分类号_____ 密　　级_____

UDC _____ 学校代码 10497

武汉理工大学

学　位　论　文

题　目 跨域访问控制研究

英　文

题　目 The Research of Cross-Domain Access Control

研究生姓名 林 智 鑫

指导教师 姓　名 龙毅宏 职称 教　授 学位 博　士

单位名称 信息工程学院 邮 编 430070

申请学位级别 硕　士 学科专业名称 通信与信息系统

论文提交日期 2008 年 4 月 论文答辩日期 2008 年 5 月

学位授予单位 武汉理工大学 学位授予日期_____

答辩委员会主席 刘泉 评阅人 周祖德，刘泉

2008 年 4 月

摘 要

随着互联网技术和分布式计算能力的不断进步，处在不同授权体系下的系统对共享资源的需求越来越强烈，用户经常需要跨越不同的授权体系来进行资源访问，因此系统之间的安全互操作就变得极其重要。在互操作过程中，系统既是服务的提供者，也是服务的使用者，因此系统在保护本地资源的同时，必须遵守其它系统的安全规则。但是在分布式环境下进行跨域访问时，由于各域中采用的访问控制机制和安全策略都各不一样，使得安全管理面临更为复杂的情况。因此，如何在确保系统安全的同时，为外域用户提供访问控制成为至关重要的问题。

本文的研究重点是为了解决在跨域授权中遇到的这些难题。为此本文采用了三种基于映射的跨域授权理论模型：首先在依据前人研究的基础上本文应用一种基于角色映射的跨域授权模型，该模型只能解决 RBAC 访问控制系统间的安全互操作；本文接着提出了一种基于用户角色（组）交叉映射的跨域授权模型，该模型解决了 ACL 与 RBAC 访问控制系统之间的安全互操作；最后本文在上述两种模型的基础上进行抽象，并首次提出了一种基于属性映射的跨域授权理论模型，它可适用于多种基于主题属性授权的访问控制机之间的安全互操作。

在本研究的基础上，研究开发了一套支持跨域授权的访问控制系统，首次提出引入跨域授权中介系统、结合上述跨域授权理论模型以及全局身份管理来实现跨域访问控制。本跨域访问控制系统基于 J2EE 平台，它包括基本访问控制系统和跨域授权中介系统。基本访问控制系统为本域内的资源和服务提供了集中的授权管理和授权服务。跨域授权中介系统是实现跨域访问控制的桥梁，它保存有全局用户和全局属性并负责为各授权域提供属性映射服务。通过跨域授权中介系统可以支持两种方式的跨域授权，一是将各外域用户的属性映射为目标域可以识别的属性，然后进行授权；二是通过跨域授权中介系统中定义的全局用户和全局属性进行跨域授权。

关键字：访问控制，跨域授权，属性映射

Abstract

With the improvement of internet and the distributed computing technology, the requirements for shared resources among systems under different domains are more and more strong. People often have to access to resources on the other domain. So the interoperations among systems become more and more important. In the process of the interoperation, these systems are not only service providers but also service consumers of other systems. Therefore, the systems should protect one's own resources and comply with the security rules of other's systems. But in the distributed environment, the security problem can get magnified because of heterogeneous security policy, different authorization mechanism. Therefore, how to configure appropriate access control for supporting interoperation and ensuring system security has become the very important technology.

This thesis is focused on solving such problems. We adopt three theoretical models to solve cross-domain authorization. In the first we based on the basis of previous studies and apply a role mapping model, but it is only applicable to solve the interoperation between the RBAC authorization systems, the second one is based on role and group cross mapping, this model solve the security interoperation between ACL and RBAC authorization systems, the last one is abstracted from the above models, it is based on attribute mapping, it can applicable to such authorization mechanisms which is based on subject's attributes.

On the basis of this research, we study and develop an access control system which supports cross-aomain authorization. This thesis first introduces the cross-domain mediator system, combined with the above theoretical models and global identity management to realize the cross-domain access control. This system is based on the J2EE platform; it includes Fundamental access control system and cross-domain mediator system. The fundamental access control system provides centralized authorization management and authorization service. The cross-domain mediator system is the bridge of cross-domain authorization; it not only stores global

users and global attributes but also provides mapping service for member authorization domain. Based on mediator system this system supports cross-domain authorization via two ways: one is mapping the foreign domain user's attribute to the recognizable attribute, the other is authorizing by the global user and attribute.

Key words: access control, cross-domain authorization, attribute mapping

目 录

第1章 绪 论

1.1 研究背景和意义

随着互联网技术的不断进步和分布式计算能力的不断增强，使得大范围的资源共享成为可能。但是资源的共享必然会带来一些如秘密信息外泄等安全问题，网络安全的形势也变得日益严峻，因此，在我们进行信息共享和资源访问的同时，必须兼顾到系统的安全性，而访问控制正是一种通过约束用户访问行为而达到对敏感信息进行隔离的目的的安全服务与机制，它决定用户及代表一定用户利益的程序能做什么，以及做到什么程度，从而使计算机系统在合法范围内使用。

访问控制技术发展到今天，目前存在有 MAC（强制访问控制机制），DAC（自主访问控制机制），RBAC（基于角色的访问控制机制），ABAC（基于属性的访问控制机制）等多种类型。由于存在这么多访问控制机制，并且每种访问控制机制的实现方法也各不相同，随着资源共享需求的不断扩大，处在不同授权机制下的企业或机构间越来越多的业务上的往来，经常需要通过网络来交换机密的资料或数据，所以建立可以跨域不同授权机制的授权管理系统就变得十分重要。目前各个企业、机构大部分都已经建立了自己的授权管理系统，如何在此基础上实现不同授权管理系统的整合，最终实现跨授权域的授权管理成为目前的热点和难点问题。

目前授权管理类的产品在国外相对比较成熟，一些大厂商都有自己相应的产品（如 IBM WebSeal[1]、Microsoft Authorization Manager[2, 3]、SUN Access Manager[4]、BEA Weblogic[5]等等）；而在国内，在这方面虽然已有一些软件产品，但同国外同类产品相比，无论在功能还是性能方面都有相当的差距，并且在现有的授权管理类产品中还没有一款可以真正解决跨域的授权问题。本文所研究的跨域授权系统正是基于这种背景下提出的。

1.2 国内外研究现状

访问控制技术是国际标准化组织 ISO 提出的五大安全服务之一，是保证信

息安全的常用技术。计算机系统的广泛应用，使得访问控制技术的研究一直是国内外研究的热点。

早在20世纪60年代，在某些早期分时计算机系统中，安全问题就已引起了人们的关注，但是，直到20世纪70年代，计算机安全学才开始快速发展。1983年提交的可信计算机系统评估准则橙皮书（TCSEC）[6]，在TCSEC中描述了两类传统的访问控制策略，即自主式访问控制（DAC）和强制式访问控制（MAC）。1992年，美国国家标准与技术研究所(NIST)的David Ferraiolo和Rick Kuhn在综合前人研究的基础上，率先提出了基于角色的访问控制（Role Based Access Control，RBAC）模型框架[7]，随后在1996年，Ravi Sandhu等人提出了著名的RBAC96模型[8]，将传统的RBAC模型拆分成四种嵌套的模型并给出了形式化定义，成为以后RBAC研究和应用中的经典模型。目前，RBAC已经变成最流行的访问控制解决方案，并且由它衍生出来很多新的访问控制模型，如基于属性的访问控制机制（Attribute Based Access Control，ABAC）和基于任务和角色的访问控制模型（T-RBAC）[9]等等。

随着信息技术的不断发展，进入21世纪，随着分布式系统的大量涌现，在分布式领域内对访问控制的要求也日益显现出其重要的地位，相对于集中式系统的访问控制，分布式系统的访问控制更为复杂，涉及多个方面的内容，包括在分布式系统中域是高度自治的，但是彼此之间又有着千丝万缕的联系，需要安全的进行协作；由于分布式系统的可伸缩性，需要信任管理技术的支持等。针对域间的安全互操作问题，出现了一些较为完善的分布式授权方案，比较典型的有组授权服务（Community Authorization Services，CAS）[10]、虚拟组织管理服务（Virtual Organization Management Service，VOMS）[11]、公钥基础设施环境下基于证书的授权策略（certificate-based authorization policy in a PKI environment）[12]、隐私和角色管理基础架构标准(PrivilEge and Role Management Infrastructure Standards，PERMIS)[13]等。虽然这些方案在分布式环境中引入了角色的概念，但是它们并不是严格的RBAC实现机制并且只能适用于特定的分布式系统。2000年美国伊利诺斯州大学的Kapadia等人提出了一个基于角色转换的域间互操作模型IRBAC 2000（Interoperability Role Based Access Control）[14]，它通过在两个采用RBAC角色层次模型的域之间提供角色的转换关系，实现用户的跨域访问。接着多种基于RBAC的通过角色映射实现跨域授权的模型被提出来，如Liang Chen，Jason Crampton提出的基于最小权限原则的域间角

色映射机制[15]，本模型建议在用户进行跨域访问时限制用户所能启用的角色数量。T.L.Prasanna Venkatesan 等人提出的基于等级划分的跨域角色映射和授权模型[16]，该模型引入等级机制，一个等级代表一个全局角色，跨域授权基于全局角色来进行。在我国清华大学徐云和肖田元在基于角色映射的跨平台授权研究一文中提出了一种基于角色映射的跨平台授权方式[17]，它包含了联邦角色（全局角色）和局部角色映射两种机制，并且引入了约束条件，是一套较为完整的跨域授权方案。目前比较流行并且得到了很大关注的是OASIS XACML标准[18]，XACML 是基于 XML 的访问控制策略语言，它提供了可移植的，统一的方法描述这些访问控制策略。OASIS SAML 为 XACML 提供了标准格式在系统之间交换权限信息[19]。SAML 规范和 XACML 规范之间这种关系可以为分布式系统之间带来灵活的授权和访问控制机制。目前 XACML 结合 SAML 的跨域授权方案还在研究和改善中。本文将对现有的跨域授权模型进行分析比较，提出一种基于属性映射的跨域授权模型。

1.3 课题研究内容

本项目是科技部 “国家科技支撑计划” （2006BAH02A03），“现代服务业共性服务集成化技术”的子项目“授权管理与访问控制系统”。该共性服务集成化技术将为各种应用服务提供所需的、共性的服务功能，如身份鉴别、访问控制、计费等。整个共性服务集成化技术的开发将有助于推动现代服务业应用的更快、更便捷、更安全的部署，具有广阔的应用前景。本系统为基于 Web 的、面向公众提供服务的应用系统提供了可跨域授权的访问控制功能。

本文作者的主要任务是对跨域授权管理系统中存在和需要解决的问题进行分析和研究。对它的结构层次进行研究，进而对整个方案的总体架构进行分析和设计，并参与跨域授权管理系统的具体实现。

具体工作内容主要包括：

（1）查阅了大量有关网络安全和访问控制方面的文献，认真深入的研究了当前存在的多种访问控制机制。

（2）对现有的授权管理类产品和前人研究的跨域授权模型进行了仔细的学习和分析。

（3）在实际项目中，参与了跨域授权管理系统的系统分析、详细设计和实现。

本文主要的创新点是：

（1） 提出了一套用户组和用户角色交叉映射的跨域授权模型。

（2） 提出一套基于属性映射的跨域授权模型。

（3） 引入跨域中介系统来实现属性映射模型。

在项目实施期间，作者对大型系统的开发实现有了一定的认识，同时也对访问控制的实现有了更深的理解，但还有许多工作需要进一步完成和改进。

1.4 本文的组织结构

本文共分为六章，组织结构如下：

第二章访问控制和跨域授权技术，首先介绍了几种常见的访问控制基本概念及其优缺点，最后介绍了跨域授权的背景和目前存在的一些跨域授权模型。

第三章基于映射的跨域授权模型，本章在上章基础上提出了三种基于映射的跨域授权理论模型。

第四章基本访问控制系统，研究实现了一种支持多种访问控制机制的集中授权管理系统。

第五章跨域访问控制，本章在基本访问控制系统中引入跨域授权中介系统，对第三章的理论模型进行具体实现。

第六章总结与展望，对本论文的工作进行了总结，并提出了今后的改进方向。

第 2 章　访问控制与跨域授权技术

国际标准化组织在其制定的有关开放系统互连参考模型的安全体系结构（ISO7498-2）中定义了五项标准安全服务：

（1）认证服务（Authentication）：为一个实体身份的合法性提供保证；

（2）访问控制服务（Access Control）：限制访问主体对访问客体的访问权限，保护资源使其不能被非法使用或操纵；

（3）机密性服务（Confidentiality）：防止信息原有内容泄漏给未经授权的实体；

（4）完整性服务（Integrity）：确认信息在传输过程中没有被篡改、删除或替换；

（5）抗抵赖服务（Non-Repudiation）：防止参与安全通信的某一方在发出（或收到）信息后否认该曾经发出（或收到）该信息。

访问控制服务作为安全服务中一个重要的组成部分，在安全体系结构中具有不可替代的作用。它可以限制系统的活动者（包括用户和软代理）对系统关键资源的访问，防止非法活动者的入侵和合法活动者的不慎操作引起对安全计算机系统的破坏。本章将针对访问控制的发展历程，主要对自主访问控制（DAC）、强制访问控制（MAC）、基于角色的访问控制（RBAC）三种典型模型进行了分析和研究，并对各种模型的优缺点进行了分析和比较，指出了其各自的应用环境。

2.1 传统的访问控制模型

访问控制是信息安全保障机制的核心内容，它是实现数据保密性和完整性机制的主要手段。访问控制是为了限制访问主体（或称为发起者，是一个主动的实体；如用户、进程、服务等），对访问客体（需要保护的资源）的访问权限，从而使计算机系统在合法范围内使用；访问控制机制决定用户及代表一定用户利益的程序能做什么，及做到什么程度[20]。

由于网络传输的需要，访问控制的研究方发展很快，有许多访问控制模型被提出来。建立规范的访问控制模型，是实现严格访问控制策略所必须的。20

世纪 70 年代，Harrison, Ruzzo 和 Ullman 提出了 HRU 模型。接着，Jones 等人在 1976 年提出了 Take-Grant 模型。具有里程碑意义的是美国国防部于 1983 年提交的可信计算机系统评估准则橙皮书（TCSEC），在 TCSEC 中描述了两类传统的访问控制策略，即自主式访问控制（DAC）和强制式访问控制（MAC ）。在 DAC 中，客体的安全由客体的属主或具有指定特权的用户来制定，主要是规定别的用户以怎样的方式访问该客体；而 MAC 对于客体的安全，则由系统规则确定一个主体能否访问一个客体。从这两类访问控制策略出发，形成了一系列的访问控制模型。

2.1.1 自主访问控制

自主访问控制是根据主体或主体所属的组来限制主体对客体的访问权限，是一种最为普遍的访问控制手段[21]。自主访问控制的主体可以按自己的意愿决定哪些用户可以访问他的资源，将他所拥有的权限直接或间接地传递给其他主体，即主体有自主的决定权，一个主体可以有选择地与其它主体共享他的资源，主体全权掌握客体的访问权限，故称为自主型。通常 DAC 通过授权列表（或访问控制列表）来限定哪些主体针对哪些客体可以执行什么操作。如此将可以非常灵活地对策略进行调整。由于其易用性与可扩展性，自主访问控制机制经常被用于商业系统。在 DAC 系统中，一般利用访问控制矩阵或访问控制列表来实现访问权限的控制。

（1）访问控制矩阵（Access Control Matrix）[22]

1971 年 Lampson 提出了访问控制矩阵的概念，它的所有访问控制策略都可以转化成为访问控制矩阵的形式。下图 2.1 所示为访问控制矩阵的例子。

客体 / 主体	Resource1	Resource2	Resource3
User1	R,W	R	W
User2	W	R,W	R
User3	R	W	R,W

图 2.1 访问控制矩阵

访问矩阵是一种简易的概念表示法，它用矩阵（i，j）中的值来代表主体 i 拥有访问客体 j 的权利。

（2）访问控制列表（Access Control List）

访问控制列表是 DAC 系统中经常采用的另一种安全机制。ACL 是以客体为中心建立的访问权限表，对每个客体单独指定对其有访问权限的主体，还可以将有相同权限的主体分组，授予组的访问权限。如针对客体分析 Resource1，用户 Userl 有“读/写”权限，用户 User2 有“写”权限，用户组 Group 中的所有成员都有“读”权限，利用 ACL 我们可以进行如下定义：Resource1：（User1：Read，Write），（User2：Write），（Group1：Read）。ACL 策略表述直观，易于理解，适用于被用户数较少而这些用户的授权状态相对比较稳定的情况。对于用户数量多、客体对象复杂的系统中，当组织内的人员发生人事变动时，管理员需要修改用户对所有资源的访问权限，这使得访问控制的授权管理变得十分复杂，并且容易出错，造成权限的失控状态。

在自主访问控制中，访问控制是基于主体的，主体可以自主地把自己所拥有客体的访问权限授予其它主体或者从其它主体收回所授予的权限，这使得访问控制具有较高的灵活性，DAC 是多用户环境下常用的一种访问控制技术，在 Unix 类操作系统中被普遍采用。

