

在RBAC模型中“责权分离约束”的冲突检测与消解

Conflict Checking and Solving of Separation of Duty Constraints in RBAC

崔中杰/CUI Zhong-jie^{1,2}, 胡昌振/HU Chang-zhen^{1,2}

1. 北京理工大学信息安全与对抗技术研究中心, 北京 100081

2. 北京理工大学机电工程学院, 北京 100081

1. Information Security and Confrontation Technology Research Center, Beijing Institute of Technology, Beijing 100081, China

2. School of Mechatronic Engineering, Beijing Institute of Technology, Beijing 100081, China

[摘要] 详细论述了责权分离约束在基于角色的访问控制(RBAC)中的冲突检测与解决方案。研究了该约束在“权限-角色授权”(PRA)、“权限-主体授权”(PSA)、“角色-主体授权”(RSA)、“角色-角色授权”(RRA)等各类授权关系中的典型示例,并结合数学中的有向图理论给出冲突检测的算法分析,为实际应用奠定了基础。此外,还对冲突产生后的消解方法进行了深入讨论,总结出多种方案并进行仿真比较,根据实验结果给出一套优化后的解决途径。

[关键词] 责权分离,冲突检测,冲突消解,有向无环图,角色分层

[中图分类号] TP301

[文献标识码] A

[文章编号] 1000-7857(2005)12-0040-04

Abstract: The conflict checking and solving of separation of duty constraints were particularly discussed in RBAC we researched these constraints' typical examples in authorization relations of the permission-to-role assignment (PRA), the permission-to subject assignment (PSA), the role-to-subject assignment (RSA), and the definition of a role hierarchy (role-to-role assignment, RRA). The arithmetic analysis of conflict checking which links with the theory of directed acycline graph (DAG) was provided. It will be a basis for practical application. Furthermore, the conflict solving was further discussed in the paper. Several schemes are summarized and achieved imitating experiment. Subsequently, we provided an optimized solving method for the further work.

Key Words: separation of duty, conflict checking, conflict solving, directed acycline graph, role hierarchy

CLC Number: TP301

Document Code: A

Article ID: 1000-7857(2005)12-0040-04

收稿日期:2005-10-20

作者简介:崔中杰,男,北京中关村南大街5号北京理工大学,博士生,研究方向为信息安全;E-mail:bjcuizj@163.com

胡昌振(通讯作者),男,北京理工大学机电工程学院,教授,研究方向为信息安全;E-mail: chzhoo@bit.edu.cn

4 结论

1) 在光照培养条件下,与尿素和氯化铵相比,硝酸钾是支持小球藻持续快速生长的最好氮源。

2) 在光照培养条件下,与尿素和氯化铵相比,硝酸钾是促进小球藻生物合成叶黄素的最好氮源,小球藻细胞中的叶黄素含量可以达到0.85 mg/g。

3) 植物激素 IAA 和 IBA 不但不能明显促进小球藻的生长,反而明显抑制了小球藻对叶黄素的生物合成。

(致谢:感谢北京科技大学校基金和北京科技大学方兴科技孵化器项目的资助。)

参考文献(References)

- [1] CHEN F. High cell density culture of microalgae in heterotrophic growth [J]. Tibtech, 1996, 14: 421-426.
- [2] KYLED. Production and use of lipids from microalgae [J]. Lipid Technol, 1992, 4(3): 59-64.
- [3] BOROWITZKAME. Microalgae source of pharmaceuticals and other biologically active compounds[J]. J of Appl Phyco, 1995, 7: 3-15.

