角色访问控制

毛碧波　孙玉芳

（中国科学院软件研究所开放系统与中文信息处理中心　北京100080）

Role based Access Control Model

MAO Bi-Bo　SUN Yu-Fang

(Open System &Chinese Information Processing Center, Institute of Software Chinese Academy of Sciences, Beijing 100080, China)

Abstract　Role based access control (RBAC) was proposed in 70's, and prevailed in 90's, and then Sandhu etc proposed formal RBAC model. Now RBAC is attracting increasing attention, and many governmental and commercial organizations have adopted it, its importance is more and more apparent. In this paper we illuminates the distinctions and similarities of role and user groups, and based the model that was proposed by Sandhu, we examine the relationship of role hierarchies and role constraints and formally describes that, and explain the most important part of role constraints ,which is separation of duties.

Keywords　RBAC, Role constraint, Role hierarchy, MAC, DAC

1　引言

角色访问控制是一种介于传统的自主访问控制和强制访问控制机制之间的安全策略，越来越引起人们的重视。角色访问控制机制很自然映射到组织机构中，每个用户可以赋予一些角色，每个角色负责一些任务。在最近20年中，角色访问控制机制已经有各种不同形式的应用。本文介绍 Sandhu 在96年提出的角色访问控制形式化的定义和模型，指出了角色访问控制和用户组的不同之处，并结合 Sandhu 提出的模型，给出了角色继承和角色限制的定义以及形式化的描述，对角色限制中最重要的部分也就是任务分离作了详细的介绍。

2　角色访问控制模型

人们常常混淆角色和用户组，本节给出了两者的区别。本节还介绍了传统的自主访问控制机制和强制访问控制机制，指出了角色访问控制是介于自主访问控制和强制访问控制两者之间的安全机制，不属于任何一类。

2.1　自主访问控制

在自主访问控制机制中，不管客体属于什么级别的类型，其所有者都有对来自主体访问的自主权，因此在自主访问控制机制中，信息总是可以从一个实体流向另一个实体，即使对于高度机密的信息也是这样，因此如果自主访问控制不加以控制就会产生严重的安全隐患。具体地说，假设有 A、B 和 C 三个不同等级的用户，每个用户安全级别由高到低，A 需要通知 B 一机密文件 secret，但是 A 又不想让 C 了解此文件，所以 C 不可以直接访问文件 secret。但是 B 在得到文件 secret 的同时，他可以拷贝一个同样内容的文件 copy_secret。由自主访问控制机制可知，copy_secret 的所有者 B 有自主权决定让 C 访问文件 copy_secret。特别是，如果用户 B 里有一个特洛木马，即使用户 B 是可信任的，此木马也可以泄漏所有发给或来自 B 的机密信息。

2.2　强制访问控制

由此需要一种控制机制，在此机制中客体的所有者不能决定来自主体的访问权限，这就是强制访问控制机制。强制访问机制最典型的例子就是由 Bell and LaPadula 提出的模型，这个模型特别复杂，我们用 BLP 模型简单描述它。在 BLP 模型中，所有的主体和客体都有一个安全标签，并且此安全标签只能由安全管理员赋值，普通用户不能改变。

在 BLP 模型中[1]，我们用 λ 标志主体或客体的标签，当主体访问客体时，需要满足如下两条规则：

简单安全属性：如果主体 s 能够读客体 o，则 $\lambda(s)\geqslant\lambda(o)$

*-属性：如果主体 s 能够写客体 o，则 $\lambda(s)\leqslant\lambda(o)$

在此模型中写不包括读，例如平时所用的 append 操作。BLP 模型保证了客体的高度安全性，它保证了信息流总是低安全级别的实体流向高安全级别的实体，因此避免了在自主访问控制机制中的敏感信息泄密的情况。但是 BLP 模型也有个问题，由高安全级别所有者拥有的文件永远不能被低安全级别的人访问，因此即使是级别不同的可信任实体也不可以相互通信。因此需要可信任的安全管理员对一些客体的安全级别进行调整，这样才能保证高安全级别的实体与低安全级别的实体相互通信，例如在网络环境中，连接的建立需要双方的互相确认。同时 BLP 模型不能消除隐藏通道；由于低级别的实体可以任意写高级别客体，BLP 模型也不能保证客体的整体性。

同时 Biba 提出了与之相反的模型，称之为 Biba 模型。我们用 w 表示主体或客体的安全标签，当主体访问客体时，需要满足如下两条规则：

简单整体属性：如果主体 s 能够读客体 o，则 $w(s)\leqslant w(o)$

*-属性：如果主体 s 能够写客体 o，则 $w(s)\geqslant w(o)$

Biba 模型保证了系统的整体性，但是可能有不信任实体故意泄漏高安全级别的信息。因此可以把 BLP 模型和 Biba 模型结合起来。对于普通的实体采用 BLP 模型，对于可信任的实体采用 Biba 模型。

