(2019春季　课程编号：011184)

信息安全导论

第4章　授权与访问控制技术

中国科学技术大学 曾凡平

billzeng@ustc.edu.cn

课程回顾：第3章 身份认证

- 3.1 用户认证
  - 3.1.1 基于口令的认证
  - 3.1.2 基于智能卡的认证
  - 3.1.3 基于生物特征的认证
- 3.2 认证协议
  - 3.2.1 单向认证
  - 3.2.2 双向认证
- 3.3 Kerbores
  - 3.3.1 Kerberos版本4
  - 3.3.2 Kerberos版本5
- 3.4 PKI技术
  - 3.4.1 PKI体系结构
  - 3.4.2 X.509数字证书
  - 3.4.3 认证机构
  - 3.4.4 PKIX相关协议
  - 3.4.5 PKI信任模型

第4章 授权与访问控制技术

4.1 授权和访问控制策略的概念	
4.2 自主访问控制	• 基本概念，授权管理，不足之处 • 完善自主访问控制机制
4.3 强制访问控制	• 基本概念，授权管理，不足之处
4.4 基于角色的访问控制	• 基本概念，授权管理，RBAC的优势
4.5 基于属性的访问控制	• ABAC模型，ABE基本概念
4.6 PMI技术	• PMI基础，PKI和PMI的关系 • PMI授权管理模式、体系及模型 • PMI基础设施的结构和应用模型

4.1 授权和访问控制策略的概念

- **给已通过认证的用户授予相应的权限，这个过程被称为授权(authorization)**。在信息系统中，可授予的权限包括读 / 写文件、运行程序和网络访问等，实施和管理这些权限的技术称为授权技术。
- 目前，主要有两种授权技术，即访问控制技术和PMI技术。

- **资源**主要指信息数据、计算处理能力和网络通信资源等。在访问控制中，通常将它们称为客体，

相关的概念

- **“访问”一词可以概括为系统或用户对这些资源的使用**，如读取数据、执行程序、占用通信带宽等，这些“访问者”通常被称为**主体**，而有的实体既可以作为主体，也可以作为客体，如计算机程序，因此也常用**实体统一指代客体和主体**。
- **授权是指资源的所有者或控制者准许别的主体以一定的方式访问某种资源，访问控制(access control)是实施授权的基础，它控制资源只能按照所授予的权限被访问。**从另一个角度看，由于对资源的访问进行了控制，才使得权限和授权得以存在。但是，在特定的访问控制基础上，可能存在不同的授权方式。

访问控制策略及访问控制矩阵

- 访问控制策略是在系统安全较高层次上对访问控制和相关授权的描述，它的表达模型常被称为访问控制模型，是一种访问控制方法的高层抽象和独立于软硬件实现的概念模型。由于访问控制涉及主体授予针对客体的权限问题，因此本质上任何访问控制策略都可以用矩阵直观表示(图4-1)

	客体o1	客体o2	客体o3
主体sl	读取、修改、执行	读取、执行	读取
主体s2	读取	读取	
主体s3	读取、执行		执行

图4-1 一个简单的访问控制矩阵

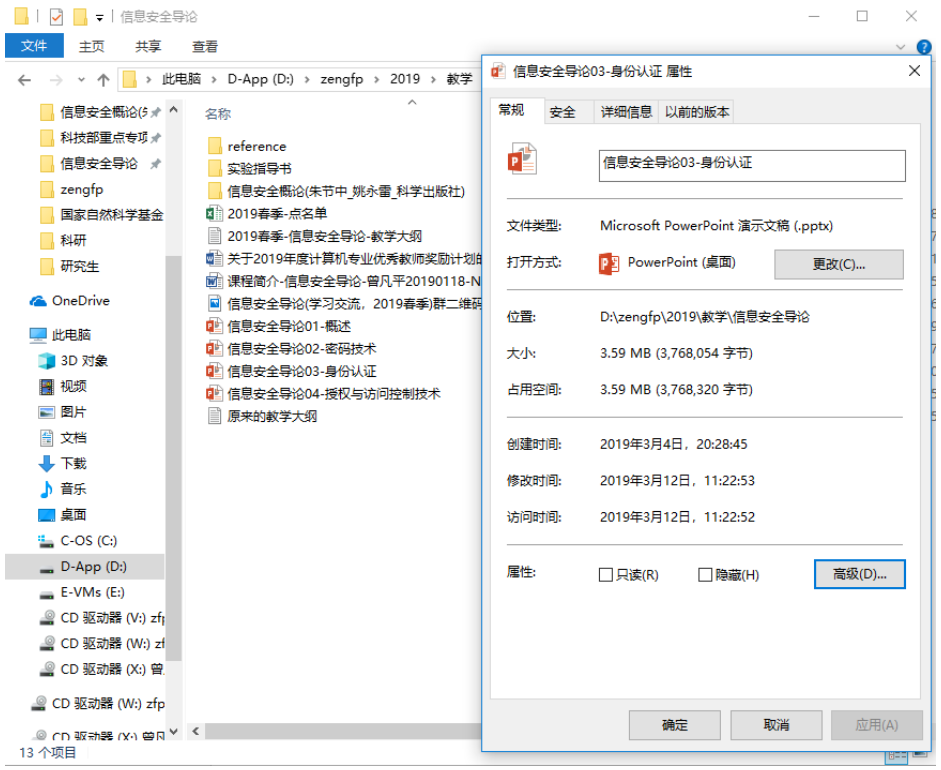
1)主体属性

- **用户的级别或种类是主要的主体属性**，操作系统一般将用户分为多种普通用户和管理员用户，用户还可以分成组，因此具有组别属性。在其他系统中，用户被授予各种角色属性，如是局长或科员。
- **主体属性还可能包括相关执行程序的性质、所处的网络或物理地址等**，它们也可能是授权的依据。例如，很多单位规定，从家中不能访问办公室的资源。
- **在安全性要求更高的情况下，主体的属性可能还包括其安全状态**。例如，在可信网络连接(trusted network connection，TNC)应用中，访问控制系统在允许某计算机接入前，可以首先评估它的漏洞补丁版本，若版本不是最新的，表明计算机已经遭到攻击或感染病毒的概率较大，因此不予授权连接。

2)客体属性

- **客体的主要属性是所允许的操作及其信息级别**。
- 操作系统一般将资源分为是否可读、是否可写、是否可执行、是否可连接等属性。
- 在普通信息系统中，这些属性还可能包括密级、是否可查询、是否可删除、是否可增加等。
- **在安全性要求更高的情况下，客体的属性也可能包括其安全状态**，例如，系统可能认为某些客体已经感染计算机病毒或来源不可信，因而不允许用户访问；有些计算机系统可能被评估为较低的安全等级，管理者不允许高等级的计算机访问它们。

实例：Windows系统的主体和客体



4.2 自主访问控制

- 在自主访问控制(discretionary access control，DAC)中，由客体的所有者（或控制者）对自己的客体进行管理，由所有者决定是否将自己客体的访问权或部分访问权授予其他主体。
- 一般地，自主访问控制策略是基于主体的身份和先行规定的访问规则来对访问进行控制，也就是说，系统允许客体的所有者按照自己的意愿去制定谁以何种访问模式去访问该客体的策略。
- 自主访问控制在C2级操作系统中应用广泛，是根据自主访问控制策略建立的一种模型，允许合法用户以用户或用户组的身份访问策略规定的客体，同时阻止非授权用户访问客体，某些用户还可以自主地把自己所拥有的客体的访问权限授予其他用户。

4.2.1 基本概念

- 自主访问控制策略根据来访主体的身份，以及“谁能访问、谁不能访问、能在哪些资源上执行哪些操作”等事先声明的访问规则，来实施访问控制。之所以被称为自主策略，是因为它基于这样的思想：**客体的主人（即资源所有者）全权管理有关该客体的访问授权**，有权泄露、修改该客体的有关信息。
- 资源的所有者将访问权限授予其他用户或用户组后，被授权的用户便可以自主地访问资源，或者将权限传递给其他的用户。有些学者也称自主访问控制为**“基于主人的访问控制”**。这种自主一方面使得访问控制的灵活性很高，另一方面由于信息在移动过程中其访问权限关系会被改变，使得原本不具有访问权限的主体也可能获得访问权限，造成安全方面的隐患。

1．传统DAC策略

- 特点：访问权限的管理依赖于所有对客体具有访问权限的主体。明显地，自主访问控制主要存在以下三点不足。

① 资源管理比较分散。

② 用户间的关系不能在系统中体现出来，不易管理。

③ 不能对系统中的信息流进行保护，容易泄露，无法抵御特洛伊木马。

- 其中，第三点不足对安全管理来说是非常不安全的，针对自主访问控制的不足，许多研究者对其提出了一系列的改进措施。

2．HRU、TAM、ATAM策略

- HRU与传统DAC最大的不同在于它将访问权限的授予改为半自主式：主体仍然有权利将其具有的访问权限授予其他客体。但是，这种授予行为要受到一个调整访问权限分配的安全策略的限制，通常这个安全策略由安全管理员来制定。
- 在HRU中，每次对访问矩阵进行改变时（包括对主体、客体以及权限的改变），先生成一个临时的结果，然后用调整访问权限分配的安全策略来对这个临时结果进行判断。如果这个结果符合此安全策略，才允许此次访问权限的授予。
- 可以说，HRU模型用得好的话，可以完全不用担心非授权者会“意外”获得某个不应获得的访问权限。但这种设定当主体集和客体集发生改变时，需要依赖安全管理员对访问权限的扩散策略进行更新。

