Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 2935638, 16 pages
http://dx.doi.org/10.1155/2016/2935638

Research Article

A 4D-Role Based Access Control Model for Multitenancy Cloud Platform

Jiangfeng Li, Zhenyu Liao, Chenxi Zhang, and Yang Shi

School of Software Engineering, Tongji University, Shanghai 201804, China

Correspondence should be addressed to Chenxi Zhang; zhangcx2000@163.com

Received 15 August 2015; Revised 26 January 2016; Accepted 31 January 2016

Academic Editor: Anders Eriksson

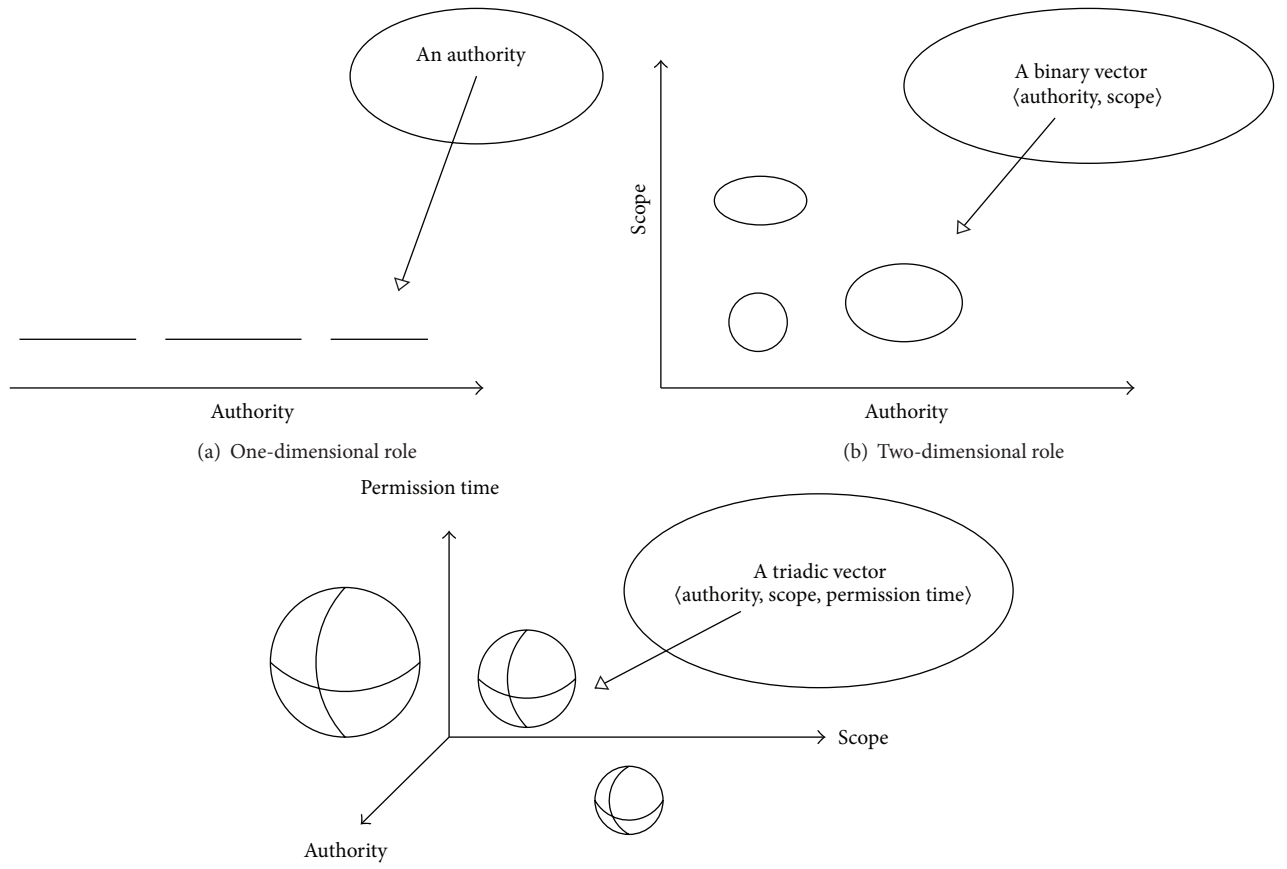
Since more and more applications and services have been transferred from servers in the B/S architecture to cloud, user access control has become a significant part in a multitenancy cloud platform. Role based access control model makes users participate in an enterprise system as particular identities. However, in a multitenancy cloud environment, it has a high probability that the information of tenants has been leaked by using existing role based access control (RBAC) model. Moreover, management problems may emerge in the multitenancy platform with the increment of the number of tenants. In this paper, a novel concept of 4D-role is presented. With a detailed definition on the concept of 4D-role, a 4D-role based multitenancy model is proposed for running various applications and services in the multitenancy cloud platform. A theoretical analysis indicates that the model has the characters of tenant isolation, role hierarchy, and administration independence. The three characters are also verified by experimental evaluation. Moreover, the evaluation results indicate that the model has a good performance in using cloud resources when large-scale users are operating in the cloud platform simultaneously.

1. Introduction

With the rapid development in computer technology, more and more applications and services have been transferred from servers in the B/S architecture to cloud platforms. In the last decade, cloud computing has attracted plenty of enterprises by its powerful processing capabilities, large storage, and low cost. As cloud computing gains popularity, it is a necessity that cloud-service providers must consider several issues when they offer services to customers, such as the safety of customer privacy, and the shifting of the original authorization. Multitenancy is a key technology for nearly every cloud computing platform, and the main purpose of it is to separate user data and ensure that users are not influenced by each other [1]. The technology makes a cloud provider able to rent its services to multiple tenants simultaneously. However, how the tenants in a multitenancy environment can access resources safely is a significant problem that needs to be solved. Moreover, it is quite necessary to find solutions that make the cloud provider manage different tenants efficiently.

Role based access control (RBAC) model has been widely applied to authorize certain users to access certain data or resources within complex systems [2–6]. In the role based access control model [7–10], role is a collection of access rights or permission. It is a concept of one-dimensional space. Role is also considered as a binary vector of authority and scope in [11], which is a concept of two-dimensional space. Moreover, considering that users participate in an enterprise system as a particular identity, it results in that roles of different users have different authorities, and each authority is valid in a relevant scope during its permission time. So, a three-dimensional role based user management model is proposed [12]. The three-dimensional role is defined as a vector composed of authority, scope, and permission time. Figure 1 gives a sketch of multidimensional roles in the existing access control models.

Although the existing RBAC model has considered three dimensions, it still has a high probability of information leakage in the multitenancy environment. Since RBAC based systems administer user authorization using a centralized control mechanism [13–15], there may be various RBAC strategies serving in the multitenancy environment. This often results in information leakage. On the other hand, management problems may emerge in the multitenancy cloud



(a) One-dimensional role

(b) Two-dimensional role

(c) Three-dimensional role

Figure 1: A sketch of multidimensional roles in access control models.

platform employing the existing RBAC model. In the multitenancy environment, RBAC authorizing mechanisms established by different enterprise managers are applied to a distribution environment [16], which also causes management problems.

In order to solve the problems above, it is necessary for the multitenancy cloud platform to have characters of tenant isolation, role hierarchy, and administration independence. The character of tenant isolation, which is the isolation for the operational fields of tenants, prevents a customer's data from being violated by other tenants. Role hierarchy means that roles of tenant users and administrator users construct hierarchical structures. It makes the multitenant cloud provider manage tenants who establish role authorizing mechanism efficiently. Administration independence ensures the cloud administration independent of any tenancy application. It not only prevents tenant information from being leaked by cloud provider, but also brings benefits for solving management problems in multitenancy cloud platform.

In this paper, we focus on designing an authorization mechanism in multitenancy environment, in order to protect cloud tenant's privacy and improve the efficiency of the cloud management. The contributions of the paper include the following: (1) a novel concept of 4D-role is proposed; (2) a 4D-role based multitenancy model is constructed in Section 2; (3) three characters of the model, tenant isolation, role hierarchy, and administration independence, are analyzed theoretically by proving in mathematical approaches in Section 3; (4) Experimental evaluation was performed in Section 4 to verify the model.

2. 4D-Role Based Multitenancy Model

In the multitenancy environment in cloud platforms, a user's role is not invariable. It varies dynamically in terms of the changing of enterprise activities. So, the system should grant multiple roles dynamically with the development of enterprise. In our definition, role is a combination of permission, scope, valid time, and user category, which is composed of a four-dimensional space. In this section, we introduce the 4D-role and the 4D-role based multitenancy model.

2.1. Basic Concepts

Definition 1 (basic permission). Basic permission is an access operation that a system owns, which is an atomic operation and cannot be divided any more. Let $B = \{b_1, b_2, b_3, \ldots, b_{n_b}\}$ be the set of all basic permissions, where b_i is the ith basic permission.

Definition 2 (permission). Permission is a union of finite basic permissions. A permission that contains m ($m \leq n_b$)

basic permissions is formulated as $p = \bigcup_{i=1}^{m} b_i$ ($b_i \in B$). Let $P = \{p_1, p_2, p_3, \ldots, p_{n_p}\}$ be the set of all permissions, where p_i is the ith permission.

The basic permission and the permission give the operation types that are using the cloud platform.

Definition 3 (scope). Scope is an operation place where the permitted operation is valid in a cloud platform. The scope is related to a permitted operation. Let $S = \{s_1, s_2, s_3, \ldots, s_{n_s}\}$ be the set of all scopes, where s_i is the ith scope.

The scope specifies the area where an operation is valid in the cloud platform.

