SaaS平台访问控制研究

朱养鹏[1,2],张 璟[1]
ZHU Yangpeng[1,2],ZHANG Jing[1]
1.西安理工大学 计算机学院,西安 710048
2.西安石油大学 经济管理学院,西安 710065
1.School of Computer,Xi'an University of Technology,Xi'an 710048,China
2.School of Economic & Management,Xi'an Shiyou University,Xi'an 710065,China

ZHU Yangpeng,ZHANG Jing.Research on access control of SaaS platform.Computer Engineering and Applications,2011,47(24):12-16.

Abstract: Software as a service is a new software deliver model which provides software services for customers and reduces hardware purchase and system management cost.As tenant's data being kept in service provider,how to protect the tenant data while keeping high resource use rate is a challenge problem.To deal with the problem about complex tenant role and meet the request of all tenants' data saving together and accessing separately,a supporting multi tenants,multi roles and easy access control SaaS model based on RBAC is proposed.Comparing with traditional RBAC,the conception of tenants is added in this model,which develops platform access control based on tenants and enhances the security and management for SaaS platform.This paper analyzes the process of user accessing to the SaaS platform and provides formal language description for this model,realizes a SaaS hotel management platform access control's database physical model and provides suggestion for SaaS platform development.

Key words: software as a service;multi tenant;access control

摘 要: Saas平台软件交付模式将应用软件以服务的形式提供给客户,可缩减硬件采购、系统管理上的开销。由于租户数据统一存储于服务提供商处,如何在维持较高资源利用率的同时保障租户的数据安全是一个挑战性问题。针对租户角色复杂、各租户数据共存而又独立访问的要求,结合基于角色的访问控制模型,构建了支持多租户、多角色、方便租户权限管理的SaaS平台的访问控制模型。和传统基于角色的访问控制模型相比,该模型增加了租户的概念,以租户为基本单元实施平台的访问控制,提高了SaaS平台访问控制的安全性和可管理性。分析了用户访问SaaS平台的具体流程,给出了模型的形式语言描述,实现了SaaS餐饮管理平台访问控制的数据库的物理模型,为SaaS平台开发提供参考。

关键词:软件及服务;多租户;权限控制
DOI:10.3778/j.issn.1002-8331.2011.24.004 **文章编号:**1002-8331(2011)24-0012-05 **文献标识码:**A **中图分类号:**TP309

1 引言

SaaS是指软件服务提供商为企业搭建信息化所需要的所有网络基础设施以及软硬件运作平台,并负责所有前期的实施,以及租户使用过程中软件的升级与维护等一系列的服务。租用的企业无需购买任何软硬件、建设相关机房以及招聘工作人员等,即可通过互联网使用信息管理系统,对企业日常事务进行有效的管理[1-2]。随着互联网技术的发展,SaaS正在全球兴起,特别是中小型企业,由于规模小,普遍存在着资金匮乏、管理水平低下、技术人才缺乏的问题,SaaS的多租户、按需付费的方式正好可以满足中小企业的IT管理需求。

SaaS模式的一个典型特征是"单实例多租户",即多个租户共享服务提供商的一个应用实例,不同租户的数据、服务在物理上共享,而在逻辑上完全隔离,对于每个租户来说这个实例好像只为自己服务一样。SaaS平台用户比较复杂,包括平台管理员,平台角色,租户管理员,租户用户,租户角色。平台管理员只能管理租户的账号和相关信息,不能操作租户的内部业务。各租户拥有自己的角色和权限,相互不能影响。为了确保系统中数据的安全性、一致性、完整性,使客户能够放心地将具有重要性、机密性的商业数据交给SaaS服务提供商进行管理和控制,在开发一个SaaS系统的过程中,作为SaaS系统的重要组成部分——权限管理变得尤为重要。由于增加了租户的概念,增加了SaaS平台安全访问管理的难度,本文在研究基