还有其它的强制访问控制模型，如中国墙策略模型、信息流模型和 clark－wilson 模型。这些模型由于所需要的安全级别很高，实施起来比较困难，需要较大的代价，一般只是在安全要求比较严格的系统中才采用这样的机制。在一些政府机

构和大型的商业机构中[2]，由于资金有限，同时所需要的安全级别也稍微低一点，因此在这些机构中，采用强制访问控制机制的很少，都在寻找一种容易实现但安全级别又比较高的机制。

2.3 角色访问控制

角色访问控制模型是一种介于自主访问控制和强制访问控制之间的模型，角色访问在70年代就有人提出，只是最近才有人提出形式化的角色模型。其中最有影响的是 Sandhu 在96年提出的形式化模型[3]，如图1所示，其中双箭头代表对应于多个，单箭头对应于单个。在此模型中，包括四个基本的要素，用户、角色、权限和授予，分别用 user，role，permission 和 session 代表。用户就是正在操作的人或主机，角色就是用户在系统和组织中执行的交易的集合，执行这些交易时就代表着系统和组织的一个角色。权限是主体访问客体时认可，权限代表了主体是否可以对客体进行某种操作。RABC 的基本思想是用户被授予角色，角色被授予权限，用户通过所授予的角色来决定其权限。RBAC 中一个角色可以授予给多个用户，一个用户可以授予多个角色，同样角色权限之间也是多对多的关系。RBAC 要求容易计算指定用户所授予的角色和指定角色所计算的用户，对于角色权限之间的关系也是如此。

用户执行任务时通过激活过程来激活所授予的角色，每个激活过程对应一个角色，每个角色可以有多个激活过程，用户可以在运行时同时激活多个角色，因此用户可以同时有多个激活角色的权限。

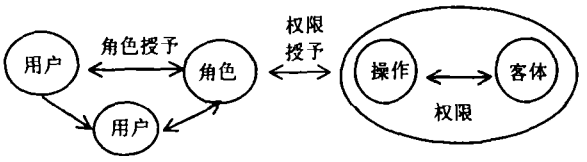

图1 角色访问控制模型

在角色模型中，有如下操作：

权限授予：$P: role \rightarrow 2^{permission}$

p[i]角色 i 所授予的权限

角色授予：$R: user \rightarrow 2^{role}$

R[i]为用户 i 所授予的角色的集合

$A: user \rightarrow 2^{role}$

A[i]为用户 i 当前所激活的角色的集合

$M: role \rightarrow 2^{user}$

M[i]为授予角色 i 的用户集合

角色激活规则：用户当前所激活的角色必须为他所授予的角色，也就是：

$$(\forall u)(\forall r): r \in A[u] \Rightarrow r \in R[u]$$

角色访问控制中，用户的权限由当前的活动角色决定，而不是由用户的权限位或用户的安全级别决定。而在许多大型网络系统，数据库系统和操作系统，以及商业机构中，用户可能经常改变，可是整个系统中角色改变并不多，角色权限之间的对应关系改变也不多。因此添加、删除或更改用户时只需要改变用户对应的角色即可，而不需要改变用户所对应的权限。因此角色访问控制能够降低操作的复杂度，同时降低了管理员误操作的可能性，同时角色访问控制更具有弹性，管理员能够更方便地管理。

很多人混淆角色和用户组的概念，认为用户可以属于不同的用户组，用户组有多个用户，每个用户组也有多种权限。两者还是有区别的，从定义上说，用户组是一组用户的集合，这些用户由于共同的利益可以共享一些权限。而角色不仅是用户的集合，同时也是权限的集合，角色一个很重要的用途是便于管理。任何角色很容易得到此角色的权限和用户的集合，因此很容易得到任何用户的权限。同时角色也是一个更抽象的概念，在不同的系统中有不同的权限，如在数据库系统中角色有 select，delete，grant 等权限，而用户组主要是用于操作系统中。