Definition 4 (valid time). Valid time is the period of time when a permitted operation is allowed to be operated in a cloud platform. A valid time $t = [t_{\text{begin}}, t_{\text{end}}]$ is a closed interval from a starting time t_{begin} to an ending time t_{end}. Let $T = \{t_1, t_2, t_3, \ldots, t_{n_t}\}$ be the set of all valid time, where t_i is the ith valid time.

The valid time points out a period of time when an account is allowed to operate in a cloud platform. Time is increasing without a bound. However, an account in the cloud platform may have an expiration date or time. For instance, accounts of an enterprise that rents cloud resources for one year expire after one year. In another case, the valid time of a staff's account ends at the day he leaves the enterprise.

Definition 5 (user category). User category is a class that a user belongs to in cloud platforms. Let $C = \{c_1, c_2, c_3, \ldots, c_{n_c}\}$ be the set of all categories, where c_i is the ith category.

The user category makes sure to which type a user account belongs.

There are various users in the cloud platform. Such users can be divided into some categories according to the user type. For example, users may be put into categories of normal user, senior user, and administrator user.

2.2. 4D-Role and 4D-Role Based User Group

2.2.1. 4D-Role. In a multitenancy cloud platform, users are divided into two categories. One is the category of tenant user. The tenant users, such as enterprise staffs, are those users who rent the cloud resources and use the cloud platform. The other is the category of platform user. The platform users operate and manage the cloud platform. Such works include assigning new resources to tenant users and managing the accounts of tenant users. So, according to Definition 5 in Section 2.1, there are two user categories in our cloud platform. The set of user category is $C = \{c_1, c_2\}$, and c_1 = tenant and c_2 = platform.

Definition 6 (role, tenant role, and platform role).

(1) Role is a set of finite tuples formed as $\langle p, s, t, c\rangle$. A role is represented as $r = \langle p, s, t, c\rangle$, where $p \in P$, $s \in S$, $t \in T$, $c \in C$. Let $R = \{r_1, r_2, r_3, \ldots, r_{n_r}\}$ be the set of all roles, where r_i is the ith role.

(2) Let $r = \langle p, s, t, c\rangle$ be a role:

(a) The role r is a tenant role, if and only if c = tenant.

(b) The role r is a Platform Role, if and only if c = platform.

From Definition 6, we can find that the role is a four-dimensional concept. A role has permissions in relevant scopes during the period of valid time. One kind of permission comprises a few basic permissions. Also, there are two types of role, tenant role and platform role.

2.2.2. 4D-Role Based User Group. A multitenancy cloud platform is a user operation oriented platform. There are a large number of users in a practical cloud platform, which makes user management complicated. In order to make things simpler, we could consider a group of users who have the same permissions, scopes, valid time, and user category.

Definition 7 (user group, tenant user group, and platform user group).

(1) User group, which is composed of finite roles, is a set of unions of finite roles. A user group that includes k ($m \le n_r$) roles is formulated as $g = \bigcup_{i=1}^{k} r_i$ ($r_i \in R$). Let $G = \{g_1, g_2, g_3, \ldots, g_{n_g}\}$ be the set of all user groups, where g_i is the ith user group.

(2) Let $g = \bigcup_{i=1}^{k} r_i$ ($r_i \in R$) be a user group.

(a) The user group g is a tenant user group, if and only if every r_i is a tenant role.

(b) The user group g is a platform user group, if and only if every r_i is a platform role.

Definition 8 (tenant). A tenant is a set of tenant user groups which are in the same enterprise. A tenant has the following properties:

(1) Let G_A be a tenant whose user groups are in the enterprise A:

$$\forall g_m^{(a)} = \bigcup_{m_i} \left\{ \left\langle p_{k_{m_i}}^{(a)}, s_{n_{m_i}}^{(a)}, t_{l_{m_i}}^{(a)}, c_{h_{m_i}}^{(a)} \right\rangle \mid p_{k_{m_i}}^{(a)} \in P,\ s_{n_{m_i}}^{(a)} \in S,\ t_{l_{m_i}}^{(a)} \in T,\ c_{h_{m_i}}^{(a)} = \text{Tenant} \right\} \in G_A. \tag{1}$$

(a) The union of all permissions in G_A is $P_A = \bigcup_m (\bigcup_{m_i} p_{k_{m_i}}^{(a)}) = \bigcup_j b_j^{(a)} \subseteq B$, where $b_j^{(a)}$ is the jth basic permission in the set of all basic permissions in Tenant G_A.

(b) The union of all scopes in G_A is $S_A = \bigcup_m (\bigcup_{m_i} s_{n_{m_i}}^{(a)}) \subseteq S$.

(2) Let G_A and G_B be two tenants whose user groups are in enterprises A and B, respectively:

$$\forall g_m^{(a)} = \bigcup_{m_i} \left\{ \left\langle p_{k_{m_i}}^{(a)}, s_{n_{m_i}}^{(a)}, t_{l_{m_i}}^{(a)}, c_{h_{m_i}}^{(a)} \right\rangle \mid p_{k_{m_i}}^{(a)} \in P,\ s_{n_{m_i}}^{(a)} \in S,\ t_{l_{m_i}}^{(a)} \in T,\ c_{h_{m_i}}^{(a)} = \text{Tenant} \right\} \in G_A,$$

$$\forall g_{m'}^{(b)} = \bigcup_{m_i'} \left\{ \left\langle p_{k_{m_i'}}^{(b)}, s_{n_{m_i'}}^{(b)}, t_{l_{m_i'}}^{(b)}, c_{h_{m_i'}}^{(b)} \right\rangle \mid p_{k_{m_i'}}^{(b)} \in P,\ s_{n_{m_i'}}^{(b)} \in S,\ t_{l_{m_i'}}^{(b)} \in T,\ c_{h_{m_i'}}^{(b)} = \text{Tenant} \right\} \in G_B. \tag{2}$$

$G_A = G_B$, if and only if

$$\begin{aligned} \bigcup_m \left(\bigcup_{m_i} p_{k_{m_i}}^{(a)} \right) &= \bigcup_{m'} \left(\bigcup_{m_i'} p_{k_{m_i'}}^{(b)} \right), \\ \bigcup_m \left(\bigcup_{m_i} s_{n_{m_i}}^{(a)} \right) &= \bigcup_{m'} \left(\bigcup_{m_i'} s_{n_{m_i'}}^{(b)} \right), \\ \bigcup_m \left(\bigcup_{m_i} t_{l_{m_i}}^{(a)} \right) &= \bigcup_{m'} \left(\bigcup_{m_i'} t_{l_{m_i'}}^{(b)} \right). \end{aligned} \tag{3}$$

(3) Let $G_1, G_2 \ldots, G_w$ be w tenants in the cloud platform. Each S_i $(1 \le i \le w)$ is the union of all scopes in G_i. S is the universal set of scopes in the cloud platform. The set $W = \{S_1, S_2, \ldots, S_w,\ S - \bigcup_{i=1}^{w} S_i\}$ is a partition to S.

From the definitions above, we know that the union of all permissions in a tenant has boundary. It is the same as the union of all basic permissions in the tenant. The scope in a tenant also has boundary. A partition of scopes in the cloud platform is defined to ensure that user groups in each tenant have isolated environment in operating in the cloud platform.

In addition, we can find the relations among user group, role, permission, and basic permission. A user group is a set of roles, and a tenant is made up of several user groups which are in the same enterprise. As a role consists of permission, scope, valid time, and user category, in the 4D-role based multitenancy model, permissions are assigned to a user group through 4D-role. Basic permissions are able to be assigned to a user group according to the mapping from permission to basic permission. Figure 2 shows the relation among user group, role, permission, and basic permission.

2.3. User in the 4D-Role Based Multitenancy (4D-RBMT) Model

2.3.1. 4D-Role Based User

Definition 9 (user, tenant user, and platform user).

(1) User is a member of user group. A user belongs to one of user groups. Let G be the set of all user groups and let $u_{i,j}$ be a user. $\forall g \in G$, if $u_{i,j} \in g_i$, then $u_{i,j}$ is called the jth user of g_i.

(2) $u_{i,j}$ is a tenant user, if and only if g_i is a tenant user group.

(3) $u_{i,j}$ is a platform user, if and only if g_i is a platform user group.

Let U be the set of all users:

$$U = \{u_1, u_2, u_3, \ldots, u_n\}, \tag{4}$$

where $|U|$ is the number of elements in set U and u_i is the ith user.

Theorem 10. *User is a union of finite roles.*

Proof. Let U be the set of all users and let G be the set of all user groups.

$\forall u \in U, \exists g_i \in G$, such that $u \in g_i$.

Note that

$$g_i = \left\{ \bigcup_j r_j \mid \left(r_j \in R,\ 1 \le j \le |R| \right) \right\}. \tag{5}$$

Therefore

$$u = \bigcup_{j_0} r_{j_0}, \tag{6}$$

where $r_{j_0} \in R$, $1 \le j_0 \le |R|$. □

2.3.2. Hierarchy of 4D-Role Based Users. There are six types of users in a multitenancy cloud platform. Such users belong to two kinds of users: platform user and tenant user.

(1) Platform user is described as follows.

(a) General Admin. It is the super administrator in the model. New applications and user categories in the cloud platform are set by the General Admin.

(b) Developer. It sets the basic permissions in the multitenancy cloud platform. There are two kinds of basic permissions in the platform. One is the kind of basic permission that is used to run programs for managing the multitenancy cloud platform. The other is the kind of basic permission that is used to run programs for managing applications in the cloud platform. The account of a developer is approved by the General Admin.

(c) Platform Senior Admin. It is responsible for configuring the permission, scope, valid time, role, and user group in the cloud platform. The highest level account, in the hierarchy of the Platform Senior Admin, is approved by a Developer. Also, a Platform Senior Admin is able to add accounts of lower level Platform Senior Admin.

