管理信息系统中基于角色的访问控制与应用

王必友

(南京师范大学数学与计算机科学学院, 210097, 南京)

[摘要]　介绍了基于角色的安全访问控制(RBAC)基本模型, 并根据管理信息系统特点, 对模型进行了改进, 为操作许可 P 定义一关于环境参数的布尔函数 f_p, 增强了 RBAC 基本模型的控制粒度. 同时, 引入先决操作许可概念, 有效地防止由于操作许可设置不当, 而导致的已授权操作许可无法执行的现象. 并给出了应用实例.

[关键词]　权限控制, RBAC, 角色

[中图分类号] TP393.07,　[文献标识码] B,　[文章编号] 1672-1292-(2003)04-0041-04

0　引言

随着信息系统向多用户网络化方向发展, 信息系统的安全访问控制日益受到重视. 基于角色访问控制(RBAC)是近些年在信息访问控制领域的研究热点, 它在满足信息系统安全需求方面显示了优势, 有效地克服了传统访问控制技术——自主访问控制(DAC)及强制访问控制(MAC)存在的不足, 可以减少授权管理的复杂性, 降低管理难度, 为系统管理员提供一个比较友好的安全策略管理环境.

但是在实际使用 RBAC 模型进行大型信息系统访问控制系统的设计中, 还存在以下问题:

(1) 目前广泛使用的 RBAC 模型(NIST RBAC 和 RBAC96)是完全基于角色的系统, 用户获得对信息资源的操作授权完全依赖于对一个或多个角色拥有授权的继承. 实际上, 用户对某一信息资源的操作有时还要受到环境因素的影响. 如删除记录操作, 删除可能是正常的删除, 也可能是操作员误录入而需要的删除. 在信息系统中, 删除操作往往是要严格限制的, 通常需要更高级别的用户才能进行. 因此普通操作员不能拥有删除的权限直接删除自己误录入的数据, 这给系统的使用带来了不便. 如何既要符合授权管理中的最小特权原则, 又要方便用户使用, 这是在设计信息系统时值得考虑的问题.

(2) 在可视化面向对象的应用程序系统中, 系统的功能往往集成在一个图形界面中, 它们之间存在某种联系. 如果角色的操作许可授权不恰当, 将导致已授权的操作无法进行. 如在一个客人资料查询列表中, 集成了增加、删除记录功能. 根据 RBAC 模型查询、增加和删除功能是彼此独立的操作, 但是若只给一角色赋予增加功能而不赋予查询功能, 显然将无法进行删除操作.

针对这些问题, 笔者在开发一企业信息系统时, 对 RBAC 模型提出了改进意见, 并对系统的访问控制方案进行了设计和实现.

1　RBAC 访问控制模型

90 年代以来, RBAC 模型得到了广泛关注和深入研究. 在 Sandhu 等提出一系列 RBAC 参考模型的基础上, 美国国家标准与技术局(NIST)研究小组定义了 RBAC 模型的标准.

1.1　基本定义

RBAC 基本模型如图 1 所示. 模型中有 User、Role、Permission 和 Session 4 个组成部分, 以下是模型中用到的基本定义.

收稿日期: 2003-11-04.

作者简介: 王必友, 1962-, 南京师范大学数学与计算机科学学院讲师, 主要从事软件工程、软件方法和数据库应用的研究.

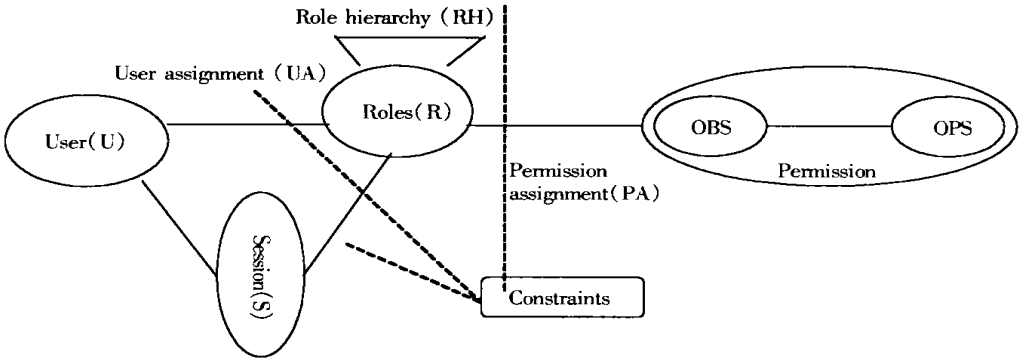

图1　RBAC基本模型

(1) U: user (用户), 信息系统的使用者, 主要是指人, 也可以是机器人、计算机或网络.

(2) R: role (角色), 对应于企业组织结构中一定的职能岗位, 代表特定的权限, 即用户在特定语境中的状态和行为的抽象, 反映用户的职责.