基金项目:国家高技术研究发展计划(863)(the National High-Tech Research and Development Plan of China under Grant No.2007AA010305863);陕西省科技计划项目(No.2006K04-G10);西安市科技局应用发展研究项目(No.YF07022)。
作者简介:朱养鹏(1975—),男,博士研究生,讲师,主要研究方向:服务计算、网络技术;张璟(1958—),男,教授,博士生导师。E-mail:zyp_hello@163.com
收稿日期:2011-01-13;**修回日期:**2011-05-31

于角色的访问控制（RBAC）模型的基础上，提出了SaaS平台的一种权限管理模型，并以餐饮管理系统为例实现了该模型。

2　访问控制研究现状

随着网络技术的发展，大型网络应用系统所面临的一个难题就是数据资源的安全问题，访问控制为解决信息系统安全性问题提供了重要保障，它能使经过授权的用户正常合法地使用已授权的功能，而将那些未授权的非法用户拒之门外[3]。访问控制是指限制主体可以访问哪些客体。在访问控制模型中，主体和客体是两个重要概念。主体是指可以授予或拒绝访问某个对象的人或事物，如用户、程序、系统进程。客体的例子如文件、打印机、程序、系统进程等。目前的主流访问控制技术有：自主访问控制（DAC）、强制访问控制（MAC）、基于角色的访问控制（RBAC）。

自主访问控制DAC（Discretionary Access Control）[4]是主体对自己的客体进行管理，由主体自己决定是否将自己的客体访问权或部分访问权授予其他主体。DAC的优点是其自主性为用户提供了极大的灵活性，但是DAC主要针对用户个人授予权限，随着用户的增多，用户授权将会变得非常复杂，不能适应SaaS平台访问控制的需要。

强制访问控制MAC（Mandatory Access Control）[5]是由管理员来设定主体和客体的安全级别来决定主体是否有对客体的访问权，而普通用户不能对自己的资源进行授权供其他用户使用。自主访问控制采用集权式管理用户权限，缺乏灵活性，不能适应SaaS平台访问控制的需要。

基于角色的访问控制RBAC（Role-Based Access Control）是由Ferraiolo等人提出的[6]。RBAC的基本思想是在用户和访问权限之间引入角色的概念，将用户和角色联系起来，通过对角色的授权来控制用户对系统资源的访问[7]。通过将权限指定给角色而不是用户在权限分派上提供了极大的灵活性和极细的权限指定粒度[8-9]。

国内学者对访问控制也做了大量研究，张学敏[10]等人提出了针对MIS系统进行动态权限管理的方法，解决了企业动态权限管理的问题；蔡昭权[11]提出了基于业务无关的权限管理方法，将权限管理和业务管理分开，以达到权限管理功能的复用；邓集波[12]等人提出基于任务的访问控制模型，能够很好解决工作流管理中随着任务的执行而动态进行权限分配的问题。由于SaaS平台中加入了租户的概念，平台的数据模型和管理方式和传统的管理系统不同，以上方法不能很好地解决SaaS平台下的用户访问控制问题。当前，针对SaaS平台访问控制研究较少，Danchen Li[13]基于RBAC模型采用分层的方法针对SaaS平台提出了一种访问控制方法，解决了SaaS访问角色命名冲突等问题，但该方法缺少实现方法的探讨。Jing Xu[14]也针对SaaS平台提出了一种访问控制方法，该方法禁用了角色的继承关系，并通过UML图展示用户访问SaaS平台的流程，但该方法中角色和租户是多对关系，如果不同租户对统一角色进行不同权限设置，则容易产生冲突，并且该模型未涉及用户对资源操作的控制。

在SaaS系统中，租户组成了一个个独立的安全域，SaaS平台管理员只可以对租户的账号进行管理，不能影响租户的内部业务操作，各租户由于内部业务分工不同，用户又划分为不同的角色。各租户内部的角色和用户只能访问该租户拥有的资源，不能超于租户的资源范围进行操作。本文在研究基于角色的访问控制（RBAC）模型的基础上，提出了SaaS平台的一种权限管理模型，解决了多租户环境下用户权限分配以及安全访问的问题。