DAC 的自主性给用户提供了灵活的访问控制方式，但同时带来的是系统的安全性相对较低，信息在传递的过程中可能会被修改或破坏。用户可以自由地将自己的访问权限授予他人，系统对此无法控制。如用户 A 对信息资源 R 具有读和写的访问权限，用户 B 对信息资源 R 只有读的权限。用户 A 将自己的访问权限传递给用户 B，从而使得用户 B 也具备了对 R 的写权限。访问权限的传递容易产生安全漏洞，造成信息数据的泄漏或错误修改，这样以来，系统的安全性不能得到充分的保证。

2.1.2 强制访问控制

MAC 用来保护系统确定的对象，对此对象用户不能进行更改。也就是说，系统独立于用户行为强制执行访问控制，用户不能改变他们的安全级别或对象的安全属性。这样的访问控制规则通常对数据和用户按照安全等级划分标签，访问控制机制通过比较安全标签来确定的授予还是拒绝用户对资源的访问。强制访问控制进行了很强的等级划分，所以经常用于军事用途。

在强制访问控制系统中，所有主体（用户，进程）和客体（文件，数据）都被分配了安全标签，安全标签标识一个安全等级。主体 （用户，进程）被分

配一个安全等级，客体（文件，数据）也被分配一个安全等级，访问控制执行时对主体和客体的安全级别进行比较。MAC 将访问控制关系分为“上读/下写”和“下读/上写”并通过安全标签实现单向信息流通模式。典型的 MAC 模型主要有：

（1）BLP 模型

1973 年，David Bell 和 Len LaPadula 提出了第一个正式的安全模型[23]，该模型基于强制访问控制系统，以敏感度来划分资源的安全级别。将数据划分为多安全级别与敏感度的系统称之为多级安全系统。BLP 保密模型基于两种规则来保障数据的机秘度与敏感度：上读（NRU），主体不可读安全级别高于它的数据；下写（NWD），主体不可写安全级别低于它的数据。BLP 模型很好地描述了信息的“机密性”，但是忽略了“完整性”的要求。

（2）Biba 模型

上世纪 70 年代，Ken Biba 提出了 Biba 访问控制模型，该模型对数据提供了分级别的完整性保证，类似于 BLP 保密性模型，BIBA 模型也使用强制访问控制系统。BIBA 模型基于两种规则来保障数据的完整性的保密性：下读(NRU)属性，主体不能读取安全级别低于它的数据；上写(NWD) 属性，主体不能写入安全级别高于它的数据。Biba 模型是一个与 BLP 相对立的模型，强调对信息完整性的保护，却没有考虑机密性要求。

强制访问控制通过控制信息的单向流动来保证信息数据的安全性和完整性，对主体和客体进行了很强的等级划分，只有符合安全级别要求的用户才可以操作数据，从而提供了更高级别的安全性。这种严格的权限管理和高度的保密特性使得强制访问控制在军事领域得到了广泛的应用。但 MAC 的保密性使得其在授权方面缺乏灵活性，有时会限制高密级用户向非敏感客体写数据的合理请求，从而降低了系统的可用性。此外，访问级别的划分不够细致，缺乏同级别间的控制机制。因此，MAC 系统中高保密性的优点同时带来了低灵活性的缺陷。

2.2 基于角色的访问控制

1992 年 David Ferraiolo 和 Richard Kuhn 在美国国家标准技术研究所(NIST)举办的第十五届美国计算机安全大会(NCSC)上提交了一篇文章，提出了 RBAC 的基本描述和结构，通称为 RBAC92，其基本思想是：用户和权限通过角色相

关联，管理员创建“角色”，并为角色分配权限，用户通过饰演不同的角色从而获得角色所拥有的权限。但是当时并没有引起巨大的关注，直到 1996 年 Sandhu 提出了 RBAC96 标准。

2.2.1 RBAC96 模型

Sandhu 的 RBAC96 将 RBAC 标准进行了规范化，RBAC96 模型由四个层次模型组成，分别为 RBAC0、RBAC1、RBAC2 和 RBAC3，各模型间的关系如图 2.2 所示[24]。

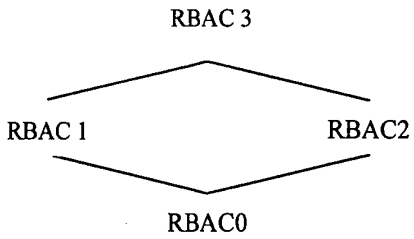

图 2.2　RBAC96 模型成员间关系

RBAC0 是最基本的模型，即核心 RBAC，它规定了 RBAC 系统的最小需求，包括五个基本要素集：用户（USERS）、角色（ROLES）、对象（OBS）、操作（OPS）和权限（PRMS）。RBAC 在用户和权限之间引入了角色的概念，安全管理员根据实际需要定义各种角色，并设置和角色相对应的访问权限，而用户根据其职责被指派为不同的角色。这样，访问权限和角色相关联，角色再与用户关联，从而实现了用户与访问权限的逻辑分离。核心 RBAC 中也引入了会话（Session）的概念，会话是用户与激活的角色集合之间的映射。

RBAC1 在 RBAC0 的基础上增加了角色层次，引入了类似于面向对象中的角色“继承”的概念，使得角色之间派生出层次关系（Role Hierarchies, RH），这样就可以把一些较为基本的权限分配给层次较为低的角色(父角色)，把附加的权限分配给较高层次的角色(子角色或后代角色)，再利用继承的方式继承那些基本权限，免去了每个角色都要重复一些基本的权限分配行为，从而减轻了维护上的负担。角色的层次化反映了一个组织的授权和责任的自然方式。考虑到角色继承的可能违反“最小权限原则”而导致权限滥用，故又将 RBAC 1 分成两类：一般继承关系和受限继承关系。一般继承关系仅要求角色继承关系是

一个绝对偏序关系，允许角色间的多重继承。而受限继承关系则进一步要求角色继承关系是一个树结构。

由于一个用户可以拥有多个角色，而角色所拥有的权限之间可能产生“利益冲突”（Conflict of Interests）或“互斥”（Mutually Exclusive），因此需要引入一种机制来克服这个问题。RBAC2 是在另一个方面对 RBAC0 进行扩展，它添加了“约束”（Constraints）的限制条件，成为带约束条件的 RBAC （Constrained RBAC），从而满足了“最小权限原则”和“权责分离原则”。RBAC 建议标准中引入两种职责分离机制：静态职责分离和动态职责分离。

（1） 静态权责分离（Static Separation of Duties，SSD）

又称为“强互斥”，在为用户分配角色的时候，不能将有利益冲突的角色分配给同一个用户。

（2） 动态权责分离（Dynamic Separation of Duties，DSD)

又称为“弱互斥”，允许将有冲突的角色分配给同一用户，但是该用户不能在一个会话中同时激活这些角色。

RBAC3 是同时包含 RBAC0，RBAC 1 和 RBAC2 全部特性的 RBAC 模型，既提供层次结构又具有约束条件。

下面是完整的 RBAC96 模型，如图 2.3 所示。

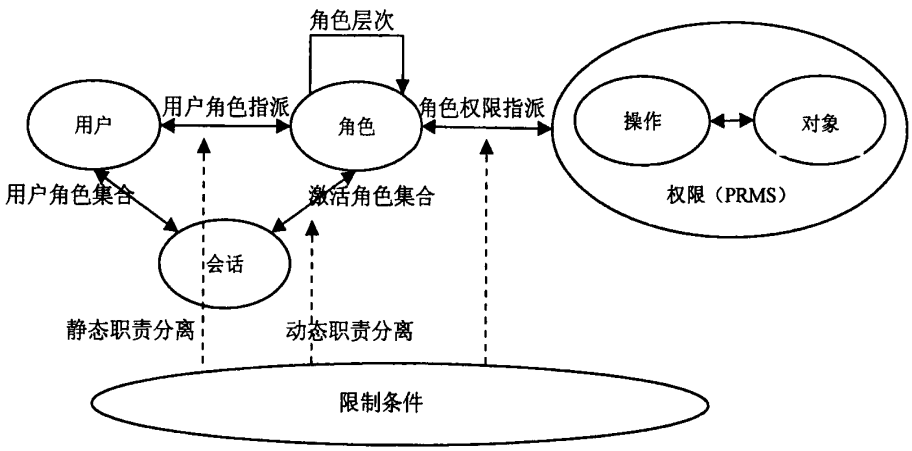

图 2.3 RBAC96 模型

上述模型中的基本概念如下：

（1） 用户(User)：与系统交互的主体，一般情况下指人，也可为计算机进程。

（2） 角色(Role)：是一个组织机构或任务环境中的工作职责，代表特定的权限，反映用户的职责。角色与用户之间、权限之间都是多对多的关系。

（3） 权限(Permission)：表示用户对客体资源进行访问的操作许可，可细化为操作和对象。对象是客体资源，操作是访问方式。如对文件(客体)进行读、写的操作(访问方式)。

（4） 会话(Session)：表示用户对角色的激活过程，是一个动态概念。一次会话代表用户与系统交互的一个过程，用户与会话之间是一对多的关系。用户只有通过会话激活角色，才能获得角色所对应的权限。一次会话可以同时激活多个角色。

（5） 角色层次 (Role Hierarchies)：反映了角色的继承关系。角色层次可以用角色树来表示，上层角色可以继承下层角色的权限。在 RBAC96 模型中，继承为全部继承，上层角色继承下层角色的所有权限。

（6） 用户角色指派 (User Assignment)：给用户分配角色的过程。

（7） 角色权限指派 (Permission Assignment)：给角色分配权限的过程。

（8） 限制(Constrains)：是整个模型上的一系列约束条件，是个抽象的概念。限制有很多种，典型的限制主要有静态职责分离、动态职责分离、角色互斥、角色基数限制等。一一介绍如下。

1）静态职责分离(Static Separation of Duty，SSD)：在用户角色指派阶段的限制，与角色激活无关。

2）动态职责分离（Dynamic Separation of Duty）：在角色激活阶段的限制。

3）角色互斥（Mutually Exclusive Roles）：根据职责分离，分为静态角色互斥和动态角色互斥。静态角色互斥是指的在用户角色指派阶段，不能赋予同一个用户的两个角色就为静态互斥角色；动态角色互斥是指在角色激活阶段，一个用户不能同时激活的两个角色就为动态互斥角色。

4）角色基数限制（Role Cardinality Constrains）：一个用户可以被赋予的最大角色数或一个角色可以被分配的最大用户数。根据职责分离，同样有静态基数限制和动态基数限制。

2.2.2 RBAC 模型的优点

与传统的自主访问控制和强制访问控制相比，基于角色访问控制(RBAC)模型是一个策略中立的模型，可以通过适当的配置，使 RBAC 能够执行自主访

问控制和强制访问控制的策略，RBAC 主要具有以下优点：

（1） 简化授权管理。传统的访问控制实现方法通常是直接为每个用户赋予一组权限，将用户和访问权限直接联系起来，若人员流动量比较大的话，将会给权限管理带来巨大的麻烦。而在 RBAC 中，角色作为一个桥梁，沟通于用户和资源之间。对用户的访问授权转变为对角色的授权，然后再将用户与特定的角色相关联，一旦 RBAC 系统建立，主要的管理工作即授权或取消用户的角色。

（2） 适应范围广泛的安全策略。RBAC 与具体的安全策略无关，所以能适应范围广泛的安全策略，包括前面讨论过的 DAC 和 MAC，因而系统管理员能够按照不同的安全策略需要定义角色，适应不同应用领域的安全需要。

（3） 支持最小权限的原则。将用户的权限限制在其完成某项任务的必须具有的权限范围之内，可有限减少越权操作的情况，管理的负担减轻，系统的安全性却随之提高。

（4） 支持职责分离原则。当角色之间存在着互斥关系时，例如“会计”角色与“出纳”角色不能同一用户拥有，在角色间定义约束关系可以实现用户间的制约。

2.3 基于属性的访问控制

如今分布式结构越来越普遍，大大增强了不同平台之间的互操作性。不过它也给传统的安全模型带来了许多新的安全挑战。最主要的是访问控制模型，开放环境的分布式访问控制对传统的访问控制模型和机制提出了严峻的挑战：(1)对传统的基于资源请求者身份的访问控制的挑战，即基于身份的方式如何认证和授权；(2)对访问控制策略集中管理模式的挑战，即跨不同安全系统的通信中，如何解决互操作问题。依赖主体属性授权，是为陌生双方建立信任关系的一种有效方法，并基于此提出了基于属性的访问控制。在 ABAC 中，利用相关实体(如主体、资源、环境)的属性而不只是身份作为授权的基础。这种基于属性的方法尤其适合于开放和分布式系统中的授权和访问控制。

基于属性的访问控制（Attribute Based Access Control，ABAC）[25, 26, 27]是以参与决策的相关实体的属性（而不仅仅是身份）为基础进行授权决策的一种访问控制机制。属性是指某实体相关的一些特性。这里的实体主要有三类：主体、客体和环境。主体属性包括主体的身份、角色、年龄、邮政编码、IP 地址、雇

员职位、已验证的 PKI 证书等；客体属性包括客体的身份、位置(URL)、大小、值等，服务的参数就是一种典型的客体属性；环境属性是与事务处理关联的属性,它通常与身份无关，但适用于授权决策，如时间、日期、系统状态、安全级别等。利用主体、客体和环境的属性来定义授权，既简化了管理，又增加了灵活性。

ABAC 的基本观点是不直接在主体和客体之间定义授权，而是利用他们的属性作为授权决策的基础。基本的 ABAC 授权模型如图 2.4 所示。

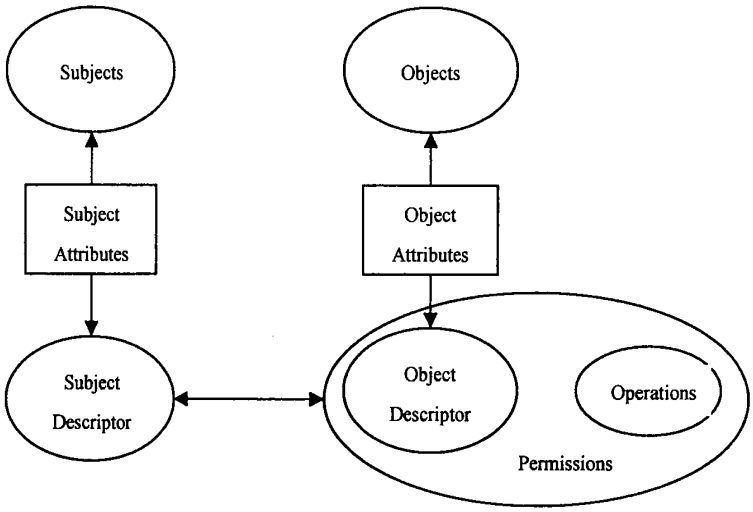

图 2.4 ABAC 授权模型

在 ABAC 基本授权模型中，主体和客体(资源)均用一组属性和对应的属性值表示。权限(permission)由客体描述器(Object Descriptor)和操作(operation)组成,授权是在主体描述器(Subject Descriptor)与客体描述器之间定义的，Subject Descriptor 或 Object Descriptor 由关于主体或客体的属性条件组成,如“年龄>30”等。除了主体属性和客体属性外，在许多情况下，访问还需要受到一定环境和系统状态的约束。例如，只有在工作日或在特定地点才能访问某个数字资源。系统负荷很重时，只有高级用户才能得到它提供的服务。授权系统需要检查目前环境和系统的状态,这种决策因素被称为“环境上下文(环境属性)”。

2.4 几种访问控制模型间的比较

本章详细介绍了四种访问控制模型，发现了各种模型的优缺点及适用的应用环境。在此基础上将这四种模型进行比较：

（1）自主访问控制模型的优点是具有较高的灵活性，某一主体可以将自己对某客体的访问许可传递给其他主体，实现了授权自主；但访问权限的传递使得其安全级别较低，并且，系统管理员难以确定哪些用户对哪些资源具有访问权限，不利于实现统一管理，适合于保密性不高、用户数量较少的应用环境中，在 Unix 类操作系统中被普遍采用；

（2）强制访问控制模型的优点是管理比较集中，根据严格的安全级别定义来实现权限管理，安全性较高；但其授权策略缺乏灵活性，不适用于主体或客体经常更新的应用环境，一般用于保密级别要求较高的军事领域；

（3）基于角色的访问控制模型由于引入了角色，使得用户和权限得到了逻辑分离，授权变得简单而灵活，访问控制框架有了较强的扩展性；其缺点是不能处理工作流中的数据信息，被动的访问控制特点使其不能满足企业动态业务流程处理的需要；