TAM策略和ATAM策略

- TAM策略对此作出了改进：每当产生新主体时，管理员就得要对新主体的访问权限和它本身所拥有权限的扩散范围进行限定。每当产生新客体时，其所属主体和管理员就需要对其每一种权限的扩散范围进行限定。这样一来，只要前期系统架构合理，TAM就能极为方便地控制住访问权限的扩散范围。
- ATAM策略则是在TAM策略的基础上，为了描述访问权限需要动态变化的系统安全策略而发展出来的安全策略。

3．基于角色／时间特性的DAC策略

- 对于严格的DAC，管理角色只有OWN_O，正规角色可以包括READ_O、WRITE_O和EXECUTE_O，分别表示有权读、写或执行的用户集。

- 2000年左右，有研究者提出使用基于角色的访问控制来模拟自主访问控制，讨论了将角色和自主访问控制结合的方法，针对三种DAC类型，设计了文件管理角色和正规角色。管理角色根据DAC类型不同，可包括OWN_O、PARENT_O和PARENTwithGRANT_O。正规角色根据访问方式不同，可包括READ_O、WRITE_O和EXECUTE_O。OWN_O角色有权向PARENT_O添加或删除用户，PARENTwithGRANT_O角色有权向PARENT_O添加或删除用户。正规角色中用户具有相应的读、写或执行的权限。

基于时间特性的DAC策略

- 在许多基于时间特性的DAC策略中，时间点和时间区间的概念被引到DAC中并与访问权限结合，使得访问权限具有时间特性。换句话说，用户只能在某个时间点或者时间区间内对客体进行访问。该方法使主体可以自主地决定其他哪些主体可以在哪个时间访问它所拥有的客体，实现了更细粒度的控制。
- 在一些客体对访问许可有严格时间要求的系统中，如军事信息、情报、新闻等，基于时间特性的DAC策略就比较适合。当然为了更加严格地控制信息流的传递，通常此策略也会和其他访问控制策略相结合。

4.2.2 授权管理

- 自主访问控制的授权管理大致有以下几种。
- (1)集中式管理：单个的管理者或组对用户进行访问控制授权或授权撤销。
- (2)分级式管理：一个中心管理员把管理责任分配给其他管理员，这些管理员再对用户进行访问授权和授权撤销。分级式管理可以根据组织结构实行。
- (3)所属权管理：如果一个用户是一个客体的所有者，则该用户可以对其他访问该客体的用户进行访问授权和授权撤销。
- (4)协作式管理：对于特定系统资源的访问不能由单个用户授权决定，而必须要其他用户的协作授权决定。
- (5)分散式管理：在分散管理中，客体所有者可以把权力权限授权给其他用户。

访问控制实现技术

(1)保护位(owner/group/other)机制

- 在此机制中，每个操作系统客体都附有一个位集合，以便为不同安全类别的用户指定不同访问模式。常见的实现中其类别包括Owner、Group、Other三类，保护位分别指定这三类用户的读、写、执行权限。由于保护位与客体相关联，显然它可决定哪些用户对客体拥有自主访问权限和在需要时撤销权限。访问权的复制和扩展可简单地通过将此客体的保护位的修改权限授予某些用户来实现。
- 比如Linux系统的文件访问控制机制。

(2)能力表(capabilities)机制

- 能力表机制将每一个操作系统的主体与一个客体访问列表（能力表）相联系，它指定了主体可以访问的客体以及此主体对此客体相应的访问方式。
- 由于能力表与主体相关联，故在一个特定时刻判断哪些主体对一个特定客体具有访问权限比较困难，这使得访问权限的撤销变得复杂。典型的是：一个用户可以通过提供一个必需的能力表的拷贝将访问权限授予其他用户，结果是访问权限的扩展过于复杂而难以控制。能力表机制提供了一种在运行期间实行访问控制的方式（例如，它在DBMS中可能发挥作用：只要能够检索用户／主体模版，就可以判断其对一个表的特定视图是否具有访问权限）。然而，这一方法对每个用户都需要很多的项来实现这种检索。

(3)访问控制表(access control lists，ACLs)机制

- 访问控制表是目前最流行、使用最多的访问控制实现技术。每个客体有一个访问控制表，是系统中每一个有权访问这个客体的主体的信息。这种实现技术实际上是按列保存访问矩阵。
- 访问控制表提供了针对客体的方便的查询方法，通过查询一个客体的访问控制表很容易决定某一个主体对该客体的当前访问权限。删除客体的访问权限也很方便，把该客体的访问控制表整个替换为空表即可。但是同能力表类似，用访问控制表来查询一个主体对所有客体的所有访问权限是很困难的，必须查询系统中所有客体的访问控制表来获得其中每一个与该主体有关的信息。同样地，删除一个主体对所有客体的所有访问权限也必须查询所有客体的访问控制表，删除与该主体相关的信息。
- 保护位机制就是这样一种简化形式的访问控制表。

(4)授权关系表(authorization relation)机制

- 访问矩阵也有既不对应于行也不对应于列的实现技术，那就是对应访问矩阵中每一个非空元素的实现技术——授权关系表。授权关系表的每一行（或者说元组）就是访问矩阵中的一个非空元素，是某一个主体对应于某一个客体的访问权限信息。如果授权关系表按主体排序，查询时就可以得到能力表的效率；如果按客体排序，查询时就可以得到访问控制表的效率。
- 虽然授权关系表需要更多的资源空间，但由于它的访问的高效性，像安全数据库这类系统通常采用授权关系表来实现其访问控制安全机制。

4.2.3 不足之处

- 自主访问控制策略在UNIX、WindowsNT等流行操作系统和许多数据库系统中得到了广泛应用。人们在实践中发现了该策略的许多不足，主要体现在以下方面。
- (1)既然用户可任意在系统中规定谁可以访问它们的资源，那么系统管理员就难以确定哪些用户对哪些资源有访问权限，不利于实现统一的全局访问控制。
- (2)在许多组织中，用户对他们所能访问的资源并不具有所有权，组织本身才是系统中资源的真正拥有者。而且，各组织希望访问控制与组织内部的安全策略相一致，并由管理部门统一实施访问控制，不允许用户自主地处理，而DAC却存在用户滥用职权的问题。
- (3)用户间的关系不能在系统中体现出来，不易管理。
- (4)信息容易泄露，不能抵御特洛伊木马(trojan horse)的攻击。

- 除了上述不足之处以外，自主访问控制还存在一些安全风险。
- 按照访问许可机制的不同，自主访问控制又分为三个类型，即**自由型、等级型和宿主型**。在自由型自主访问控制机制中，不同主体之间可以自由转让客体访问控制表的修改权限，意味着任何主体都有可能对某一客体进行操作，系统安全性很难得到保障；在等级型自主访问控制机制中，用户可以将拥有修改客体访问控制表权限的主体组织成等级型结构。例如，按照等级将不同的主体排列成树型结构，高等级主体自动获得低等级客体的控制权限。这种方案的优点是可以选择值得信任的人担任各级领导，从而实现对客体的分级控制，缺点是同时有多个主体有能力修改某一客体的访问权限。

- 从市场应用情况看，等级型自主访问控制是使用范围最为广泛的安全机制，现有大型商用服务器操作系统（如AIX、HP-UX、Solaris、Windows Server、Linux Server等）中的访问控制机制均为**等级型自主访问控制**，涉及金融、能源、军工等国家命脉行业。在这些系统中，位于树型结构顶端的超级用户拥有无上的权限，可以对其他用户拥有的资源进行任意修改和访问。权限的高度集中，客观上放大了系统的安全风险。针对等级型自主访问控制，攻击者可以通过暴力破解、系统漏洞利用、木马攻击等多种方式窃取管理员权限，进而实现对目标系统的完全控制。事实证明确实如此，无论是曾经肆虐全国的“灰鸽子”木马，还是震‘惊全球的“震网”、“火焰”等病毒，都将获得管理权限作为一种重要手段，在此基础上成功入侵系统并实施破坏行为。

4.2.4 完善自主访问控制机制

- 为了提升信息系统的安全防护能力，我国颁布了《信息安全等级保护管理办法》，并制定了一系列国家标准，为用户开展信息安全等级保护工作提供指导意见。其中，GB/T20272-2006《信息安全技术——操作系统安全技术要求》是专门针对操作系统安全防护的国家标准，该标准在“自主访问控制”部分提出了明确的要求：“客体的拥有者应是唯一有权修改客体访问权限的主体，拥有者对其拥有的客体应具有全部控制权，但是，不允许客体拥有者把该客体的控制权分配给其他主体。”
- 从技术要求的细节上看，满足等级保护标准的自主访问控制机制实质上是宿主型自主访问控制。