3　SaaS访问控制模型

3.1　RBAC模型简介

RBAC96模型是典型的基于角色的访问控制模型。RBAC96模型由于系统全面地描述了RBAC多方面、多层次的意义而得到了广泛的认可。RBAC96模型包括RBAC0、RBAC1、RBAC2和RBAC3四个不同层次的模型。其中RBAC0是基础模型，定义了支持RBAC的最小需求，如用户、角色、权限、会话等概念，RBAC0如图1所示。在RBAC1中加入了角色继承关系，可以根据组织内部权力和责任的结构来构造角色与角色之间的层次关系；在RBAC2中加入了各种用户与角色之间、权限与角色之间以及角色与角色之间的约束关系，如角色互斥、角色最大成员数等。RBAC3是对RBAC1和RBAC2的集成，它不仅包括角色的层次关系，还包括约束关系。在RBAC模型中，角色是系统根据管理中相对稳定的职权和责任来划分，每种角色可以完成一定的职能。用户通过饰演不同的角色获得角色所拥有的权限，一旦某个用户成为某角色的成员，则此用户可以完成该角色所具有的职能。通过将权限指定给角色而不是用户在权限分派上提供了极大的灵活性和极细的权限指定粒度。本文基于RBAC模型提出了SaaS平台访问控制模型SRBAC。

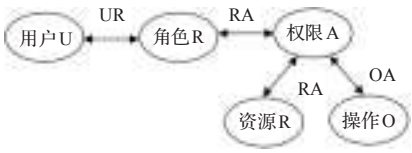

图1　RBAC0模型图

3.2　SaaS平台用户分析

由图2可以看出，在SaaS平台中，各租户可以通过Internet独立访问SaaS平台，各租户之间相互没有影响，就像使用一套独立的软件系统。各租户又由多个用户来共同使用SaaS平台，这些用户由于分工不同又可以划分为不同的角色。SaaS平台由平台供应商统一管理包括用户租费的管理、带宽的分配、数据的备份等。SaaS平台用户可分为以下几种。

平台管理员：负责平台的日常维护和管理，包括用户日志的管理、租户账号审核、租户状态管理、租户费用的管理，租户权限的管理，要注意的是平台管理员不能对租户的具体业务进行管理。如果租户数量大，还可以对平台管理员划分角色，可以按地域划分，比如西北地区、东北地区等，让平台管理员分别管理不同的租户；也可以根据业务进行划分，比如租户管理员，租费管理员等。

租户：指访问SaaS平台的用户企业，在SaaS平台中各租户之间信息是独立的。租户信息包括租户的名称、地址等租户企业的相关信息，主要用来区别各租户，并且由平台管理员