3 角色继承

角色继承定义了角色之间的相互关系，角色继承是表示组织中不同权限角色之间的很自然的方法。通常的角色继承可理解为角色间权限的继承，如果角色 r1拥有角色 r2的所有权限就说角色 r1继承角色 r2，而可能角色 r1的一些权限 r2可能没有，显然角色继承具有偏序关系，具有自反性、传递性和反对称性，这是角色限制最常见的定义。还有一种定义是包含关系，角色 r1继承 r2，那么所有赋予角色 r1的用户也必须被赋予角色 r2，包含关系比前一种关系要求更强烈，在前一种关系中 r1继承 r2并不一定意味着赋予用户 r1角色时一定要赋予 r2角色。本文所指的角色继承如不作特殊说明，一般指的权限继承关系。

角色继承有两种情况[4]，一种是一般继承，支持角色继承间任意的偏序关系，支持角色多种继承，一角色可以有多个直接子角色，也可以有多个直接父角色。还有一种情况是有限制的继承，例如一个角色可以有多个子角色，但是只能有一个父角色，常见的有限制的继承结构有树和反树结构。

在图2中，它是一个具有反树结构的有限制继承关系，每个角色最多有一个子角色，但是可能有多个父角色，图2中每个部门都具有作为一个部门的基本权限，部门中的研发部和测试部又都具有各自部门的权限。图3是一个具有树状结构的有限制继承，每个角色最多有一个父角色，但是可能有多个子角色，图3中总经理有财务主管和项目主管的权限，项目主管又具有研发部经理和测试部经理的权限。

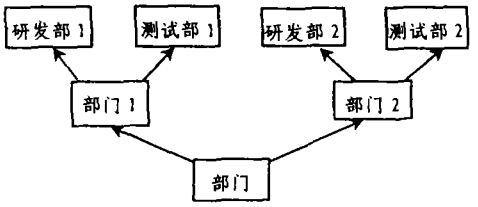

图2 反树结构的有限制继承

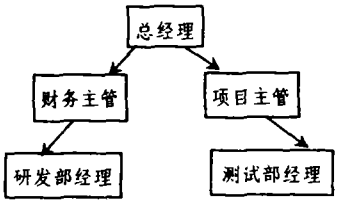

图3 树状结构的有限制继承

反树结构的角色继承有利于权限共享，父角色具有子角色所有的权限，因此两个父角色可以同时共享子角色的权限，然而由于没有一个角色有两个或两个以上的子角色，所以反树结构不利于权限的聚集。但是树结构正好相反，然而树结构有利于资源的聚集而不利于资源的共享。

一般角色继承对于角色之间的继承关系没有限制，反树结构和树结构只不过是它的一种特殊形式。一般角色继承中一个角色可以有多个父角色也可以有多个子角色，一般角色继承有两个好处：首先可以从多个子角色中构造父角色，使

父角色同时具有子角色的权限，其次一般角色继承还具有反树结构。但是一般角色继承有时候角色之间的关系复杂，特别是在大型系统中角色比较多，角色之间的复杂关系可能会引起误操作而带来系统的不一致性。

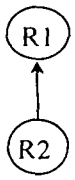

图4a 角色权限完全继承

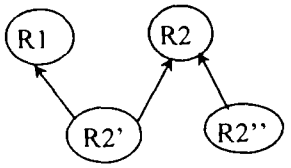

图4b 角色权限部分继承

还有一些角色，他们只代表一些权限的集合，但是并没有授予给任何用户，就像面向对象的虚类一样，我们称这样的角色为虚角色。虽然虚角色没有实例化的用户，但虚角色可以被其它角色继承，所有继承虚角色的角色都有其权限。

两个角色 R1和 R2还有这样的情况，R1由 R2继承，但是R2的一部分权限不想被 R1继承。在 Sandhu 模型中，把 R2分成两部分 R2' 和 R2''，分别是 R2和 R1共有的权限 R2' 和 R2的私有权限 R2''，如图4a、4b 所示。

4 角色限制

限制是角色模型中的重要部分，角色模型的安全性就是要求角色间的相互限制。在许多系统和组织中，角色间或角色—用户之间和角色权限间有各种各样的限制。根据角色的状态可以分为静态限制和动态限制[5]。静态限制是在角色被授权给用户时的限制，动态限制是角色被激活时的限制。静态限制比动态限制的范围要大，角色被激活时不仅要满足动态限制还要满足静态限制。

角色访问控制的一个重要的功能就是任务分离，同时角色访问控制还支持最小特权，任务分离和最小特权一直是完全问题几十年来的话题。有时候为了防止误操作或者欺诈行为发生，两个不同的任务不能由同一个人完成，角色访问控制通过利用角色之间的互斥实现任务分离，同一用户不能授予给互斥的两个角色。同时用户执行任务时，只需要赋予用户相应的角色从而实现最小权限。