(3) P: permission (许可), PERMISSIONS= $2^{(OPS\times OBS)}$, 表示操作许可的集合, 即用户对信息系统中的对象(OBS)进行某种特定模式访问(OPS)的操作许可. 操作许可的类别取决于其所在的应用系统. 在文件系统中, 许可包括读、写和运行, 而在数据库管理系统中, 许可包括插入、删除、添加和更新.

(4) S: session (会话), 会话是一个动态概念, 用户激活角色集时建立会话. 会话是一个用户和多个角色的映射, 一个用户可以同时打开多个会话. 如在网络环境下, 一个用户可在多个工作站登录, 这样一个用户就建立了多个会话.

1.2　RBAC基本模型

RBAC的基本思想是根据组织结构中不同职能岗位划分角色, 操作许可映射到角色上, 用户被分配给角色, 并通过会话激活角色集, 能够间接访问信息资源. 用户与角色以及操作许可与角色是多对多的关系, 因此一个用户可以分配多个角色, 一个角色可以拥有多个用户. 同理, 一个操作许可可以分配多个角色, 一个角色可以赋予多个操作许可. 角色可以划分等级, 即角色的层次化, 反映企业组织的结构和人员责权的分配, 并且角色通过继承形成偏序关系.

RBAC模型描述如下:

$PA \subseteq P \times R$, 是授权到角色的多对多的关系;

$UA \subseteq U \times R$, 是用户到角色的多对多的关系;

User: $S \rightarrow U$, 将各个会话映射到一个用户的函数 user(S_i);

Roles: $S \rightarrow 2^R$, 将各个会话 S_i 与一个角色集合联接起来的映射, 随时间变化可以变化. 一个会话 S_i 拥有的角色 $roles(S_i) \in \{r \mid (user(S_i), r) \in UA\}$.

因此, 一个用户的一个会话 S_i 通过直接激活角色获得的实际操作许可集为:

$$\bigcup_{r \in roles(S_i)} \{p \mid (p, r) \in PA\}$$

$RH \subseteq R \times R$, 是角色上的一个偏序关系, 称之为角色层次关系或者支配关系, 一个会话 S_i 所继承的子角色 $roles(S_i) \subseteq \{r \mid (\exists r' \geqslant r)[(user(S_i), r') \in UA]\}$ (其中 $r' \geqslant r$ 表示角色 r' 继承角色 r).

因此, 一个用户的一个会话 S_i 通过角色继承获得的实际操作许可集为:

$$\bigcup_{r \in roles(S_i)} \{p \mid (\exists r'' \leqslant r) \wedge (p, r'') \in PA\}$$

另外, RBAC模型还提供角色静态互斥、角色动态互斥、角色容量等约束概念.

2　RBAC模型改进

在基于角色访问控制RBAC模型中, 用户与操作许可是通过角色来建立联系的, 能够表达企业组织

结构中的“职– 责”关系, 而且能够降低存取控制管理的复杂度. 但是, 由于 RBAC 模型的高度抽象, 割裂了用户与操作对象之间的某些联系, 导致引言(1)中的现象, 本质上是 RBAC 模型无法实现与环境有关的细粒度的访问控制.

另外, 引言(2)中的问题, 实际上是操作许可之间的相关性, 即某用户拥有操作许可 p_j, 则必须拥有 p_k, 否则是奇异的, 称操作许可 p_k 先决于操作许可 p_j. 实际上, 它是一种关于操作许可的约束.

在不影响 RBAC 模型整体框架(保持 RBAC 模型原有优点)情况下, 为了解决这些问题, 本文给出如下解决方案:

(1) 针对操作许可 P 定义一布尔函数 $fp(par_1, par_2, \cdots, par_n)$. 其中, 环境参数 $par_1, par_2, \cdots, par_n$ ($n>=0$) 可以是操作对象(OBS)的某些属性, 如创建者、创建日期等; 也可是系统的参量, 如系统时间, 用户等. 当 fp= True 时, 操作许可 P 处于激活状态即有效, 反之, 则处于休眠状态即无效.

因此, 一个用户的一个会话 S_i 通过直接激活角色获得的实际操作许可集为:

$$\bigcup_{r\in roles(S_i)}\{p \mid (p,r)\in PA \wedge fp(par_1, par_2, \cdots, par_n)=true\}$$

通过角色继承获得的实际操作许可集为:

$$\bigcup_{r\in roles(S_i)}\{p \mid (\exists r'' \leqslant r) \wedge (p,r'')\in PA \wedge fp(par_1, par_2, \cdots, par_n)=true\}.$$

值得特别强调的是, 只有在用户对操作对象施加操作时, 才能激活对应的函数 fp 的执行, 以决定操作许可是处于激活还是处于休眠状态.

例如, 在一个信息系统中, 若允许操作员 u_j 在 Δt 时间内可以删除自己误录入的数据(由用户 u_x 在 t_x 时刻创建)的 fp 函数可表达为:

$u_x = u_j \wedge (t_c - t_x) <= \Delta t$(其中 t_c 为系统当前时间).