- [4] 陈峰,姜悦主编.微藻生物技术[M]. 北京:中国轻工业出版社,1999.
- [5] 刘学铭,梁世中.小球藻的保健和药理作用研究进展 [J]. 中草药, 1999, 30(5): 383-386.
- [6] SHI X M, CHEN F. Stability of lutein under various storage conditions[J]. Nahrung/Food, 1997, (41): 38-41.
- [7] 宋昊,何泽超,章杰,等.万寿菊中叶黄素的提取[J]. 化工设计,2003, 13(4): 10-12.
- [8] SHI X M, CHEN F, YUAN J P. et al. Heterotrophic production of lutein by selected Chlorella strains [J]. J. Appl. Phycol, 1997, 9: 445-450.
- [9] BOROWITZKAME M A, BOTOWITZKA L J. Micro-algal biotechnology[M]. Cambridge: Cambridge University Press, 1992.
- [10] GLADUE R M, MAXEY J E. Microalgal feeds for aquaculture[J]. J. Appl. Phycol, 1994, 6: 131-141.
- [11] 闫海,周洁,何宏胜,等.小球藻的筛选和异养培养[J]. 北京科技大学学报, 2005, 27(4): 408-412.
- [12] MCAULEY P. Nitrogen limitation and amino-acid metabolism of Chlorella symbiotic with green hydra[J]. Planta, 1987, 171: 532-538.
- [13] 刘世名,梁世中.植物激素在小球藻异养培养中的作用 [J]. 植物学通报, 1999, 16(6): 696-700.

(责任编辑 王 芷)

1 引言

基于角色的访问控制(RBAC)^[1]被认为是一种具有广泛应用前景的模型。近年来,RBAC 及其扩展模型已经发展成工业和研究领域中的一种事实上的标准。一个完善的 RBAC 服务可以配置完成许多不同的访问控制策略,包括自主和强制策略。RBAC 一个重要思想是在该模型中所有组件均支持约束,以便达到较高的灵活性。

责权分离^[2]是 RBAC 模型中十分重要的基本原则。在访问控制中,责权分离约束强调了重要策略的冲突。例如:在财务管理系统中,用户被同时授予会计与出纳两个角色,将会导致金融安全问题。这些利益冲突是同时将两个互相排斥的权限或角色授予同一主体或客体所导致的。互斥的角色或权限是由于权利或责任的分离而产生的。如果没有主体可以获得未经授权的权限,则访问控制体系被认为是安全的。此体系是由一组访问控制策略规则集和相应的主体授权关系组成的。换言之,互斥权限是不能通过直接或角色授权的方式指定到主体权限集合中。然而,对于一般访问控制模型如 RBAC,其安全属性的验证是较难判定的。因而,约束作为一种尝试,可以通过清晰的手工建模来增加安全属性。

本文着重分析了责权分离约束在 RBAC 模型中所引发的各种类型冲突示例,进而根据其所含有的共同特征并结合数学中图论的思想,抽象出将冲突检测利用有向图表示的方法。最后,文章给出约束冲突的解决方案。

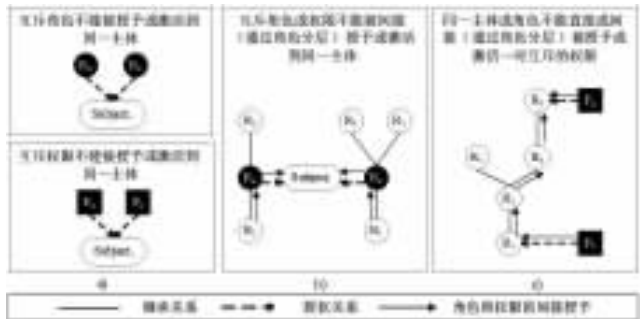


图1 责权分离约束检测关系图

2 责权分离约束

互斥关系是通过责权分离约束^[3]来定义的。责权分离约束分为静态责权分离约束和动态责权分离约束两类。

静态责权分离约束是指互斥角色或权限不能同时授予相同的主体或客体。

动态责权分离约束定义为两个互斥的角色或权限不能同时被激活到相同主体或客体。这意味着两个动态互斥角色可以被授予相同的主体。然而,相关的主体在同一时刻只能激活动态互斥角色或权限中的一个。