4.1 静态限制

静态限制是在授予角色给用户时必须满足的条件，根据用户、角色和权限三者之间的关系可分为如下：

数目限制 角色所授权的用户的个数有限，例如每个系的系主任只能由一个用户担任。还有用户所授权的角色数有限，例如一个用户只能担任有限个职位。可形式化描述为：

max_uses [i]＝角色 i 所能授予的最大用户数目，默认值是系统的用户数目。
authorized_users [i]＝角色 i 所授予的用户数目。
authorized_users [i] <= max_users [i]
∀ i∀ j j≤i⇒authorized_users[i] <= max_users[j]

静态任务分离 由于内在的利益冲突或防止欺骗行为，需要任务分离。根据用户、角色和权限之间的关系可以分为角色互斥和权限互斥。角色互斥指的是两个不同的角色不能由同一用户执行，常见的有角色 i，j 不能授权给同一用户。权限互斥指的是权限 p，q 不能授权给同一角色。

角色互斥 〈i，j〉∈mutex_role，当且仅当角色 i 与 j 不能授权给同一用户，显然〈i，j〉∈mutex_role 时有〈j，i〉∈mutex_role。形式化描述如下：

(∀ i∀ j)(∀ u)〈i，j〉∈mutex_role⇔(i∈R[u]⇒j∉R[u])⇔(j∈R[u]⇒i∉R[u])

权限互斥 〈p，q〉∈ mutex_permission，当且仅当权限 p 与 q 不能授给同一角色，显然〈p，q〉∈mutex_permission 时有〈q，p〉∈utex_permission。形式化描述为：

(∀ p∀ q)(∀ r)〈p，q〉∈mutex_permission⇔(p∈P[r]⇔q∉P[r])⇒(q∈P[r]⇒q∉P[r])

把角色限制和角色继承结合起来，可以得出：分别拥有互斥权限的角色不能被同一个角色所继承，形式化描述如下：

(∀ p∀ q)(∀ i∀ j)〈p，q〉∈mutex_permission ∧p∈P[i]∧q∈P[j]⇒¬∃ k(i≤k∧j≤k)

4.2 动态限制

动态限制的条件比静态限制的条件要弱，有时角色被激活时不仅要满足静态限制条件，也要满足动态限制条件，常见的动态限制有下面几点。

数目限制：由于用户的安全级别或系统的性能，用户所能激活角色的最大数目有限。可形式化描述为：

max_roles [i]＝用户 i 所能激活的最大角色数目。
authorized_roles [i]＝用户 i 所激活角色的数目。
authorized_roles [i] <= max_roles [i]

动态任务分离：只是在角色被激活时才有一些限制，因此称之为动态任务分离，常见的有角色 i，j 不能同时被一个用户激活。例如任务 t1和 t2可以由一个用户执行，但是用户不能同时执行任务 t1和 t2。常见的有两个互斥角色 i，j 不能被用户同时激活。形式化描述为：

(∀ i∀ j)(∀ u)〈i，j〉∈mutex_role⇔(i∈A[u]⇒j∉A[u])⇔(j∈A[u]⇒i∉A[u])

小结 角色访问控制由于不是直接授权给用户，而是先授予权限给角色，然后再授予用户角色，这样在用户和权限之间引入角色，从而大大降低了系统的复杂度，同时角色访问控制体现了系统的组织结构，简洁具有灵活性，大大降低了系统管理员误操作的可能性。角色之间的互斥关系可以很容易地实现任务分离，角色访问控制还支持最小权限。

角色模型现在还只是一种抽象模型，实现时需要跟具体的控制策略结合在一起。角色访问控制虽然已经模型化，由于后来不断地增添和修改，经过许多发展，现在角色访问控制还没有统一的形式。例如在角色继承中，有的系统定义角色权限的包含关系为继承关系，有的系统定义不仅要求如此，还要求角色授予的包含关系才为继承关系。本文所叙述的部分都是角色访问控制的静态部分，角色访问控制的动态部分，例如角色的创建和删除，角色权限的授予，用户角色的授予等，现在都没有统一的规范。

本文从角色访问最初的模型出发，明确地论述了角色访问控制机制，分析了角色控制中角色之间的关系，角色继承是为了更好地描述角色之间权限的关系，角色继承就像根树一样，父节点拥有子节点的所有权限，更加清晰明白地说明了角色间的关系。角色限制根据系统中安全的要求而产生，角色限

（下转第89页）

表1 两种老算法的性能

类数	鉴别矢量数	训练样本数	错误识别数	
			文[8]的方法	文[9]的方法
4	2	4	0	0
4	3	4	0	0
5	3	4	0	6
5	4	4	0	6
6	5	4	0	5
7	6	4	1	8
8	7	4	3	20
9	8	4	3	6
10	9	4	5	16