对租户账号状态进行管理。各租户可根据需要自行选择SaaS平台功能模块并依此付费。

租户用户:根据租户管理员分配的权限以及自己的角色进行相关的业务管理。各租户用户只能访问该租户选择的SaaS平台的功能模块。

租户角色:根据业务功能分由租户管理员进行角色划分,划分好角色后,租户管理员可以对相应的角色进行权限分配。

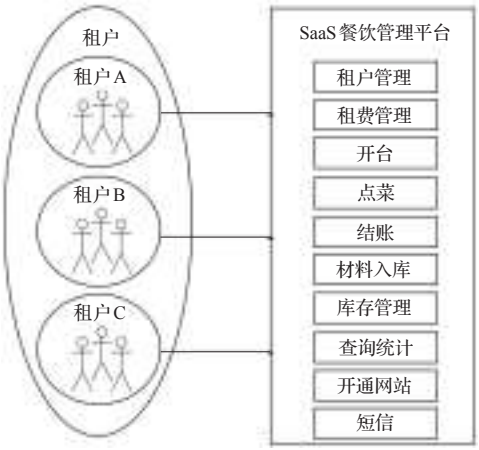

图2 Saas餐饮管理平台功能结构图

3.3 形式化描述

基于角色的访问控制是一种非常有潜力的访问控制技术,其基本观念是:分配给每一个用户合适的角色,每一个角色都具有其对应的权限,角色是安全控制策略的核心[6-7],这极大地简化了安全管理,特别适用于大规模的企业应用。由于SaaS平台引入租户的概念,各租户之间不能互相访问资源,数据保持独立,各租户用户只能在自己租户的范围内使用系统资源,因此需要对基于角色的访问控制进行改造以适应SaaS平台的权限管理的需要。

和RBAC模型相比,SaaS平台权限管理包括六个基本元素集合。

(1)租户:SaaS平台的使用企业,各租户用户只能在租户许可的范围内使用系统,记作$Tenants=\{t_1,t_2,\cdots,t_n\}$,表示所有租户的集合。

(2)用户:可以独立访问系统中的数据的主体,记作$Users=\{u_1,u_2,\cdots,u_n\}$,表示所有用户的集合,在SaaS平台中用户包括SaaS平台管理员和租户用户。

(3)角色:指一个组织或任务中的工作或岗位,记作$Rolses=\{r_1,r_2,\cdots,r_n\}$,表示所有角色的集合,用户拥有自己所属的角色的权限的并集,在SaaS平台中角色包括平台管理类角色和租户自定义角色。

(4)资源:所有需要设置权限的窗口的通称,例如一个页面、某个窗口、某一部分数据,都是一种资源,记作$Resources=\{res_1,res_2,\cdots,res_n\}$,在SaaS平台中表示所有页面的集合。

(5)操作:对资源的操作,比如删除、新增、修改、打印等,记作$Operations=\{Op_1,Op_2,\cdots,Op_n\}$,表示所有操作的集合。

(6)访问权限:表示允许对资源进行的各项操作,记作$Auths=\{a_1,a_2,\cdots,a_n\}$,表示所有访问权限的集合。

SaaS平台基于角色的访问控制方法的显著的两大特征是:

(1)在每个租户内部,由于角色/权限之间的变化比角色/用户关系之间的变化相对要慢得多,减小了授权管理的复杂性,降低管理开销。

(2)灵活地支持企业的安全策略,并对企业的变化有很大的伸缩性。

SaaS平台基于角色的访问控制授权模型如图3所示。在该模型中,用户U和角色R,角色R和权限A,资源和操作之间都是多对多关系,即:同一用户可以有多种角色,同一角色可以赋予多个用户;同一角色可对多个资源有访问权限,同一资源可赋权给多个角色。角色是系统根据管理中相对稳定的职权和责任来划分,每种角色可以完成一定的职能。用户通过饰演不同的角色获得角色所拥有的权限,一旦某个用户成为某角色的成员,则此用户可以完成该角色所具有的职能。通过将权限指定给角色而不是用户在权限分派上提供了极大的灵活性和极细的权限指定粒度。

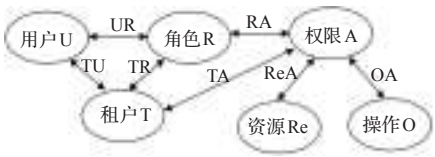

图3 SaaS平台访问控制模型图

下面给出适合于SaaS平台的访问控制模型的形式化描述:

租户权限指派:租户集*Tenants*和权限集*Auths*之间的一个二元关系,$TA \in Tenants \times Auths$。租户和权限之间是多对多关系,即一个租户可以拥有多个权限,一个权限也可以分配给多个租户。在SaaS平台中,租户可以根据自己的业务选择权限并依此付费。

租户用户创建:租户集*Tenants*和用户集*Users*之间的一个二元关系,$TU \in Tenants \times Users$。租户和用户之间是一对多关系,一个租户可以有多个用户,一个用户只能属于某一个租户。用户只能访问所属租户下的相应权限,不能超越跨租户访问系统。