（4）基于属性的访问控制模型包含了基于角色的访问控制模型，ABAC 更适合于开放和分布式系统中的授权和访问控制。

2.5 跨域授权

2.5.1 背景介绍

随着网格，Web 服务等分布式技术的大量使用，分布式结构的安全问题也越来越引起关注，由于在分布式环境下强调资源共享，这也给传统的访问控制技术带来了新的需求，也就是跨域访问控制。跨域访问控制，即当来自其他域的用户访问本域的系统资源时对其进行的访问控制。这里所说的“域”是指授权管理作用域有效的范围，即授权域，它的范围可以是单个系统，一个部门的所有系统，或者一个组织机构的所有系统。此处的“授权域”同另一个相似而且密切相关的“身份域”概念并不相同。身份域是指身份管理作用和有效的范围，如身份标识和身份鉴别作用的范围，与之相对应的跨域身份鉴别要解决的是一个域中的用户身份在另一个域中被识别、获得认可、接受的问题。也就是说，跨域身份鉴别与跨域访问控制要解决的问题不一样，一个是解决你是谁的问题，一个是解决你能干什么的问题。在有些情况下，身份域和授权域的范围是一样的，如采用用户名/口令进行身份鉴别时，二者通常是一样的；在有些情况，二者是不一样的，如当不同组织机构采用一个共同的第三方 CA 签发的数

字证书进行身份标识时，它们的身份域是相同的，而授权域各自不同。

跨域的访问控制机制要求进行跨授权域边界的授权，然而不同授权域策略异构性使问题变得较为复杂。每个授权域中都有自己的用户系统和访问控制策略，在进行跨域访问的时候，由于资源提供者和使用者处在不同的授权域中，这给域间安全互操作带来了困难。基于这个前提，我们的跨域授权模型必须满足如下几点要求[28]：

（1）自治控制。自治控制是指在各个授权域中，访问控制策略的制定不受其它授权域访问控制策略的干涉或影响。

（2）透明性。对于使用资源的终端用户来说，他们并不关心资源是怎样共享过来的，以及使用这些资源时权限验证和交互的细节，用户经过单一登录进入系统后对于可用的资源不必再重复的输入密码进行身份验证等。

（3）异构性。不同授权域的安全策略是异构的，如角色、权限命名差异和角色层次异构和职责分离约束差异等，系统应该能够解决异构性。

（4）可扩展性。当一个域或组织加入或退出协同组织时，不对其它域的访问控制策略产生影响，而其自身的访问控制策略也不因加入或退出而改变。

（5）动态性。域的加入和离去是动态变化的，另外域中用户也是动态变化的。

（6）松散耦合。跨域授权系统与跨域的身份鉴别间松散耦合，跨域授权系统和授权管理系统可以与多种跨域身份鉴别模块进行整合。

2.5.2 跨域授权模型

目前，虽然分布式系统的安全性引起人们很大的关注，但是更多的研究工作集中在了跨域的身份鉴别上面，而在跨域授权这部分的研究相对还比较少。跨域授权必须建立在跨域身份鉴别的基础上，目前的跨域身份鉴别主要是采用身份联合(Identity Federation)的方式，如基于 OASIS SAML[29]的自由联盟 ID-FF（Identity Federation Framework）[30]和 IBM，Microsoft 等倡导的 WS-Federation[31]，WS-Trust[32]系列中都是基于用户帐户之间的联合，即在两个异构系统中通过用户 ID 之间的映射，来实现跨域身份鉴别。然后在基于这种跨域身份鉴别的基础上对跨域用户进行授权。基于用户 ID 联合的跨域授权可能存在下列问题：若发生帐户修改、删除等操作时，系统间的映射关系就必须重设；若用户数量庞大，就必须维护大量的映射数据；身份鉴别与访问控制紧耦合。

虽然我们也可以引入假名映射、临时假名映射或基于属性的身份映射机制[33]，也不能完全解决上述问题并且身份鉴别和访问控制模块紧耦合在一起。我们希望能够像 RBAC 一样引入一个中介（角色），既减少了映射关系也能够降低身份鉴别与访问控制之间的耦合。

由于 RBAC 的灵活性和良好的管理性，目前有许多跨域授权的研究都是基于角色映射来实现。2000 年美国伊利诺斯州大学开发的一种跨域之间用户互操作的 RBAC 模型（IRBAC，Interoperability Role Based Access Control），该工作第一次将 RBAC 模型成功地扩展到多个安全域中，通过域间角色关联，实现了跨域之间的 RBAC 系统。该模型在各个安全域中加入了一个代理（Agent）模块，负责对本域用户进行身份认证，接收本域用户对该域的应用请求，协商域－域之间的事务处理规则，接收并按规则执行其他域代理的请求。IRBAC 2000 的结构图如图 2.5 所示：

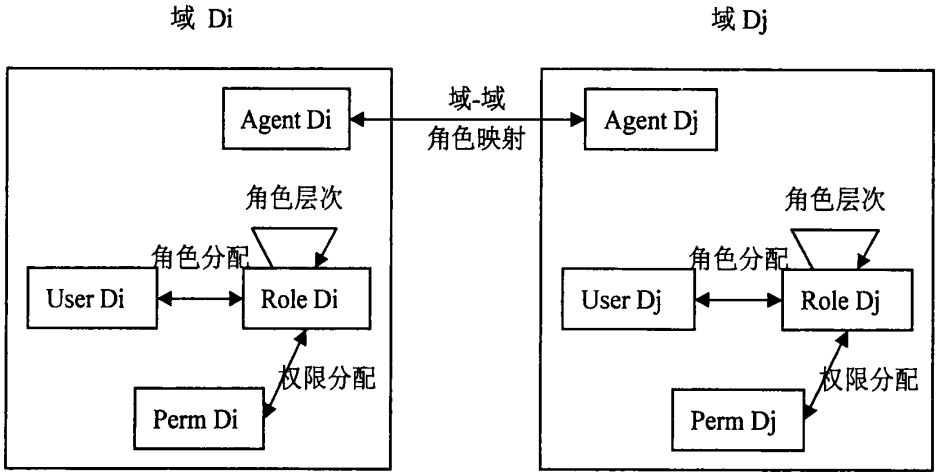

图 2.5 IRBAC 2000 结构图

IRBAC 2000 的具体步骤为：首先由两域间的代理模块（Agent）协商好角色映射关系，若代理模块发现有访问来自外域，代理模块执行相应的角色映射条件，然后将映射后的本域角色分配给外域用户，用户依据这些角色对安全应用进行授权访问。该模型提出了基于角色的映射在一定程度上实现了跨域授权，但也存在一些安全问题[34]，如 IRBAC2000 模型中的动态角色转换可能会违背职责分离，这是在角色映射中必须考虑的安全问题之一；删除角色，所有与该角色相关的角色关联都必须重新设置。

T.L.Prasanna Venkatesan 等人提出的基于等级划分的跨域角色映射和授权模型，该模型引入等级机制，所有的域通过一个 Ranking Server 进行联合。各联合域将本地角色映射成一个全局等级，用户在进行跨域访问时，将本地角色转化为相应的全局等级，而后应用系统根据相应的全局等级对用户进行授权。本模型引入了一个类似全局角色的等级机制，但是只能提供一种粒度相对较粗的访问控制机制。

目前在国内，跨域授权的研究也得到越来越多的关注，清华大学徐云和肖田元在基于角色映射的跨平台授权研究一文中提出了一种基于角色映射的跨平台授权方式，本模型具有上述两种模型的优点，并且引入联邦角色和角色约束等新的概念。该模型中根据是否引入联邦角色集作为中介，以及是否对用户直接进行角色指派，将角色映射分为了以下四种。

（1）平台角色和联邦角色映射。将各授权域中的角色与联邦角色进行映射，用户访问时，首先将本地角色映射为联邦角色，然后再将联邦角色映射为所要访问的授权域角色。

（2）两个平台角色之间映射。不引入联邦角色集，直接进行两个授权域角色集间的映射。

（3）跨平台用户角色映射（指派）。由外授权域为本域用户分配跨域角色。

（4）用户联邦角色映射（指派）。直接赋予用户联邦角色集，无需经过角色映射。

角色映射是本跨平台授权模型的核心，模型中还引入了角色映射约束来保证角色映射的合理性，降低系统的运营风险是安全保障的关键问题。角色映射的约束分为合同约束，静态约束和动态约束三种。

合同约束实际上是一种合同关系,主要用于日后产生纠纷时作为协调或仲裁的证据。静态约束在角色映射时起到资格过滤的作用，如某些重要角色信息可以设为对普通集成平台不可视。动态约束的范围是随着跨平台业务过程不断发生变化的属性，如平台或用户的消费历史和付费信誉，这些属性体现了平台在集成系统中的共享能力，可以通过一定算法量化为平台声望或用户声望。动态约束是集成系统对平台和用户动态监管的重要部分。本模型不仅提出了一套较为完善的基于角色映射跨域访问控制模型，并且包含了角色间的约束条件。

2.6 本章小结

本章首先介绍了目前流行的几种访问控制模型，并对这些模型的优缺点进行了比较，最后介绍了目前存在的一些跨域授权模型，为下文的分布式环境下的访问控制模型研究奠定了坚实的理论基础。由于不同的访问控制模型适应的环境各不相同，在现代企业信息系统中随着需求的不同，每种访问控制模型都有其利用价值。在后文中，我们将实现一种支持跨域授权和多种访问控制机制的授权管理系统。

第 3 章　基于映射的跨域授权模型

当前在跨域授权中面临的最大问题是：不同授权域间采用不同的访问控制机制，现在流行的访问控制机制有：ACL，RBAC 等等，即使采用的是相同的访问控制机制但是由于策略格式以及权限的组织方式也大不相同，比如同为采用 RBAC 的两个授权域其角色名称与角色层次也各不一样。为解决上述问题本章首先在总结前人研究的基础上给出一个基于角色映射的跨域授权模型，本模型解决了 RBAC 访问控制系统间的跨域授权问题。由于 RBAC 和 ACL 是目前最常见的访问控制机制，本章接着提出了一套基于用户角色（组）交叉映射的跨域授权模型，本模型在用户角色跟用户组之间搭建映射桥梁解决了基于 ACL 访问控制系统与基于 RBAC 访问控制系统之间的跨域授权问题。最后在对上述两种模型的基础上我们首次提出了一种基于属性映射的跨域授权模型，该模型可适用于多种基于主体属性授权（如 MAC，DAC，RBAC，ABAC 等等）的访问控制系统之间进行跨域授权。

3.1 角色映射模型

由于 RBAC 的众多优点和其在授权管理系统中的普遍使用，因此基于角色映射的跨域授权模型研究也得到较多的关注，基于角色映射的机制目前已较为成熟，本节将在前人研究的基础上探讨一种基于角色映射的跨域授权模型。

下图 3.1 中给出了基于角色映射的跨域授权过程。用户首先进行身份鉴别，如果是跨授权域的服务请求，授权系统就将服务请求发到角色映射模块进行角色映射，角色映射将用户所在域的角色转换成服务提供者可以识别的角色，最后由服务提供者的授权系统决定用户权限。

基于角色的访问控制主要包含了用户集（*USERS*），角色集（*ROLES*）和权限集（*PERMS*）。在角色映射模型中我们只考虑用户集和角色集之间的关系。在基于角色的访问控制系统中，角色是权限的载体，管理员给角色赋权限，用户通过饰演不同的角色从而获得角色所拥有的权限。角色是 RBAC 访问控制系统的核心，因此在两个 RBAC 访问控制系统中，只需要进行角色间的映射就可以实现两个系统间的安全互操作。下面将在 2.5.2 节中介绍的跨域授权模型的基

础上给出一种角色映射模型。

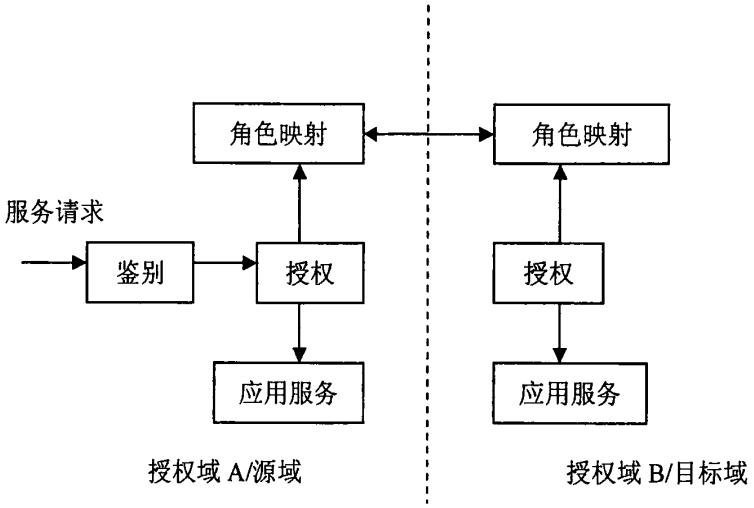

图 3.1 基于角色映射的跨域授权过程图

假设存在两个授权域 $DOMAIN_a$，$DOMAIN_b$。$USERS_a$，$USERS_b$ 分别表示各授权域中的用户集，$ROLES_a$，$ROLES_b$ 表示各授权域中的角色集。$URA \subseteq USERS \times ROLES$ 表示授权域内用户与角色的对应关系。

角色映射双方的关系是单向的映射关系，参加映射的双方角色可以是一对一，多对一，多对多或者一对多的关系。如定义 $DOMAIN_a$ 到 $DOMAIN_b$ 的映射关系如下：

若授权域 A 的角色 $\{R_{a1}, R_{a2}, R_{a3}\ldots\} \in ROLES_a$，

授权域 B 的角色 $\{R_{b1}, R_{b2}, R_{b3}\ldots\} \in ROLES_b$。

则 $\{R_{a1}, R_{a2}, R_{a3}\ldots\} \Rightarrow \{R_{b1}, R_{b2}, R_{b3}\ldots\}$ 表示将授权域 A 中 $R_{a1}, R_{a2}, R_{a3}\ldots$ 映射为授权域 B 中的 $R_{b1}, R_{b2}, R_{b3}\ldots$，若有来自授权域 A 并且具有 $R_{a1}, R_{a2}, R_{a3}\ldots$ 角色的用户，授权域 B 将按照 $R_{b1}, R_{b2}, R_{b3}\ldots$ 的本地角色权限对外域用户进行授权。

如图 3.2 中所示，授权域 B 中的经理角色映射到授权域 A 中的出纳角色，而转关主管角色被映射到会计角色。这样处在授权域 B 中且拥有经理角色的用户可以在授权域 A 中获得出纳角色的所有权限。为了保证安全性，在本模型中角色映射为单向的映射关系，如图 3.2 中所示，经理角色被映射为出纳角色，但是经理角色拥有的权限要大过出纳角色，因此出纳角色并不一定在 B 域中具有经理角色的权限。

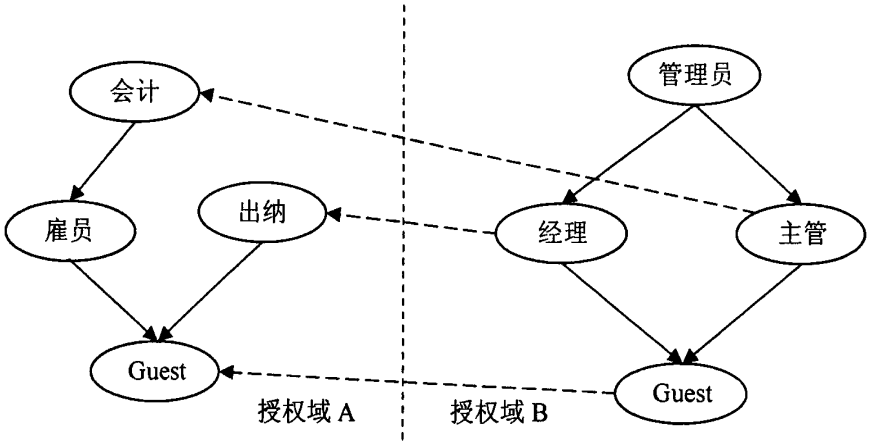

图 3.2 角色映射图

角色映射模型中的动态角色转换是否会违背职责分离是必须考虑的安全问题之一。以上图 3.2 为例，授权域 B 中的角色经理被映射为授权域 A 的出纳角色，主管角色被映射为授权域 A 中的会计角色，而在授权域 A 中的管理员角色是经理和主管的子角色，根据 RBAC 中角色的继承性管理员角色将同时拥有经理和主管两个角色的所有权限，通过角色映射则管理员将在授权域 A 中同时获得会计和出纳两个角色。而出纳和会计角色在授权域 A 中是一对互斥的角色，因此当管理员访问授权域 A 中的资源时就会引发安全问题。由于角色的映射转换是一个动态的过程，模型中我们可以采用动态职责分离策略对跨域用户的行为进行限制。

3.2 用户角色（组）交叉映射模型

目前最常用的访问控制机制有访问控制列表（ACL）和基于角色的访问控制（RBAC）。基于 ACL 的访问控制，通过在资源 ACL 中列明允许访问的用户 ID（标识）和组 ID 及允许的相应操作，规定用户对资源的访问权限；RBAC 则通过角色定义及相应的角色权限和访问控制策略规定，确定用户的资源访问权限。上节介绍的角色映射模型只能适用于基于角色的访问控制系统间的安全互操作，在与基于访问控制列表的访问控制系统间就无法适用。本节介绍的模型包含了上节中介绍的角色映射模型，我们将对角色模型进行扩展，提出一套基于用户角色（组）交叉映射的跨域授权模型。本模型针对目前最常用的 ACL 和