表2 本文方法的实验结果

类数	鉴别矢量数	训练样本数	错误识别数
2	1	3	0
2	1	4	0
3	2	3	0
3	2	4	0
4	3	3	0
4	3	4	0
5	4	3	0
5	4	4	0
6	5	3	0
6	5	4	0
7	6	4	0
8	7	3	0
9	8	3	1
9	8	4	1
10	9	3	3
11	10	4	1

结论 本文对统计不相关最佳鉴别矢量集的求解方法进行研究，获得如下结论：

(1)本文提出了一种新的统计不相关最佳鉴别矢量的求解方法。

(2)本文提出的广义DKL变换较成功地解决了人脸识别中的小样本问题，并且特征抽取和识别速度较快。

(3)在ORL人脸数据库的数值实验验证了本文方法的有效性。

(4)新方法不仅对人脸的特征提取有效，而且对手写体数字识别、汉字识别以及基于内容的检索等模式识别领域的研究都有一定的意义。

(5)在实验中我们发现，当类别数目比较小时识别率均为100%，但当类别数目较大时仍然要保持高识别率，还有待于进一步的研究，作者正在致力于这一问题的研究。

参考文献

1 Turk M, Pentland A. Eigenfaces for face recognition. J. Cognitive Neuroscience, 1991, 3(1): 71～86

2 Foley D H, Sammon J W. An optimal set of discriminant vectors. IEEE Trans. Computers, 1975, 24(3): 281～289

3 Duchene J, Leclercq S. An optimal transformation for discriminant and principal component analysis. IEEE Trans. PAMI, 1988, 10(6): 978～983

4 Hong Z Q. Algebraic feature extraction of image for recognition. Pattern Recognition, 1991, 24(3): 211～219

5 洪子泉，杨静宇. 用于图像识别的图像代数特征抽取. 自动化学报，1992, 18(2): 232～238

6 洪子泉，杨静宇. 基于奇异值特征和统计模型的人像识别算法. 计算机研究与发展，1994, 31(3): 60～65

7 Cheng Y Q, Yang J Y, et al. A novel feature extraction method for image recognition based on similar discrminant function. Pattern Recognition, 1993, 26(1): 115～125

8 Liu K, Yang J Y, et al. An efficient algorithm for Foley-Sammon optimal set of discriminant vectors by algebraic method. International Journal of Pattern Recognition and Artificial Intelligence, 1992, 6(5): 817～829

9 Liu K, Yang J Y, et al. A generalized optimal set of discriminant vectors. Pattern Recognition, 1992, 25(7): 731～739

10 郭跃飞，杨静宇. 求解广义最佳鉴别矢量的一种迭代算法及人脸识别. 计算机学报，2000, 23(11)

11 郭跃飞. 人脸图像代数特征提取与最佳鉴别矢量的研究：[博士学位论文]. 南京：南京理工大学，2000

12 Guo Yue-Fei, Yang Jing-Yu, et al. Feature Extraction Method Based on the Generalized Fisher Discriminant Criterion and Facial Recognition. Pattern Analysis & Application, 2001, 4(1): 61～66

13 吴小俊，杨静宇，王士同，等. A new algorithm for solving optimal discriminant vectors. Journal of Computer Science and Technology, 2002, 17(3): 324～330

14 金忠. 人脸图像特征抽取与维数研究：[博士学位论文]. 南京：南京理工大学，1999

15 金忠，杨静宇. 一种具有统计不相关性的最佳鉴别矢量集. 计算机学报，2000

16 Swets D L, John (Juyang) Weng. Using discriminant eigenfeatures for image retrieval. IEEE Trans. PAMI, 1996, 18(8)

(上接第123页)

制描述了系统安全的要求。

参考文献

1 Bell D, Padula L. Secure computer systems: unified exposition and MULTICS: [Report ESD-TR-75-306]. The MITRE Corporation, Bedford, Massachusetts, March 1976

2 Clark D, Wilson D. A comparison of commercial and military computer security policies. In: proc. of the Symp. on Security and Privacy, IEEE Press, 1987. 184～194

3 Sandhu R, Coyne E, Feinstein H, Youman C. Role-based access control models. IEEE Computer, 1996, 29(2)

4 Ferraiolo D F, Sandhu R. Proposed NIST Standard for Role-Based Access Control. ACM Transactions on Information and Systems Security, 2001, 4(3)

5 Ahn G, Sandhu R. Role-Based Authorization Constraints Specification. ACM Transactions on Information and System Security, 2000, 3(4)