第5章　访问控制

访问控制是信息安全保障机制的重要内容，它是实现数据保密性和完整性机制的主要手段之一。访问控制是在身份认证的基础上，根据身份对提出的资源访问请求加以控制，其目的是为了保证网络资源受控、合法地使用，用户只能根据自己的权限来访问系统资源，不能越权访问，同时，访问控制也是记账、审计的前提。广义地讲，所有的计算机安全都与访问控制有关。

本章5.1节介绍了访问控制的概念和组成要素；5.2节具体介绍三种访问控制机制，即自主访问控制、强制访问控制和基于角色的访问控制机制，详细介绍了自主访问控制的三种实现方式以及强制访问控制的安全模型。

5.1　访问控制概述

5.1.1　访问控制机制与系统安全模型

James P. Anderson在1972年提出的引用监控器(The Reference Monitor)的概念是经典安全模型的最初雏形，如图5.1所示。

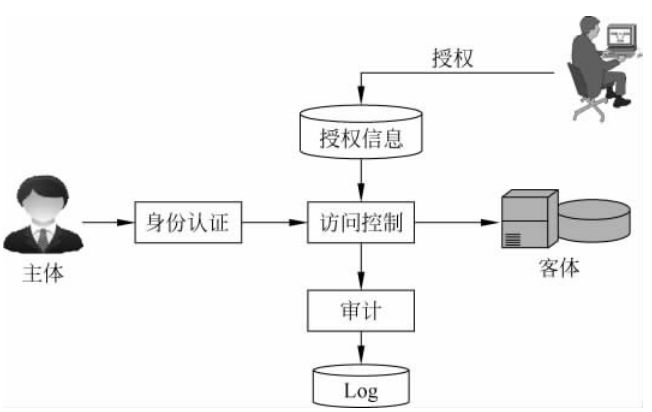

图5.1　引用监控器模型

经典安全模型包括如下基本要素：

(1) 明确定义的主体和客体；

(2) 描述主体如何访问客体的一个授权数据库；

(3) 约束主体对客体访问尝试的引用监控器；

(4) 识别和验证主体和客体的可信子系统；

(5) 审计引用监控器活动的审计子系统。

从图5.1中可以看出，实现计算机系统安全的基本措施(安全机制)包括身份认证(识别和验证)、访问控制和审计。身份认证是验证用户的身份与其所声称的身份是否一致的过

程。访问控制是在主体身份得到认证后，根据授权数据库中预先定义的安全策略对主体行为进行限制的机制和手段。审计作为一种安全机制，它在主体访问客体的整个过程中都发挥作用，为安全分析提供了有力的证据支持。本章主要讨论访问控制技术。

5.1.2　访问控制的基本概念

访问控制技术起源于 20 世纪 70 年代，当时是为了满足管理大型主机系统上共享数据授权访问的需要。随着计算机和网络技术的发展，访问控制技术在信息系统的各个领域得到了越来越广泛的应用，先后出现了多种重要的访问控制技术，如自主访问控制、强制访问控制、基于角色的访问控制等。

访问控制常常以身份认证作为前提，在此基础上实施各种访问控制策略来控制和规范合法用户在系统中的行为，身份认证解决的是“你是谁，你是否真的是你所声称的身份”，目的是阻止非法用户进入系统，而访问控制技术解决的是“你能做什么，你有什么样的权限”，目的是限制合法用户的操作权限。

访问控制包括两个重要的过程，其一是系统通过授权（Authorization）设定合法用户对资源的访问权限规则集；其二是根据预先设定的规则对用户访问某项资源（目标）的行为进行控制，只有规则允许时才能访问，违反预定安全规则的访问行为将被拒绝。资源可以是信息资源、处理资源、通信资源或者物理资源，访问方式可以是获取信息、修改信息或者完成某种功能，一般情况可以理解为读、写或者执行。

访问控制是针对越权使用资源的防御措施，通过限制对关键资源的访问，防止非法用户的侵入或因为合法用户的不慎操作而造成的破坏，从而保证网络资源受控、合法地使用。访问控制中涉及的主要概念包括以下几个：

1. 主体（Subject）

主体是指访问操作的主动发起者，它造成了信息的流动和系统状态的改变，主体可以是用户或其他任何代理用户行为的实体，如进程、作业等。

2. 客体（Object）

客体是被访问的对象，客体在信息流动中的地位是被动的，处于主体作用之下。凡是可以被操作的对象都可以认为是客体，客体通常包括文件、目录、消息、程序、库表等，还可以是处理器、通信信道、时钟、网络节点等。