租户角色创建:租户集*Tenants*和角色集*Roles*之间的一个二元关系,$TR \in Tenants \times Roles$。在传统编程模式下,用户和角色之间是多对多关系,即一个用户可以拥有多种角色,一种角色也可以属于多个用户。在SaaS模式下,为保证各租户自定义的角色不发生冲突,租户和角色之间是一对多关系,即一个租户可以设置多个角色,一个角色只能属于一个租户,在SaaS平台中允许每个租户自己定义角色以方便权限管理,为避免各租户角色定义混乱,租户角色只在所属租户范围内有效。

角色权限配置:角色集*Roles*和访问权限集*Auths*之间的二元关系,$RA \in Roles \times Auths \in TA$。角色和权限之间是多对多关系,一个角色可以有多种权限,每种权限可以属于多个角色。在SaaS平台中角色只能拥有所属租户权限内的相关权限。

用户角色分配:角色集*Roles*和用户集*Users*之间的二元关系,$UR \in Users \times Roles$。在SaaS平台中,用户和角色之间是多对多关系,用户只能拥有所属租户的权限,该租户的权限也只能分配给此租户所有的用户。

资源操作配置:资源集合*Resources*和操作集合*Operations*之间的二元关系,$ReO \in Resources \times Operations$,那么$(re,o) \in ReO$表示资源$r$具有$o$操作。

返回指定角色的用户集:$return\_users(r:Roles)=\{u \in Users|$

$(u,r) \in UR\}$。

返回指定角色的权限集：$return\_auths(r:Roles)=\{a \in Auths|(r,a) \in RA\}$。

所有权限的集合：$Auths=2^{(Resources \times Operations)}$。

如果直接对用户授权，则用户的权限集合为：$UA=2^{Users \times Auths}$。

采用角色授权，则角色权限的集合为：$RA=2^{Roles \times Auths}$，由于在一个企业或租户内部，用户数量远远大于角色数量，因此$UA=2^{Users \times Auths}>RA=2^{Roles \times Auths}$，因此对角色授权工作量远远小于对用户直接授权。并且，对企业或租户而言，角色相对比较固定，如果发生用户变更，只需调整用户和角色的关系，不牵扯到角色权限的变更，降低了权限管理的复杂性，并对企业变化有很大的伸缩性。

利用此模式，即使租户的权限体系变化导致用户、角色和权限及其相互关联的变化，系统也可以很简单地通过配置解决这个问题，并且很好地解决了平台管理员、租户、租户用户之间复杂的访问控制问题。

4　应用实例

本文所设计的SaaS平台权限管理模型已成功应用于SaaS餐饮管理平台(http://www.xa001.net)。该SaaS平台支持多租户的访问，并可以对用户进行灵活的权限管理。该餐饮管理平台主要包括餐桌使用情况、客户开台、点菜、结账、餐厅菜品的入库、库存管理、查询统计等基本功能和宣传网站、网上订餐、短信提醒等可选功能。由于中小型餐饮企业规模小，对信息化投入低，因此基于SaaS的餐饮管理系统既可以满足中小型餐饮企业日常管理的需求，又可以根据用户的需求支付较低的使用费即可。

4.1　操作流程

SaaS餐饮管理平台用户可分为以下几种：

平台管理员：负责所有餐饮企业账户、权限、租费管理。

餐饮企业管理员：负责本餐饮企业角色划分、权限分配、日志管理、数据备份等工作。

前台管理员：开台、点菜、结账等业务。

库房管理员：菜品入库、库存管理、菜品补库、盘库等工作。

大堂经理：营业情况查看。

餐馆老板：每日、周、月、季、年的经营情况、费用情况、盈利情况查看。下面来看一下SaaS餐饮管理平台访问控制的两个重要流程。

(1)注册流程

在SaaS餐饮管理平台中，租户注册流程如图4所示。首先用户登记租户信息，包括企业名称、地址、联系电话、所在省市等企业基本信息，接下来填写管理员账号(以后就可以用这个管理员账号创建租户内部用户以及分配权限)，平台管理员审核通过后租户就可以通过创建的租户管理员账号登录平台进行功能试用，如果租户试用满意就可以选择自己需要的功能进行付费，管理员收到用户付费信息后就可以将用户转为正式用户。此时租户用户可以根据自己企业需求来管理用户，为租户用户分配权限。