RBAC 访问控制机制，通过用户组和角色的交叉映射策略，在具有不同访问控制机制和不同安全策略的授权域间实现角色（组）映射，从而完成跨域的权限信息转换。

假设存在两个授权域 $DOMAIN_a$，$DOMAIN_b$。$USERS_a$，$USERS_b$ 分别表示各授权域中的用户集，$ROLES_a$，$ROLES_b$ 表示各授权域中的角色集。$GROUPS_a$，$GROUPS_b$ 表示各授权域中的用户组集，在 ACL 的访问控制系统中，用户组代表具有相同权限的一群用户。$URA \subseteq USERS \times ROLES$ 表示授权域内用户与角色的对应关系。

参与映射的双方可以是一对一，多对一，多对多或者一对多的关系。在本模型中映射方式可分为下列 4 种：

（1）角色到角色的映射

参见 3.1 节。

（2）用户组到用户组的映射

若授权域 A 和授权域 B 都采用了基于 ACL 的访问控机制，且

授权域 A 的用户组 $\{G_{a1}, G_{a2}, G_{a3}\ldots\} \in GROUPS_a$，

授权域 B 的用户组 $\{G_{b1}, G_{b2}, G_{b3}\ldots\} \in GROUPS_b$。则

$\{G_{a1}, G_{a2}, G_{a3}\ldots\} \Rightarrow \{G_{b1}, G_{b2}, G_{b3}\ldots\}$ 表示将授权域 A 中的用户组 G_{a1}, G_{a2}, G_{a3}... 映射为授权域 B 中的用户组 G_{b1} , G_{b2}, G_{b3}... ，若有来自授权域 A 并且具有 G_{a1}, G_{a2}, G_{a3}...用户组的用户，授权域 B 将按照 G_{b1} , G_{b2}, G_{b3}...的本地用户组权限对外域用户进行授权。用户组的映射双方关系是单向的映射关系。例如：某公司 A 的管理用户组被映射到公司 B 的销售和人力资源两个用户组，则公司 A 中属于管理用户组的用户将会同时拥有 B 公司的销售和人力资源用户组的权限。该映射方式实现了基于 ACL 的访问控制系统之间的跨域授权，本映射为单向映射关系，也就是 B 域中同属于销售和人力资源用户组并不能被映射为 A 域的管理员组。

（3）用户组到角色的映射

若授权域 A 采用的是基于 ACL 的访问控制机制，

其中包含用户组 $\{G_{a1}, G_{a2}, G_{a3}\ldots\} \in GROUPS_a$，

授权域 B 采用的是基于 RBAC 的访问控制机制，

其中包含角色 $\{R_{b1}, R_{b2}, R_{b3}\ldots\} \in ROLES_b$，

$\{G_{a1}, G_{a2}, G_{a3}\ldots\} \Rightarrow \{R_{b1}, R_{b2}, R_{b3}\ldots\}$ 表示将授权域 A 中的用户组 G_{a1}, G_{a2}, G_{a3}...

映射为授权域B中的角色R_{b1}, R_{b2}, R_{b3}…，若有来自授权域A并且具有G_{a1}, G_{a2}, G_{a3}…用户组的用户，授权域B将按照R_{b1}, R_{b2}, R_{b3}…的本地角色权限对外域用户进行基于角色的授权。用户组和角色之间映射双方的关系是单向的映射关系。例如：授权域A采用ACL的访问控制机制，其中存在一个经理用户组；授权域B采用RBAC的访问控制机制，其中存在销售经理角色和人力资源经理角色。若A域中经理用户组被映射到B域的销售经理角色和人力资源经理角色，则属于A域中经理用户组的用户都会具有B域中的销售经理角色和人力资源经理角色。该映射方式通过用户组与用户角色之间的映射实现ACL到RBAC访问控制系统之间的跨域授权，本映射为单向映射关系，也就是B域中同属于销售经理和人力资源经理角色的用户并不能映射为A域的经理用户组。

（4）角色到用户组的映射

若授权域A采用的是基于RBAC的访问控制机制，

其中包含用户组$\{R_{a1}, R_{a2}, R_{a3}\ldots\} \in ROLES_a$，

授权域B采用的是基于ACL的访问控制机制，

其中包含角色$\{G_{b1}, G_{b2}, G_{b3}\ldots\} \in GROUPS_b$，

$\{R_{a1}, R_{a2}, R_{a3}\ldots\} \Rightarrow \{G_{b1}, G_{b2}, G_{b3}\ldots\}$ 表示将授权域A中的角色R_{a1}, R_{a2}, R_{a3}… 映射为授权域B中的用户组G_{b1}, G_{b2}, G_{b3}…，若有来自授权域A并且具有R_{a1}, R_{a2}, R_{a3}…角色的用户，授权域B将按照G_{b1}, G_{b2}, G_{b3}…的用户组权限对外域用户进行基于ACL的授权。用户角色和用户组之间映射双方的关系是单向的映射关系。例如：授权域A采用RBAC的访问控制机制，其中存在一个经理角色；授权域B采用ACL的访问控制机制，其中存在销售经理和人力资源经理用户组。若A域中经理角色被映射到B域的销售经理和人力资源经理用户组，则属于A域经理角色的用户都会具有B域中的销售经理和人力资源经理用户组。该映射方式通过用户角色与用户组之间的映射实现RBAC到ACL访问控制系统之间的跨域授权，本映射为单向映射关系，也就是B域中同属于销售经理和人力资源经理用户组的用户并不能被映射为A域的经理角色。

3.3 属性映射模型

本节我们将根据上面讨论的模型中抽象出一种更具一般性的跨域授权模型——基于属性映射的跨域授权模型。在2.3节中介绍了基于属性的访问控制机制更适用于分布式结构，基于属性的访问控制机制不仅可以涵盖所有基于主体

属性（用户组，角色等）进行授权的访问控制机制（如 ACL，RBAC 等等），并且还可以利用客体（资源）属性，环境属性来定义授权策略，基于属性的访问控制机制具有很强的灵活性。通过属性的映射机制，不仅可以在具有不同访问控制机制的授权域中进行跨域访问，还可以制定一系列属性条件，只有在外域用户满足这一系列属性条件的情况才可以拥有本域的某个属性，例如，属性映射策略可以设为：岁数大于 30 且信用度大于 100 的源域用户可以映射到目标域的 vip 角色。

假设存在两个授权域：源域 $DOMAIN_S$，目标域 $DOMAIN_T$。S，R 分别表示主体（Subject）和资源（Resource）。

*ATTR(S)*表示主体属性集合。例如，用户 EMAIL=xxx@yyy.com；用户的信任级别 CredentialLevel=10 等等。

*f(ATTR(S))*表示由主体属性组成的属性条件。例如，用户年龄 age<50；用户的信任级别 CredentialLevel>10；用户 EMAIL=xxx@yyy.com 等组成的属性条件集合。单个属性条件（如用户年龄 age<50）除了使用上面简单的数学表达式表述外，属性条件也可以由用户自定义函数来表述源域到目标域属性之间的映射转换关系，如正则表达式，通配符匹配等等。除此以外，属性条件（即 *f(ATTR(S))*）亦可以是各属性条件之间的逻辑组合（与，或，非等）。总之属性条件具有很强的灵活性，用户可以通过自定义函数和映射器来实现各种所需的属性条件及其映射实施机制。

属性映射主要针对主体属性，本模型属性映射方式如下：

ATTR (S_T)，*ATTR* (S_S)分别表示目标域主体属性集和源域的主体属性集。

$\{f(ATTR(S_S) \vee f(ATTR(S_S)) \vee \ldots\} \Rightarrow ATTR(S_T)$ 表示由源域属性组成的属性条件集合（各属性条件间采用“并”关系，属性条件中可以支持其它逻辑关系，如或，非等等）映射到目标域的属性集，目标域授权系统将根据映射结果 $ATTR(S_T)$（目标域属性集）实施授权。

在映射过程中，有些属性可能不需要映射即目标域直接认可来自源域的属性值，例如用户的 EMAIL 属性在映射过程中无需进行源域到目标域的转换，因此只需把源域的 EMAIL 属性直接赋予跨域用户。

假设存在两个授权域 A，B。授权域 A 中的用户 User 欲访问授权域 B 中的资源，在本基于属性映射的模型中就必须进行 User 属性从授权域 A->B 的映射转换。User 在 A 域中的属性有：EMAIL=xxx@yyy.com；CredentialLevel=11；

age=40。授权域 A->B 的映射策略表示为：

{ CredentialLevel >10 && age>30}=> {role=manager && EMAIL=#EMAIL}

#属性名称表示该属性在映射过程中采用来自源域的属性值。

如上面策略所示用户 User 将在授权域 B 中获得 manager 角色，且 EMAIL 属性将依然为 xxx@yyy.com。同样，为了保持映射的安全性，属性的映射采用单向映射关系（即不可逆关系）。

用户的源域属性经过属性映射后将存放于用户会话中，访问目标域资源时，授权系统将根据用户会话中的用户属性对资源请求进行访问控制。基于属性映射的跨域授权模型具有 ABAC 的灵活性，并且可以在多种基于主体属性授权（如 ACL，RBAC，ABAC 等）的访问控制系统中实现跨域授权。

3.4 本章小结

本章中我们首先根据现有的基于角色映射的跨域授权模型提出了一种基于角色映射的跨域授权模型并讨论喝解决了角色映射过程中可能出现的一些安全问题，然后在角色映射的基础上提出了一种基于用户组和角色交叉映射的跨域授权模型。最后在上述两种模型基础上抽出共性，首次提出了一种支持用户属性映射的跨域授权模型。下文中我们将根据本章提出的跨域授权理论模型实现一个具有跨域授权功能的授权管理系统。

第 4 章　基本访问控制系统

4.1 系统总体架构

本系统为独立的授权管理服务系统。通过授权管理系统，应用系统可以采用灵活的权限管理策略，企业可以将多个应用系统中的资源进行集中管理，实现集中的权限管理和授权。

系统的总体架构如图 4.1 所示。

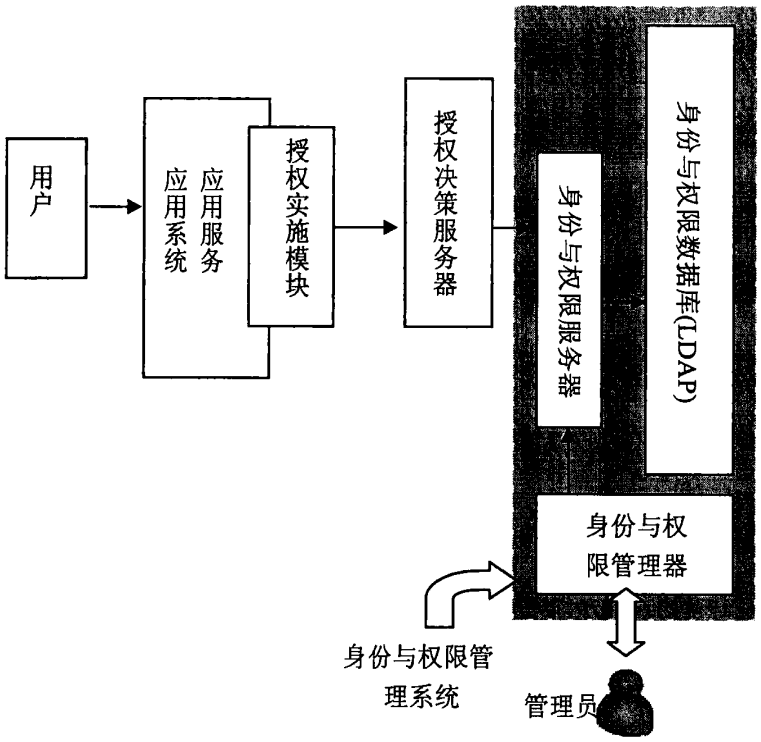

图 4.1　基本访问控制系统总体架构

基本访问控制系统主要包括如下几部分：身份与权限数据库，身份与权限管理器（Identity & Privilege Manager，IPM）和身份与权限服务器(Identity & Privilege server，IPS)、授权决策服务器（Authorization Decision Server，ADS），授权实施模块。系统中各个模块的描述如下：

身份与权限管理器：是一个基于 Web 的应用服务程序，它向身份与权限管理员提供了基于 Web 的身份与权限信息管理服务，身份与权限管理员可通过该管理器进行身份与权限信息的创建、查询、修改、更新、删除等。

身份与权限数据库：是一个 LDAP（Lightweight Directory Access Protocol，轻量级目录协议）[35]数据库，用于存放用户身份信息、角色信息和资源权限信息。身份与权限信息的创建、查询、修改、更新、删除，通过身份与权限服务器（IPS）、身份与权限管理器（IPM）进行。

身份与权限服务器：通过提供 Java 本地，RMI[36]，Web Services[37]远程接口对外向应用进程提供身份与权限信息的创建、查询、修改、更新、删除等服务功能。

授权决策服务器：根据用户的身份、权限信息和访问控制规则决定用户是否能够访问某个资源、进行某项操作、获得某项服务等。

授权实施模块：采用过滤器的方式（如 Servlet Filter、ISAPI、NSAPI 过滤器）拦截用户的资源请求并发送到授权决策服务器进行决策。

授权管理系统的执行流程为：当用户访问应用程序或应用服务时，授权实施模块将拦截用户的资源请求，转发到授权决策服务器，授权决策服务器依据身份与权限数据库的权限信息和用户请求进行授权决策，授权实施模块将根据授权决策的结果允许或拒绝用户的请求。身份与权限管理系统向管理员提供 Web 方式对身份与权限信息进行管理。身份与权限服务器为其他模块提供 LDAP 数据库的访问接口，身份与权限管理器和授权决策服务器通过调用身份与权限服务器接口来操作 LDAP 数据库。

4.2 系统实现

4.2.1 身份与权限信息组织结构

身份与权限数据库被用来存储身份与权限信息，鉴于身份与权限数据库中数据查询的次数要远大于写入，本系统采用 LDAP 数据库进行存储。LDAP 的英文全称是 Lightweight Directory Access Protocol，即轻量级目录协议，它是由早期的 X.500 目录标准发展而来的。目录是一个为查询、浏览和搜索而优化的专业分布式数据库，它成树状结构组织数据。目录数据库有优异的读性能，但写性能差，没有事务处理、回滚等功能，不适于存储修改频繁的数据。由于身

份与权限信息的特性，查询和读取操作要远多于写入操作，所以本系统的数据库采用 LDAP 数据库。

身份与权限信息数据库中包含用户目录、角色目录和资源目录。用户目录由层次性的用户组织单元和用户组组成。系统管理员可以根据此用户组织单元查找、修改、删除用户、组信息。并且 ACL 策略可以根据用户的组信息进行授权，所有加入到该组的用户将拥有用户组的权限，方便了 ACL 策略的管理和分配。角色目录中，角色条目的上下级关系体现了角色之间的继承关系。本文的角色信息在逻辑上还将以角色作为结点在数据库中存放，称之为角色树。资源目录采用资源策略树对资源、策略信息进行存放。对于多种访问控制方法的权限与策略进行统一管理，针对不同的访问控制方法（本文主要是 ACL、RBAC、ABAC 三种访问控制方式）同时定义不同类型的授权策略并以通用的形式存放于数据库中。系统可以根据访问控制方法的需要灵活的选择使用授权策略。

本系统以开源的 OpenLDAP[38]作为 LDAP 数据库。图 4.2 所示为本系统中 LDAP 的数据组织结构。

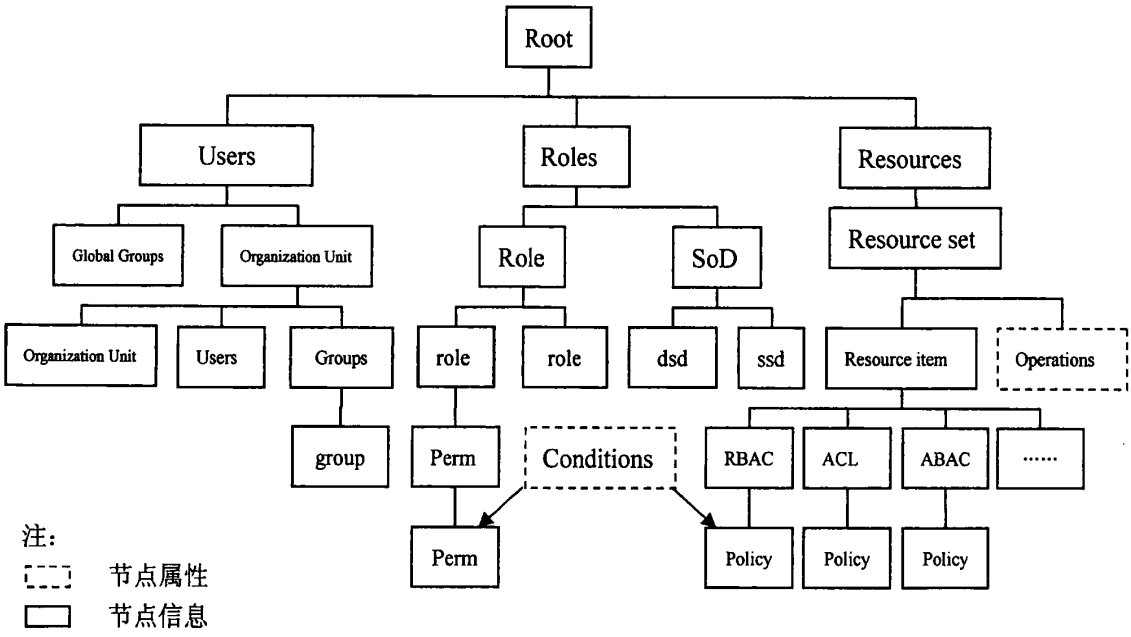

图 4.2 LDAP 数据组织结构图

LDAP目录服务器中包含了用户目录、角色目录、资源目录。

（1）用户目录

包含用户信息和用户组信息。用户信息包含了用户的基本信息，如用户名，口令，状态，邮箱等，还包括用户组和用户角色信息。主要内容和功能有：

1）用户以单位—部门—部门的树形目录结构进行存放。单位目录中可包含部门，组，用户三种信息。

2）用户组信息存放在单位或部门目录下的 Groups 子目录中，全局组信息存放在用户目录的根目录下的 Groups 子目录中。用户组信息包含组名称，组信息介绍，组用户 URL 和组成员。ACL 策略可以根据用户的组信息进行授权，所有加入到该组的用户将拥有用户组的权限，方便了 ACL 策略的管理和分配，用户组不具备继承性。