3. 访问（Access）

访问是使信息在主体和客体之间流动的一种交互方式。访问包括读取数据、更新数据、运行程序、发起连接等。

4. 访问控制策略

访问控制策略是主体对客体的访问规则集，这个规则集直接定义了主体对客体的作用行为和客体对主体的条件约束。访问策略体现了一种授权行为，也就是客体对主体的权限允许，这种允许不超越规则集中的定义。

访问控制策略的制定需要考虑以下原则：

1）最小特权原则

最小特权原则是指主体执行操作时，按照主体所需权利的最小化原则分配给主体权利。

最小特权原则的优点是最大限度地限制主体行为，可以避免来自突发事件、错误和未授权主体的危险。也就是说，为了达到一定目的，主体必须执行一定操作，但他只能做他所被允许做的。

2）最小泄露原则

最小泄露原则是指主体在执行任务时，按照主体所需知道的信息最小化的原则分配给主体权利。

3）多级安全策略

主客体分配一定的安全级别，安全级别通常包括绝密、秘密、机密、限制和无级别 5 级。主客体的数据流向和权限控制以不允许信息从高级别向低级别流动为原则，采用多级安全策略可以避免敏感信息的扩散。

访问控制在信息系统中的应用非常广泛，例如对用户的网络接入过程进行控制、操作系统中控制用户对文件系统和底层设备的访问。另外当需要提供更细粒度的数据访问控制时，可以在应用程序中实现基于数据记录或更小的数据单元访问控制。例如大多数数据库管理系统（如 Oracle）都提供独立于操作系统的访问控制机制，Oracle 使用其内部用户数据库，且数据库中的每个表都有自己的访问控制策略来支配对其记录的访问。

5.2 访问控制策略

1985 年美国军方提出了可信计算机系统评估准则 TCSEC，其中描述了两种著名的访问控制策略：自主访问控制和强制访问控制。基于角色的访问控制（RBAC）由 Ferraiolo 和 Kuhn 在 1992 年提出，考虑到网络安全和传输流，又提出了基于对象和基于任务的访问控制。

各种访问控制策略之间并不相互排斥，现存计算机系统中通常都是多种访问控制策略并存，系统管理员能够对安全策略进行配置使其达到安全政策的要求。

5.2.1 自主访问控制

自主访问控制（Discretionary Access Control，DAC）是指资源的所有者（往往是创建者），对于其拥有的资源，可以自主地将访问权限分发给其他主体，即确定这些主体对于资源有怎样的访问权限，是最常用的访问控制机制。在这种访问控制机制下，客体的拥有者可以按照自己的意愿精确指定系统中其他用户对其客体的访问权，从这种意义上来说，是“自主的”。Linux、UNIX、Windows NT/SERVER 版本的操作系统，SQL Server、Oracle 等数据库管理系统都提供了自主访问控制的功能。自主访问控制通常有三种实现机制，即访问控制矩阵（Access Control Matrix）、访问控制列表（Access Control Lists，ACLs）和访问控制能力表（Access Control Capabilities Lists，ACCLs）。

1. 访问控制矩阵

访问控制矩阵是最初实现访问控制机制的概念模型，它利用二维矩阵规定了任意主体和任意客体间的访问权限。矩阵中的行代表主体的访问权限属性，矩阵中的列代表客体的访问权限属性，矩阵中的每一格表示所在行的主体对所在列的客体的访问授权。如表 5.1 所示，其中 Own 表示所在行主体是所在列客体的属主，可以自主授予或回收其他用户对其

拥有客体的访问权限，即拥有对客体管理的权限，R 表示读操作，W 表示写操作。

表 5.1　访问控制矩阵示例

	File1	File2	File3	File4
张三	Own，R，W		Own，R，W	
李四	R	Own，R，W	W	R
王五	R，W	R		Own，R，W

访问控制矩阵清晰地描述了任意主体对任意客体的访问权限，但是，在较大的系统中，访问控制矩阵将变得非常巨大，而且矩阵中的许多任务格可能为空，造成很大的存储空间浪费，因此在实际应用中，访问控制很少利用矩阵方式实现，目前大部分系统实现的自主访问控制是用基于访问控制矩阵的行或列来表达访问控制信息。

2. 访问控制列表

访问控制列表实际上是按访问控制矩阵的列实施对系统中客体的访问控制，是从客体角度进行设置的、面向客体的访问控制。每个客体有一个访问控制列表，用来说明有权访问该客体的所有主体及访问权限，如图 5.2 所示。利用访问控制列表，能够很容易地判断出对于特定客体的授权访问。