4.3 强制访问控制

- 强制访问控制(mandatory access control，MAC)是一种多级访问控制策略。它的主要特点是系统对访问主体和受控对象实行强制访问控制，系统事先给访问主体和受控对象分配不同的安全级别属性。在实施访问控制时，系统先对访问主体和受控对象的安全级别属性进行比较，再决定访问主体能否访问该受控对象。
- MAC用于将系统中的信息依据密级和类进行管理，以保证每个用户只能访问到资源中被标明他可以访问的信息的一种访问约束机制。通俗地说，在强制访问控制下，用户（或其他主体）与资源（或其他客体）都被标记了固定的安全属性（如安全级、访问权限等），在每次访问发生时，系统检测安全属性以便确定一个用户是否有权访问。
- 其中，多级安全(multilevel secure，MLS)就是一种强制访问控制策略。

4.3.1 基本概念

- 强制访问控制模型基于与每个数据项和每个用户关联的安全性标识(security label)。安全性标识被分为若干级别：绝密(top secret)、机密(secret)、秘密(confidential)、一般(public)。**数据的标识称为密级(security classification)，用户的标识称为许可级别证(security clearance)**。
- 在计算机系统中，每个运行的程序继承用户的许可证级别，也可以说，用户的许可证级别不仅仅应用于作为人的用户，而且应用于该用户运行的所有程序。当某一用户以某一密级进入系统时，在确定该用户能否访问系统上的数据时应遵守如下规则。

→(1)当且仅当用户许可证级别大于或等于数据的密级时，该用户才能对该数据进行读操作。

→(2)当且仅当用户许可证级别小于或等于数据的密级时，该用户才能对该数据进行写操作。

强制访问控制(MAC)的主要特征

- **强制访问控制(MAC)的主要特征是权威制定访问规则，对所有主体及其所控制的客体（进程、文件、段、设备等）实施强制访问控制。**
- 访问控制是“强加”给访问主体的，即系统强制主体服从访问控制策略。系统首先为这些主体及客体指定敏感标记，这些标记是等级分类和非等级类别的组合，它们是实施强制访问控制的依据。
- 系统通过比较主体和客体的敏感标记来决定一个主体是否能够访问某个客体。用户的程序不能改变他自己及任何其他客体的敏感标记。

强制访问控制策略

- 强制访问控制一般与自主访问控制结合使用，并且实施一些附加的、更强的访问限制。一个主体只有通过了自主与强制性访问限制检查后，才能访问某个客体。用户可以利用自主访问控制来防范其他用户对自己客体的攻击，由于用户不能直接改变强制访问控制属性，所以强制访问控制提供了一个不可逾越的、更强的安全保护层以防止其他用户偶然或故意地滥用自主访问控制。
- 多级安全策略(multilevel security policy)是最为常见的强制访问控制策略，它基于系统中主体与客体的分级来决定是否允许访问。多级安全策略指预先定义好用户的可信任级别和资源的安全级别，当用户提出访问请求时，系统对两者进行比较以确定访问是否合法。

多级访问控制

- 在多级访问控制系统中，所有主体和客体都被分配了安全标签，安全标签对其自身的安全等级进行了标识，其作用过程是：
- (1)主体被分配一个安全等级；
- (2)客体也被分配一个安全等级；
- (3)执行访问控制时，对主体和客体的安全级别进行比较。
- 在强制式策略中，资源访问授权根据资源和用户的相关属性确定，或者由特定用户（一般为安全管理员）指定。它的特征是强制规定访问用户允许或者不允许访问资源或执行某种操作。

主体对客体的访问方式

- (1)向下读(rd，readdown)：主体安全级别高于客体信息资源的安全级别时允许查阅的读操作；
- (2)向上读(ru，readup)：主体安全级别低于客体信息资源的安全级别时允许的读操作；
- (3)向下写(wd，writedown)：主体安全级别高于客体信息资源的安全级别时允许执行的动作或是写操作；
- (4)向上写(wu，writeup)：主体安全级别低于客体信息资源的安全级别时允许执行的动作或是写操作。
- 由于MAC通过分级的安全标签实现了信息的单向流通，因此它一直被军方采用，其中最著名的是Bell-LaPadula模型和Biba模型。

4.3.2 授权管理

- 在强制访问控制中，访问控制完全是根据主体和客体的安全级别决定。其中，主体的安全级别是由系统安全管理员赋予用户，而客体的安全级别则由系统根据创建它们的用户的安全级别决定。

- 因此，强制访问控制的授权管理策略比较简单，**只有安全管理员能够改变主体和客体的安全级别。**

4.3.3 不足之处

- 强制访问控制策略最早应用于1965年由AT&T和MTT联合开发的安全操作系统Multics系统中，在1983年美国国防部的可信计算机系统评估标准中被用做B级安全系统的主要评价标准之一。强制访问控制策略目前主要应用于军事系统或是安全级别要求较高的系统之中。该策略去除了自主访问控制策略中由用户来自由分配的特点，而采用集中控制的方法。它对特洛伊木马攻击有一定的抵御作用，即使某用户进程被特洛伊木马非法控制，也不能随意扩散机密信息。但同时也存在一些不足之处，主要表现在两个方面。
- (1)完整性方面控制不够。
- (2)应用领域比较窄。由于强制访问控制的规则制定严格并且缺乏弹性，所以无法适应于复杂的现实环境。

4.4 基于角色的访问控制

- 以上两种访问控制模型都存在的不足是将主体和客体直接绑定在一起，授权时需要对每对（主体、客体）指定访问许可，这样存在的问题是当主体和客体达到较高的数量级之后，授权工作将非常困难。

- 20世纪90年代以来，随着对在线的多用户、多系统研究的不断深入，角色的概念逐渐形成，并产生了以角色为中心的访问控制模型(role-based access control，RBAC)，被广泛应用在各种计算机系统中。

4.4.1 基本概念

- 基于角色访问控制(RBAC)策略基于系统中用户所具有的角色，以及这些角色所规定的访问规则来对访问进行控制。基于角色的访问控制(RBAC)是实施面向企业安全策略的一种有效的访问控制方式。其基本思想是，对系统操作的各种权限不是直接授予具体的用户，而是在用户集合与权限集合之间建立一个角色集合，每一种角色对应一组相应的权限。
- 一旦用户被分配了适当的角色后，该用户就拥有此角色的所有操作权限。这样做的好处是，不必在每次创建用户时都进行分配权限的操作，只要分配用户相应的角色即可，而且角色的权限变更比用户的权限变更要少得多，这样将简化用户的权限管理，减少系统的开销。

RBAC的核心思想

- RBAC的核心思想是将权限与角色联系起来，在系统中根据应用的需要为不同的工作岗位创建相应的角色。同时根据用户职责指派合适的角色，用户通过所指派的角色获得相应的权限，实现对文件的访问。

- 也就是说，传统的访问控制是直接主体即传统的访问控制直接将访问主体（发出访问操作，有存取要求的主动方）和客体（被调用的程序或欲存取的数据访问）相联系，而RBAC在中间加入角色，**通过角色沟通主体和客体**。

RBAC的基本思想

- RBAC的基本思想是：授权给用户的访问权限，通常由用户在一个组织中担当的角色来确定。RBAC中许可被授权给角色，角色被授权给用户，用户不直接与许可关联。RBAC对访问权限的授权由管理员统一管理，用户不能自主地将访问权限传给他人。
- RBAC在主体和权限之间增加了一个中间桥梁——角色。权限被授予角色，而管理员通过指定用户为特定角色来为用户授权，从而大大简化了授权管理，具有强大的可操作性和可管理性。角色可以根据组织中的不同工作创建，然后根据用户的责任和资格分配焦色，用户可以轻松地进行角色转换。而随着新应用和新系统的增加，角色可以分配更多的权限，也可以根据需要撤销相应的权限。