3）用户可以被本级管理员分配到其本级或下级部门的组当中，若要将用户赋到上级单位或部门的用户组中则需由上级目录管理员进行指派。

4）用户信息中还包含与其他系统进行映射的用户列表，本属性用来映射已有应用系统的用户，为已有应用系统提供授权服务。

（2）角色目录

角色目录包含角色信息和职权分离策略信息。主要内容和功能有：

1）角色信息定义了角色的名称，权限等。资源策略目录中的资源+操作组成了角色的权限信息。

2）权限条目存放在角色的 Perms 子目录中，每个权限条目包含资源信息，操作信息，权限条件和审计标识属性。权限条件限制了该权限信息只能在满足本条件的情况下才能启用。本系统中支持时间条件，只有在时间条件定义的区间内，权限才会生效。权限条件具有可扩展性。

3）角色动态成员属性允许管理员定义类似“具有什么属性的用户可以拥有该角色”的条件，从而实现角色的动态赋予。

4）职责分离策略定义了在企业、应用内部具有安全冲突的角色集。本策略包含静态职责分离和动态职责分离两种策略。静态职责分离在管理员进行角色—用户分配的时候进行检查；动态职责分离在激活用户角色的时候进行检查。

（3）资源目录

资源目录包含资源的基本信息，操作信息和策略信息。资源的定义涵盖了各种应用程序资源，包括自定义的各种操作，各类服务，服务内容，文件，数据库内容等等。主要内容和功能有：

1）资源目录结构以资源集—资源子集—资源条目进行划分，资源条目是资

源的最小单位。

2）操作信息对应资源条目所允许的操作；操作信息对其下级目录是可继承的，顶级资源节点对应的操作被定义为全局操作，全局操作对所有资源条目有效。操作信息存放在资源的 actions 属性中。

3）资源目录中包含三种类型访问控制策略，ACL、RBAC、ABAC。不同的访问控制策略存放在资源目录下相应的目录中（如 ACL 策略存放在 ACL 目录下）。

4）策略条件，存放在每个策略条目（ACL，RBAC，ABAC）对应的策略条件属性中。策略条件与上述的全线条件相同，在策略条目中它限制了该策略信息只能在满足本条件的情况下才能生效。本系统中支持时间条件，只有在时间条件定义的区间内，权限才会生效。策略条件具有可扩展性。

5）ACL 策略（访问控制列表）由多条 ACI（访问控制条目）构成，不同的 ACI 对应着 ACL 目录下的一个条目，每条 ACI 包含用户、组信息，操作信息，策略条件，审计标识和继承标识等属性。继承标识为 TRUE 的 ACI 允许被下级的资源条目继承否则不能被下级资源节点继承，每条策略中包含审计（Audit）标识，标识为 True 的策略在授权过程中需要做审计。

6）RBAC 策略（基于角色的访问控制策略）由多条 RBAC 策略条目构成，不同的 RBAC 策略条目对应着 RBAC 目录下的一个条目，每条 RBAC 策略包含角色集信息，操作信息，策略条件，审计标识和继承标识等属性。每条策略条目允许定义多个角色才能具有的操作权限。每条策略都带有继承标识，继承标识为 TRUE 的策略条目允许被下级的资源条目继承否则不能被下级资源节点继承，每条策略中包含审计（Audit）标识，标识为 True 的策略在授权过程中需要做审计。

7）ABAC 策略（基于属性的访问控制策略）由多条 ABAC 策略条目构成，ABAC 策略条目根据用户属性进行授权。不同的 ABAC 策略条目对应着 ABAC 目录下的一个条目，每条 ABAC 策略包含属性条件信息，操作信息，策略条件，审计标识和继承标识等属性。每条策略都带有继承标识，继承标识为 TRUE 的侧策略条目允许被下级的资源条目继承否则不能被下级资源节点继承，每条策略中包含审计（Audit）标识，标识为 True 的策略在授权过程中需要做审计。

8）资源属性包含一个启用继承标识，标识本级资源是否允许从上级资源条目继承的访问控制策略在本资源条目有效。

4.2.2 身份与权限服务器

身份与权限服务器（IPS）通过非 LDAP 接口对外向应用进程提供身份与权限信息的创建、查询、修改、更新、删除等服务功能。IPS 提供本地接口和远程接口方式供外部调用，为外部应用提供数据访问功能。身份与权限服务器结构如图 4.3 所示，主要划分为：请求/响应接口，鉴别与授权，消息加解密和执行组件几个组件。

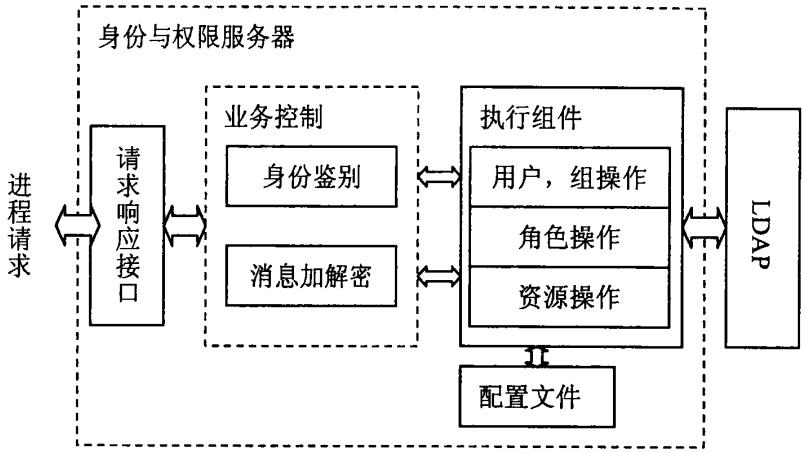

图 4.3 身份与权限服务器模块结构图

身份与授权服务器在授权管理系统中充当数据访问接口，由于调用该服务接口的模块可能是授权决策服务器，身份与权限管理器或采用各种技术的应用程序/服务，所以 IPS 需要提供多种形式的调用方式，本模块支持本地调用，RMI 调用，Web Services 调用三种方式。本地和 RMI 针对各种 J2EE 系统，Web Services 则针对来自其他异构系统的调用。所有进行 LDAP 操作的行为都将通过 IPS 来完成。

请求响应接口负责接收来自进行进程或用户的各种类型请求（包括本地，RMI，WS 等），若传递的消息具有较高的安全性请求响应接口还负责对消息进行相应的加解密，由于身份与权限信息的保密性是衡量授权管理系统安全性的重要标志，鉴别和授权组件负责对服务请求进行身份鉴别和授权，只有得到授权的用户才能执行相应的 LDAP 操作。

本模块以 Spring 2.0 作为容器，各个系统组件由 Spring 框架通过配置文件来管理。所用技术包含 Spring Acegi[39]，Spring LDAP[40]，XFire[41]等，Acegi 框

架负责解决系统的鉴别和授权，Spring LDAP 提供了访问 LDAP 的 API，XFire 为我们提供了 Web Services API。这些框架或技术都可以通过 Spring 无缝的粘合在一起。身份与权限服务器中各个功能组件的实现方案如下：

（1） 请求响应接口

提供支持多种协议的请求消息，如 SOAP/HTTP，RMI，Java 本地调用。应用程序可自己选择相应的调用方式。基于 J2EE 的应用系统和授权决策服务器（ADS）使用 RMI 进行调用；对于非 J2EE 的系统采用 Web Services 进行调用，IPS 服务接口发布其 WSDL（web 服务描述语言）供远程异构系统调用。各种调用方式实现方法：

1） 本地调用方式：IPS 存放于 J2EE 系统的类路径中；

2） RMI 调用：采用 Spring RMI 方案，Spring 对 RMI 进行了简化，可以很方便的将 Java 接口发布为 RMI 服务；

3） Web Services 调用：XFire 是一个流行的 Java Web Services 开发框架，它与 Spring 可以进行无缝结合，支持将 Spring 中管理的 JavaBean 发布为 Web Services。

（2） 身份鉴别与授权

由于 IPS 是对 LDAP 中的身份与权限信息等重要信息进行操作，而身份与权限信息是授权管理系统的关键，因此我们必须对调用者进行身份鉴别和授权。

本系统中采用 Acegi 框架来解决身份鉴别和授权问题。Acegi 安全系统，是一个用于 Spring Framework 的安全框架，能够和目前流行的 Web 容器无缝集成。它使用了 Spring 的方式提供了身份鉴别和授权安全服务。在本授权系统中，身份鉴别方式支持用户名口令和数字证书，下面我们将 Acegi 与上文提到的接口调用方式相结合：

1） 本地调用：在 LDAP 中我们设置 Admin 用户组，只有在 Admin 用户组中的用户才具有使用 IPS 的权限。

2） RMI 调用：由于 RMI 协议的无状态性，因此如果采用 RMI 的调用方式，就必须在每次调用中都进行一次身份鉴别，这样系统的效率势必会受到影响。Acegi 安全框架专门为 RMI 的身份鉴别提供了一个方便的类 ContextPropagatingRemoteInvocationFactory。Acegi 提供了支持 RMI 身份鉴别的 API，客户端只需要将用户身份信息（用户名口令或数字证书）存放在

SecurityContextHolder类中，服务端的Acegi拦截器将拦截SecurityContextHolder中的用户身份信息进行鉴别，所有的操作对用户来说都是透明的。

3） Web Services 调用：Web Service Security (WSS) 规范[42]定义了 Web Services 的身份鉴别和 SOAP 消息签名，加密。在 Web Services 的调用中客户端遵循 WSS 规范，将用户的身份信息存放在 SOAP Header 中，身份信息可以是用户名口令，数字证书，Kerberos ticket，SAML 断言等。在基于 XFire 的服务端中，我们通过扩展 XFire 的拦截器将其与 Acegi 进行整合。

具体方案为[43]：

```
public class AuthHandler extends AbstractHandler{
//注入 Acegi 验证模块
private AuthenticationManager authenticationManager;
//从头部获取身份信息登陆 LDAP 并执行操作
public void invoke(MessageContext context) throws Exception{}
}
```

在 XFire 拦截器类中，首先分离出 SOAP Header 中的用户身份信息，封装到 SecurityContextHolder，然后通过 Spring 向 AuthHandler 中注入 Acegi 的 AuthenticationManager 对用户进行身份鉴别。

授权：用户完成身份鉴别后，Acegi 拦截器获得已鉴别的 LDAP 用户信息，并使用该用户信息来实现访问 LDAP 数据库和执行 LDAP 操作。用户所获的权限由 LDAP 中定义的 ACL 策略决定。

（3） 消息加解密

对于一些保密的消息，为防止在传输的过程中泄漏，系统提供了如下机制来对传输消息进行加解密。

本地调用：Web 服务器配置双向 SSL；

RMI 调用：使用 Spring RMI 配置双向 SSL；

Web Services 调用：IPS 服务端使用 XFire+WSS4J，可以通过 SSL 或 WSS 来实现信息的安全传输。

（4） 执行组件

执行组件即 LDAP 数据库的数据访问接口，包含三个部分，分别为用户和用户组操作接口，角色操作接口和资源权限操作接口。分别对相应的 LDAP 目录进行操作。

Spring LDAP 提供了简便的 LDAP 数据库访问 API，这个框架能够帮助开发人员简化 looking up, closing contexts, looping through NamingEnumerations，encoding/decoding values 与 filters 等 LDAP 操作。Spring LDAP 的核心类是 LdapTemplate，身份与权限信息的创建、查询、修改、更新、删除等操作都是通过 LdapTemplate 来完成（LdapTemplate 被注入到 IPS 接口中）。

身份与权限服务器的执行流程如图 4.4 所示：

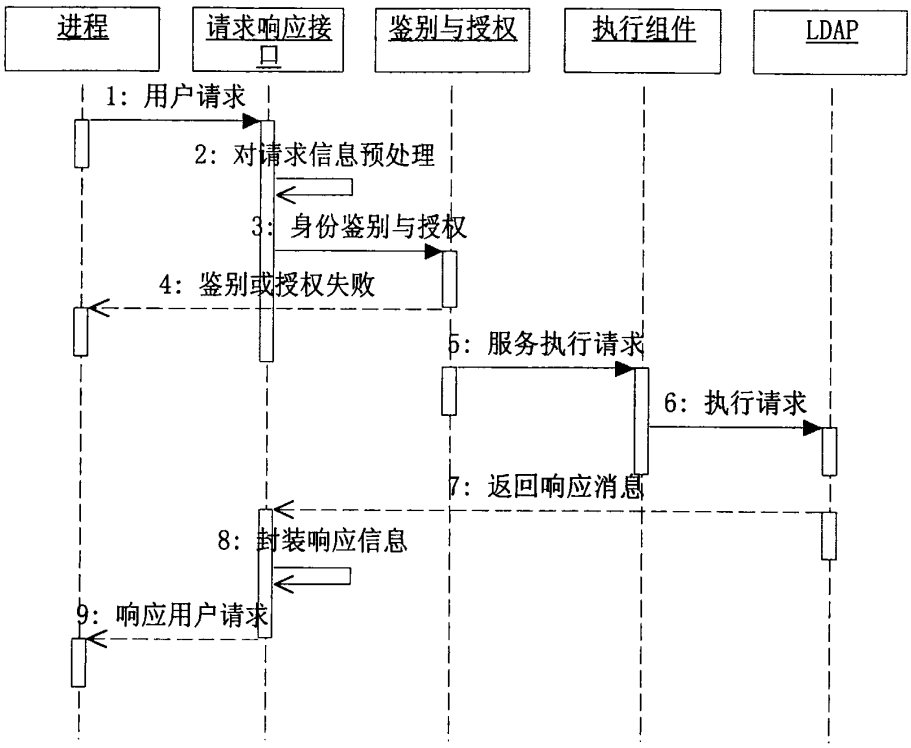

图 4.4 身份与权限服务器系统流程图

4.2.3 身份与权限管理器

身份与权限管理器（IPM）是一个基于 MVC 框架 Struts 2 的 Web 的应用服务程序，它向身份与权限管理员提供了基于 Web 的身份与权限信息管理服务，身份与权限管理员可通过该管理器进行身份与权限信息的创建、查询、修改、更新、删除等。身份与权限管理器模块结构图如图 4.5 所示。

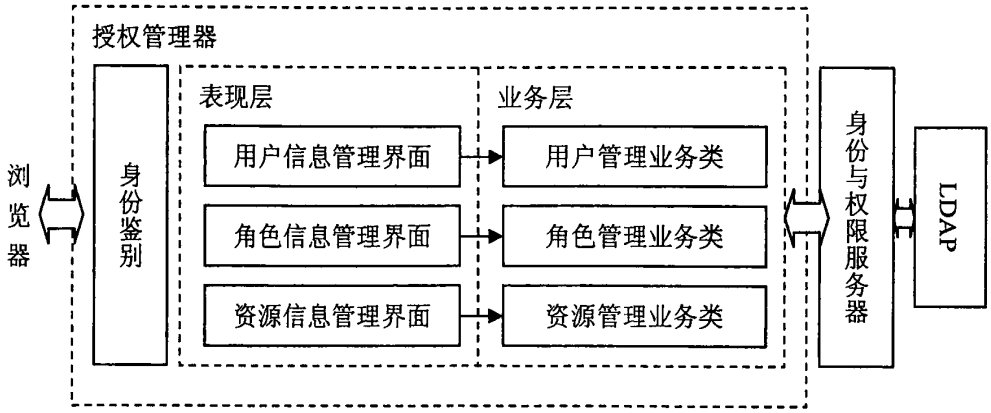

图 4.5 身份与权限管理器模块结构图

身份与权限管理器采用三层结构划分，包括表现层，业务层和数据访问层。表现层分为用户信息管理界面，角色信息管理界面，资源信息管理界面，所使用的技术包含 JSP/Servlet，AJAX 等；业务层分为用户业务管理类，角色管理业务类，资源管理业务类，Struts Action 负责处理业务逻辑；数据访问层由身份与权限服务器（IPS）来完成。于是我们可以将系统主要划分为三个子模块：**用户身份和组信息管理模块**（用户信息管理界面—用户管理业务类），**角色管理模块**（角色信息管理界面—角色管理业务类），**资源及权限管理模块**（资源信息管理界面—资源管理业务类）。