(2)登录流程

用户登录流程如图5所示。用户在SaaS平台输入账号和密码，如果用户通过平台验证首先获取该用户所属租户信息，然后获取该用户所属角色信息，最后通过该用户角色获取其对应的权限，这时用户就可以正常使用平台功能了。

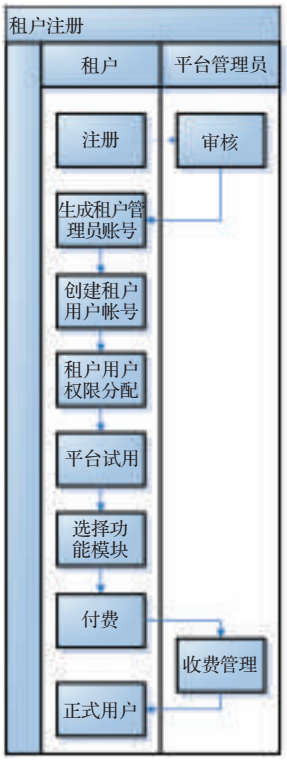

图4　注册流程图

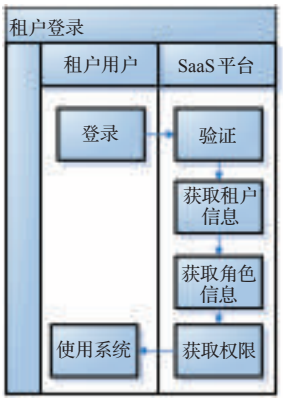

图5　登录流程图

4.2　访问控制数据库设计

根据上面的用户和操作流程分析，设计基于SaaS的餐饮管理平台访问控制部分的数据库。访问控制部分主要包括用户表、租户表、角色表、权限表、资源表和操作表。对这些表的定义如下：

(1)用户表(User)，存储SaaS平台用户信息。需要注意的是用户分为租户用户和平台管理员，租户用户必须隶属于某一租户，只能访问该租户拥有的资源

表1　用户表(User)

字段名	数据类型	中文名	说明
UserID	int	用户编号	自动编号
UserName	varchar(250)	用户名称	
UserPW	varchar(20)	用户密码	
tenanted	int	租户编号	

(2)租户表(Tenant)，用来保存租户相关信息。租户编号用来唯一标记一个租户企业。

表2　资源表(Resource)

字段名	数据类型	中文名	说明
tenanted	int	租户编号	自动编号
tenantname	varchar(50)	租户名称	
description	varchar(100)	描述	

(3)角色表(Role)，存储系统所有角色信息，角色可以由用户自由划分。需要注意的是在本模型中，租户和角色是一对多关系，因此租户用户可以自由划分角色，不会和其他租户角色混淆。角色表信息如表3。

表3 角色表(Role)

字段名	数据类型	中文名	说明
RoleID	int	角色编号	自动编号
RoleName	varchar(50)	角色名称	
Discription	varchar(2100)	角色说明	
Tenanted	int	租户编号	

(4)权限表(Auth),存储每个资源的操作。比如资源用户管理的编号(ResourceID)为1,操作对应为1添加,2修改,3删除,4打印。如果用户管理需要有添加、修改、删除操作,那么权限表(Auth)就应该存储1,1,1;2,1,2;3,1,3三组数据。

表4 权限表(Auth)

字段名	数据类型	中文名	说明
AuthID	int	资源编号	自动编号
ResID	varchar(50)	资源名称	
OpeID	varchar(100)	资源路径	