RBAC核心模型包含的5个基本静态集合

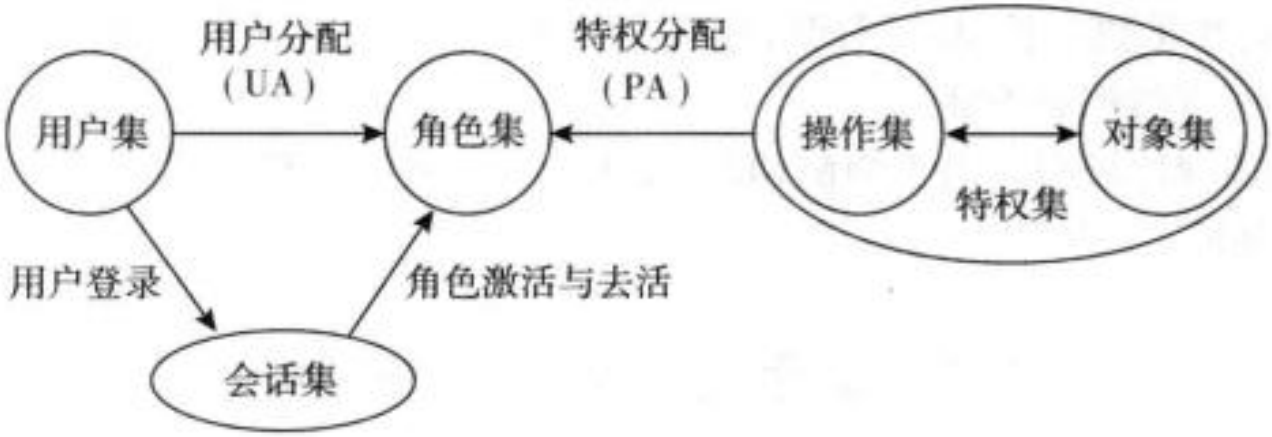


图4-2 会话集

- 用户集包括系统中可以执行操作的用户，是主动的实体；对象集是系统中被动的实体，包含系统需要保护的信息；操作集是定义在对象上的一组操作，对象上的一组操作构成了一个特权；角色则是RBAC模型的核心，通过用户分配(UA)和特权分配(PA)使用户与特权关联起来。

- RBAC属于策略中立型的存取控制模型，既可以实现自主存取控制策略，又可以实现强制存取控制策略。它可以有效缓解传统安全管理处理瓶颈问题，被认为是一种普遍适用的访问控制模型，尤其适用于大型组织的有效的访问控制机制。
- 基于角色访问控制根据对合法的访问者进行角色认证来确定访问者在系统中对哪类信息有什么样的访问权限。系统只问用户是什么角色，而不用管用户是谁。基于角色的访问控制策略的基本思想是：在用户与权限之间引入角色的概念，利用角色来实现用户和权限的逻辑隔离，即用户与角色相关联，角色与权限相关联，通过给用户分配角色而使用户获得相应的权限。

基于角色访问控制策略中的关键概念

- **(1)主体(subject)：**可以对其他实体实施操作的主动实体。通常是系统用户或代理用户行为的进程，也被称为访问的发起者(initiator)。
- **(2)客体(object)：**接受其他实体动作的被动实体。通常是可以识别的系统资源，如文件。一个实体在某一时刻是主体而在另一时刻又可能成为客体，这取决于该实体是动作的执行者还是承受者。
- **(3)用户(user)：**试图使用系统的人员。每个用户都有一个唯一的用户标识(UID)，当注册进入系统时，用户要提供其UID，系统进行用户身份认证以确证用户身份。一个可以独立访问计算机系统中的数据或数据表示的其他资源的主体指使用计算机系统的人，或指计算机里的进程、账号等。一个用户与系统对话框的实例称为一个会话(session)。

用户大致可以分为以下几类

- **①普通的用户：**其访问操作受到一定限制，由系统管理员分配。
- **②特殊的用户：**系统管理员，具有最高级别的特权，可以访问任何资源，并具有任何类型的访问操作能力。该用户有以下权限：
  - 可以创建和删除计算机上的用户账户。
  - 可以为计算机上的其他的用户账户创建账户密码。
  - 可以更改其他用户的账户名、图片、密码和账户类型。
- **③作废的用户：**曾经有权使用系统，但当前被系统拒绝的用户。
- **④作审计的用户：**负责整个安全系统范围内的安全控制与资源使用情况的审计。

基于角色访问控制策略中的关键概念

- **(4)角色(role)：是系统中一组职责和权限的集合**。角色的划分涉及组织内部的岗位职责和安全策略的综合考虑。角色的创建方式与用户账户的常规创建方式相同。角色具有起始目录、组指定和口令等，权限配置文件和授权可为角色提供管理权能。
- 用户承担某种角色时，此角色的属性将取代所有用户属性。

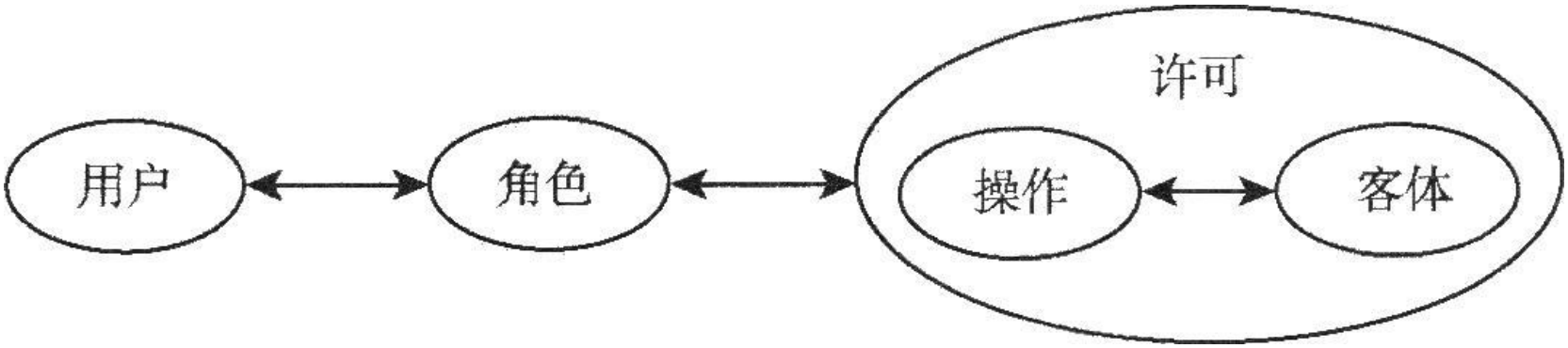


图4-3 用户、角色、许可的关系

基于角色访问控制策略中的关键概念

- **(5)权限(permission)：**在受系统保护的客体上执行某一操作许可。在客体上能够执行的操作常与系统的类型有关，表示对系统中的客体进行特定模式的访问操作，与实现的机制密切相关。权限指对计算机某些资源、某些操作的许可，也可以理解为是用户在系统中进行任何一个操作，对资源的任何一种访问都会受到系统的限制，用户对特定的资源进行特定操作的许可称为权限。
- **(6)用户角色分配(user-to-role assignment)：**为用户分配一定的角色，即建立用户与角色的多对多关系。系统管理员通过为用户分配角色，取消用户的某个角色等操作管理角色分配。

基于角色访问控制策略中的关键概念

- **(7)角色权限分配(permission-to-role assignment)：**为角色分配一组访问权限，即建立角色与访问权限的多对多关系。这样通过角色把用户与访问权限联系起来。用户具有其所属诸角色的访问权限的总和。

- **(8)会话(session)：**用户是一个静态的概念，而会话是一个动态的概念。一次会话是用户的一次活跃进程，它代表用户与系统进行交互。用户与会话是一对多关系，一个用户可同时打开多个会话。

基于角色访问控制策略中的关键概念

- **(9)活跃角色集(active roleset，ARS)**：一个会话构成一个用户到多个角色的映射，即会话激活了用户授权角色集的某个子集，使子集成为活跃角色集。ARS决定了本地会话的许可权限集。一次会话是用户的一个活跃进程，它代表用户与系统交互。用户与会话是一对多关系，一个用户可同时打开多个会话。
- **(10)保护域(protection domain)**：保护域是一系列权限的集合，描述一个主体在给定时间可能执行的所有操作的集合。它的表示形式为：（权限1，权限2，…，权限n），在一个特定会话期间控制用户行为的保护域叫做活动保护域(active protection domain)。

4.4.2 授权管理

- **授权是指可以授予角色或用户的独立权限，授权在用户中应用程序级别强制执行策略。**
- 尽管可以将授权直接指定给角色或用户，但最佳做法是将授权归到权限配置文件中。然后，将该权限配置文件添加到某个角色中，再将该角色指定给用户，包含单词delegate或assign的授权允许用户或角色将安全属性指定给其他账户。
- 要阻止升级特权，就不要将assign授权指定给某个账户。delegate授权仅允许委托方将其拥有的安全属性指定给其他账户。遵循RBAC的应用程序可以先检查用户的授权，然后再授予其访问该应用程序或应用程序中特定操作的权限。

1．用户间的授权关系

- 依据角色指派关系，运行系统中的用户自身可以对角色进行管理。通常，角色指派的权力都在系统中具有管理责任的用户手中。在增强RBAC中，授权是与安全相关的功能或者命令相关联的文本字符串。授权提供了一种机制，以便为用户授予相应的权限以执行某些特权操作，并对不同类别的用户提供不同的功能级别。通常先将授权分配给角色，然后再将角色分配给用户。当执行由某个授权所管理的命令时，仅在调用该命令的用户具有所需授权的情况下，才能够为其授予访问权限。因此，可以将授权看成解锁一个或者多个命令的钥匙。
- 特权在内核中强制执行安全策略，授权与特权之间的差别与强制执行安全策略的级别有关。如果没有适当的特权，内核可能会阻止进程执行特权操作，用户可能无法使用特权应用程序或在特权应用程序内执行与安全相关的操作。