(d) Platform Admin. It is the ordinary administrator to manage the cloud platform. The account of a Platform Admin is approved by a Platform Senior Admin.

(2) Tenant user is described as follows.

(a) Application Administrator. It is responsible for configuring the permission, scope, valid time, role, and user group in the application system. An Application Administrator is able to add accounts of lower level Application Administrators. The highest level account, in the hierarchy of Application Administrator, is approved by a Developer.

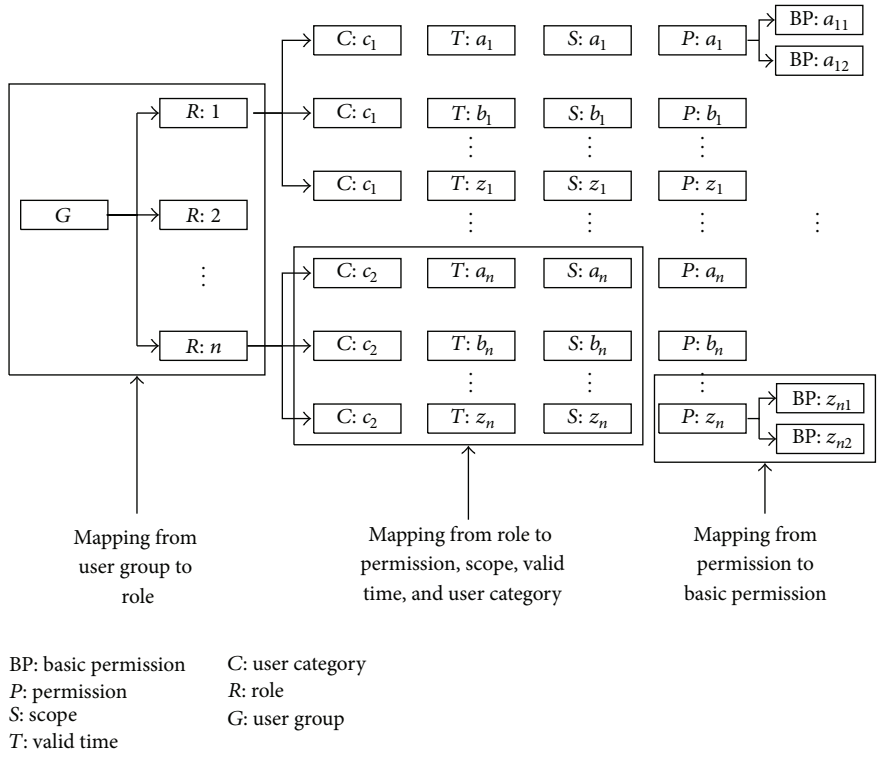


Figure 2: Relations among user group, role, permission, and basic permission.

(b) Application User. It is the terminal user to use an application.

The six types of users above are divided into four levels, according to their duties in the cloud platform. General Admin is in level 1. Developer belongs to level 2. Level 3 contains Platform Senior Admin and Application Administrator. Level 4 includes Platform Admin and Application User. Figure 3 shows the user hierarchy in the 4D-RBMT model.

3. Verification of 4D-RBMT Model Using Mathematical Approaches

A cloud platform with the 4D-role based multitenancy (4D-RBMT) model has characters of independence of platform administration, isolation of tenant operation, and hierarchy of user relations. In this section, we will prove the properties in mathematics approaches. Firstly, relations between two user groups are defined.

3.1. User Group Relation

Definition 11 (relations between user groups). Let G be the set of user groups, $\forall g, g' \in G$; the relation of g and g' is (1) disjoint, if $g \cap g' = \varnothing$, (2) inclusive, if $g \subseteq g'$, and (3) mixed, if $g \cap g' \neq \varnothing$, $g \subsetneq g'$ and $g' \subsetneq g$.

Under the basic fact of set theory, we can find that the three relations, disjoint, inclusive, and mixed, are existing between two user groups in the 4D-RBMT model. Figure 4 shows the relations of projections on the three-dimensional space composed of permission, scope, and valid time, between two user groups which have the same user categories.

3.2. Independence of Platform Administration.

In the 4D-RBMT model based cloud platform, user groups are divided into platform user groups and tenant user groups. According to definitions in Section 2, for any platform user group g_p and tenant user group g_t,

$$\begin{aligned} g_p &= \bigcup_{i=1}^{k} \{r_i \mid r_i \in R\} = \bigcup_{i=1}^{k} \left\{ \left\langle p_{m_i}, s_{n_i}, t_{l_i}, c_{h_i} \right\rangle \mid p_{m_i} \right. \\ &\quad \left. \in P,\ s_{n_i} \in S,\ t_{l_i} \in T,\ c_{h_i} = \text{Platform} \right\}, \\ g_t &= \bigcup_{j=1}^{k'} \left\{ r'_j \mid r'_j \in R \right\} = \bigcup_{j=1}^{k'} \left\{ \left\langle p'_{m_j}, s'_{n_i}, t'_{l_i}, c'_{h_i} \right\rangle \mid p'_{m_i} \right. \\ &\quad \left. \in P,\ s'_{n_i} \in S,\ t'_{l_i} \in T,\ c'_{h_i} = \text{Tenant} \right\}. \end{aligned} \tag{7}$$

Obviously, $g_p \cap g_t = \varnothing$. It means that the intersection of a platform user group and a tenant user group is empty.

So, the disjoint relation is the only relation between a platform user group and a tenant user group. Neither inclusive relation nor mixed relation exists between them. Moreover, operation fields of a platform user are absolutely separated from those of a tenant user. It means that a tenant user cannot operate the fields that a platform user operates. Figure 5 shows the 4D-Role Based User Group's projections on the dimension category. From the figure, we can know that a platform user group is strictly independent on a tenant

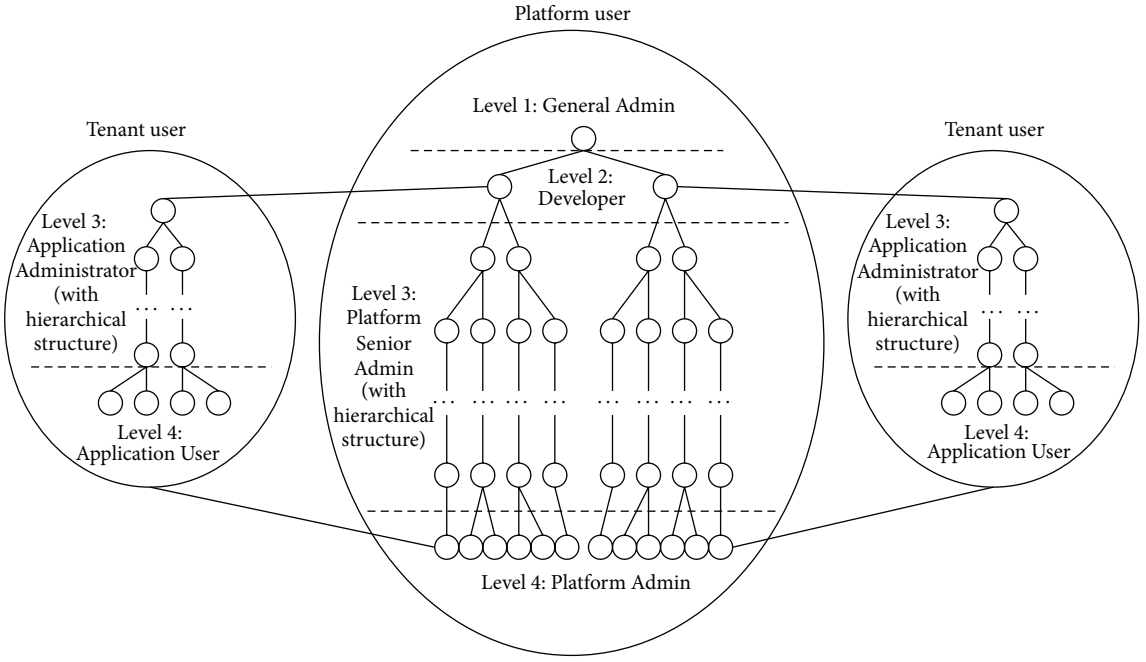


Figure 3: User hierarchy in the 4D-RBMT model.