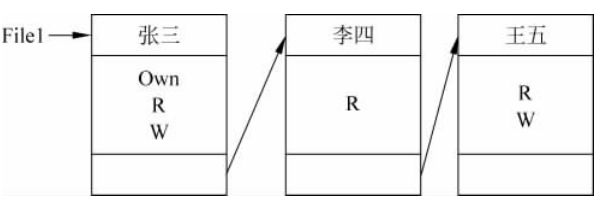

图 5.2　访问控制列表

由于访问控制列表简单、实用，虽然在查询特定主体能够访问的客体时，需要遍历查询所有客体的访问控制列表，它仍然是一种成熟且有效的访问控制实现方法，许多通用的操作系统使用访问控制表来提供访问控制服务。

【例 5-1】 Linux 中实现了访问控制列表的简略方式，将系统中的所有用户划分为三类：属主用户、同组用户、其他用户，系统按这三类用户进行授权，权限主要包括 r：读，w：写，x：执行，这样可以使得访问控制列表只需要 9 位就可描述。如某个文件的访问控制列表为“rwxr-x---”，从左往右每三位为一组，第一组“rwx”表示文件的属主拥有可读、可写、可执行权限，第二组“r-x”表示文件属主的同组用户拥有读和执行权限，第三组“---”表示其他用户对该文件没有访问权限。

3. 访问能力表

访问能力表（Access Capabilities List）实际上是按访问控制矩阵的行实施对系统中客体的访问控制，如图 5.3 所示。能力（Capability）是为主体提供的、对客体具有特定访问权限的不可伪造的标志，它决定主体是否可以访问客体以及以什么方式（如读、写、修改或运行）访问客体。主体可以将能力转移给为自己工作的进程，在进程运行期间，还可以动态地添加或修改能力。能力的转移不受任何策略的限制，所以对于一个特定的客体，不能确定所有有权访问它的主体，利用访问能力表实现自主访问控制的系统并不多。

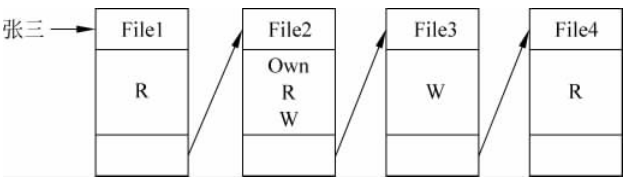

图 5.3 访问控制能力表

自主访问控制的最大特点是自主，即资源的拥有者对其资源的访问策略具有决策权，因此是一种限制比较弱的访问控制策略，这种方式给用户带来灵活性的同时，也带来了安全隐患。这种机制允许用户自主地将自己客体的访问操作权转授给别的客体，权力多次转授后，一旦转授给不可信主体，那么客体的信息就会泄露。DAC 的另一个缺点是无法抵御特洛伊木马的攻击，木马窃取敏感文件的方法有两种，一是通过修改敏感文件的访问权限来获取敏感信息，在 DAC 机制下，某一合法用户可以任意运行一段程序修改自己文件的访问权限，系统无法区分这是合法用户的修改还是木马程序的非法修改；二是躲在用户程序中的木马利用合法用户身份读敏感文件的机会，把所访问文件的内容复制到入侵者的临时目录下，而 DAC 无法阻止，因而无法抵挡特洛伊木马的攻击。

【例 5-2】 假设用户 SOS 将其重要信息存放在文件 important. doc 中，并且将文件权限设置成只有自己可以读写。SPY 是一个恶意攻击者，试图读取 important. doc 文件的内容，他首先准备好一个文件 pocket. doc，并将其权限设置成为 SOS：w，SPY：rw，同时设计一个有用的程序 use_it_please，该程序除了有用部分，还包含一个木马。当诱使 SOS 下载并运行该程序时，木马会以 SOS 用户的身份执行，将 important. doc 中的信息写入 pocket. doc 文件，这样 SPY 就窃取了 important. doc 的内容。

对安全性要求更高的系统，仅采用 DAC 是不够的，需要采用更安全的访问控制技术——强制访问控制。

5.2.2 强制访问控制

1. 强制访问控制的概念

强制访问控制（Mandatory Access Control，MAC）是比 DAC 更为严格的访问控制策略，最早出现在 20 世纪 70 年代，是美国政府和军方源于对信息保密性的要求以及防止特洛伊木马攻击而研发的。