2．授权策略

- 授权策略规定何人在何种情况下能访问何种目标。权限管理策略由策略管理机构制定、管理，由SOA签发，用于权限的委派和分配。
- 访问控制和授权策略展示了一个机构在信息安全和授权方面的顶层控制。策略应当包括一个机构如何将它的人员和资源进行分类组织，这种组织方式必须考虑到具体应用的实际运行环境，如资源的敏感性、人员权限的明确划分以及必须和相应人员层次相匹配的管理层次等因素。
- 授权策略包括：应用系统中的所有用户和资源信息以及用户和信息的组织管理方式；用户和资源之间的权限关系；保证安全的管理授权约束；保证系统安全的其他约束。

授权策略的基本内容

- (1)委托策略：规定谁可以将什么权力委托给谁。
- (2)SOA策略：它规定受信任分配角色的SOA，并允许进行角色分配的分散管理。SOA策略明确了哪些SOA是可信的，并限定了SOA委派和分配角色的方式和范围。
- (3)角色指派策略：它规定何种SOA可以分配何种角色至何种主体，而不论角色代表是否进行或角色会被指派多久。
- (4)动作策略：它规定目标支持的动作，以及与动作一起通过的参数。
- (5)用户策略：它规定主体域，规定了需要赋予权限的用户的范围及分类方法，主要目的是确定和识别与权限相关的具体的用户（组），它可以是一组用户，也可以是具体一个用户。
- (6)目标访问策略：它规定在何种条件下，何种角色有许可对何种目标进行何种操作。
- (7)角色继承策略：它规定不同的角色及其相互的继承关系。

4.4.3 RBAC的优势

- **(1)简化权限管理**。不同于DAC和MAC需要将权限明确地授权给用户或主体，RBAC只需将权限分配给特定角色，然后将该角色授权给相应用户。如果一组权限需要改变，只需修改分配给角色的权限，所有被授予该角色的全部用户的安全域将自动地反映对角色所做的修改。
- **(2)灵活表达和实现组织的安全策略**，接近日常生活。
- **(3)安全性高**。该策略可以有效实现最小权限管理。如用户虽然被赋予了高级身份但是可以限制其权限，只有必要时才能拥有特权。
- **(4)实用性强**。该策略为系统管理员提供了一种比较抽象的、与企业通常的业务管理相类似的访问控制层次。

4.5 基于属性的访问控制

- 云计算、物联网等新型计算范型为我们提供了便捷的数据共享、高效计算等服务，极大提高了数据的处理效率，提升了计算和存储资源的利用能力。同时，这些计算范型存储并融合了大量具有“所有权”特征的数据，需要提供可靠的保护。基于属性的访问控制有效解决了具有大规模、强动态性和强隐私性特点的新型计算环境下的细粒度访问控制问题，为云计算、物联网等新型计算环境提供了理想的访问控制策略。
- ABAC将主体和客体属性作为决策的基本依据，灵活地利用资源访问者所具有的属性决定是否授予其访问权限，能够很好地将策略管理和权限判定分离。同时，由于属性是主体和客体内在固有的，无需手工分配，使得ABAC管理上相对简单。

4.5.1 ABAC模型

- 属性是ABAC的核心概念，ABAC中的属性可以通过一个**四元组(S，O，P，E)**进行描述。

① **S表示主体(subject)属性**，即主动发起访问请求的所有实体具有的属性，如年龄、姓名、职业等；

② **O表示客体(object)属性**，即系统中可被访问的资源具有的属性，如文档、图片、音频或视频等数据资源；

③ **P表示权限(permission)属性**，即对客体资源的各类操作，如文件或数据库等的读、写、新建、删除等操作；

④ **E表示环境(environment)属性**，即访问控制过程发生时的环境信息，如用户发起访问的时间、系统所处的地理或网络位置、是否有对同一信息的并发访问等信息，这一属性独立于访问主体和被访问资源。