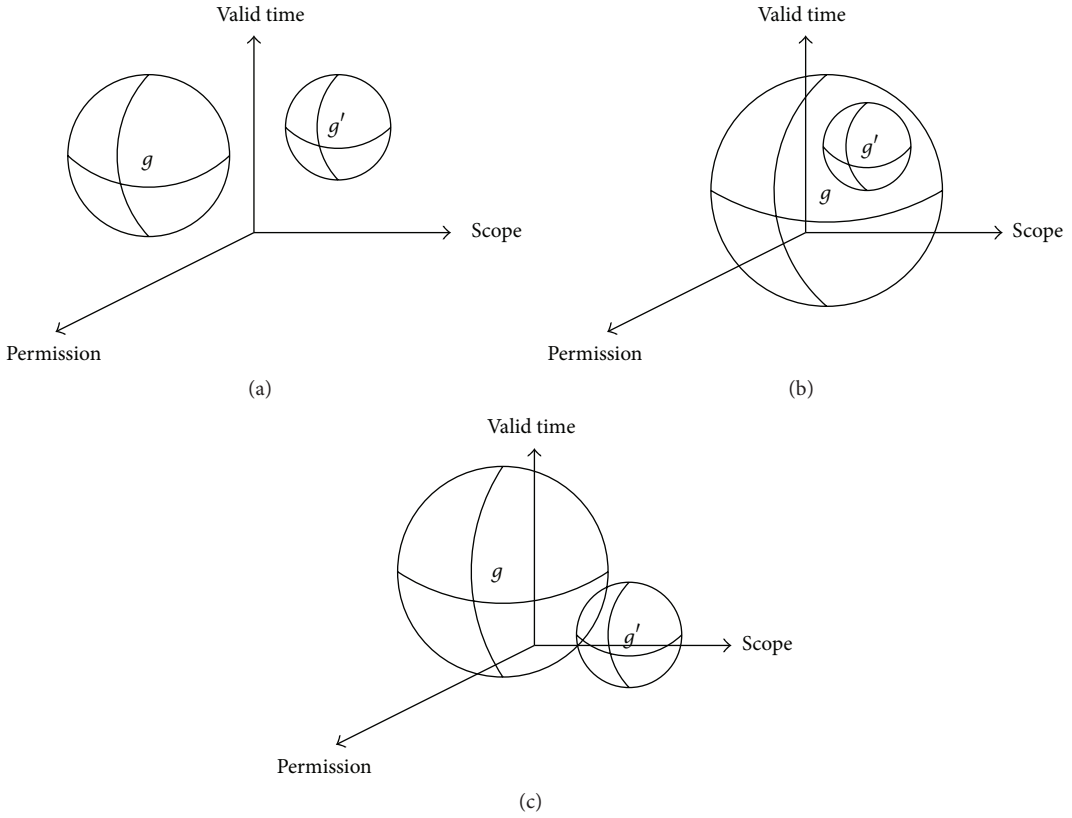


Figure 4: Relations of projections on the three-dimensional space composed of permission, scope, and valid time between two user groups, which have the same user categories. (a) Disjoint relation. (b) Inclusive relation. (c) Mixed relation.

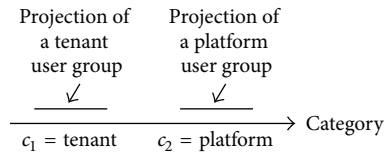


Figure 5: User group's projection on the dimension category.

user group, which means that the platform administration is independent of any tenancy application.

3.3. Isolation of Tenant Operation. In the 4D-RBMT model, operation field of user groups in a tenant is strictly isolated from that in another different tenant. It is attributed to the fact that two user groups which belong to different tenants have disjoint relation only, which means that the two user groups have an empty intersection. Theorem 12 below proves it.

Theorem 12. *Let G_A and G_B be two tenants whose user groups are in enterprises A and B, respectively. $\forall g_m^{(a)} \in G_A$, $\forall g_{m'}^{(b)} \in G_B$, if $G_A \neq G_B$, then $g_m^{(a)} \cap g_{m'}^{(b)} = \varnothing$.*

Proof. A method of proof by contradiction is used to prove the theorem.

Suppose that $\exists g_m^{(a)} \in G_A$, $g_{m'}^{(b)} \in G_B$, such that $g_m^{(a)} \cap g_{m'}^{(b)} \neq \varnothing$, where

$$\begin{aligned} g_m^{(a)} &= \bigcup_{m_i} \left\{ \left\langle p_{k_{m_i}}^{(a)}, s_{n_{m_i}}^{(a)}, t_{l_{m_i}}^{(a)}, c_{h_{m_i}}^{(a)} \right\rangle \mid p_{k_{m_i}}^{(a)} \in P,\ s_{n_{m_i}}^{(a)} \right. \\ &\quad \left. \in S,\ t_{l_{m_i}}^{(a)} \in T,\ c_{h_{m_i}}^{(a)} = \text{Tenant} \right\}, \\ g_{m'}^{(b)} &= \bigcup_{m_i'} \left\{ \left\langle p_{k_{m_i'}}^{(b)}, s_{n_{m_i'}}^{(b)}, t_{l_{m_i'}}^{(b)}, c_{h_{m_i'}}^{(b)} \right\rangle \mid p_{k_{m_i'}}^{(b)} \in P,\ s_{n_{m_i'}}^{(b)} \right. \\ &\quad \left. \in S,\ t_{l_{m_i'}}^{(b)} \in T,\ c_{h_{m_i'}}^{(b)} = \text{Tenant} \right\} \end{aligned} \tag{8}$$

that means $\exists r_0 = \langle p_0, s_0, t_0, \text{Tenant} \rangle$, such that $g_m^{(a)} \cap g_{m'}^{(b)} = r_0$.

Thus

$$s_0 \in s_{n_{m_i}}^{(a)} \cap s_{n_{m_i'}}^{(b)}. \tag{9}$$

It implies that

$$s_0 \in \left(\bigcup_m \left(\bigcup_{m_i} s_{n_{m_i}}^{(a)} \right) \right) \cap \left(\bigcup_{m'} \left(\bigcup_{m_i'} s_{n_{m_i'}}^{(b)} \right) \right). \tag{10}$$

$S_1 = (\bigcup_m (\bigcup_{m_i} s_{n_{m_i}}^{(a)}))$ is the scope of G_A and $S_2 = (\bigcup_{m'} (\bigcup_{m_i'} s_{n_{m_i'}}^{(b)}))$ is the scope of G_B.

Since $G_A \neq G_B$, according to Definition 8, the set $W = \{S_1, S_2, \ldots, S_w,\ S - \bigcup_{i=1}^{w} S_i\}$ is a partition to S, where S is the universal set of scopes in the cloud platform.

So,

$$S_1 \cap S_2 = \varnothing. \tag{11}$$

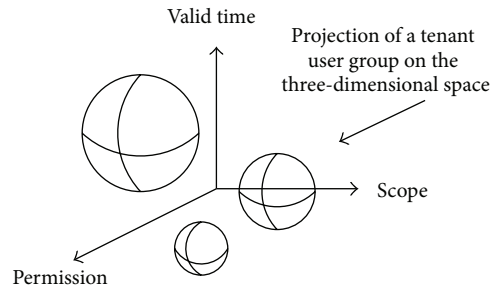


Figure 6: User group's projection on the three-dimensional space composed of permission, scope, and valid time.

We know that

$$\left(\bigcup_m \left(\bigcup_{m_i} s_{n_{m_i}}^{(a)} \right) \right) \cap \left(\bigcup_{m'} \left(\bigcup_{m_i'} s_{n_{m_i'}}^{(b)} \right) \right) = \varnothing. \tag{12}$$

At this time, there is a contradiction between (10) and (12), which means that the hypothesis $g_m^{(a)} \cap g_{m'}^{(b)} \neq \varnothing$ is false.

Thus, $g_m^{(a)} \cap g_{m'}^{(b)} = \varnothing$. □

Theorem 12 proves that the 4D-RBMT model has a character of tenant isolation. Figure 6 shows the 4D-Role Based User Group's projections to the three-dimensional space composed of permission, scope, and valid time. From the figure, we can find that two user groups which belong to two different tenants only have the relation of disjoint, which means that user groups in one tenant are absolutely isolated from any user groups in another tenant.

3.4. Hierarchy of User Relations. In Section 3.1, we have known that there are three relations existing between user groups, such as disjoint relation, inclusive relation, and mixed relation. Moreover, in the two user groups which are inclusive, the permissions, scopes, and valid time of the subgroup are not more than those of supergroup, and the two categories are the same.

Theorem 13. *Let G be the set of user groups, $\forall g, g' \in G$; if $g \subseteq g'$, then $P \subseteq P'$, $S \subseteq S'$, $T \subseteq T'$, and $c_{k_i} = c'_{k_j}$, where P and P' are the permissions of g and g', S and S' are the scopes of g and g', T and T' are the valid time of g and g', and c_k and c'_k are the categories of g and g', respectively.*

Proof. Consider $\forall g, g' \in G$:

$$\begin{aligned} g &= \left\{ \bigcup_i r_i \mid (r_i \in R,\ 1 \leq i \leq |R|) \right\}, \\ g' &= \left\{ \bigcup_j r'_j \mid \left(r'_j \in R',\ 1 \leq j \leq \left|R'\right| \right) \right\}. \end{aligned} \tag{13}$$

Since $g \subseteq g'$, so that $\forall 1 \leq i \leq |R|$, $\exists 1 \leq j_i \leq |R'|$, we have $r'_{j_i} = r_i$.

Note that

$$r_i = \left\{\left\langle p_i, s_i, t_i, c_{k_i}\right\rangle \mid \left(p_i \in P,\ s_i \in S,\ t_i \in T,\ 1 \le i \le |P|,\ c_{k_i} \in C\right)\right\},$$
$$r'_{j_i} = \left\{\left\langle p'_{j_i}, s'_{j_i}, t'_{j_i}, c'_{k_{j_i}}\right\rangle \mid \left(p'_{j_i} \in P',\ s'_{j_i} \in S',\ t'_{j_i} \in T',\ 1 \le j_i \le \left|P'\right|,\ c'_{k_{j_i}} \in C\right)\right\}. \quad (14)$$

This implies that

$$\left\langle p'_{j_i}, s'_{j_i}, t'_{j_i}, c'_{k_{j_i}}\right\rangle = \left\langle p_i, s_i, t_i, c_{k_i}\right\rangle. \quad (15)$$

Therefore $\forall 1 \le i \le |R|$, $\exists 1 \le j_i \le |R'|$, such that $p'_{j_i} = p_i$, $s'_{j_i} = s_i$, and $t'_{j_i} = t_i$.

Also $c_{k_i} = c'_{k_{j_i}}$.

So $P \subseteq P'$, $S \subseteq S'$, and $T \subseteq T'$.

Now we prove $c_{k_i} = c'_{k_{j_i}}$.

Since g, g' are two nonempty user groups, and $g \subseteq g'$, it means that $g \cap g' \neq \varnothing$.

