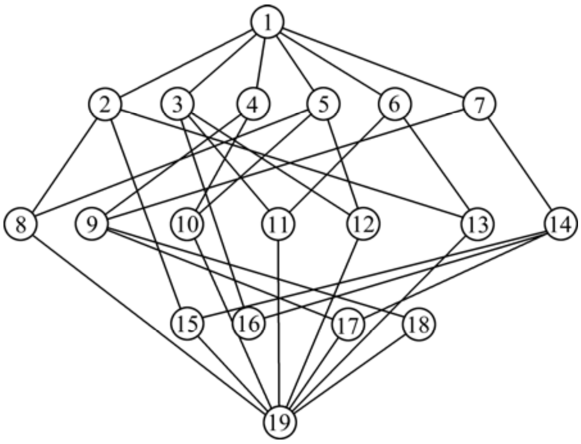

Fig. 3 Hasse graph of $C_S(C_1)$

all requests of access control. Furthermore, the attribute-based policy is not restricted to identity of subject by using the attributes of requester to determine the permission whether authorized or not. During the system running, policy is more stable than attribute. So, the attribute-based policy is good at the separation of attribute managing and access control decision-making. The RBAC model can simplify the permission assignment by classifying the permission set assign to roles and roles assign to users, and RBAC becomes a single attribute special case of ABAC when taking a role as an attribute. The limitations of ABAC are that all the entities must be described by attributes and relations between subject attributes and object attributes must be explicitly explained.

Role engineering [11] focuses on the role mining, optimizing, constraints, and role hierarchy of RBAC using data analysis, graph theory, and other methods [12−15, 21−34]. ZHANG et al [12] used graph optimization to process decomposed matrix and found out the hierarchy roles; SCHLEGELMILCH and STEENS [14] introduced roles generated based on clustering using expert's domain knowledge; VAIDYA et al [15] proposed roles mining using subset enumeration; MOLLOY et al [13] evaluated some popular roles mining methods. VAIDYA et al [21] gave the definition of a role mining problem to express the role optimization. Based on Ref. [21], ENE [22] presented a fast exact and heuristic methods for role minimization problems by using biclique cover and lattice-based postprocessing methods. Aiming at the goal that all permissions are treated evenly in previous approaches, MA et al [33] proposed an algorithm of role mining based on permissions with weights given to reflect their importance to the system, which can find frequent permission sets based on weights scanning the database only once. While the traditional role mining methods need to scan database many times, and the experiments illustrate the effectiveness of algorithm.

The difference between our method and upper algorithms is that we analyze the relations of permission, role, user and attribute sets, and generate a mapping between permission and attribute sets. Then, we use the mapping to generate the information of automatical permission and role assignments when a new user enters. Other works focused on role mining by different methods, such as data mining, graph theory, and Boolean matrix decomposing. Experiments illustrate the effectiveness of this method.

6 Conclusions

1) The relations among permission, role and user sets is analyzed by generating mappings, and the relation between user and attribute sets is described using a concept lattice model

2) An valuable mapping between attribute and permission sets is generated and makes the expression and meaning of role natural and operational, i.e., roles are determined by a permission set and user's attributes.

Acknowledgments

The authors would like to acknowledge Dr. PAN Rong for the helpful advices.

References

[1] SANDHU R, COYNE E J. Role based access control models [J]. IEEE Computer, 1996, 29(2): 38−47.

[2] FOCARDI R, GORRIERI R. Access control: Policies, models, and mechanisms [C]// Proceedings of Foundations of Security Analysis and Design. Bertinoro, Italy, 2000: 137−196.

[3] PARK J H, SANDHU R. The $\mathrm{UCON}_{\mathrm{ABC}}$ usage control model [J]. ACM Transactions on Information and System Security, 2004, 7(1): 128−174.

[4] ZHANG X, LI Y, NALLA D. An attribute-based access matrix model [C]// Proceedings of the 2005 ACM Symposium on Applied Computing. Santa Fe, USA, 2005: 359−363.

[5] LI Xiao-feng, FENG Deng-guo, CHEN Chao-wu, FANG Zi-he. Model for attribute based access control [J]. Journal on Communications, 2008, 29(4): 90−98. (in Chinese)

[6] THOMAS R K, SANDHU R S. Task-based authentication controls (TABC): A family of models for active and enterprise-oriented authentication management [C]// Proceedings of the IFIP WG11.3 Workshop on Database Security. Lake Tahoe, California, 1997: 11−13.

[7] BARKER S, SERGOT M J, WIJESEKERA D. Status-based access control [J]. ACM Transactions on Information and System Security, 2008, 12(1): 1−47.

[8] YANG Qiu-wei, HONG Fan, YANG Mu-xiang. Security analysis on administrative model of role-based access control [J]. Journal of Software, 2006, 17(8): 1804−1810. (in Chinese)

[9] SASTURKAR A, YANG Ping, STOLLER S D. Policy analysis for administrative role based access control [C]// Proceedings of the 19th IEEE Workshop on Computer Security Foundations. Venice, Italy, 2006: 183−196.

[10] LIU Qiang, JIANG Yun-fei, RAO Dong-ning. Safety analysis of ARBAC policy based on graphplan [J]. Chinese Journal of

Computers, 2009, 32(5): 910−921. (in Chinese)

[11] COYNE E J. Role-engineering [C]. Proceedings of 1st ACM Workshop on Role-Based Access Control. Maryland, USA, 1995.