(5)资源表(Resource),用来保存系统所有资源,对于本系统而言,就是指平台页面。

表5 资源表(Resource)

字段名	数据类型	中文名	说明
ResID	int	资源编号	自动编号
ResName	varchar(50)	资源名称	
ResURL	varchar(100)	资源路径	

(6)操作表(Operation),存储平台页面上所有需要有权限控制的操作,比如增加、修改、删除、打印等。

表6 操作表(Operation)

字段名	数据类型	中文名	说明
OpeID	int	操作编号	自动编号
OpeName	varchar(50)	操作名称	
Description	varchar(100)	操作说明	

除了以上基本表,访问控制部分还要涉及到表和表之间的关系表,如用户租户表、租户权限表、用户角色表、租户角色表、角色权限表等。

5 总结

分析了SaaS平台访问控制的特点,结合基于角色的访问控制模型,设计了一种适合SaaS平台的访问控制模型,并以该模型为基础实现了基于SaaS平台的餐饮管理系统(http://www.xa001.net)。该模型可以灵活地处理租户和租户、租户和用户、用户和角色等之间的关系,使用该模型,可以方便进行用户的权限管理并保证了用户的数据安全。在SaaS平台安全设计时除了权限控制,还要考虑用户数据的隐私性保护[15]、数据传输层的安全性、数据库的冗余备份、入侵检测[16]等诸多安全因素,这也是后续工作的研究重点。

参考文献:

[1] 叶伟.互联网时代的软件革命—SaaS架构设计[M].北京:电子工业出版社,2009:12-13.

[2] 金珊,吴国芳.基于SaaS模式的SOA服务分析与设计[J].信息化建设,2009:115-118.

[3] Ferraiolo D F,Barkley J F,Kubn D R.A role based access control model and reference implementation within a corporate intranet[J].ACM Transactions on Information Systems Security,1999,2(1):34-64.

[4] Sandhu,R,Samarati P.Access control:principles and practice[J].IEEE Commun,1994,32(9):40-48.

[5] Nyanchama M,Osborn S.Modeling mandatory access control in role-based security systems[C]//Proceedings of the 9th Annual IFIP TC11 WG11.3 Working Conference on Database Security IX:Status and Prospects,Rensselaerville,New York,United States,1996:129-144.

[6] Ferraiolo D F,Kuhn D R.Role based access control[C]//15th National Computer Security Conference,1992:554-563.

[7] Sandhu R S,Coyne E J.Role-based access control models[J].Computer,1996,29(2):38-47.

[8] Ahn G J,Sandhu R.Role-based authorization constraints specification[J].ACM Transactions on Information and System Security,2000,3(4):207-226.

[9] Wainer J,Kumar A.A fine-grained,controllable user-to-user delegation method in RBAC[C]//Proc of the 10th ACM Symp on Access Control Models and Technologies.New York:ACM Press,2005:59-66.

[10] 张学敏,熊曾刚,陈建新,等.基于MIS系统的用户动态权限管理[J].计算机工程,2005,33(6):231-233.

[11] 蔡昭权.基于业务无关的权限管理的设计与实现[J].计算机工程,2008,34(9):183-185.

[12] 邓集波,洪帆.基于任务的访问控制模型[J].软件学报,2003,14(1):76-82.

[13] Li Danchen,Liu Cheng,Wei Qiang.RBAC-based access control for SaaS systems[C]//2nd International Conference on Information Engineering and Computer Science,Wuhan,China,2010:1-4.

[14] Jing Xu,Tang Jinglei,He Dongjian.Research and implementation on access control of management-type SaaS[C]//2nd International Conference on Information Engineering and Computer Science,Wuhan,China,2010:388-392.

[15] 张坤,李庆忠,史玉.面向SaaS应用的数据组合隐私保护机制研[J].计算机学报,2010,33(11):2045-2056.

[16] 赵玉霞.基于SaaS模式下的系统数据安全策略研究[J].软件导刊,2010,9(1):143-144.