ABAC系统按的两个阶段：准备阶段和执行阶段

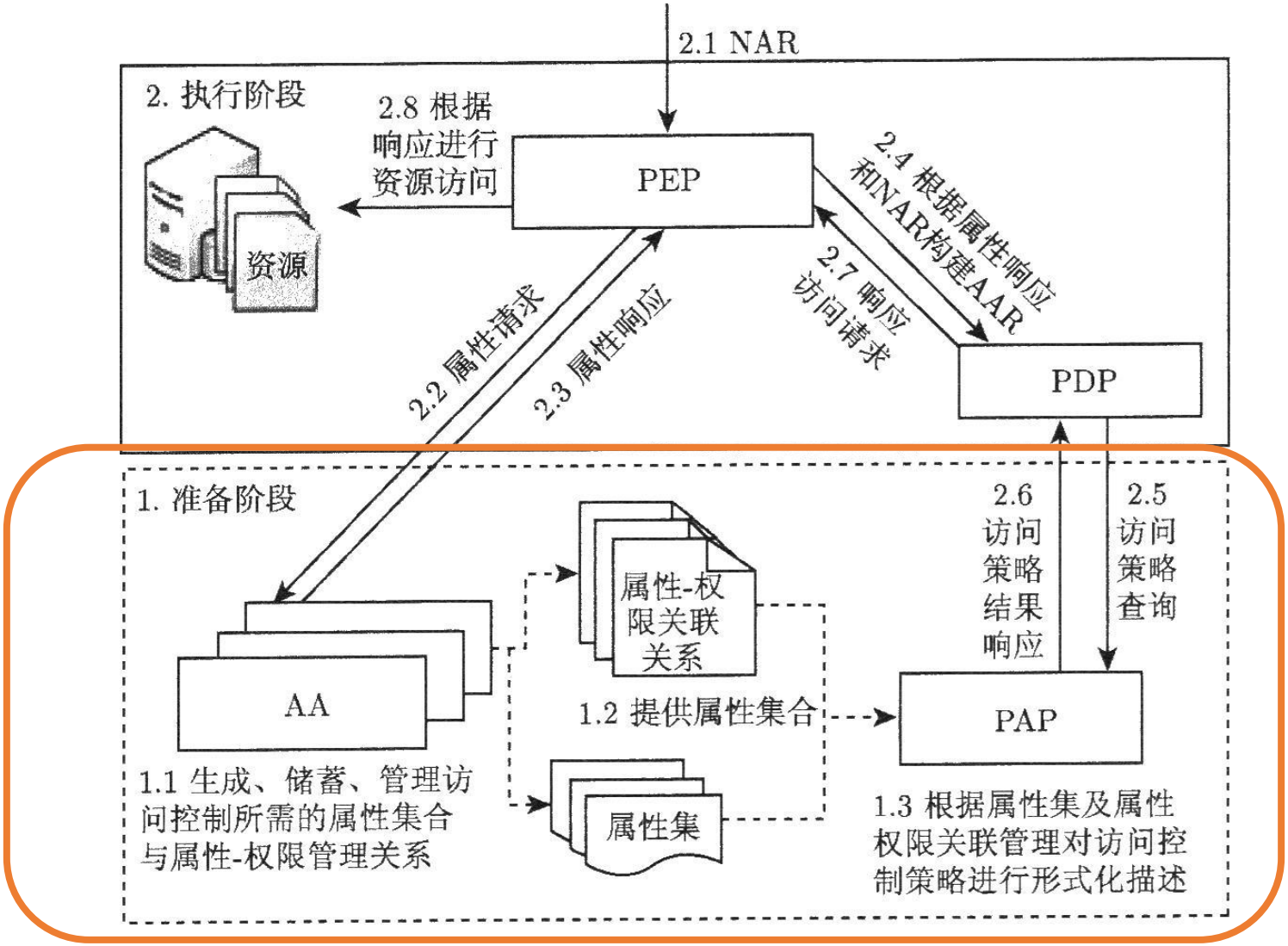


图4-4 ABAC机制框架示意图

ABAC系统按的两个阶段：准备阶段和执行阶段

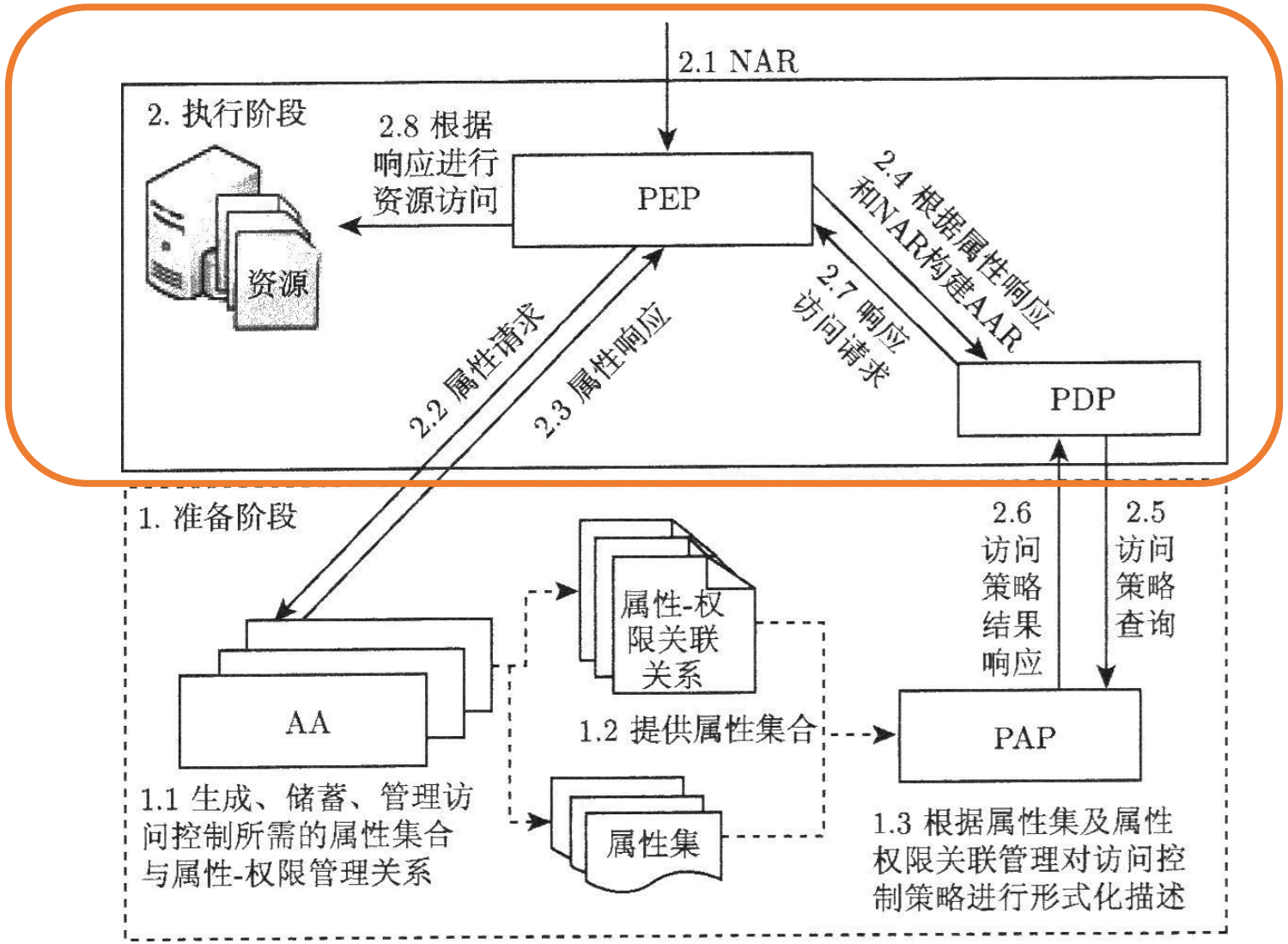


图4-4 ABAC机制框架示意图

4.5.2 ABE基本概念

- 虽然传统的ABAC有效控制了用户对资源的访问操作，但其仅实现了对用户访问过程的控制。而随着云计算和物联网等新型计算环境产生以及存储的敏感隐私信息日益增多，由信息泄露所导致的安全威胁也不断增加。因此，为了最大限度的保护数据的隐私安全，实现更细粒度的访问控制，研究者们提出了**基于属性的加密机制(attribute-based encryption，ABE)**。
- ABE实现了对数据机密性的访问控制，其采用非对称密码机制并利用属性作为加解密的关键要素，将属性同密文和用户密钥相结合。当用户属性与密文属性的公共集合满足加密和访问结构所规定的参数时才能解密相应数据。

ABE机制也可分为准备阶段和执行阶段

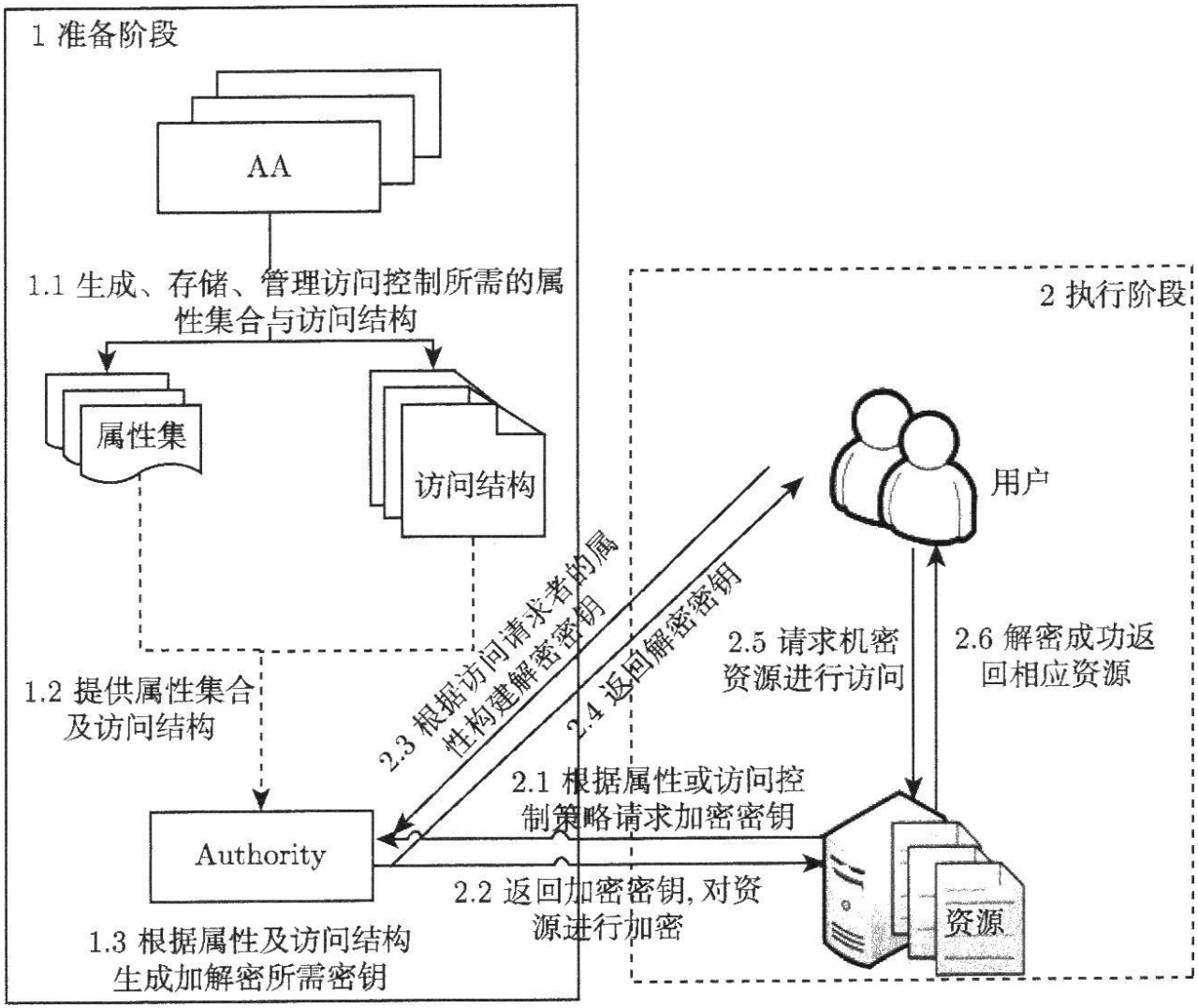


图4-5 ABE机制框架示意图

KP-ABE和CP-ABE对比

		KP-ABE	CP-ABE
Setup(λ, U)	输入	安全参数	安全参数
		属性空间大小	属性空间大小
		用户空间大小	用户空间大小
	输出	公钥参数 PK	公钥参数 PK
Encrypt(PK, M, A)	输入	主密钥 MK	主密钥 MK
		公钥参数 PK	公钥参数 PK
		信息 M	信息 M
		属性集合 γ	访问结构 A
	输出	加密数据CT	加密数据CT
KeyGen(MK, S)	输入	主密钥 MK	主密钥 MK
		访问结构 A	属性集合 γ
		公钥参数 PK	NA
	输出	解密密钥 D	用户私钥 SK
Decrypt(PK, CT, SK)	输入	公钥参数 PK	公钥参数 PK
		加密数据 CT	加密数据 CT
		解密密钥 D	用户私钥 SK
	输出	原始数据 M	原始数据 M

4.6.1 PMI基础

- **PMI(privilege management infrastructure)**即权限管理基础设施或授权管理基础设施，是属性证书、属性权威、属性证书库等部件的集合体，用来实现权限和证书的产生、管理、存储、分发和撤销等。
- AA(attribute authority)，即属性权威，是用来生成并签发属性证书的机构。它负责管理属性证书的整个生命周期。
- **AC(attribute certificate)**，即属性证书，对于一个实体的权限的表示是由一个进行了数字签名的数据结构来提供的，这种数据结构称为属性证书，由属性权威签发并管理。