与 DAC 相比，强制访问控制不再让众多的普通用户完全管理授权，而是将授权归于系统管理，并确保授权状态的变化始终处于系统的控制下。在强制访问控制中，每个主体（进程）和客体（文件、消息对列、共享存储区等）都被赋予一定的安全属性，并且安全属性只能由管理部门（如安全管理员）或操作系统按照严格的规则进行设置，当一个进程访问一个客体（如文件）时，强制访问控制机制通过比较进程的安全属性和文件的安全属性来决定访问是否允许，如果系统判定拥有某一安全属性的主体不能访问某个客体，那么即使是客体的拥有者都不能使该主体有权访问客体。MAC 主要用于保护敏感数据（例如，政府、军队敏感文件等）。

在系统中实现 MAC 时，需要根据总体安全策略和需求为系统中的每个主体和客体分

配一个适当的安全级别，且安全级别是不能轻易改变的，它由管理部门（如安全管理员）或由操作系统自动按照严格的规则设置。在 MAC 下，即使是客体的拥有者，也没有对自己客体的控制权，并且系统安全管理员修改、授予、撤销主体对客体的访问权的管理工作，也要受到严格的审核与监控。

2. 强制访问控制模型

1）BLP 模型

1973 年，David Bell 和 Len Lapadula 提出了第一个也是最著名的安全策略模型 Bell-LaPadula 安全模型，简称 BLP 模型，BLP 模型是遵守军事安全策略的多级安全模型，主要用于解决面向机密性的访问控制问题，已实际应用于许多安全操作系统的开发中。

在 BLP 模型中主客体的安全属性由两部分构成。

（1）保密级别（又称为敏感级别或级别）：例如公开、秘密、机密、绝密等。

（2）一个或多个范畴：该安全级涉及的领域，例如陆军、海军、空军等。

因此一个安全属性包括一个保密级别、一个范畴集，而范畴集包含任意多个范畴，安全属性通常写作保密级后随一个范畴集的形式，例如{机密；陆军，海军，空军}。

在安全属性中，保密级别是线性排列的，例如公开＜秘密＜机密＜绝密，范畴则是相互独立和无序的，两个范畴集之间的关系是包含、被包含或无关。

BLP 模型有两个基本规则。

（1）规则 1（简单安全性）：一个主体对客体进行读操作的必要条件是主体的安全级支配客体的安全级，即主体的保密级别不小于客体的保密级别，主体的范畴集包含客体的全部范畴，即主体只能向下读。

（2）规则 2（＊特性）：一个主体对客体进行写访问的必要条件是客体的安全级支配主体的安全级，即客体的保密级别不小于主体的保密级别，客体的范畴集包含主体的全部范畴，即主体只能向上写。

BLP 模型的强制访问控制可以概括为不允许“上读，下写”，这种规则是由信息的保密性的安全要求决定的。保密性要求只有高保密级的主体能够读低保密级客体的内容，否则会造成高保密级的客体的信息泄密；反过来，高保密级的主体对低保密级的客体进行写操作也会造成信息泄密，如图 5.4 所示。

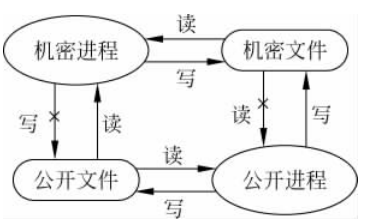

图 5.4　多级安全规则

【例 5-3】　客体 LOGISTRIC 文件的敏感标签为 SECRET[VENUS ALPHA]，主体 Jane 的敏感标签为 SECRET[ALPHA]，虽然主体的敏感等级满足上述读写规则，但是由于主体 Jane 的类集合当中没有 VENUS，所以不能读此文件，而写则允许，因为客体 LOGISTIC 的敏感等级不低于主体 Jane 的敏感等级，写了以后不会降低敏感等级。

运用 BLP 模型的＊特性可有效防范特洛伊木马。前面介绍过木马窃取敏感文件的方法有两种，一是通过修改敏感文件的访问权限来获取敏感信息，在 DAC 机制下，某一合法用户可以任意运行一段程序修改自己文件的访问权限，系统无法区分这是合法用户的修改还是木马程序的非法修改，但在 MAC 下，杜绝了用户修改客体安全属性的可能，因此木马利用这种方法窃取敏感信息是不可能的；二是特洛伊木马伪装成正常的程序，例如一个小

游戏、一个小工具，诱使用户下载运行，而实际当运行带有木马的程序时，木马会利用合法用户的身份读取敏感信息，把所访问的文件复制到入侵者的临时目录下，这在 DAC 机制下是完全可以做到的，然而在 * 特性下，能阻止正在机密安全级上运行的木马，把机密信息写到一个低安全级别的文件中，因为机密级进程写的每条消息的安全级至少是机密级的。