If $c_{k_i} \neq c'_{k_{j_i}}$, that means, in the two user groups g, g', one is tenant user group and the other is platform user group. According to the independent property introduced in Section 3.2, we know that $g \cap g' = \varnothing$. It is a contradiction against the fact $g \cap g' \neq \varnothing$.

Thus, $P \subseteq P'$, $S \subseteq S'$, $T \subseteq T'$, and $c_{k_i} = c'_{k_{j_i}}$. □

Theorem 13 above indicates that if two user groups are inclusive, the permissions, scopes, and valid time of the two user groups also have inclusive relations.

4. Performance Evaluations

For demonstrating our proposed 4D-RBMT model, we firstly implement a prototype system considering typical enterprise operations that are running in multitenancy cloud environment. The proposed prototype system is developed using Java language. With Spring MVC framework, it is convenient to provide necessary APIs which can be visited by other systems. Then, hotel and restaurant business are separately running in the multitenancy platform provided by a cloud provider, which is used to evaluate the characters of platform administration independence, role hierarchy, and tenant isolation in the 4D-RBMT model. Last, performance of the prototype system using cloud resources is evaluated.

The experiments are conducted on a Vmware Cloud environment. The computer resources consist of 20 VMware-server virtual machines (each has 2 GB memory and runs CentOS7) hosted on two vSphere servers with dual 2.6 GHz Xeon CPU and 96 GB memory. Each virtual machine is one of the tenants in the cloud. The prototype system is deployed on every virtual machine and provides remote access through APIs relative to roles.

4.1. Introduction to the Prototype System

4.1.1. System Structure of the Prototype System. There are four modules in the 4D-RBMT model, 4D-Role Based Power Configuration Module (4D-Role PCM), User Account Module (UAM), Power Validation Module (PVM), and Apply and Permit Interface (API). In the structure, 4D-Role PCM and UAM are communicated through the API, which is supported by the PVM. Figure 7 is the structure of the 4D-RBMT model.

4D-Role PCM is the key module in the 4D-RBMT model. It deals with the operation of power configuration management, which involves the power configurations, such as Basic Permission Configuration (BPC), Permission Configuration (PC), Role Configuration (RC), and User Group Configuration (UGC).

In BPC, a basic permission is one of the operations in adding, deleting, and updating. The configuration of basic permission in BPC is operated by a system operator (SYSOP). In PC, permission is a union of finite basic permissions. Fetched from BPC, basic permissions are combined for certain requirements of the business applications. In RC, role is a set of finite four-dimensional vectors composed of permissions, scopes, valid time, and user categories. Obtained from AC, an authority is matched with a scope and permission time for certain requirements of the business. In UGC, user group is a set of unions of finite roles. Obtained from RC, roles are combined for certain requirements.

The User Account Module (UAM) contains the operation that users register to get accounts and update their personal information. Also, user information can be imported from or exported to files. Finally, every user belongs to a user group which is configured in the Power Configuration Module (PCM).

Power application and power permission are two key operations in Power Validation Module (PVM). A user previously applies for joining in a user group through operation of power application. An administrator permits a user's application to join in a user group after verifying the information of the applicant through operation of power permission. In addition, an administrator can remove a user from any user group.

Apply and Permit Interface (API), supported by the PVM, is an interface connecting UAM and PCM.

4.1.2. Workflow of 4D-Role Based User Assignment. As we know, users in the 4D-RBMT model have six types. Users in different types have their own duties in the model. The General Admin works at setting cloud application and user categories. A developer is responsible for setting basic permissions. Platform user groups and tenant user groups are configured by Platform Senior Admin and Application Administrator, respectively. A Platform Admin or an Application User needs to register first and then be assigned to a user group. Figure 8 shows the workflow of user group configuration and assignment in the 4D-RBMT model.

4.2. Experimental Evaluation of Performance in Business Functions

4.2.1. Business Functions

(a) User Functions. Functions of each user in the prototype are listed in Table 1. Depending on the defined users and

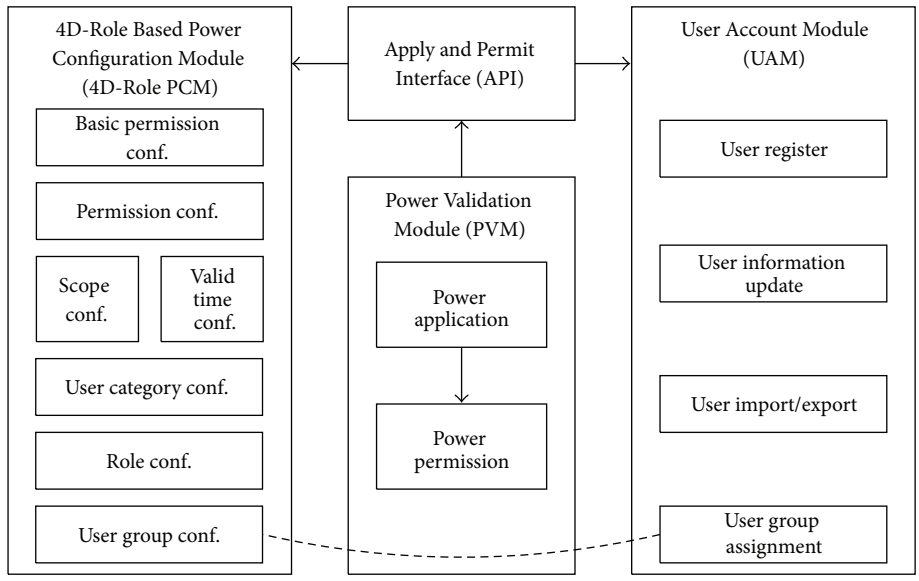


Figure 7: Structure of the 4D-RBMT model.

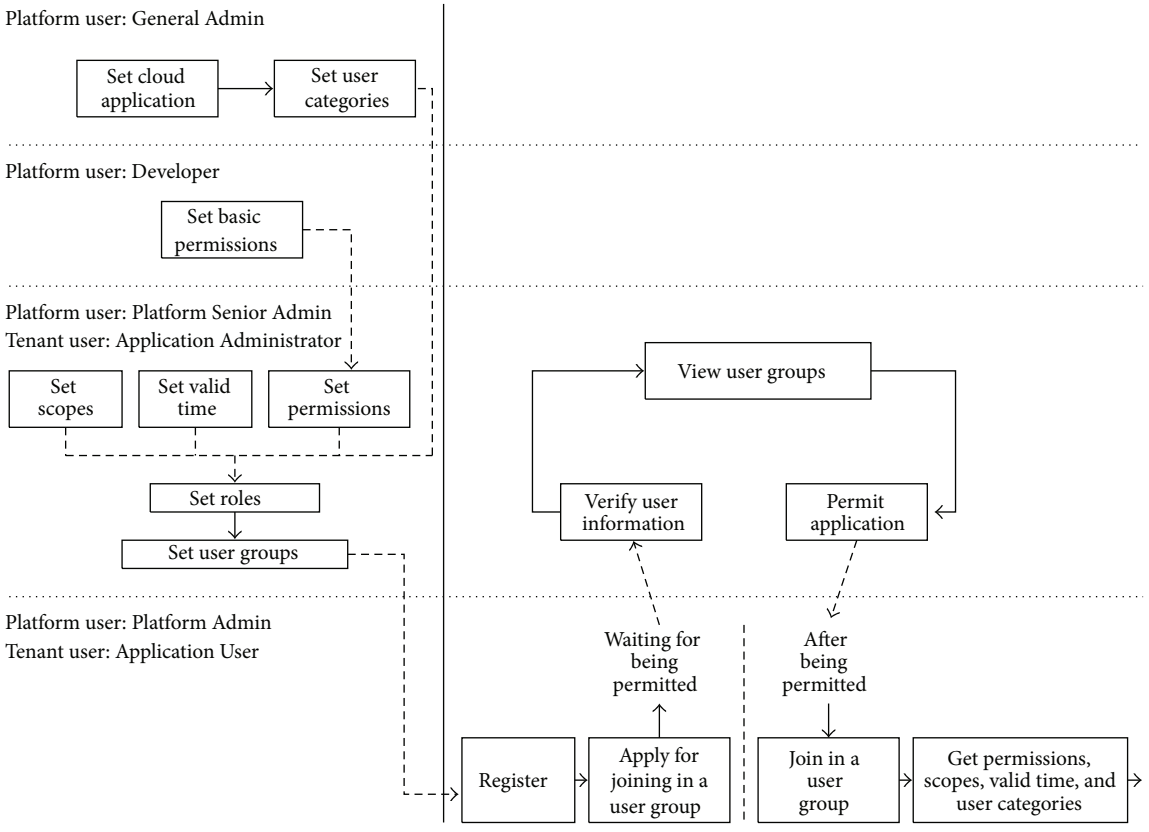


Figure 8: Workflow of the 4D-RBMT model.

functions, a user graph describing the structural relations among the users is shown in Figure 9.

(b) Configuration of 4D-Role Based User. As the 4D-role in the 4D-RBMT model is composed of permission, scope, valid time, and user category, we configure the four elements of 4D-Role starting with the configuration of basic permission, while permission is a union of one or more basic permissions.

We consider three types of basic permissions (BP) in the prototype. The first type is the management of information, including create, delete, update, and view on the conventional data items. The second type is related to the affairs about

Table 1: Users and functions.