身份与权限管理器主要包含如下功能：

（1）用户和用户组信息管理

1）添加，修改，删除用户信息和用户组信息；

2）定制多条搜索条件查询用户信息，查询条件包含多种用户属性，可对条件进行逻辑组合；

3）维护用户和用户组对应关系；

4）添加和删除用户单位，部门；

5）给用户分配角色。

（2）角色和职责分离策略信息管理

1）添加，修改，删除角色信息；

2）制订角色启用条件允许角色动态赋予；

3）赋予角色权限，引用资源目录中的资源和操作信息；

4）制订角色的启用条件，系统预置了时间等基本条件，条件可以由用户

自行扩展；

5） 制定，修改，删除职责分离策略；

（3）资源和授权策略管理

1） 添加，删除资源集和资源条目；

2） 管理员可以添加，删除，修改资源集和资源条目下的操作信息；

3） 制定，删除，修改资源集或资源条目对应的 ACL 访问控制策略，RBAC 访问控制策略（基于多角色的）和 ABAC 访问控制策略。

4） 制定针对于第 3 点中策略的策略条件。

（4）分层次，分区域管理

由于在集中授权管理系统中，存在多个应用系统和资源，为了避免管理员的权限过大，本系统采用分层次，分区域管理。如企业、部门，一个区域的管理员只能对所负责域的身份与权限信息进行操作。

身份与权限管理系统实现采用 Struts + Spring 的架构。Spring 负责对系统使用的 JavaBean 进行依赖注入和管理。IPM 通过加载相应的配置文件来选择本地调用，RMI 方式或者 Web Services 方式调用身份与权限服务器（IPS）。身份与权限管理器中各个功能组件实现方案如下：

（1） 身份鉴别模块：身份与权限管理系统支持表单方式和证书方式两种身份鉴别方式。

1） 表单方式

管理员提交用户名口令，系统调用 IPS 的身份鉴别和授权模块对管理员身份进行鉴别，管理员的操作权限由 LDAP 数据库的访问控制策略定义，若管理员没有足够的权限来完成某次操作，系统将返回拒绝页面。

2） 证书登陆

证书登陆要求系统与浏览器之间进行双向的 SSL 认证，身份鉴别方式基于 IPS 的身份鉴别方式，其余操作与表单方式相同。

（2） 分层次的身份与权限信息管理

为避免管理员的权限过大，在本系统中身份与权限信息的管理采用分层次管理，即特定的管理员只能管理特定的区域。管理员在使用授权管理系统时，所做的操作必须由 IPS 来执行， IPS 执行权限是由 LDAP 数据库的访问控制策略来定义，通过 LDAP 自带的访问控制策略机制，授权系统可以良好的支持分层次的身份与权限信息管理。

（3） 用户身份和组信息管理模块

本模块主要实现用户身份信息和用户组信息的管理功能。模块业务逻辑基于 Struts 2 的 Action 来完成。Action 类由 Spring 依赖注入 IPS 服务接口。

图 4.6 所示为部门跟用户的管理页面。本页面中包含了本部门中的用户和用户组管理。

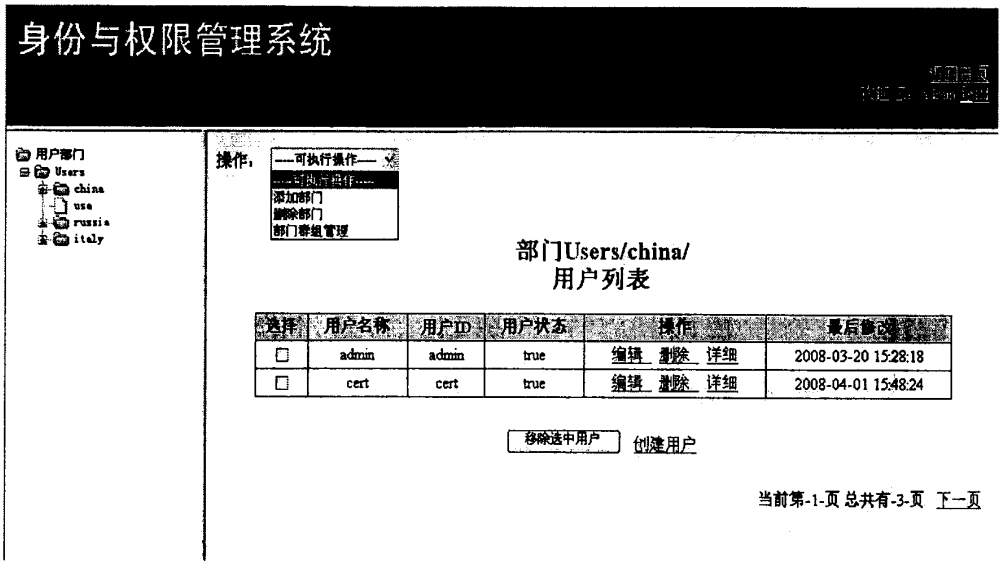

图 4.6 用户部门管理图

图 4.7 所示为用户编辑界面。

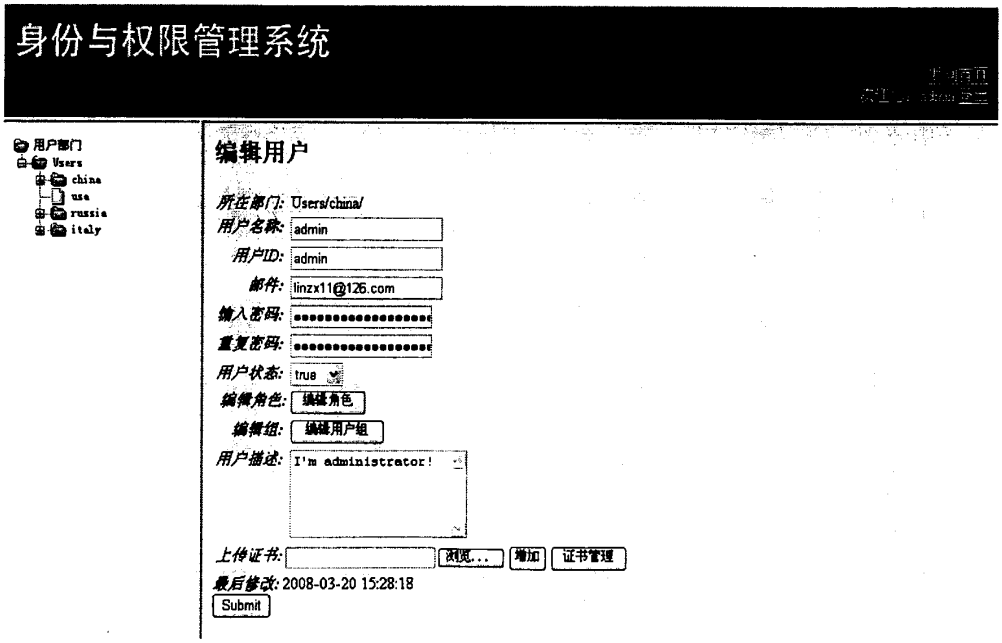

图 4.7 用户编辑界面

用户只能被加入到用户所在部门的本级或下级部门的用户组中，若要加入上级部门的用户组需要由上级部门的管理员来添加。管理员能添加的角色只有其在 LDAP 有管理权限的角色。每个用户支持保存多张数字证书，数字证书间通过证书的序列号进行查找。

图 4.8 所示为用户组编辑页面。

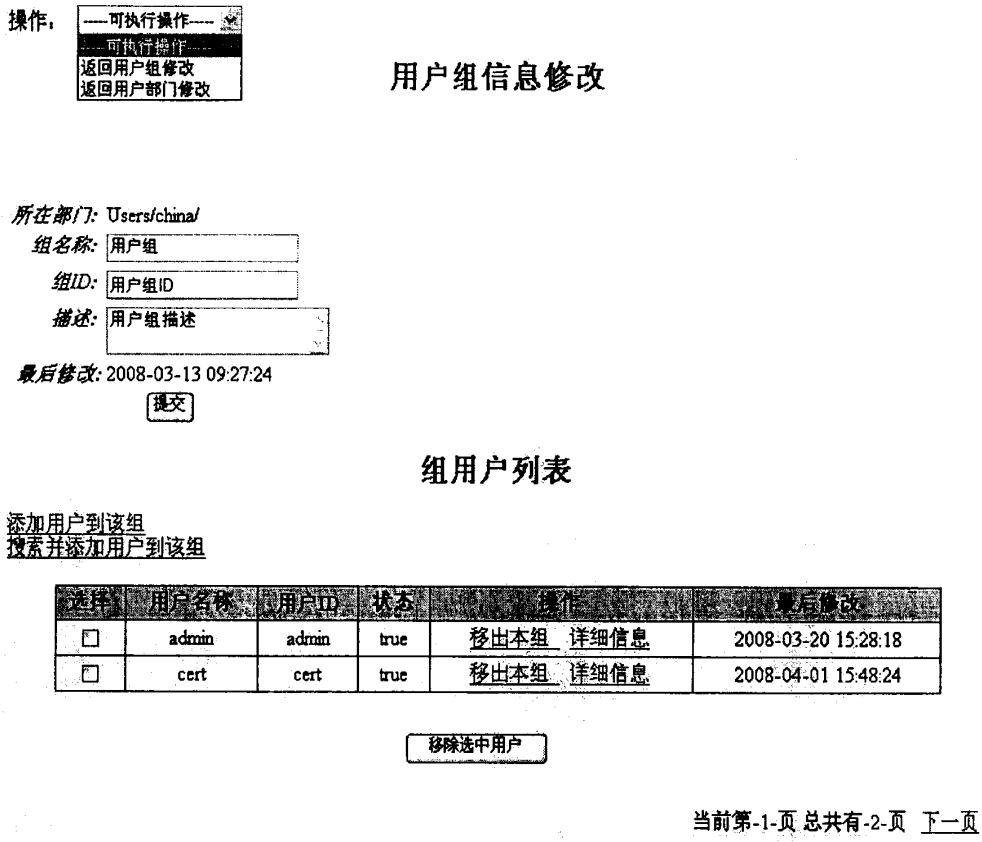

图 4.8 用户组编辑页面

用户组管理中支持两种方式添加用户到用户组，一种是通过搜索用户，一种是通过按部门列出未加入本用户组的用户。

（4）角色管理模块

本模块主要负责角色相关信息的管理，包含角色信息管理和职责分离策略管理。

职责分离策略包含动态职责分离和静态职责分离。二者具有相同的策略格式。策略格式包含角色集和基数，任何用户所拥有的角色（包括子角色）与策略中角色集的交集个数必须小于基数。

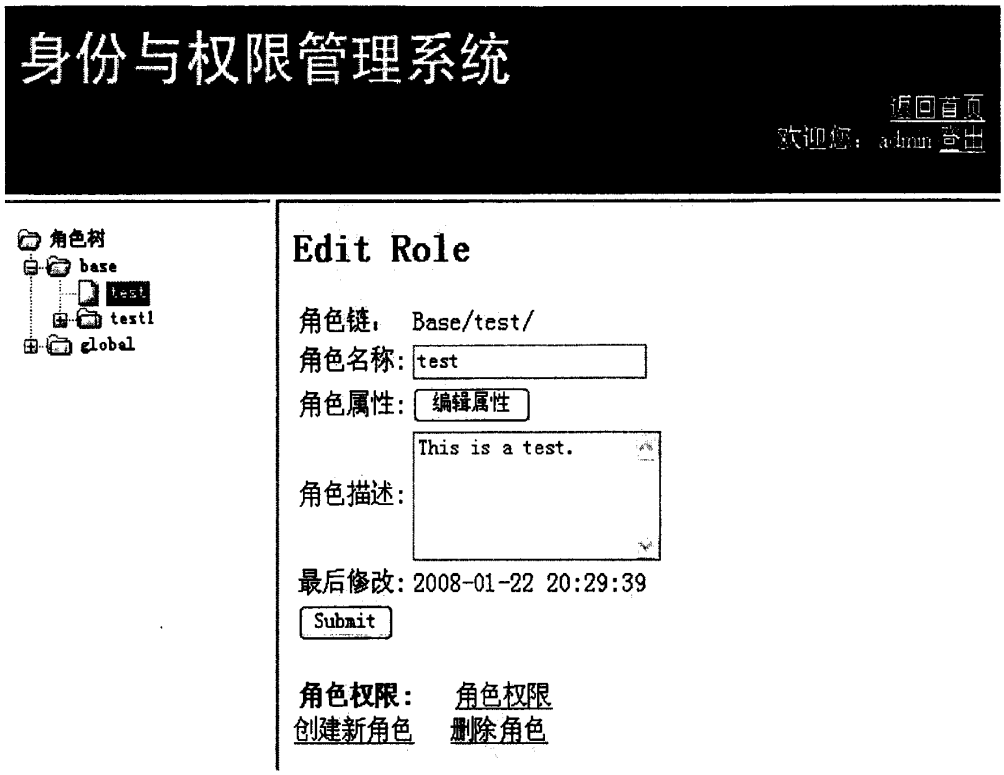

图 4.9 角色管理页面

图 4.9 所示为角色管理页面，角色属性为制定一系列用户属性组合，满足本组合的用户将获得该角色权限。在角色权限中可以编辑角色的权限信息如图 4.10 所示。

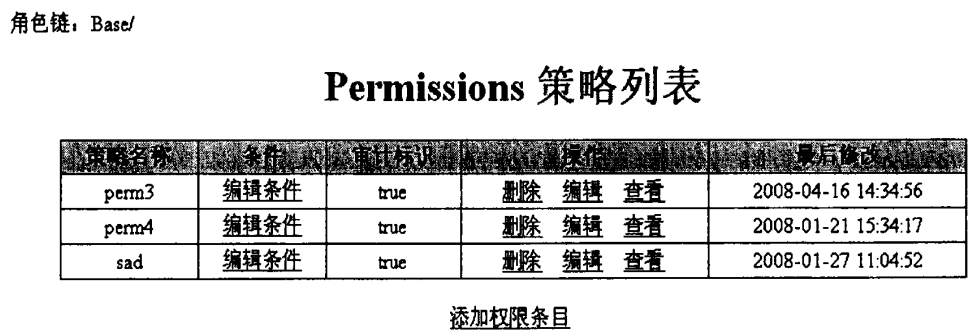

图 4.10 角色权限编辑页面

角色中权限信息由资源+操作构成，权限本身具有继承性，其策略格式如表 4.1 所示：

表 4.1 角色权限格式

资源	资源信息（资源 DN）
操作	资源操作
条件	策略条件
审计	本策略是否需要审计(true/false)

权限策略条件负责对权限进行限制，目前系统支持对权限或策略的时效控制，包括权限的权限的生效时间、权限的失效时间以及权限的启用时间段。例如自 2008/01/01 权限生效，自 2008/12/31 停用权限，该权限在每个工作日的 9 点至 17 点为启用时间段。用户可自定义扩展权限策略条件。