图2 RBAC 模型的授权关系

图2描述了RBAC中定义的主体、角色、权限授予关系。图中每一种关系都将影响责权分离约束的定义,并且其自身也会受

到责权分离约束的影响。此后,我们将详细讨论“权限-角色授权”(PRA)、“权限-主体授权”(PSA)、“角色-主体授权”(RSA)、“角色-角色授权”(RRA)中怎样及何时来影响责权分离约束的定义或者受已存在的约束所影响。

图1中描述了新建一个责权分离约束时需要优先进行的校验。如图1a)中所示,不能将两个静态互斥的角色或权限授权到同一主体,同时被激活到同一主体的两个角色或权限也不能是动态互斥的。同样的原则适用于直接或间接(通过角色分层)授权至同一主体的两个角色或权限。如图1b),描述互斥角色的 R_x 和 R_z 不能被定义到主体 Subjecti。同样的,低级别的角色 R_1 和 R_5 不能通过角色分层来间接地授予主体 Subjecti。图1c)中定义的两个互斥权限 P_x 和 P_z 会引发冲突,因为角色 R_5 将通过继承同时拥有 P_x 和 P_z 。在此,采用类似面向对象技术中的继承方法将有助于解决责权分离冲突。

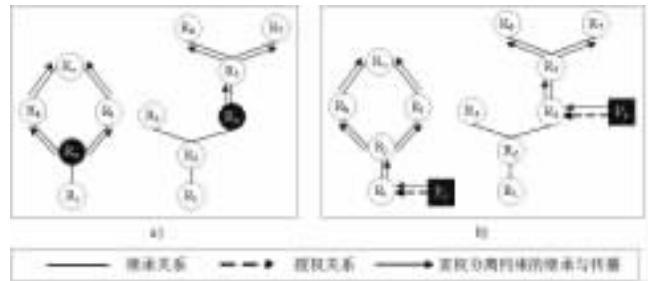


图3 角色分层中责权分离约束的继承与遗传关系图

图3给出角色分层^[4]中责权分离约束的继承与遗传示例。本文用实心圆代表角色、实心矩形表示权限来说明两个角色或权限定义为互斥的。在图3a)中角色 R_x 和 R_z 是互斥的。这对互斥的约束由相应的高级角色所继承。因此,在 R_x 和 R_z 之间存在着的责权分离约束也可以阻止 R_x 的高级角色(如角色 R_m)和 R_z 的高级角色(如角色 R_e)被同时授予或激活到同一主体上。换言之,如果一个给定角色 R_x 定义为角色 R_z 的互斥角色,那么 R_x 将与 R_z 的所有高级角色互斥。事实上,基于角色分层理论,当一个主体被授予 R_z 的一个高级角色时,它将自动获得其自身及所有低级别角色(包括 R_z)的属性。因此,两个角色互斥是有可能因为它们的低级别角色互斥所引起的,如图3a)中示例。

相对于角色中责权分离约束,权限的责权分离约束在角色分层中亦被继承。图3b)中定义 P_x 和 P_z 为互斥的权限。因此, R_1 与 R_4 及相对应的所有高级角色均继承了这种互斥约束。换言之, R_1 与 R_4 是互斥的,由于它们拥有互斥的权限,因此, R_1 与 R_4 及两者相应的高级角色均不能被同时授予同一主体。然而,如果 P_x 或 P_z 从其角色 R_1 与 R_4 中撤销,那么其角色间的互斥约束将同时撤销。

3 责权分离约束检测的模型分析

本文将主要讨论静态责权分离约束(SSD),动态约束与静态约束的处理方式几乎相同。唯一的区别是由于SSD所拥有的静态属性,使其约束将存在于所有的会话的进程中,而动态约束仅应用于被激活的一个指定会话之中。

3.1 责权分离约束检测在“权限-角色授予”(PRA)中的应用

图4展示了PRA中两类型的诱发冲突示例。图4a)描述了一个新权限 P_n 将被授予角色 R_1 的情形。可是由于权限 P_x 相对于权限 P_n 是互斥的,因此该授权操作将不被接受。否则,角色 R_3 将获取到 P_x 和 P_n ,引起模型一致性混乱。图4b)描述了一个更加复杂的情况。主体 Subjecti 拥有角色 R_4 和权限 P_x ,约束将检测一个新的权限 P_n 是否允许授予角色 R_1 。与图4a)中示例类似的是,