基于 BLP 模型的 MAC 阻止了信息由高级别的主/客体流向低级别的主/客体，保证了信息的机密性，适用于保密性要求比较高的军事、政府部门和金融等领域，但该模型不能保证信息的完整性。而在商业领域，以加强数据完整性为目的的强制访问控制模型也有广泛的应用。

2）Biba 模型

对于政府发布的公告，允许所有用户阅读，但绝对不允许被篡改，在这种应用场合下，更强调数据的完整性保护。

Biba 模型是 BLP 模型的变体，由 Biba 等人于 1977 年提出，它的主要目的是保护数据的完整性。

在 Biba 模型中，每个主体和客体都被分配一个完整性属性，类似于 BLP 模型，该完整性属性是由一个完整性级别和一个范畴集构成的。Biba 模型规定，信息只能从高完整性等级向低完整性等级流动，就是要防止低完整性的信息“污染”高完整性的信息。

Biba 模型并未约定具体采用的策略，而是将策略分为非自主策略和自主策略两类，在每类下给出了一些具体的策略以适应不同的需求，下面简单介绍非自主策略。

非自主策略是指主体是否具有对客体的访问权限取决于主体和客体的完整性级别，具体规则为：

主体对客体进行读访问的必要条件是客体的完整级不低于主体的完整级，即主体只能向上读。

主体对客体进行写操作的必要条件是主体的安全级不低于客体的安全级，即主体只能向下写。

3）Dion 模型

Dion 于 1981 年提出了同时面向机密性和完整性的 Dion 模型，该模型结合 BLP 模型中保护数据机密性的策略和 Biba 模型中保护数据完整性的策略，模型中的每一个客体和主体被赋予一个安全级别和完整性级别，安全级别定义同 BLP 模型，完整性级别定义如 Biba 模型，因此可以有效地保护数据的机密性和完整性。

强制访问控制是比自主访问控制功能更强的访问控制机制，但是这种机制也给合法用户带来许多不便。例如，在用户共享数据方面不灵活且受到限制。因此，当敏感数据需在多种环境下受到保护时，就需要使用 MAC，如需对用户提供灵活的保护且更多地考虑共享信息时，则使用 DAC。

在高安全级（TCSEC 标准的 B 级）以上的计算机系统中常常将自主访问控制和强制访问控制结合在一起使用。自主访问控制作为基础的、常用的控制手段；强制访问控制作为增强的、更加严格的控制手段。一些客体可以通过自主访问控制保护，重要客体必须通过强制访问控制保护。对于通用型操作系统，从用户友好性出发，一般还是以 DAC 机制为主，适当增加 MAC 控制，目前流行的操作系统（如 Windows、UNIX、Linux）、数据库管理系统（SQL Server、Oracle）均属于这种情况。

5.2.3　基于角色的访问控制

1. 概述

MAC 和 DAC 属于传统的访问控制模型，通常为每个用户赋予对客体的访问权限规则集，如果系统中用户数量众多，且系统安全需求处于不断变化中，就需要进行大量烦琐的授权操作，系统管理员的工作将变得非常繁重，更主要的是容易发生错误，造成安全漏洞。

在现实的工作中，绝大多数情况并不是针对每个人设定其工作职责，而是根据这个人在工作单位中所承担的角色设定其工作职责的，例如医院包括医生、护士、药剂师等角色，而银行则包括出纳员、会计、行长等角色，用户的职责完全由其承担的角色来决定，当其承担的角色发生改变，其职责也会随之改变。信息系统是现实世界的反映，因而在信息系统中一个用户所能访问资源的情况应该随着用户在系统中角色的改变而改变。

基于角色的访问控制(Role-Based Access Control，RBAC)是 20 世纪 90 年代 NIST (National Institute of Standards and Technology)提出的访问控制策略，这种技术能够减少授权管理的复杂性、降低管理开销，而且还能为管理员提供一个比较好的实现复杂安全政策的环境。目前这一访问控制模型已被广为接受。

RBAC 中的基本元素包括用户、角色和权限，RBAC 的核心思想是将访问权限分配给一定的角色，用户通过饰演不同的角色获得角色所拥有的权限。所谓角色(Role)是一个或一群用户在组织内可执行的操作的集合。用户通过角色与相应的访问权限相联系，用户权限是其所拥有角色权限的并集，脱离了角色用户将不存在任何访问权限。角色相当于工作部门中的岗位、职位或分工。一个角色可以有多个权限(对多个资源的访问权)；一个角色可以对应多个用户(相当于一个岗位可以有多个职员)。