Number	User	User type	Functions
$U1$	General Admin	General Admin	Set applications and user categories
$U2$	Application Developer	Developer	Set basic permissions of applications
$U3$	Platform Developer	Developer	Set basic permissions of the platform
$U4$	Cloud platform manager	Platform Senior Admin	Manage cloud resource lease affairs for all the enterprises
$U5$	Manager of admin for manufacture enterprises	Platform Senior Admin	Manage cloud resource lease affairs for manufacture enterprises
$U6$	Manager of admin for service enterprises	Platform Senior Admin	Manage cloud resource lease affairs for service enterprises
$U7$	Admin of hotel enterprises	Platform Admin	Manage cloud resource lease affairs for hotel enterprises
$U8$	Admin of restaurant enterprises	Platform Admin	Manage cloud resource lease affairs for restaurant enterprises
$U9$	General Manger (hotel)	Application Administrator	Manage all the affairs in the hotel and approve management report
$U10$	Manager of room division	Application Administrator	Manage room affairs and submit room management report
$U11$	Room receptionist	Application User	Manage room reception affairs
$U12$	Room attendant	Application User	Manage room service affairs
$U13$	Manager of finance (hotel)	Application Administrator	Manage finance affairs and submit finance management report
$U14$	Accountant (hotel)	Application User	Manage hotel accounting affairs
$U15$	Cashier (hotel)	Application User	Manage hotel cashier affairs
$U16$	General Manager (restaurant)	Application Administrator	Manage all the affairs in the restaurant and approve management report
$U17$	Manager of Food and Beverage	Application Administrator	Manage food and beverage affairs and submit food and beverage management report
$U18$	Food attendant	Application User	Manage food affairs
$U19$	Beverage attendant	Application User	Manage beverage affairs
$U20$	Manager of Finance (restaurant)	Application Administrator	Manage finance affairs and submit finance report
$U21$	Accountant (restaurant)	Application User	Manage accounting affairs
$U22$	Cashier (restaurant)	Application User	Manage cashier affairs

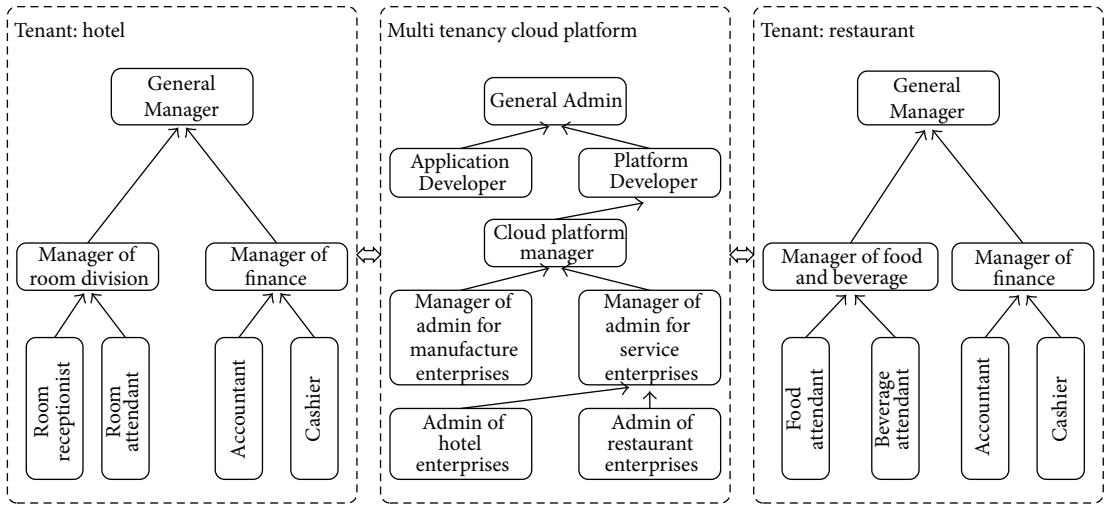


Figure 9: A user graph describing the structural relations among the users.

Table 2: Basic permissions and permissions.

Basic permission number	Basic permission	Permission number	Permission
BP1	Create information		
BP2	Delete information	$P1$	$P1 = BP1 \cup BP2 \cup BP3 \cup BP4$
BP3	Update information		
BP4	View information		
BP5	Submit management report	$P2$	$P2 = BP5$
BP6	Approve management report	$P3$	$P3 = BP6$
BP7	Assign cloud resource	$P4$	$P4 = BP7 \cup BP8$
BP8	Repossess cloud resource		

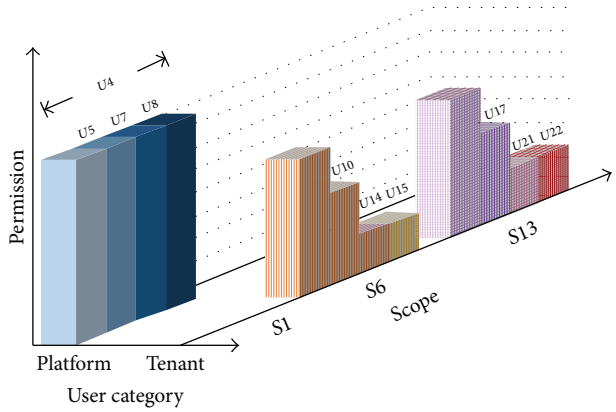


Figure 10: Fields of user duties.

management report, which is composed of basic permission of submit and approve management reports. The third type which consists of assigning cloud resource and repossessing cloud resource is something about the management of cloud resource. Table 2 shows the basic permissions and permissions in our prototype. Scopes are also configured and listed in Table 3. Relations existing among those scopes are also listed.

Besides the permissions and scopes, in our 4D-role concept, user category and valid time are needed to be considered. Platform and tenant are two categories in the prototype. To make system simpler, every valid time ranges from $-\infty$ to ∞.

4.2.2. Evaluation Results. After configuring permissions, scopes, valid time, and user categories as mentioned in Section 4.2, we run the hotel and restaurant business in the prototype implemented in Section 4.1. We examine fields of user duties and find the boundaries of the fields. The users include the ones from both platform and tenants. Figure 10 shows the fields of user duties.

In the figure, there are four platform users, $U4$, $U5$, $U7$, and $U8$. The user type of $U4$, $U5$, and $U7$ is Platform Senior Admin, while the user type of $U8$ is Platform Admin. On the other side, $U10$, $U14$, $U15$, $U17$, $U21$, and $U22$ are from two tenants. We can find from the figure that the fields of platform user are absolutely different from those of users from the two tenants. It indicates that the platform users have no rights to operate any data items in the tenants. The duties of platform users to the tenants are only operating the cloud resources such as assigning resources and repossessing resources. Moreover, the figure illustrates that the platform has an independent administration environment. It means that the administration never depends on applications or enterprises of tenant.

Next, we verify the character of user hierarchy using the prototype. Figure 11 shows the hierarchy structures of the users, no matter what user types they belong to.

In Figure 11(a), the fields of the three users, $U4$, $U5$, and $U7$, have a relation $U4 \supset U5 \supset U7$. It means that the three users build a hierarchy structure. It is attributed to the fact that the corresponding scopes of the three users have relation that $S1 \supset S4 \supset S5$, while the permissions of three users are the same. In Figure 11(b), there is a relation, $U9 \supset U13 \supset U14$, existing in the fields of the three users $U9$, $U13$, and $U14$. The reasons of becoming such hierarchy structure are from two aspects. On one hand, the permissions of the three users are $P1 \cup P2 \cup P3$, $P1 \cup P2$, and $P1$, respectively, which is an inclusive relation. On the other hand, scopes of the three users have inclusive relations that $S6 \supset S10 \supset S11$. The same as $U9$, $U13$, and $U14$ in Figure 11(b), in Figure 11(c), the users $U21$, $U20$, and $U16$ become a hierarchy structure.

Lastly, we compare the fields of the two tenants. Figure 12 shows the fields of the hotel tenant and the restaurant tenant.

Figure 12 illustrates that there is no intersection between the fields of users $U9$ and $U16$. From the hierarchy structures shown in Figure 11, we can know that $U9$ and $U16$ have the largest fields in their own tenants. It infers that the two tenants

Table 3: Scopes and relations.

Scope number	Scope	Scope relations
$S1$	Cloud resource lease affairs for all the enterprises	$S1 \supset (S2 \cup S3)$
$S2$	Cloud resource lease affairs for manufacture enterprises	
$S3$	Cloud resource lease affairs for service enterprises	$S3 = S4 \cup S5$
$S4$	Cloud resource lease affairs for hotel enterprises	
$S5$	Cloud resource lease affairs for restaurant enterprises	
$S6$	Hotel affairs	$S6 \supset (S7 \cup S10)$
$S7$	Hotel room affairs	$S7 = S8 \cup S9$
$S8$	Hotel room reception affairs	
$S9$	Hotel room service affairs	
$S10$	Hotel finance affairs	$S10 = S11 \cup S12$
$S11$	Hotel accounting affairs	
$S12$	Hotel cashier affairs	
$S13$	Restaurant affairs	$S13 \supset (S14 \cup S17)$
$S14$	Food and beverage affairs	$S14 = S15 \cup S16$
$S15$	Food affairs	
$S16$	Beverage affairs	
$S17$	Restaurant finance affairs	$S17 = S18 \cup S19$
$S18$	Restaurant accounting affairs	
$S19$	Restaurant cashier affairs	

are totally isolated. So, the character of tenant isolation is verified.

4.3. Simulation Evaluation of Performance in Using Cloud Resources. It is significant to evaluate the performance of the prototype system in using cloud resources, especially in the cases when a lot of users are using the cloud platform simultaneously. Simulations that simulate behaviors of large-scale users are used to evaluate the performance.

4.3.1. Simulation Parameters. We use simulations to evaluate the performance of our proposed prototype in using cloud resources. In the simulations, performance of tenant systems and cloud platform is evaluated under different loads. The parameters of the simulations that simulate different scenarios include number of users, max number of concurrent users, and runtime. Table 4 shows the details of parameters in the simulations.