- 建立在PKI基础之上的PMI，以向用户和应用程序提供权限管理和授权服务为目标，主要负责向业务应用系统提供授权管理服务，提供用户身份向应用授权的映射功能，实现与实际应用处理模式相对应的、与具体应用系统开发和管理无关的访问控制机制，并能极大地简化应用中访问控制和权限管理系统的开发与维护，并减少管理成本和复杂性。
- PMI以资源管理为核心，对资源的访问控制权交由授权机构统一处理，即由资源的所有者来进行访问控制。而公钥基础设施PKI(public key infrastructure)以公开密钥技术为基础，以数据的机密性、完整性和无可抵赖性为安全目的而构建的认证、授权、加密等硬件、软件的综合设施。

4.6.2 PKI和PMI的关系

- **PMI主要进行授权管理，证明这个用户有什么权限，即“你能做什么”；**
- **PKI主要进行身份鉴别，证明用户身份，即“你是谁”。**
- PKI和PMI的关系类似于签证和护照的关系。护照是身份证明，唯一标识个人信息，只有持有护照才能证明你是一个合法的人；签证具有属性类别，持有哪一类别的签证才能在该国家进行哪一类的活动。
- 授权的信息可以放在身份证书扩展项中或者直接使用属性证书表示，但是将授权信息放在身份证书中是很不方便的。独立的授权信息更易用。

表4-2 PKI与PMI术语对比

概念	PKI实体	PMI实体
证书	公钥证书	属性证书
证书签发者	认证权威	属性权威
证书用户	主体	持有者
证书绑定	主体名和公钥绑定	持有者名和权限绑定
撤销	证书撤销列表CRL	属性证书撤销列表ACRL
信任的根	根CA/信任锚	权威源SOA
从属机构	子CA	属性权威AA

- 公钥证书是对用户名称和用户的公钥进行绑定，而属性证书是将用户名称与一个或多个权限属性进行绑定。在这个方面，公钥证书可被看做特殊的属性证书。
- 数字签名公钥证书的实体被称为CA，签名属性证书的实体被称为AA。
- PKI信任源有时被称为根CA，而PMI信任源被称为SOA。
- CA可以有它们信任的次级CA，次级CA可以代理鉴别和认证，SOA可以将它们的权利授予次级AA。
- 如果用户需要废除自己的签字密钥，则CA将签发证书撤销列表。与之相似地，如果用户需要废除授权，AA将签发一个属性证书撤销列表。

4.6.3 PMI授权管理模式、体系及模型

1. PMI技术的授权管理模式

- 授权服务体系主要是为网络空间提供用户操作授权的管理，即在虚拟网络空间中的用户角色与最终应用系统中用户的操作权限之间建立一种映射关系。授权服务体系一般需要与信任服务体系协同工作，才能完成从特定用户的现实空间身份到特定应用系统中的具体操作权限之间的转换。目前建立授权服务体系的关键技术主要是授权管理基础设施PMI技术。PMI以资源管理为核心，对资源的访问控制权交由授权机构统一处理，即由资源的所有者来进行访问控制。
- PMI是一个由属性证书、属性权威、属性证书库等部件构成的综合系统，用来实现权限和证书的产生、管理、储存、分发和撤销等功能。

1．PMI技术的授权管理模式

- PMI使用属性证书表示和容纳权限信息，通过管理证书的生命周期实现对权限生命周期的管理。属性证书申请、签发、注销、验证的流程对应着权限申请、发放、撤销、和验证的过程，而且使用属性证书进行权限管理方式使得权限的管理不必依赖某个具体的应用，并有利于权限的安全分布式应用。
- **PMI技术通过数字证书机制来管理用户的授权信息，并将授权管理功能从传统的应用系统中分离出来，以独立服务的方式面向应用系统提供授权管理服务。**
- 由于数字证书机制提供了授权信息的安全保护功能，因此，作为用户授权信息存放载体的属性证书同样可以通过公开方式对外发布，由于属性证书并不提供用户身份的鉴别功能，因此属性证书中将不包含用户的公钥信息。

2. PMI系统的体系架构

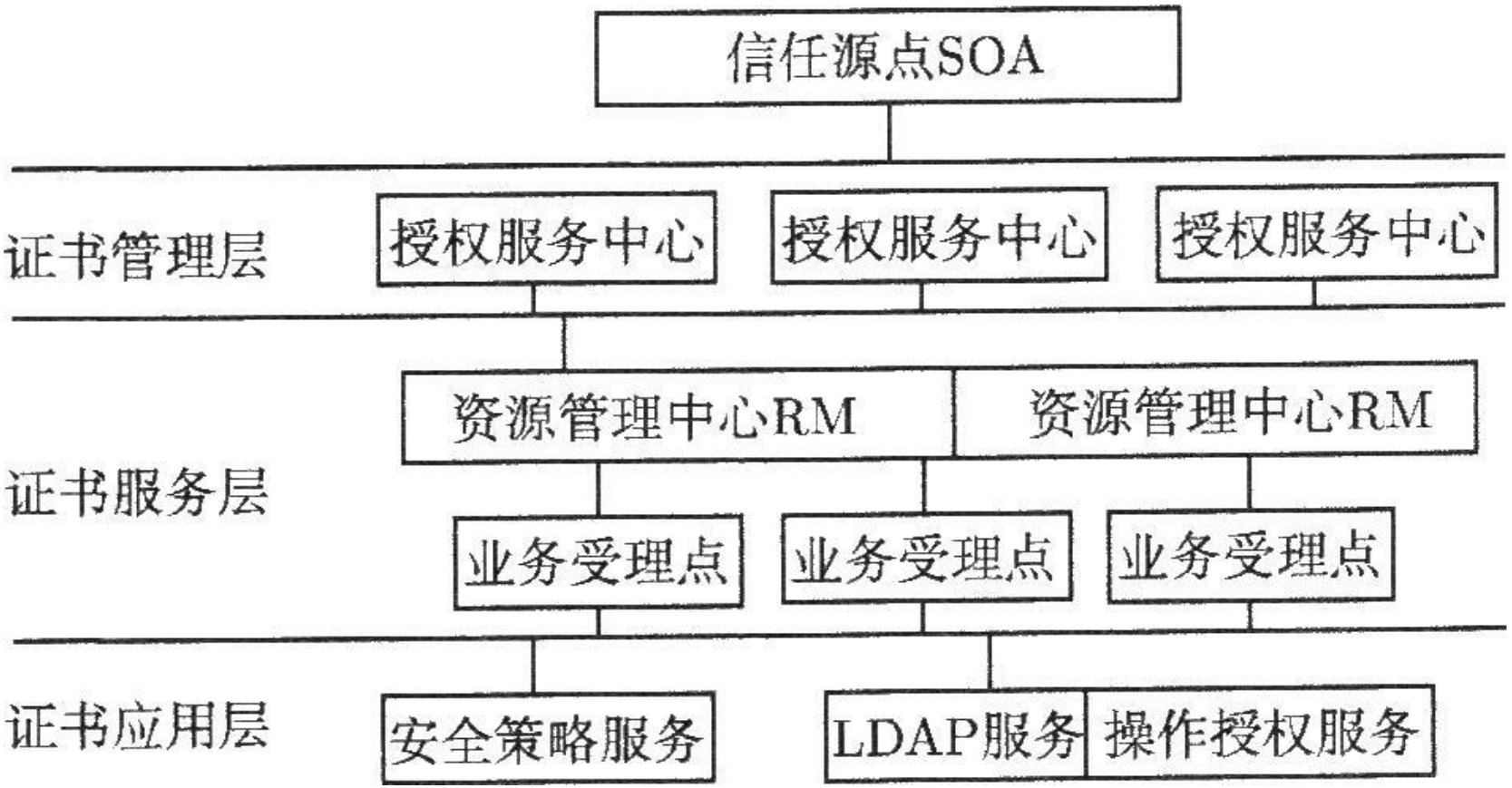


图4-6 授权管理体系框图

3．PMI模型

- 绝大多数的访问控制应用都能抽象成一般的权限管理模型，包括三个实体：对象(object)、权限声称者(privilege asserter)和权限验证者(privilege verifier)。
- 对象可以是被保护的资源，例如，在一个访问控制应用中，受保护资源就是对象。
- 权限声明者也就是访问者，是持有特定权限并声明其权限具有特定使用内容的实体。
- 权限验证者对访问动作进行验证和决策，是制定决策的实体，决定被声明的权限对于使用内容来说是否充分。
- 权限验证者根据以下4个条件决定访问通过 / 失败：
- ①权限声明者的权限；②适当的权限策略；③当前环境变量；④对象的敏感度。