【例 5-4】 在学院教务系统中，假设用户有学生 Stud1，Stud2，Stud3，…，Stud$j$，有教师 Tch1，Tch2，Tch3…Tch$i$，有教务管理人员 Mng1，Mng2，Mng3，…，Mng$k$，用户数量众多，在为用户授权时，可以定义如下角色，TchMN＝{查询成绩、上传所教课程的成绩}，StudMN＝{查询成绩、反映意见}，MngMN＝{查询、修改成绩、打印成绩清单}，为用户分配相应的角色，一旦某个用户成为某角色的成员，则此用户可以完成所具有的职能。

角色由系统管理员定义，角色成员的增减也只能由系统管理员来执行，即只有系统管理员有权定义和分配角色。

2. RBAC 模型

由于 RBAC 采用的很多方法在概念上接近于人们社会生活的管理方式，所以相关的研究和应用发展得很快。从 1996 年发展至今，专家们已经提出了一系列 RBAC 模型，这里主要探讨美国 George Mason 大学提出的 RBAC96 模型，该模型为开发实际的应用系统提供了一个总方针，并为 RBAC 用户提供了评判系统的标准，具体包括 RBAC0、RBAC1、RBAC2、RBAC3 四个模型，其中：

RBAC0—基本模型，规定了任何 RBAC 系统所必需的最小需求。

RBAC1—在 RBAC0 的基础上增加了角色等级(Role Hierarchies)的概念。

RBAC2—在 RBAC0 的基础上增加了限制(Constraints)的概念。

RBAC3—包含了 RBAC1 和 RBAC2，依传递性也间接包含了 RBAC0。

美国国家标准和技术研究所(NIST)已经基于 RBAC96 制定了 RBAC 标准,它将 RBAC 主要分为核心 RBAC、有角色继承的 RBAC 和有约束的 RBAC 三类。

1) 核心 RBAC 模型

核心 RBAC 模型包括 6 个基本集合:用户集 USERS、对象集 OBJECTS、操作集 OPERATORS、权限集 PERMISSIONS、角色集 ROLES 和会话集 SESSIONS,如图 5.5 所示。USERS 中的用户可以执行操作,是主体;OBJECTS 中的对象是系统中被动的实体,主要包括被保护的信息资源;对象上的操作构成了权限,因此 PERMISSIONS 中的每个元素涉及来自 OBJECTS 和 OPERATORS 的两个元素,ROLES 是 RBAC 的中心,通过它将用户与特权联系起来,SESSIONS 包括了系统登录或通信进程和系统之间的会话。以下具体给出将上述集合关联在一起的操作,通过这些操作,用户被赋予了相应的权限或获得了相应的状态。

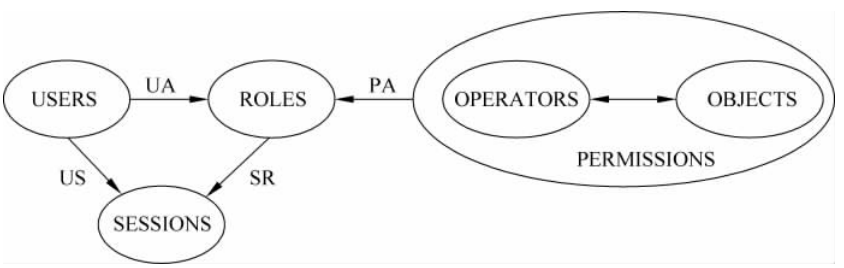

图 5.5 核心 RBAC 中集合及其关系

(1) 用户分配(UA,User Assignment)

UA⊆USERS×ROLES 中的元素确定了用户和角色之间多对多的关系,记录了系统为用户分配的角色。若对用户 u 分配角色 r,则 UA=UA∪(u,r)。

(2) 特权分配(PA,Permission Assignment)

PA⊆PERMISSION×ROLES 中的元素确定了权限和角色之间多对多的关系,记录了系统为角色分配的权限。若把权限 p 分配给角色 r,则 PA=PA∪(p,r)。

(3) 用户会话

US⊆USERS×SESSIONS 中的元素确定了用户和会话之间的对应关系,由于一个用户可能同时进行多个登录或建立多个通信连接,这个关系是一对多的。

(4) 激活/去活角色

若某个用户属于某个角色,与之对应的会话可以激活该角色,SR⊆SESSIONS×ROLES 中的元素确定了会话与角色之间的对应关系,此时该用户拥有与该角色对应的权限。用户会话也可以通过去活操作终止一个处于激活状态的角色。