In the evaluation, CPU usage and active memory are used to measure the running status of the tenant system and cloud platform. Besides CPU usage and active memory, response time is used to evaluate the system efficiency of our proposed system. There are three kinds of response time that are used to measure the performance. First one is average response time. It reflects the average experience of users. Next, max response time describes the worst case in using this prototype system. Last, line of 99% response time is the value of top 99% shortest response time in simulation, which gives the upper bound of most cases in using this prototype system.

Table 4: Simulation parameters.

Parameter	Value	Description
Number of users	2500, 5000, 7500, 10000, 12500, 15000, 17500, 20000	The number of users in simulation
Max number of concurrent users	400–1000	The max number of concurrent users in simulation
Runtime	30 s–60 s	The runtime of simulation

4.3.2. Evaluation Results. Three rounds of simulations are run under light, normal, and heavy loads. After that, we analyze the data collected from vSphere monitor and evaluate the performance. Figure 13 shows the CPU usage and active memory of one of the tenant systems and the cloud platform.

Figure 13 illustrates that the max value of CPU usage and active memory, in both tenant and cloud platform, is growing with the increase of user number. The usage of CPU in the environment of light load reaches the peak of 22.08% for a tenant and 34.97% for the whole cloud platform. In the scenario of running under the normal load, the maximum of the CPU usage for the tenant and cloud platform is 25.05% and 43.43%, respectively. In the heavy load situation, the usage of CPU increases up to 37.9% for tenant and 48.92% for the cloud platform. Under the heavy load, the value for tenant increases to 51.3% compared with the environment under the normal load and 71.6% compared with that under the light load. Meanwhile, the increase ratios of CPU usage for cloud platform under the heavy load are 12.6% and 39.9%, compared to the situations under the normal and light loads. This indicates that the CPU usage for a cloud platform increases slower than that for a tenant. As for active memory, it has almost the same trend as the CPU usage.

In addition, we evaluate the response time that indicates how long an authorization query is answered. Figure 14 shows the response time in the simulation.

The number of users in simulations ranges from 2500 to 20000. According to Figure 14, the average response time is increasing almost linearly and the line of 99% response time is also increasing smoothly. It indicates that the proposed system can keep a low latency performance in most of the scenarios. As for the max response time, the proposed system has an acceptable performance in these kinds of extreme situations, which appear at a quite low probability.

The evaluation results above imply that our proposed system performed well in situations where large-scale users are operating in the cloud platform. Our proposed system can deal with the surge of user number with limited cloud resource. Meanwhile, it is able to provide high quality and low response service for most of the users.

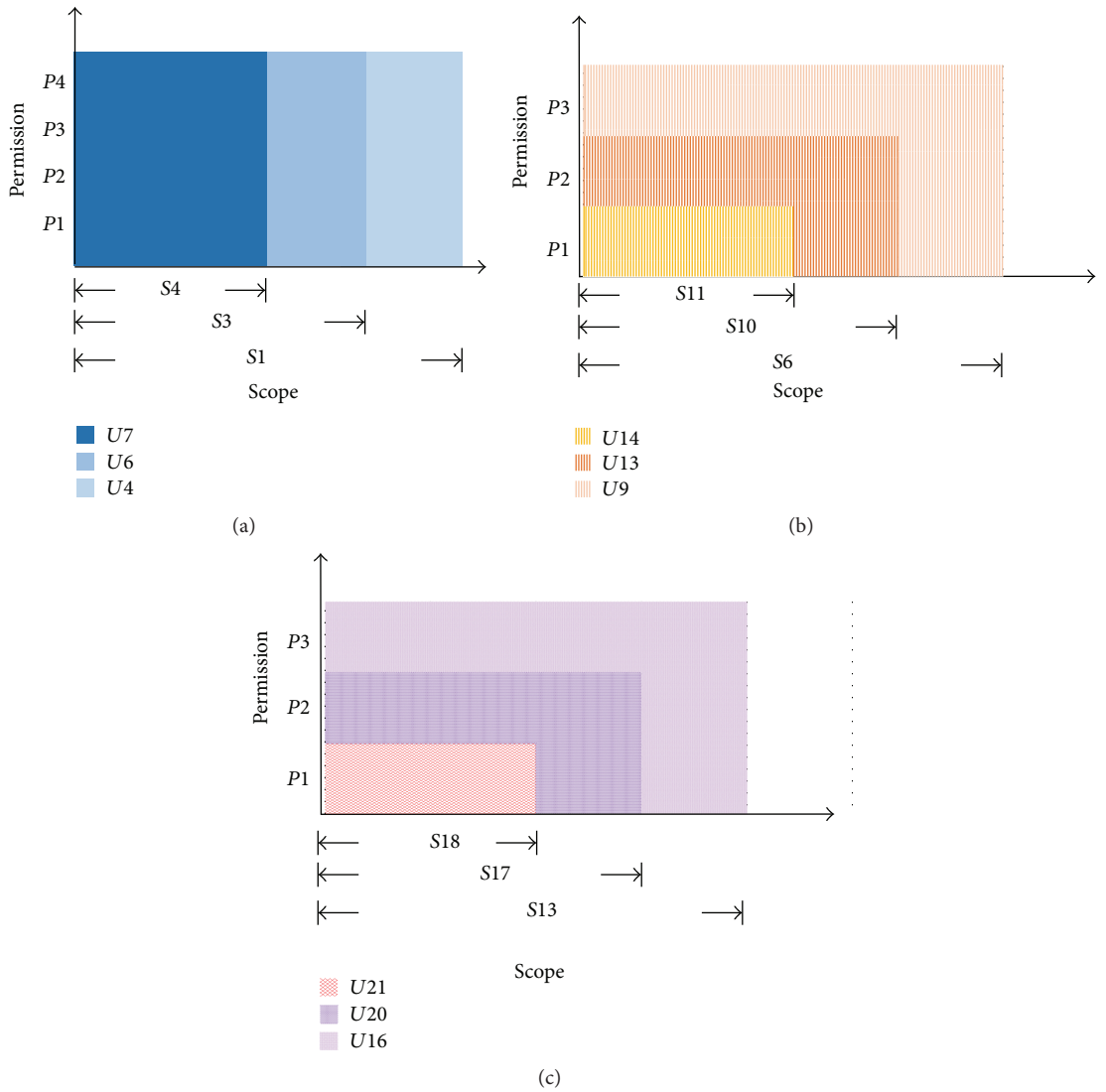


Figure 11: Hierarchy structure of the users. (a) Three users from the cloud platform. (b) Three users from the hotel tenant. (c) Three users from the restaurant tenant.

Figure 12: Fields of the two tenants.

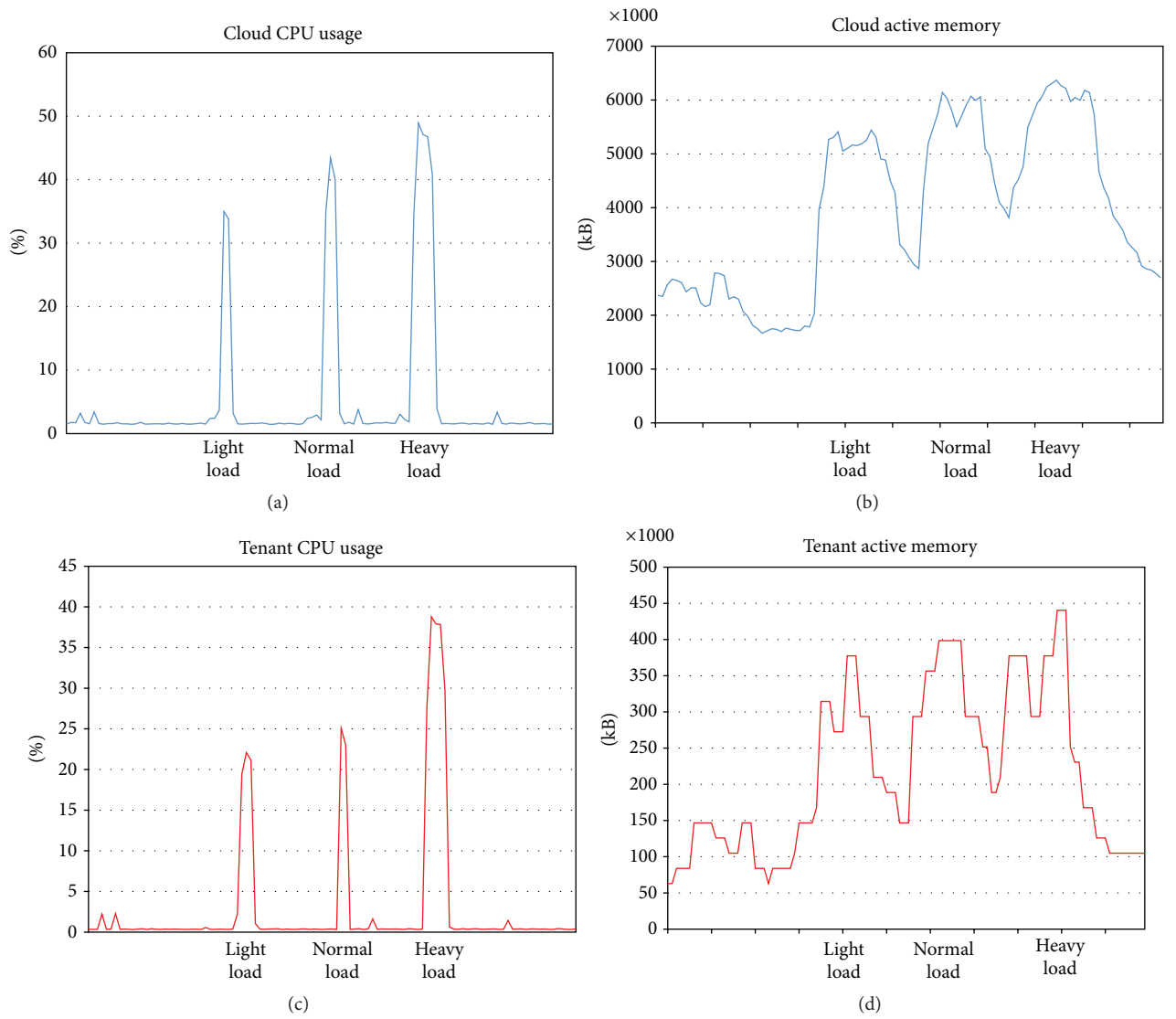


FIGURE 13: CPU usage and active memory in the simulation. (a) CPU usage of cloud platform. (b) Active memory of cloud platform. (c) CPU usage of tenant system. (d) Active memory of tenant system.