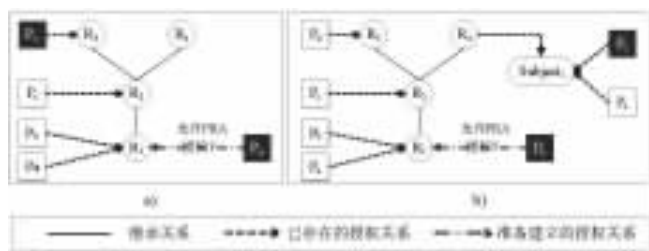


图4 责权分离约束在 PRA 中冲突检测示例

对于 PRA 关系的冲突检测方法必须拒绝相应的授权关系,否则主体 $Subject_i$ 将通过角色分层拥有权限 P_x 和 P_n ,这亦会引起模型一致性冲突的问题。

3.2 责权分离约束检测在“角色-主体授权”(RSA)及“权限-主体授权”(PSA)中的应用

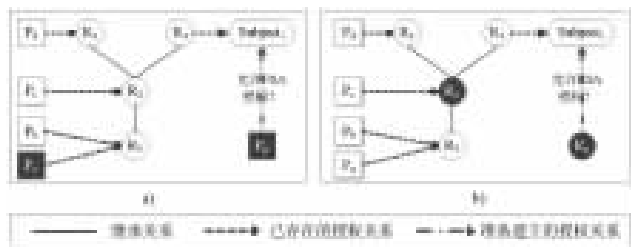


图5 责权分离约束在 RSA 和 PSA 中的示例

图5中分别给出在“角色-主体授权”及“权限-主体授权”中引起冲突的特征示例。图5a)中权限 P_x 和 P_n 互斥,主体 $Subject_i$ 拥有角色 R_4 。一个新权限 P_n 将被直接授予主体 $Subject_i$ 。但是,这个授权操作必须被相应的冲突检测方法拒绝,否则主体 $Subject_i$ 将获得互斥权限 P_x 和 P_n ,导致模型一致性故障。图5b)描述了一种相似状况,角色 R_n 将被授予主体 $Subject_i$ 。然而该操作将被拒绝,以便阻止主体将同时拥有互斥角色 R_x 与 R_n 。

3.3 责权分离约束检测在“角色-角色授予”(RRA)中的应用

角色分层的规范将影响到前面所述每种授权关系,并且自身亦会受到前述每一种授权关系的影响。因此,“角色-角色授权”是责权分离约束检测中最复杂的一种操作。在两个或多个角色间建立起分层关系后,责权冲突可能来自互斥权限或角色的集合,并且该冲突亦可能会有多种不同的方式。

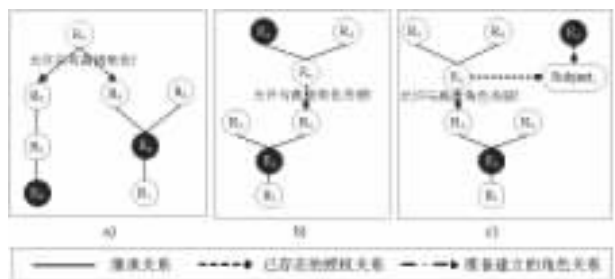


图6 责权分离约束在 RRA 中的示例

图6给出了在 RRA 中较典型的冲突示例。以下部分将不再区分责权分离中角色或权限的约束,因其在 RRA 关系中的结果是类似的。图6a)描述了两个独立角色分层,且 R_x 与 R_z 是互斥的。现在一个新角色 R_n 将这两个角色分层连接起来,冲突检测必须决定是否允许对 R_3 与 R_6 共同的高级角色的定义,其结果是被拒绝。否则, R_n 将会同时拥有互斥角色 R_x 与 R_z 。