总之,在 RBAC 中,系统将权限分配给角色,用户需要通过获得角色来得到权限。

2) 有角色继承的 RBAC 模型

有角色继承的 RBAC 模型是建立在以上核心 RBAC 基础上的,它包含核心 RBAC 的全部组件,但增加了角色继承(Role Hierarchies,RH)操作,如图 5.6 所示。如果一个角色 r1 继承另一个角色 r2,r1 也有 r2 的所有权限,并且有角色 r1 的用户也有角色 r2。

RBAC 标准包括两种方式的继承:一种是受限继承,一个角色只能继承某一个角色,不支持继承多个角色;另一种是多重继承,一个角色可以继承多个角色,也可以被多个角色继承。这样,角色的权限集不仅包括系统管理员授予该角色的权限,还有其通过角色继承获得

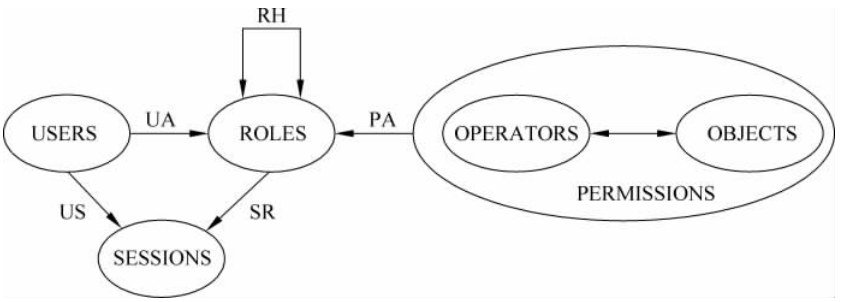

图 5.6　有角色继承的 RBAC 中集合及其关系

的权限，而对应一个角色的用户集不仅包括系统管理员分配的用户，还包括所有直接或间接继承该角色的其他角色分配的用户。

3）有约束的 RBAC 模型

有约束的 RBAC 模型通过提供职责分离机制进一步扩展了以上有角色继承的 RBAC 模型，如图 5.7 所示。职责分离是有约束的 RBAC 模型引入的一种权限控制方法，其目的是为了防止用户超越其正常的职责范围，例如在银行业务中，授权付款与实施付款应该是分开的职能操作，否则可能发生欺骗行为，职责分离主要包括静态职责分离和动态职责分离。

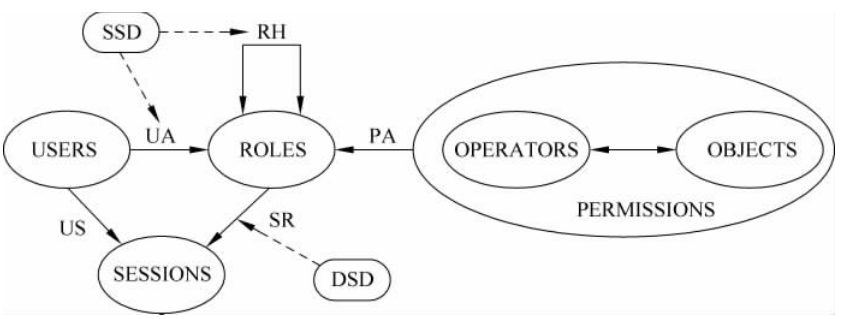

图 5.7　有约束的 RBAC 中集合及其关系

（1）静态职责分离

静态职责分离（Statistic Separation of Duty，SSD）对用户分配和角色继承引入了约束。如果两个角色之间存在 SSD 约束，那么当一个用户分配了其中一个角色后，将不能再获得另一个角色，即存在排他性。由于一个角色被继承将使它拥有继承它的其他角色的全部用户，如果在 SSD 之间的角色存在继承关系，将会违反前述的排他性原则，因此，不能在已经有 SSD 约束关系的两个角色之间定义继承关系。

（2）动态职责分离

动态职责分离（Dynamic Separation of Duty，DSD）引入的权限约束作用于用户会话激活角色的阶段，如果两个角色之间存在 DSD 约束关系，系统可以将这两个角色都分配给一个用户，但是，该用户不能在一个会话中同时激活它们。

3. RBAC 的特点和应用优势

RBAC 具有以下几大特点：

（1）便于授权管理。RBAC 将权限与角色关联起来，用户的授权是通过赋予相应的角

色来完成的。当用户的职责变化时只需要改变角色即可改变其权限；当组织的功能变化或演进时，则只需删除角色的旧功能、增加新功能，或定义新角色，而不必更新每一个用户的权限设置。这极大地简化了授权管理，降低了授权管理的复杂度。