[12] ZHANG D, RAMAMOHANRAO K, EBRINGER T. Role engineering using graph optimisation [C]// Proceedings of Symposium on Access Control Models and Technologies (SACMAT). Autipolis, France, 2007: 139−144.

[13] MOLLOY I, LI N, LI T, MAO Z, WANG Q, LOBO J. Evaluating role mining algorithms [C]// Proceedings of the 14th ACM Symposium on Access Control Models and Technologies (SACMAT). Stresa, Italy, 2009: 95−104.

[14] SCHLEGELMILCH J, STEENS U. Role mining with orca [C]// Proceedings of Symposium on Access Control Models and Technologies (SACMAT). ACM, Stockholm, Sweden, 2005.

[15] VAIDYA J, ATLURI V, WARNER J. Roleminer: Mining roles using subset enumeration [C]// Proceedings of the 13th ACM Conference on Computer and Communications Security. 2006: 144−153.

[16] ANSI, ANSI INCITS 359-2004 for Role Based Access Control, 2004.

[17] FERRAIOLO D F, GILBERT D M, LYNCH N. An examination of federal and commercial access control policy needs [C]// Proceedings of NIST-NCSC National Computer Security Conference. Baltimore, USA, 1993: 107−116.

[18] MICHALSKI R S, ROSENFELD A, DURIC Z, MALOOF M, ZHANG Q. Application of machine learning in computer vision [C]// MICHALSKI R S, BRATKO I, KUBAT M, eds, Machine Learning and Data Mining: Methods and Applications. London: John Wiley & Sons, 1997: 83−113.

[19] GANTER B, WILLE R. Formal concept analysis: Mathematical foundations [M]. Berlin: Springer-Verlag, 1999: 1−5.

[20] WANG Guo-yin, YAO Yi-yu, YU Hong. A survey on rough set theory and applications [J]. Chinese Journal of Computers, 2009, 32(7): 1229−1246. (in Chinese)

[21] VAIDYA J, ATLURI V, GUO Qi. The role mining problem: Finding a minimal descriptive set of roles [C]// Proceedings of Symposium on Access Control Models and Technologies (SACMAT). Antipolis, France, 2007: 175−184.

[22] ENE A, HORNE W, MILOSAVLJEVIC N, RAO P, SCHREIBER R, TARJAN R. Fast exact and heuristic methods for role minimization problems [C]// In The ACM Symposium on Access Control Models and Technologies. Colorado, USA, 2008.

[23] COLANTONIO A, DI PIETRO R, OCELLO A, VINCENZO VERDE N. Taming role mining complexity in RBAC [J]. Computers & Security, 2010, 29: 548−564.

[24] FRANK M, BUHMANN J M, BASIN D. On the definition of role mining [C]// Proceedings of Symposium on Access Control Models and Technologies (SACMAT). Pittsburgh, USA, 2010: 35−44.

[25] TAKABI H, JAMES B. D. JOSHI. StateMiner: An efficient similarity-based approach for optimal mining of role hierarchy [C]// Proceedings of Symposium on Access Control Models and Technologies (SACMAT). Pittsburgh, USA, 2010: 55−64.

[26] HU Jin-wei, ZHANG Yan, LI Rui-xuan, LU Zheng-ding. Role updating for assignments [C]// Proceedings of Symposium on Access Control Models and Technologies (SACMAT). Pittsburgh, USA, 2010: 89−98.

[27] ZHANG Da-na, RAMAMOHANARAO K, VERSTEEG S. Graph based strategies to role engineering [C]// Proceedings of Symposium on Access Control Models and Technologies (SACMAT). Oak Ridge, Tennessee, USA, 2010.

[28] GONCALVES G, PONISZEWSKA-MARANDA A. Role engineering: From design to evolution of security schemes [J]. The Journal of Systems and Software, 2008, 81: 1306−1326.

[29] LU H, VAIDYA J, ATLURI V. Optimal boolean matrix decomposition: Application to role engineering [C]// ICDE ’08. Washington, DC, USA. IEEE Computer Society. 2008: 297−306.

[30] MOLLOY I, CHEN H, LI T, WANG Q, LI N, BERTINO E, CALO S, LOBO J. Mining roles with semantic meanings [C]// Proceedings of Symposium on Access Control Models and Technologies (SACMAT). Colorado, USA, 2008: 21−30.

[31] FRANK M, BASIN D, BUHMANN J M. A class of probabilistic models for role engineering [C]// Proceedings of 15th ACM conference on Computers and Communications Security. Alexandria, Virginia, USA. 2008: 299−309.

[32] COLANTONIO A, DI PIETRO R, OCELLO A, VERDE N V. A formal framework to elicit roles with business meaning in RBAC systems [C]// Proceedings of the 14th ACM Symposium on Access Control Models and Technologies. Stresa, Italy, 2009: 85–94.

[33] MA Xiao-pu, LI Rui-xuan, LU Zheng-ding. Role mining based on weights [C]// Proceedings of Symposium on Access Control Models and Technologies (SACMAT). Pittsburgh, Pennsylvania, USA. 2010: 65−74.

[34] VAIDYA J, ATLURI V, WARNER J. Role engineering via prioritized subset enumeration [J]. IEEE Transactions on Dependable and Secure Computing, 2010, 7(3): 300−314.

(Edited by DENG Lü-xiang)