图6b)中已经存在的两个角色分层,其中各含有一对互斥角

色 R_x 与 R_z 之一。 R_n 为 R_z 的低级角色, R_4 为 R_x 的高级角色。现在系统将检测 R_n 是否允许定义为 R_4 的高级角色。其结论是否定的。因为根据角色分层理论, R_z 将继续继承其所有低级角色的属性,必然会导致分层模型中出现一对互斥角色 R_x 与 R_z 。图6c)已知主体 $Subject_i$ 同时拥有角色 R_n 与 R_z ,且 R_x 与 R_z 为一对互斥角色,现判断 R_n 是否允许定义为 R_3 的高级角色。根据相关的交互角色分层关系,这种授权是不会产生冲突的且被允许授予的。但是,依照图6c)中特殊的案例,角色 R_z 是不能定义为 R_3 的高级角色的,否则主体 $Subject_i$ 将同时拥有一对互斥角色 R_z 与 R_x 。

4 责权分离约束检测的数学模型

本文把第三节中所述授权关系看作是有向图上的弧,主体、角色、权限等元素看作是节点,使用统一的有向无环图(DAG)表示系统中元素间的关系及其角色(或权限)传播范围。下面所述内容仍以静态责权分离约束^[9]为主,因为动态约束与它的区别仅在于有向图中的节点为被激活的角色或权限。

4.1 责权分离约束在 DAG 中的表示法

定义1:设有两个互斥角色 R_1 与 R_2 ,策略 $A_1(u_1, R_1)$ 表示主体 u_1 被分配 R_1 角色,策略 $A_2(u_2, R_2)$ 表示主体 u_2 被分配 R_2 角色,如果 $A_1.u_1=A_2.u_2$,则称策略 A_1 与 A_2 责权分离冲突,表示成 conflict duties(A_1, A_2)。

记为: $R_1=-R_2$,且 $A_1.u_1=A_2.u_2 \rightarrow \text{conflict duties}(A_1, A_2)$

定义2:设有两个互斥权限 P_1 与 P_2 ,策略 $A_1(u_1, P_1)$ 表示主体 u_1 被分配权限 P_1 ,策略 $A_2(u_2, P_2)$ 表示主体 u_2 被分配权限 P_2 ,如果 $A_1.u_1=A_2.u_2$,则称策略 A_1 与 A_2 责权分离冲突,表示成 conflict duties(A_1, A_2)。

记为: $P_1=-P_2$,且 $A_1.u_1=A_2.u_2 \rightarrow \text{conflict duties}(A_1, A_2)$

根据第三节所述内容,主体 u_i 可以是用户、用户组、资源等元素,亦可以为角色集合中的任意元素。

4.2 责权分离约束检测的算法分析

设策略 $A_1(S_1, R_1, P_1)$ 与策略 $A_2(S_2, R_2, P_2)$ 是系统中的两条策略,且有 $A_1.R_1=-A_2.R_1$ (R_1 与 R_2 互斥)或 $A_1.P_1=-A_2.P_1$ (P_1 与 P_2 互斥)成立; $DAG(V, E)$ 是系统中构成的有向无环图,其中 S 表示用户、组或资源, R 表示角色, P 表示权限。设 $DAG_1(V_1, E)$ 表示以 v_{A1S1} 为顶点的连通分量 $V_1=\{v_{A1S1}, v_{i+1}, v_{i+2}, \dots, v_{i+m-1}\}$, 计有 m 个点, $DAG_2(V_2, E)$ 表示以 v_{A2S2} 为顶点的连通分量 $V_2=\{v_{A2S2}, v_{k+1}, v_{k+2}, \dots, v_{k+n-1}\}$, 计有 n 个点,如果 $V_1 \cap V_2 \neq \emptyset$,则可判定策略 A_1 与策略 A_2 冲突。