权限的审计标识为 true 表示，若该条权限在授权过程中被评估，则对本次授权进行日志记录以供后续审计，否则无需进行日志记录。

（5）资源及权限管理模块

本模块主要负责资源和权限信息的管理，包含了资源基本信息管理，策略信息管理（包括资源的策略信息和角色权限信息）。

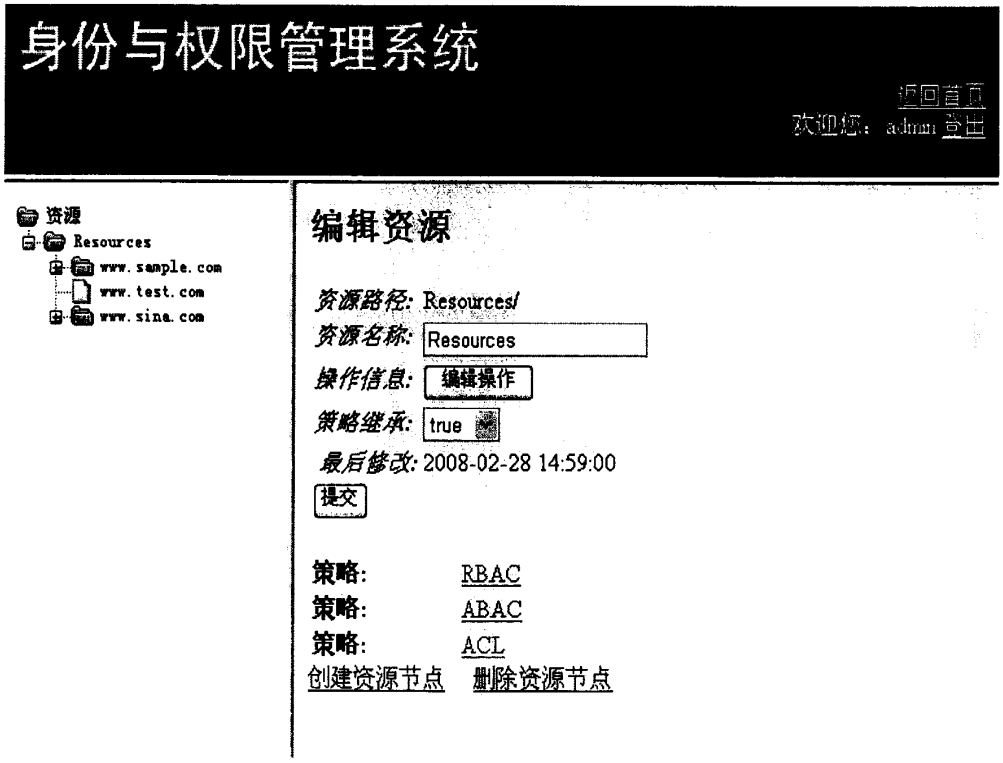

图 4.11 资源与权限管理页面

图 4.11 所示为资源与权限管理页面，在资源目录中，资源是按树形结构进行组织的，资源的操作信息对下级具有可继承性。资源的所有策略默认对下级都具有可继承性，但是可能存在某个目录并不想继承上级目录的安全策略，如图所示在资源属性中设置了一个策略继承标识，该标识表示本级资源是否允许从上级资源条目继承的访问控制策略在本资源条目有效，若设为 false 则从上级资源节点中继承的安全策略在本资源节点无效，但是对本资源节点下的子目录依然可以被继承。资源支持三种权限格式 RBAC，ABAC，ACL。此处的 RBAC 不同于角色目录下的权限策略，由于在访问控制需求中可能存在需要多个角色才能执行的操作，此处的 RBAC 策略解决了这个问题。

ACL，RBAC，ABAC 策略格式（此处 RBAC 策略用于制定需要多个角色才能完成的权限）如下表所示：

表 4.2 资源策略格式

Policy Subject	策略适用的主体
操作	策略动作
条件	策略条件
继承	策略对下级资源是否具有继承性
审计	本策略是否需要审计

资源策略的管理页面如下图 4.12 所示

资源位置：Resources/

ACL 策略列表

策略名称	条件	继承标识	审计标识	操作	最后修改
acl1	编辑条件	true	true	删除 编辑 查看	2008-03-13 10:48:20
acl3	编辑条件	true	true	删除 编辑 查看	2008-03-04 12:26:05
acl4	编辑条件	true	true	删除 编辑 查看	2008-01-18 16:55:58

添加策略条目

图 4.1 ACL 策略管理页面

其他资源策略的管理页面与 ACL 相似，资源策略内容比角色权限多了一个属性“继承标识”，只有继承标识为 true 的策略条目才会被下级资源继承（角色由于本身的可继承性所以其继承标识都是 true）。其余的策略条件，审计标识与角色的权限策略相同。

4.2.4 授权决策服务器

授权决策服务器是一个单独的服务系统，主要负责对应用系统提交的请求做出授权决策。授权决策服务器采用一个授权管理器（Manager）和多个授权提供者（Provider）的组织结构。

在 IPM 中允许管理员给资源制定多种访问控制策略，因此授权决策服务器必须支持多种访问控制技术，如 ACL、RBAC、ABAC 等。不同的访问控制技术由不同的 Provider 实施，如 ACL Provider 、 RBAC Provider 、 ABAC Provider 等，其间以松散耦合的形式聚合在一起。

授权决策服务器采用 Manager 对各 Provider 进行管理。Provider 通过 Manager 动态注册和注销。当有新的 Provider 要聚合在服务器中时，只需要继承 Provider 接口中的方法，并在 Manager 处注册信息，即可对外提供服务。授权决策服务器系统逻辑结构图如图 4.13 所示。

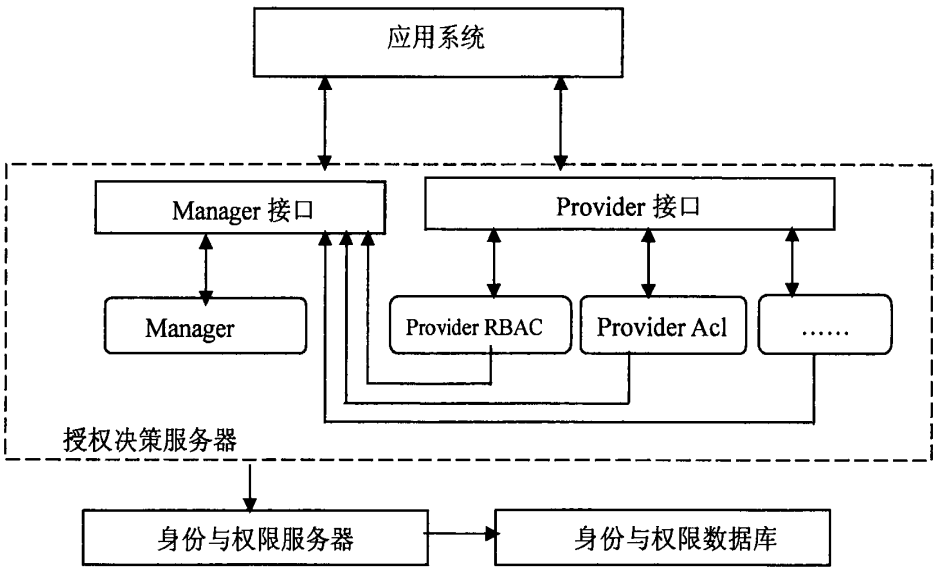

图 4.13 授权决策服务系统逻辑结构图

（1） 授权管理器（Manager）功能：

1）Provider 的注册\注销：负责 Provider 的注册登记，同时存放 Provider 的对象在 Manager 缓存中。同时也可以注销 Provider，删除与之有关的所有信息。

2）Provider 的分配与选择：由于基于不同访问控制技术的 Provider 支持的请求类型不同（如 RBAC Provider 中必须要求请求中包含角色信息），针对应用系统的不同请求类型，Manager 负责为用户分配一个 Provider 进行授权，或者

用户自己指定一个 Provider 进行授权。选择 Provider 的方式应该包括以下 3 种：

① 用户使用缺省的 Provider 授权，此 Provider 由系统定义。Manager 返回此缺省的 Provider 对象。

② 用户指定一个 Provider 授权，即用户请求中标明指定使用一个 Provider 授 Manager 通过请求中所指定这个 Provider 的 ID，在 Manager 缓存中为其找到 Provider 对象并返回。

③ 用户委托 Manager 为其寻找一个 Provider 授权，此时 Manager 将用户请求发送给各个 Provider。Providers 在内部匹配此请求类型，若发现可以对此请求类型决策，则返回成功的结果给 Manager。最后 Manager 在符合条件的 Providers 中选一个合适的并返回给用户使用。

3） 授权访问的查询与匹配：针对应用系统的请求类型，查询授权决策服务系统中哪些 Providers 可以对请求进行决策。结果中返回这些 Providers 的 ID。

（2） 授权提供者（Provider）功能：

1）授权决策：针对应用系统提交的请求，按照 Provider 中存储的策略信息作出决策，并返回结果。

2）授权访问的查询与匹配：应用系统可以查询该 Provider 是否能根据现有请求信息类型做出决策。

3）缓存策略信息：授权决策服务器在授权前将策略信息全部从 LDAP 数据库中取出并存放在缓存中。

授权决策服务器提供了本地接口，RMI 和 Web Services 三种调用方式。授权决策服务器被授权实施模块调用，授权实施模块是一个附着于应用系统的过滤器，它拦截住所有对资源的请求操作并构造相应的授权请求发送到授权决策服务器，最后根据授权服务器的返回结果允许或者拒绝用户操作。

4.3 本章小结

本章介绍了一个支持多种访问控制机制的基本访问控制系统，该系统为企业环境下的所有应用服务和资源提供集中的授权管理和授权服务。本基本访问控制系统为下文中设计的跨域访问控制系统提供原型，下文将在第三章理论模型的基本访问控制系统的基础上进行扩展，加入跨域授权功能。

第 5 章　跨域访问控制

本章将在上一章介绍的基本访问控制模型中加入跨域授权功能，实现共享资源间的安全互操作。本跨域授权模块依据第三章中我们提出的跨域授权理论模型进行具体实现。

5.1 跨域访问控制原理

在分布式技术不断发展的今天，由于业务需要很多时候我们必须去访问来自其它授权域的资源，但是由于资源提供者和使用者处在不同的授权域中，不同授权域间所采用的访问控制机制并不相同且策略的异构性使问题变得更加复杂，因此给域间的安全互操作带来了困难。本章介绍的跨域访问控制系统正是为解决这个问题而提出的。

目前解决上述问题的方法主要有两种：一种是第三章中介绍的通过映射机制来实现不同授权域间身份与权限信息的映射转换；另外一种则是引入全局用户和全局属性，各个进行联合的授权域组成了一个更大的逻辑上的授权域，全局用户和全局属性在这些联合授权域中有效。

针对第一种方法，由于身份与权限信息的映射关系错综复杂，为减轻各授权域的访问控制系统的压力和让系统间耦合程度降低，为此我们引入可信第三方跨域授权中介系统，它联合了存在联盟关系的各授权域，并为各授权域处理相关的映射转换功能。

针对第二种方法，在映射模型中引入的跨域授权中介系统作为可信第三方是所有进行联合的各授权域可信任的，因此在跨域授权中介系统中定义的用户和属性在所有授权域中都是有效的，我们称之为全局用户，全局属性。全局用户和全局属性的权限由各授权域的身份与权限管理员分配。全局用户或具有全局属性的用户可以在多个域间自由穿梭。

本章作者将在第四章基本访问控制系统的基础上实现一个支持跨域授权的访问控制系统。本跨域访问控制系统的逻辑结构图如图 5.1 所示。

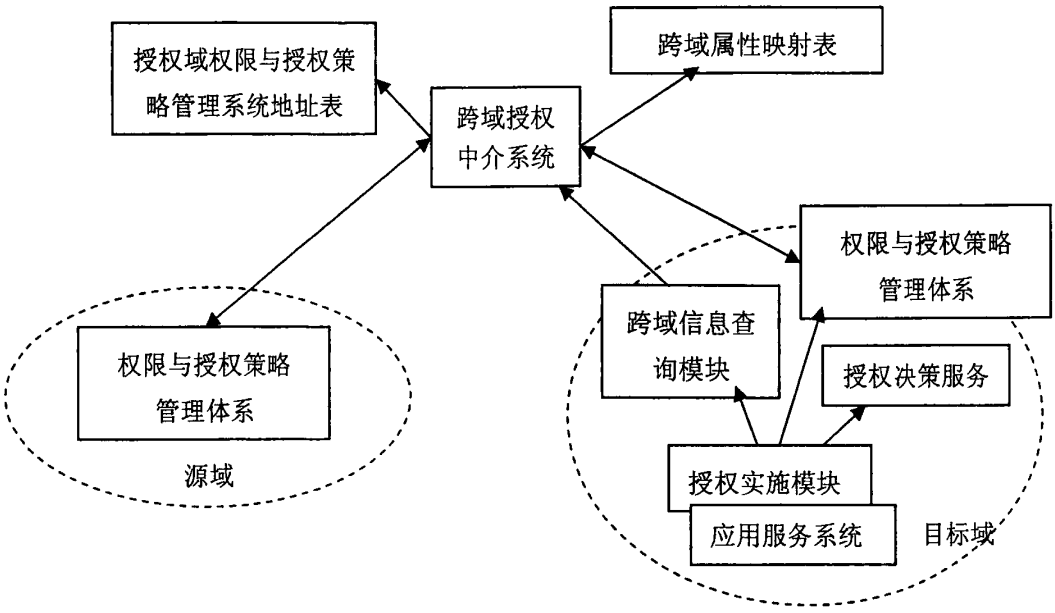

图 5.1 跨域访问控制模型的逻辑结构图

图中权限与授权策略管理体系包括 LDAP 数据库，身份与权限管理器和身份与权限服务器。

本跨域访问控制系统与基本访问控制系统的主要差别有：

（1） 身份与权限服务器中需要增加 SAML/SOAP/HTTP[44]方式对外提供用户属性信息查询的服务。OASIS 的 SAML 是一个 XML 框架，也是一组协议，可以用来在异构系统间传输安全断言，并且可以与 Web Services 的 SOAP 协议结合在一起。因此我们在身份与权限服务器中增加了一个基于 SAML/SOAP/HTTP 的用户属性信息查询服务。本模块基于 Java Servlet 和 OpenSAML 1.1[45]实现。它接收来自跨域授权中介系统的 SAML/SOAP/HTTP 请求，Servlet 接受请求后，使用 OpenSAML 解析 SAML 请求并通过调用本地的身份与权限服务器获得用户的属性信息，最后将用户属性信息封装成 SAML 断言返回跨域授权中介系统。

（2） 引入跨域授权中介系统。

跨域授权中介系统主要包含如下功能：

1）跨域授权中介系统联合了几个授权域，充当可信第三方。

2）跨域授权中介系统维持一个包含全局属性的LDAP数据库，并与各授权域同步全局属性信息，全局属性的权限由各授权域独自分配。

3）跨域授权中介系统负责制定和存储已联合的授权域间的属性映射信息。

4）通过SAML/SOAP/HTTP方式向用户所在域（源域）的身份与权限服务器查询用户属性信息。

5）包含一个映射模块，负责对用户属性进行映射转换。

6）跨域授权中介系统包含一个授权管理模块，它允许各授权域管理员添加域用户到中介LDAP，并赋予全局属性，使其成为全局用户。

7）跨域授权中介系统LDAP中只存储用户信息和全局属性信息，用户信息存放在以所在域域名标识的组织（Organization）下，各域管理员分别管理自己相应的域用户，并可以为用户分配全局属性。

（3） 授权实施模块需要增加控制跨域用户的功能。授权实施模块首先判断请求是否为外域用户提交的，若是则调用跨域信息查询模块，该模块将构造SAML 请求发送到跨域授权中介系统请求所需的外域用户属性信息；若否则与基本访问控制系统相同。

（4） 跨域信息查询模块。被授权实施功能模块调用（只有资源请求是来自外域用户才会被调用），它负责向跨域授权中介系统查询和获取跨域用户的权限信息。

5.2 跨域访问控制流程

对图 5.1，跨域访问控制的工作流程如下：

（1）来自某授权域（用户域）的用户通过网络访问本授权域（应用域）的应用系统。

（2）应用系统的授权实施模块拦截、检查服务请求，并通过 Session 机制确定该用户是否已完成身份鉴别，若是，则转入步骤 19；否则，继续到步骤 3。

（3）应用系统的授权实施模块通过一定方式完成对用户的身份鉴别，并确定用户是否为本授权域的用户，

（4）对于来自本授权域的用户，应用系统中的授权实施模块从本地的权限与授权策略管理系统查询、获得用户的角色（组 ID）信息，并将用户的身份 ID、角色（组 ID）信息保存在该用户的 Session 对象中，然后转入步骤 19；

（5）对于来自其他授权域的用户，应用系统中的授权实施模块通过一定的方式确定用户来自的授权域及身份标识（ID）信息。

（6）应用系统中的授权实施模块调用跨域授权信息查询模块，请求获取该

跨域用户的跨域权限信息。

（7）跨域授权信息查询模块向跨域授权中介系统请求查询该跨域用户的权限信息，请求中有跨域用户来自的授权域名（用户域）、用户 ID 及本授权域名（应用域）。

（8）跨域授权中介系统接收到跨域权限信息查询请求后，首先查看属性映射策略配置表中是否有从用户域（为源域）到应用域（为目标域）的映射策略，

（9）若无，则跨域授权中介系统返回查询失败，并给出失败理由（无域间映射策略）。

（10）若有，则跨域授权中介系统（依据用户域名）从授权域权限与授权策略管理系统地址表中找到用户所在域中的权限与授权策略管理系统地址，然后，向用户域中的权限与授权策略管理系统获取用户的角色、用户组信息，请求中有该用户 ID。

（11）用户域中的权限与授权策略管理系统接收到来自跨域授权中介系统的查询请求后，根据用户 ID 从其数据库中查找用户的所有属性信息，若有相应用户，则返回查询成功（查询结果空也是成功）及有关的查询结果；若找不到对应用户，则返回查询失败，给出原因（无对应用户）。

（12）跨域授权中介系统接收到来自用户域的用户属性信息查询结果后，判断查询结果是成功还是失败，

（13）若查询结果是失败，则跨域授权中介系统向应用域的跨域授权信息查询模块返回查询失败；

（14）若查询结果是成功，则跨域授权中介系统从属性映射策略配置表中，找到与用户域（为源域）和应用域（为目标域）对应的映射策略，取出一个个的属性映射元素，依据映射元素对从用户域获得的用户属性进行从源域到目标域的属性映射，最后将映射获得的所有用户属性集合进行整合，返回给应用域中的跨域授权信息查询模块。