4.6.4 PMI基础设施的结构和应用模型

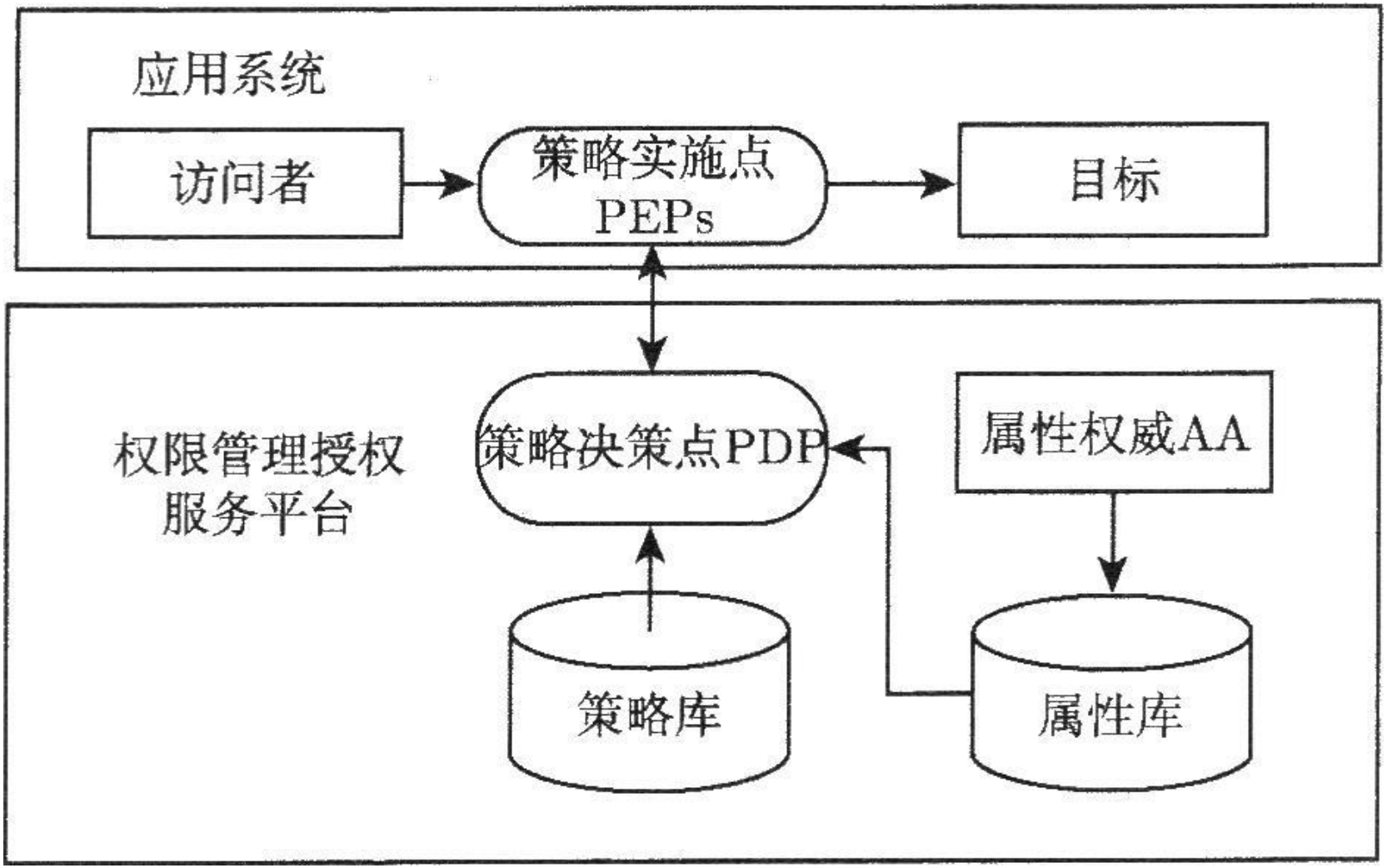


图4-7 PMI基础设施的结构和应用模型

4.6.5 属性权威与属性证书

- 1. 属性权威
- 属性权威即AA，用来生成并签发属性证书的机构，它负责管理属性证书的整个生命周期。一个属性权威AA的基本组成如图4-8所示。
- 属性权威主要包括AC签发、AA受理、AA管理、数据库服务器、目录服务器，其中数据库服务器不是必须的。

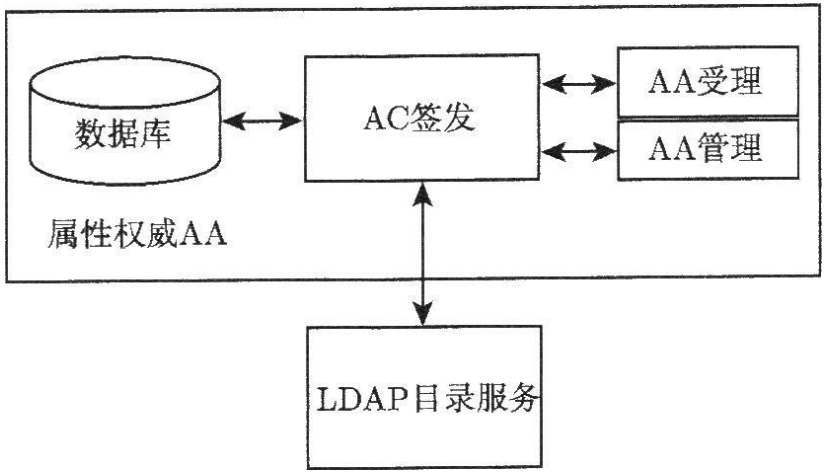


图4-8 属性权威AA结构图

2．属性证书

- PMI使用属性证书来表示和容纳权限信息，对权限生命周期的管理是通过管理证书的生命周期实现的。

1)属性证书的定义和格式

- 属性证书：对于一个实体的权限的绑定是由一个进行了数字签名的数据结构来提供的，这种数据结构称为属性证书，由属性权威签发并管理。一些应用使用属性证书来提供基于角色的访问控制。

表4-3 属性证书的格式

版本号	属性证书的版本号
持有者	属性证书持有者信息，一般是持有者公钥证书DN和公钥证书颁发者DN组合
颁发者	属性证书的颁发者信息，一般是颁发者公钥证书DN
签名算法	属性证书使用的签名算法
序列号	属性证书的序列号
有效期	属性证书的生效和失效日期
属性	属性证书持有者的属性信息
扩展项	定义诸如撤销信息、证书颁发者密码标识符、ACRL分布点、角色定义证书标识符及其它信息
签名信息	属性证书签发者对属性证书的签名

2)属性证书的特点

- 公钥证书将一个身份标识和公钥绑定，属性证书将一个标识和一个角色、权限或者属性绑定（通过数字签名）。和公钥证书一样，属性证书能被分发和存储或缓存在非安全的分布式环境中，不可伪造，防篡改。属性证书具有以下特点。
- (1)分立的发行机构。
- (2)存储介质。属性证书可以发放给用户，由用户存储在磁盘上或者USBkey上，或者委托给系统进行统一存储和管理，而不必分发给用户。
- (3)本地发放。
- (4)基于其拥有属性来决定其对某一资源或服务是否拥有访问权。
- (5)属性证书可以设置成短时效的
- (6)属性证书与身份证书的相互关联。

3)属性证书的申请与发布

- 权限的获得和发布是通过属性证书的申请和发布实现的。属性证书的申请在PMI中，用户申请属性证书应向业务应用部门进行申请，然后由相关部门提供所需材料给属性权威AA，属性权威AA在审核材料的合法性之后，签发属性证书，并将属性证书存储到证书数据库中。上述过程也支持在线的属性证书申请。

4)属性证书的分发

- 属性证书的分发有两种模式。第一种模式是“推”模式，当用户在要求访问资源时，由用户自己直接提供其属性证书，即用户将自己的属性证书“推”给资源服务管理器。
- 第二种模式是“拉”模式，是业务应用授权机构分发属性证书到目录服务系统。当用户需要用到属性证书时，由服务器从属性证书发放者——属性权威或存储证书的目录服务系统“拉”回属性证书。

5)属性证书的撤销

- 取消权限的过程通过撤销属性证书实现。在具体的应用系统中，属性证书的注销和公钥证书的注销相同，即通过属性证书撤销列表ACRL来公布已经被注销的属性证书。在使用属性证书的时候，通过查询属性证书撤销列表ACRL来判断所使用的属性证书是否已经被注销。

6)属性证书的基本验证过程

- (1)验证用户的身份证书，应用系统验证证书的有效性，包括查看证书是否经过CA的正确签名。
- (2) 根据身份信息获取属性证书。
- (3)验证属性证书。
- (4)最后应用程序检查属性证书中的内容，来确定是否允许此用户存取其所需的资源及服务。

第4章 作业

- 作业
  - ✓1.什么是授权？什么是访问控制？
  - ✓3.自主访问控制的基本思想是什么？
  - ✓4.强制访问控制的主要特点是什么？
  - ✓8.简述PMI和PKI的关系。
- 实践（自己研究，不考核）
  - 调研Linux和Windows系统的访问控制和授权的区别。
  - 在大规模的Web应用中，选用哪一种访问控制模型较合适？