对于两个连通分量可考虑使用深度优先搜索 (depth first search) 遍历进行求解。

5 责权分离约束的冲突消解

解决策略冲突最根本的方法是防患于未然,从语法上取消策略冲突。但是,彻底地取消策略冲突是不现实的,因为策略冲突的发生有时是管理员的特殊需要。于是,解决策略冲突的另一种可供选择的方案成为首选,即指定策略之间的执行优先权,在出现策略冲突时,管理系统将选择具有最高优先权的策略加以执行。

5.1 冲突消解方案

本文给出了4种指定策略执行优先权的方案:本地策略优先、近策略优先和指定优先权、时序策略优先。

5.1.1 本地策略优先

本地策略优先^[6]指的是指定给本地的、更小范围的策略具有较高的优先权。例如在域结构系统中,指定给子域的策略比指定给祖先域的策略具有更高的优先权。这个概念最早是由 Miró 系统^[7]引入的。

如:授权策略 A_1 “禁止任意用户重启服务器”和授权策略 A_2 “许可系统管理员重启服务器”,当 A_1 和 A_2 同时授予管理员用户 u_1 时便会产生责权分离约束冲突。但是,由于系统管理员是普通

用户的一部分,所以按照本地策略优先, A_2 将具有比 A_1 更高的优先权,普通用户不能重启机器,而系统管理员是可以的。

5.1.2 近策略优先

在面向对象的系统中,指定给继承层次更近对象的策略具有更高的优先权。仍以 A_1 、 A_2 为例,系统管理员类将通过角色分层,继承普通用户类的所有属性,尽管 A_1 、 A_2 两条策略都可作用于系统管理员对象,但 A_1 是通过继承关系间接指定给管理员对象的, A_2 则是直接指定,比 A_1 在类继承层次上要近,因此具有比 A_1 更高的优先权。该方案类似于本地策略优先,但前者针对的是对象继承层次,后者针对的是域的层次;并且由于类继承层次是固定不变的,可据此计算出具体的数值,在实际应用中非常方便。

5.1.3 指定优先权

指定优先权是通过给每条策略赋予严格的优先权值,来定义它们执行的优先顺序。该方案从理论上说是最完备的,因为只要指定的权值合适,就可解决任意的冲突。该方案的局限性在于仅适用于小规模网络系统,如用于大规模分布式系统中会导致策略赋值的工作相当烦琐,并且在实际操作中很容易发生优先权不一致的情况。

5.1.4 时序策略优先

在当前,大规模分布式系统的管理策略是随系统需求而不断变化着的,并不是在系统启动的时候一次性完全载入的,因此每条策略都有一个加载时间的属性。在指定策略优先权的时候可以将该属性作为参考对象,规定“新加载的策略具有更高的优先权”,当策略出现冲突的时候,选择后加载的策略予以执行。该思想来源于新近制定或修改的策略,应该是管理员更想要的结果。

然而,以上这些方案都只能部分地解决某类冲突问题,没有适应所有情况的通用方案,因此在多数不确定的情况下应将策略冲突提交给管理员,由他们决定如何取舍。

5.2 对各类方案的仿真与评估

评估优先权方案的标准有两个:成功率和正确率。成功率指的是当策略发生冲突的时候,该方案能够判定出策略优先权孰高孰低的情况所占的比率;正确率则指的是在已经判定策略优先权高低的情况下,照此执行的结果符合期望结果所占的比率。

此外,笔者认为关联性分析方法是解决上述各方案的一种有效的补充手段。考虑到本地策略优先与近策略优先类似,而指定策略优先局限性较大,故采用本地策略优先与时序策略优先关联的方案作为方案5,其判定策略优先权公式如下:

- 1) $P(A)>P(B)$, $O(A)>O(B)$ 或 $O(A)=O(B)$, $T(A)>T(B)$
- 2) $P(A)<P(B)$, $O(A)<O(B)$ 或 $O(A)=O(B)$, $T(A)<T(B)$
- 3) 不能判定 $O(A)=O(B)$ 且 $T(A)=T(B)$