The experimental evaluation in Section 4.2 shows that the 4D-RBMT model is business-oriented model for multitenant users in cloud platform. It is suitable not only for different enterprises as tenants to configure and manage users' roles in cloud, but also for administrators of cloud providers to assign resources to the tenants and manage tenants' role in the cloud. According to the evaluation in Section 4.3, the proposed system performed well in the situations of large amount of users who are operating in the cloud simultaneously.

5. Conclusion

The paper presented a 4D-role, which is a set of four-dimensional vectors, forming as ⟨permission, scope, valid time, user category⟩. According to the four-dimensional role, we proposed a 4D-role based multitenancy (4D-RBMT) model using a level based structure. The 4D-RBMT model is able to make users work in their own scopes during their valid time by assigning different permissions, scopes, valid time, and user category to each role. It is proved in mathematic approaches that the 4D-RBMT model has characters of tenant isolation, role hierarchy, and administration independence. Such three characters of the model are also verified by experimental evaluation. In addition, the evaluation results indicate that the 4D-RBMT based prototype system performs well in using cloud resources, especially in the scenarios where large-scale users are operating in the cloud platform simultaneously. The 4D-RBMT model can be used effectively in developing a tenant management system for multitenancy cloud platforms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

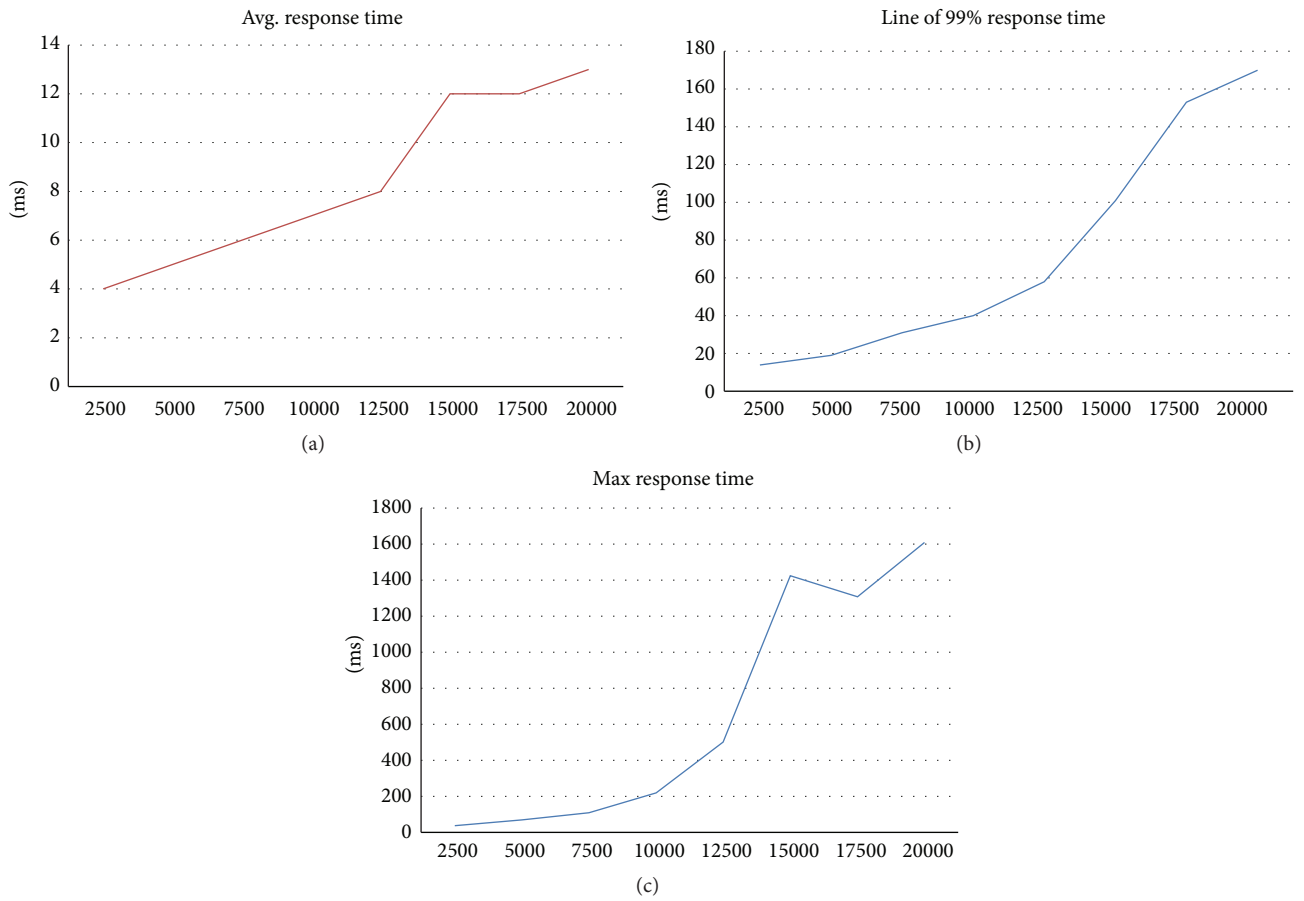


FIGURE 14: Response time in the simulation. (a) Average response time. (b) Line of 99% response time. (c) Max response time.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61202382) and the Youth Science and Technology Foundation of Shanghai, China (no. 15YF1412600) and partly supported by the Fundamental Research Funds for the Central Universities of China.

References

[1] N. W. Lo, T. C. Yang, and M. H. Guo, "An attribute-role based access control mechanism for multi-tenancy cloud environment," *Wireless Personal Communications*, vol. 84, no. 3, pp. 2119–2134, 2015.

[2] X. Feng, B. Ge, Y. Sun et al., "Enhancing role management in role-based access control," in *Proceedings of the 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT '10)*, pp. 677–683, Beijing, China, October 2010.

[3] S. Ranise, A. Truong, and A. Armando, "Scalable and precise automated analysis of administrative temporal role-based access control," in *Proceedings of the 19th ACM Symposium on Access Control Models and Technologies (SACMAT '14)*, pp. 103–114, ACM, June 2014.

[4] B. Lee, D. K. Kim, H. Yang, and H. Jang, "Role-based access control for substation automation systems using XACML," *Information Systems*, vol. 53, pp. 237–249, 2015.

[5] S. Raje, C. Davuluri, M. Freitas, R. Ramnath, and J. Ramanathan, "Using ontology-based methods for implementing role-based access control in cooperative systems," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12)*, pp. 763–764, Trento, Italy, March 2012.

[6] S. Kim, D.-K. Kim, L. Lu, S. Kim, and S. Park, "A feature-based approach for modeling role-based access control systems," *Journal of Systems and Software*, vol. 84, no. 12, pp. 2035–2052, 2011.

[7] M. E. Kabir, H. Wang, and E. Bertino, "A role-involved purpose-based access control model," *Information Systems Frontiers*, vol. 14, no. 3, pp. 809–822, 2012.

[8] A. Gupta, M. S. Kirkpatrick, and E. Bertino, "A formal proximity model for RBAC systems," *Computers & Security*, vol. 41, pp. 52–67, 2014.

[9] S. A. Mohammed and M. M. Yusof, "Towards an evaluation method for information quality management of health information systems," in *Proceedings of the 2nd International Conference on Information Management and Evaluation (ICIME '11)*, pp. 529–538, Toronto, Canada, April 2011.

[10] L. Liqing, Y. Rong, L. Hai, and L. Xudong, "Research of extended RBAC model on permission control in WEB information system," in *Proceedings of the IEEE 3rd International Conference on Communication Software and Networks (ICCSN '11)*, pp. 359–362, IEEE, Xi'an, China, May 2011.

[11] C.-X. Zhang, J.-F. Li, Y. Liu, and W.-D. Zhao, "Design and implementation of universal management system based on

roles and scopes," *Computer Engineering*, vol. 7, no. 34, pp. 47–49, 2008.

[12] J. Li and C. Zhang, "A three-dimensional role based user management model in web information systems," in *Proceedings of the 2012 International Conference on Information Technology and Software Engineering: Information Technology*, vol. 210 of *Lecture Notes in Electrical Engineering*, pp. 657–665, Springer, Berlin, Germany, 2013.

[13] X. H. Le, T. Doll, M. Barbosu, A. Luque, and D. Wang, "An enhancement of the role-based access control model to facilitate information access management in context of team collaboration and workflow," *Journal of Biomedical Informatics*, vol. 45, no. 6, pp. 1084–1107, 2012.

[14] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224–274, 2001.

[15] R. S. Sandhu, E. J. Coyne, H. L. Feinstein et al., "Role-based access control models," *Computer*, no. 2, pp. 38–47, 1996.

[16] T. C. Yang, N. W. Lo, and H. T. Liaw, "An enhancement RBAC mechanism for multi-tenancy cloud environment," in *Proceedings of the International Workshop on Advanced Information Technology and Applications*, 2012.