若令 fp 为常数 true, 则新模型与原 RBAC 模型等同. 实际上, 在一个信息系统中往往有相当数量操作许可 P 的 fp 值可定义为常数 true, 如系统功能菜单等. 这样可减少引入 fp 函数带来的系统复杂性.

(2) 增加操作许可先决操作许可二元关系 $PR \subseteq P \times P$, 显然它是一个偏序关系. 即若 $(p_j, p_k)\in PR$, 则 $p_j \in user(S_i) \rightarrow p_k \in user(S_i)$. 在 RBAC 模型中用户与操作许可是通过角色来建立联系的, 为了不违反这一原则, 可规定只有 p_k 已属于角色 r 时, p_j 才可赋予角色 r.

例如, 在一个客人资料查询列表中, 集成了增加、删除记录功能, 对客人资料的操作许可有查询、增加和删除, 显然有:

[(增加, 查询), (删除, 查询)] $\in$ PR

若给一角色赋予增加、删除操作许可时, 发现其没有查询许可, 则禁止授权. 避免了由于角色的操作许可授权不直观导致设置不当, 从而引起已授权的操作无法进行的现象.

3 应用实例

笔者在开发一企业信息系统时, 进行了以上改进 RBAC 模型访问控制方案的设计和实现.

首先, 根据系统所要实现的不同功能, 将整个系统划分为若干功能相对独立的模块, 并将系统所有操作许可(即权限)在每一模块中进一步划分, 这样所有权限都隶属于某一功能模块.

第二, 对每一操作许可, 确定其 fp 函数. 实际上, 绝大多数操作许可的 fp 函数应该定义为常数 true, 只有 RBAC 模型的访问控制粒度解决不了的操作许可需定义 fp 函数. fp 函数参数若涉及到操作对象的某些属性, 则在建立操作对象时应当包含这些属性, 如创建者、创建日期等. 另外, 在功能模块中找出所有操作许可的互斥关系及先决关系(不同功能模块中, 这种关系比较少见, 即使有也可通过角色约束来实现).

第三,根据企业工作岗位设置角色,确定每个角色容量(即可以支持的最大用户数量)以及互斥角色.角色可以是低级角色,也可以是高级角色.角色可以直接授予操作许可,对于高级角色还可授予其它低级角色和高级角色,以便继承其它角色的功能权限.

第四,将系统所有用户分为3级,即系统管理员用户、功能模块管理员用户和普通用户.系统管理员用户是一个超级用户,可以管理系统中所有用户和角色,对用户和角色进行授权,设置各种约束条件.功能模块管理员用户只能在其功能模块内进行授权,普通用户没有任何系统管理权限,只是系统的使用者.

主要数据表包括:功能模块定义表(模块编号,模块名称,模块管理员编号);操作许可表(操作许可号,所属模块编号,fp函数);角色表(角色编号,角色名称,所属功能模块编号,角色容量,角色级别);用户表(用户编号,用户名称,所属部门编号,用户级别);用户—角色对应表(角色编号,用户编号);操作许可—角色对应表(操作许可号,角色编号);角色层次表(高级角色编号,子角色编号);角色互斥约束表(角色1编号,角色2编号);操作许可互斥表(操作许可1号,操作许可2号);操作许可先决表(操作许可1号,操作许可2号).

系统采用主动权限控制和被动权限控制两种方式相结合的方法实现权限控制.对于fp函数定义为常数true的操作许可,一般采用主动权限控制,即用户进入系统后,直接隐藏没有操作许可的功能.而对于一般fp函数定义的操作许可只能采用被动权限控制,即当用户选择某功能执行时,引发fp函数的执行,若结果为true操作允许,否则,提示用户无权操作.

4 结论

文中介绍了基于角色的访问控制RBAC模型,并提出了在不影响其整体结构的情况下修改RBAC模型的方案,既保证了原有RBAC模型访问控制管理的优点,又能满足实际信息系统中访问控制系统的需要,具有一定的实用价值.

[参考文献]

[1] Ravi Sandhu, Edward J Coyne. Role-Based Access Control Models[J]. Computer, 1996,(2):38~47.

[2] 曹天杰,张永平. 管理信息系统中基于角色的访问控制[J]. 计算机应用,2001,21(8):21~23.

Role Based Access Control Model and Application in MIS

Wang Biyou

(College of Mathematics and Computer Science, Nanjing Normal University, 210097, Nanjing, PRC)

Abstract: This paper introduces Role-Based Access Control(RBAC) model, and presents an improved model based on the characteristic of MIS to define an environment function fp to enhance access control granularity of RBAC model. In addition the concept of prerequisite permission is introduced for preventing improper authorization causing non-excution of authorized permission. Finally, an application of this improved model is given.

Key words: Access Control, RBAC, role

[责任编辑:刘健]