$P(A)$ 、 $P(B)$ 分别代表策略 A 、 B 的优先权。 $O(A)$ 、 $O(B)$ 为策略 A 、 B 的本地优先级。 $T(A)$ 、 $T(B)$ 为策略 A 、 B 的时间属性。时间属性只精确到小时,因为如果过于精确,虽然可以提高成功率,却有可能造成正确率大幅下降。

笔者分别实现了方案1、2、4以及5的原型,并以图6a)相似的包含60个节点的结构作为系统背景做了相应的仿真实验,系统中冲突策略总数为15。由于方案3对输入较敏感,不同的输入可能得出迥异的结果,因此在仿真的时候未将它们考虑进去。在输入了6组不同的策略集合后,各方案的成功率和正确率如表1

表1 策略优先权方案的仿真结果

方案	成功率	正确率	有效率
1	0.85	0.79	0.67
2	0.82	0.81	0.66
4	0.90	0.65	0.59
5	0.98	0.91	0.89

所示,表的最后一栏还给出了有效率(成功率和正确率的乘积),以作为统一的评估结果。

从表中可以看出,方案1、2结果类似,其有效率虽不高,但是属于可接受的范围,并且实现简单,大多数系统都选择其一作为缺省方案。方案4的成功率较高,但正确率较低。方案5最为理想,其有效率是所有方案中最高的,证明关联后的方案明显优于关联前。其补充了已有研究的不足,在解决策略冲突时有良好的效果,在相同的系统背景下与原有的方法相比,方案5的有效率提高了30%~51%。

6 结论

本文着重分析了在RBAC模型中责权分离约束所引起的各种冲突模型,并以此为依据与有向图理论相结合,抽象概括出此类冲突的数学表达式,并且对其算法进行了进一步分析推理,使之更加便于工程实践。在此基础上,根据实际工作经验与前人的成果,总结出一套解决冲突的优先级方案。并且将各方案进行仿真试验,利用相关性分析法给出优化方案。

参考文献(References)

[1] FERRAILOLO D, KUHN R, CHANDRAMOULI R. Role-based access control. [M]. MA USA. 685 Canton St. Norwood: Artech House Inc, 2003.

[2] SIMON R, ZURKO M. Separation of duty in role-based environments: Proceedings of the 10th Computer Security Foundations Workshop [C]. Washington, D C: IEEE Computer Society Press, 1997.

[3] LUPU E, SLOMAN M. Conflicts in policy-based distributed systems management [J]. IEEE Transactions on Software Engineering(Special Issue on Inconsistency Management). 1999, 25(6): 852~869.

[4] SANDHU R, FERRAILOLO D, Kuhn R. The NIST model for role-based access control: towards a unified standard: Proceedings of the Fifth ACM Work-shop on Role-Based Access Control (RBAC'00), Berlin, Germany, July 2000 [C]. Bevlín: ACM Press, 2000.

[5] BUCKLEY F, LEWINTER M. A friendly introduction to graph theory[M]. Prentice Hall, 2003.

[6] LUPU E, SLOMAN M. Conflict analysis for management policies. Fifth IFIP/IEEE International Symposium on Integrated Network Management. San-Diego, 1997[C]. Chapman & Hall, Ltd, 1997.

[7] STEGMANN C. A framework for authorization policies [D]. 2229 Route des Cre-etes, F-06904 Sophia Antipolis, France: Institut Eurécom, 1997.

(责任编辑 王 芷)

·读者·作者·编者·

《中国学术期刊文摘(中文版)》2006 年改版

为进一步提高办刊质量,我社对所主办的刊物——《中国学术期刊文摘(中文版)》自2006年第1期起进行改版:(1)四封重新设计,采用全彩印刷;(2)增加“开篇文摘”栏目,每期摘登一篇《科技导报》曾发表的“卷首寄语”文章;(3)重新设计目录页和版权页;(4)逐步缩短各学术期刊报送的文摘条目发表的周期。

(本刊编辑部)