(2) 便于实施职责分离。通过定义角色约束，可以防止用户超越其正常的职责范围，有效地实现职责分离。

(3) 便于实施最小权限原则。最小特权是指用户所拥有的权力不能超过他执行工作所需的权限。实现最小特权原则，需要分清用户的工作职责，确定完成该工作的最小权限集，然后把用户限制在这个权限集范围之内。一定的角色就确定了其工作职责，而角色所能完成的操作蕴含了其完成工作所需的最小权限。用户要访问信息首先必须具有相应的角色，用户无法绕过角色直接访问信息。

正是由于 RBAC 具有灵活性、方便性和安全性的特点，目前在大型数据库管理系统的权限管理中得到了普遍应用，但是，在大型分布式网络环境下，通常无法确知网络实体的身份真实性和授权信息，而 RBAC 无法实现对未知用户的访问控制和委托授权机制，从而限制了 RBAC 在网络环境中的应用。

虽然 RBAC 已在某些系统中得到了应用，但 RBAC 仍处于发展阶段，RBAC 的应用仍是一个相当复杂的问题。

5.3 小　　结

访问控制在身份认证的基础上，根据身份对提出资源访问的请求加以控制。它是对信息系统资源进行保护的重要措施，也是计算机系统最重要和最基础的安全机制。访问控制机制主要包括自主访问控制机制、强制访问控制机制和基于角色的访问控制机制。自主访问控制是计算机系统中实现最多的访问控制机制，其主要特征表现在：主体可以自主地把自己所拥有客体的访问控制权限授予其他主体或从其他主体收回所授予的权限；而 MAC 则根据客体的敏感级和主体的许可级来限制主体对客体的访问，多用于多级军用系统。基于角色的访问控制的基本思想是将访问权限分配给一定的角色，用户通过饰演不同的角色获得角色所拥有的权限，在网络规模变大、用户增多、需求更复杂的情况下，传统的访问控制机制已经不能满足许多企业或组织的安全需求，基于角色的访问控制 RBAC 便明显地显示出其优越性。

习　　题

一、填空题

1. 访问控制的三要素包括_____、_____、_____。

2. 文件的拥有者可以决定其他用户对于相应的文件有怎样的访问权限，这种访问控制是_____。

3. 信息系统实现访问控制有多种方式，其中以用户为中心建立起的描述访问权限的表

格，这种方式指的是_____。

4. Bell-LaPadula 模型的出发点是维护系统的_____，而 Biba 模型与 Bell-LaPadula 模型完全对立，它修正了 Bell-LaPadula 模型所忽略的信息的_____问题。

5. 访问控制中，访问的发起者称为_____，接受访问的被动实体称为_____。

6. 引用监控器模型中涉及到的基本安全机制有_____、_____、_____。

7. 自主访问控制的实现方式有_____、_____、_____。

8. 强制访问控制模型主要有_____、_____。

9. 基于角色的访问控制中的基本元素包括_____、_____、_____。

二、选择题

1. 下列对访问控制影响不大的是(　　)。

A. 主体身份　　B. 客体身份

C. 访问类型　　D. 主体与客体的类型

2. 访问控制是指确定(　　)以及实施访问权限的过程。

A. 用户权限　　B. 可给予哪些主体访问权利

C. 可被用户访问的资源　　D. 系统是否遭受入侵

3. 信息系统实现访问控制有多种方式，其中以用户为中心建立起的描述访问权限的表格，这种方式指的是(　　)。

A. 访问控制矩阵　　B. 访问控制表

C. 访问控制能力表　　D. 授权关系表

4. 文件的拥有者可以决定其他用户对于相应的文件有怎样的访问权限，这种访问控制是(　　)。

A. 自主访问控制　　B. 强制访问控制

C. 主体访问控制　　D. 基于角色的访问控制策略

三、简答题

1. 什么是自主访问控制？自主访问控制的实现方法有哪些？

2. 什么是强制访问控制？如何利用强制访问控制抵御特洛伊木马的攻击？

3. 什么是基于角色的访问控制技术？它与传统的访问控制技术有何不同？

4. 简述访问控制的基本概念。

5. 有哪几种访问控制策略？

6. 访问控制策略制定可以遵循哪些原则？

7. 自主访问控制和强制访问控制可以在一个系统中共存吗？

8. 与传统的访问控制相比，基于角色的访问控制有哪些优点？