（15）应用域中的跨域授权信息查询模块将从跨域授权中介系统获得查询结果返回给应用系统的授权实施模块。

（16）应用系统的授权实施模块获得了查询结果后，判断结果是成功还是失败，

（17）若失败，授权实施模块阻断用户的服务访问；

（18）若成功，授权实施模块将跨域用户经属性映射后所拥有的（本地和

全局）用户属性保存到该用户的 Session 对象中，然后继续。

（19）应用系统的授权实施模块请求授权决策引擎对用户的服务（资源）访问进行授权决策，请求中有从用户 Session 对象取出的用户属性信息以及用户要访问的服务（资源）名（即 URL）和相应的操作。

（20）授权决策引擎根据用户的本地和全局属性及从本地权限与授权策略管理系统获得的访问控制策略，做出允许或拒绝用户访问的决定，并将决定返回给应用系统的授权实施模块。

（21）应用系统的授权实施模块根据返回的授权决定结果进行访问控制实施（授权实施），允许或阻断用户的服务访问。

在步骤 2，授权实施模块通过 Session 机制确定用户是否已完成身份鉴别的方法如下。应用系统、Web 容器、或授权实施模块可基于一定的 Session 维护机制识别、跟踪不同的用户，并为每个用户创建一个 Session 对象用于存放服务状态和身份（授权）信息；对初次登录用户或未完成身份鉴别用户，其 Session 对象中没有相关身份（或授权）信息，或者信息不完全；因此，授权实施模块可据此判断用户是否已完成身份鉴别。

在步骤 3 中对用户的身份鉴别并确定其是否来自本授权域可按如下两种情形进行：

（1）身份域同授权域一致

这时可使用户提交的身份鉴别信息中包含域信息，如对于基于用户名/口令（包括动态口令）的身份鉴别，除了用户名（ID）、口令外，用户还需提交来自的域信息（类似于 Windows 中的域身份鉴别）；而对于基于数字证书的身份鉴别，数字证书中将直接或间接（如通过信任链间接指示）包含域信息。这样，授权实施模块可从鉴别信息中判断出用户是否为本授权域用户。对于跨域用户，可以利用已有的跨域身份鉴别技术，如 Kerberos，或通过 SAML、WS-Trust/WS-Federation 等，进行跨域身份鉴别。

（2）身份域同授权域不一致

这时通常是有一个全局身份 ID（如数字证书或其它全局 ID）在多个授权域中有效，在这种情形下，若通过该全局身份 ID 在本授权域的权限与授权策略管理系统中找不到对应的用户，则可以认为该用户为非本授权域的用户，而对该用户的身份鉴别可基于该全局身份 ID 进行，如通过验证证书信任链或采用其他适用的跨域身份鉴别技术。

在步骤5， 应用系统中的授权实施模块获取用户来自的授权域及身份标识（ID）信息的方式也可相应地分为两种情形：

（1）身份域同授权域一致

这对应于步骤3中身份域同授权域一致的情形，这时可从身份鉴别信息中直接或间接地获得授权域名及身份标识（ID）信息（对于数字证书的情形，可通过将证书信任链对应到相应的授权域，从而间接获得授权域名）。

（2）身份域同授权域不一致

这对应于步骤3中身份域同授权域不一致的情形，这时在基于全局身份ID完成身份鉴别后，可以弹出一个用户界面，让用户填写、提交其来自的授权域名。

5.3 系统实现

5.3.1 跨域信息查询模块

跨域信息查询模块负责向跨域中介系统查询经过映射转换后的外域用户属性信息。当外域用户访问本地资源的时候，授权实施模块对所有访问本地资源的请求进行拦截，若授权实施模块判断为来自外域用户的请求时，将调用跨域信息查询模块向跨域中介系统查询用户映射到本域的的用户属性和全局属性。

跨域信息查询模块实际上是跨域授权API，由授权实施模块进行调用。跨域授权API主要包含了如下方法：

```
public class CrossDomainAPI {
    //加载默认位置的配置文件，初始化跨域授权 API
    public static Result initCrossDomainAPI() {}
    //加载自定义的配置文件路径初始化跨域授权 API
    public static Result initCrossDomainAPI(URL configPath) {}
    //发送 SAML 请求，成功获得 SAML 响应后，解析 SAML 信息返回获得相
    //应的属性键值对
    public static Map<String, List<String>> crossDomainAttributeRequest(
            HttpServletRequest request) throws Exception {}
}
```

跨域授权API使用前必须经过初始化，进行API初始化时将加载默认或自

定义的配置文件，配置文件中定义了：

本域的授权域域名：本系统中授权域的标识借用 DNS 域名的形式，如 whut.edu.cn，itrus.com.cn 分别表示不同的授权域。这里域名仅作为一种授权域标识，不与 IP 地址对应。一个机构、组织通常可以取其网络系统的域名作为其授权域的域名。

跨域中介系统的地址：跨域授权 API 将构造 SAML 请求发送到该地址。

映射类型：跨域授权系统提供了四种类型的属性映射类型，分别是：

（1）Global_Global：映射结果包含用户在跨域授权中介系统中的全局属性和用户局部属性所映射的全局属性。

（2）Global_ ：映射结果只包含用户在跨域授权中介系统中的全局属性（该用户必须是全局用户）。

（3）Global_Local：映射结果包含用户在跨域授权中介系统中的全局属性和用户局部属性所映射的目的域（目标域）局部属性。

（4）_Local ：映射结果只包含用户局部属性所映射的目的域（目标域）局部属性。

映射属性：表示目标域在进行授权时所需要的属性，这些属性从跨域中介系统那获取。

密钥设置：配置用于签署 SOAP 消息的密钥信息，系统采用 JAVA 的 Keystore 方式存储密钥。

配置文件采用 Properties 格式（键值对）进行存储，具体格式如下：

```
domainName=WHUT.edu.cn  //标识出本授权域域名
mediatorURL=http://mediatorURL/MediatorService  //跨域中介系统地址
mappingType= Global_Local  //映射类型
mappingAttributes=role, group  //授权所需属性，不同属性之间用“,”分隔
keystore=domainA.keystore  //签名使用的密钥库
keystore.storepass=****** //密钥库储存密码
keystore.keypass=******  //私钥密码
```

跨域授权 API 初始化完成后，跨域信息查询模块根据配置文件和用户的提交数据构造 SAML 请求发往跨域中介系统。SAML 请求的格式参见附录。

跨域中介系统在正确接收到来自源域的用户属性断言后，分离出用户属性信息并调用映射提供者进行属性映射，最后将映射结果封装成SAML响应返回给

目标域的跨域权限信息查询模块。SAML响应格式参照附录。

5.3.2 跨域授权中介系统

跨域授权中介系统在现实中它可能是某种行业协会或因为某种利益结合在一起的合作联盟。它是实现跨域访问控制的桥梁，它不仅可以通过定义全局用户和全局属性来实现跨域授权也可以通过属性间的映射实现不同授权域中权限信息的对应和转换。

跨域授权中介系统的结构如下图所示，由中介授权管理，中介 LDAP 数据库，中介服务模块，用户权限信息查询模块，映射提供者（Mapping Provider），授权域权限与授权策略管理系统地址表，映射策略配置管理模块，用户属性映射策略表及一系列映射策略文件组成。

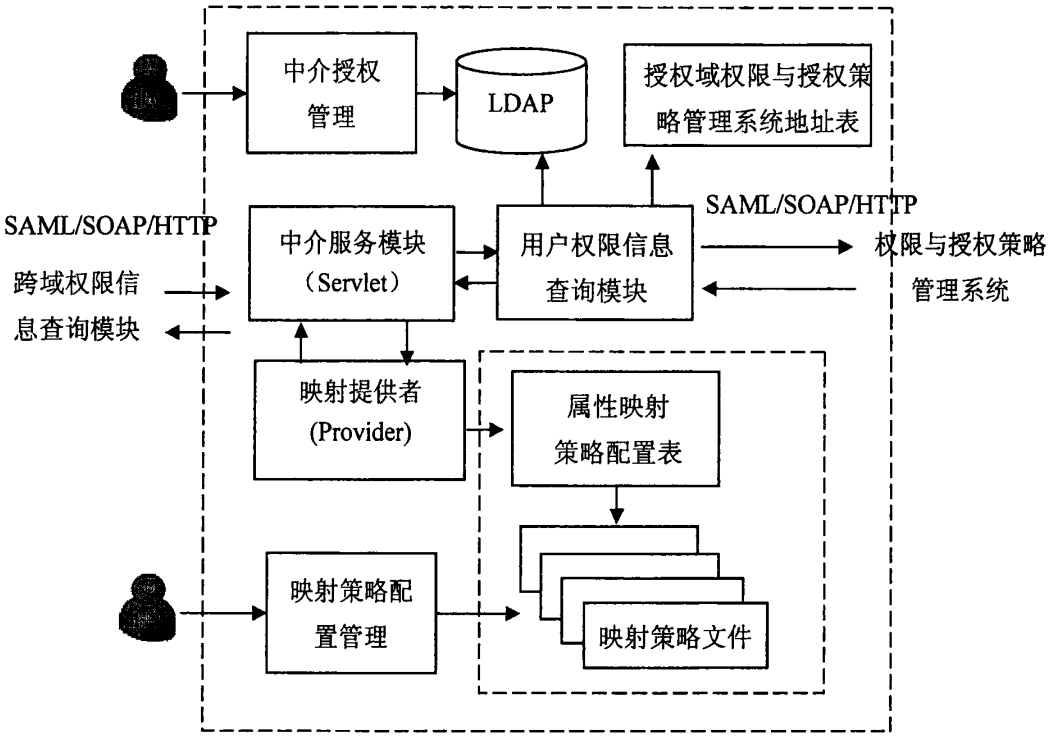

图 5.2 跨域授权中介系统的结构图

跨域授权中介系统中的各模块具体实现如下：

（1） 中介授权管理和中介 LDAP 数据库

中介 LDAP 数据库中存储有全局用户和全局属性信息，由于角色具有继承性并且角色中包含权限信息，所以全局角色信息将与中介系统中所有参与联合

的授权域中的 LDAP 进行同步，其他全局属性自身并不具备权限且无继承性，则无需进行同步。中介系统只是负责建立全局属性（如全局角色，全局组等），全局属性的具体权限由各个授权域自己定义。中介 LDAP 的结构图如下所示：

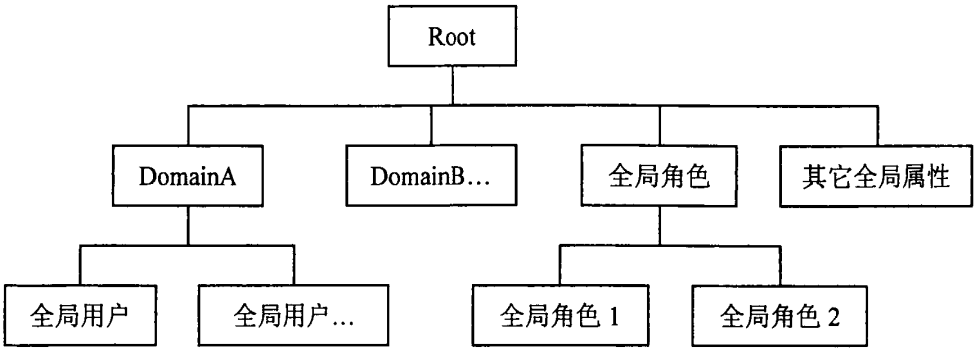

图 4.15 中介 LDAP 结构

中介授权管理负责维护全局用户，定义和分配全局属性。各个授权域管理员各自管理自己所在域目录下的全局用户，管理员可以将本域用户存放在中介 LDAP 中并为其赋予全局属性，使其成为一名全局用户。

（2） 中介服务模块

中介服务模块是跨域授权中介系统供各授权域跨域授权信息查询模块调用的接口，它基于 Java Servlet 实现，各联合授权域的跨域信息查询模块通过调用它来获取跨域用户的属性信息。中介服务模块工作的工作步骤如下：

1） 接收来自目标域跨域信息查询模块发来的 SAML/SOAP/HTTP 请求，根据目标授权域域名查找目标域公钥验证请求签名信息；

2） 验证签名有效，中介服务模块将根据请求中用户所在的授权域（源域），向用户权限信息查询模块查询请求用户属性信息；

3） 若成功返回用户属性信息，中介服务模块将根据请求中的映射方式执行不同操作，若映射方式为 Global_（即只根据全局用户的全局属性来进行授权）则直接将全局属性返回；若为另外三种方式则将从源域获得的用户属性集传入映射提供者模块；

4） 若映射提供者正确返回映射结果，中介服务模块将映射结果中目标域所需的属性封装成为 SAML 响应结果返回目标域的跨域信息查询模块。

5） 若步骤中有一步出错则返回目标域的跨域信息查询模块空值。该模块

中所使用的 SAML 请求响应消息格式参见附录。

（3） 授权域权限与授权策略管理系统地址表

中介服务模块在向用户所在的授权域中查询用户属性信息时，如何确定其所在授权域的身份与权限服务器地址？跨域中介系统中包含一个授权域权限与授权策略管理系统地址表，该表保存了与跨域授权中介系统相联的每个授权域中权限与授权策略管理系统的身份与权限服务器的 URL 地址。添加或移除授权域必须对地址表进行相应的添加删除操作。地址表采用 Java 的属性文件（键值对表示形式，通过 java.util.Properties 类加载）描述，其信息结构如下表所示。

表 5.1 授权域权限与授权策略管理系统地址表

授权域名称 （Authorization Domain）	用户信息服务查询服务地址 （Query Service Address）
whut.edu.cn（DNS 域名）	http://somehost/QueryService
itrus.com.cn	http://otherhost/QueryService
…	…

（4） 用户权限信息查询模块

用户权限信息查询模块功能：根据用户的身份标识、所在域名到用户所在授权域获取请求中所需要的用户属性信息。接口形式如下：

```
public interface IAttrbQuery {
    public Map<String, List<String>> attrbQuery(SAMLRequest request);
}
```

方法 attrbQuery(…) 负责连接用户所在授权域获取用户属性信息，方法参数为来自目标域的 SAML 请求。

用户权限信息的查询需根据目标域所采用的映射类型来决定（参见 5.3.2 节）。映射类型一共分为：Global_，Global_Global，Global_Local，_Local。

Global_：表示用户权限信息查询模块只需从中介 LDAP 中获取该全局用户的全局属性信息返回即可，若不存在该全局用户则返回空值。

其他的三种映射方式则需要从源域获取用户的属性信息。用户权限信息查询模块的工作步骤如下：

1） 分离出目标域请求中的 Subject（主体，包含用户信息，用户名或数字证书），AttributeDesignator（属性选择器，指定了目标域所需的用户属性信息）；

2） 根据 Subject 和 AttributeDesignator 元素重新构造 SAML 请求响应，并

使用中介系统的密钥进行签名；

3） 查找授权域权限与授权策略管理系统地址表中源域所对应的身份与权限服务器地址，并将上步中的 SAML 请求发送到该地址。

4） 若成功获得 SAML 响应，签名验证成功后分离出 SAML Response 中的用户属性并封装到 Map 中返回给中介服务模块。

（5） 属性映射策略配置表

属性映射策略配置表中配置了从源授权域到目标授权域的映射策略，其逻辑结构如下表所示。

表 5.2 属性映射策略配置表

源授权域	目标授权域	映射策略	映射类型	映射提供者
xxx.xxx.xxx （DNS 域名）	yyy.yyy.yyy （DNS 域名）	zzz.xml （文件名）	Global_Local （三种类型）	MapProvider
…	…	…	…	…

这里的一个映射策略项本身不包含映射策略的具体内容，而是一个或几个映射策略文件名。由于 XML 的良好可读性和扩展性，这里属性映射策略配置表采用 XML 格式。一个映射策略项中可以包含多个映射策略，一个映射策略文件中定义了一个域（源域）中用户属性到另一个域（目标域）中用户属性的映射关系。每个映射策略又由一系列的映射元素（mapping element）构成，它将源域中的一个用户属性集映射到目标域中的一个用户属性集。映射关系是不可逆的，单向的，即域 A 到域 B 的映射，不一定就是域 B 到域 A 映射的反映射。这里我们的映射策略文件也是采用 XML 文档格式。映射策略的格式如下所示：

```
<mapPolicy>
  <policyElement>
    <request-attribute>
      <attribute>level>10</attribute>
      <attribute>age>40</attribute>
    </request-attribute>
    <response-attribute>
```

```
            <attribute>role=vip</attribute>
        </response-attribute>
    </policyElement>
    <policyElement>...</policyElement>...
</mapPolicy>
```

上述 mapPolicy 元素表示一个映射策略，policyElement 元素表示映射元素，例子中所示的映射元素表示用户属性 level>10 并且 age>40 的源域用户可以映射为目的域的 vip 角色。

映射类型中由于 Global_无需执行映射操作，所以这里的取值只有三种类型分别是：

Global_Global：除了从中介 LDAP 获取全局属性外，还需要进行源域属性到全局属性的映射。

Global_Local：除了从中介 LDAP 获取全局属性外，还需要进行源域到目标域的属性映射。

_Local：只进行源域到授权域的属性映射。

（6） 映射策略配置管理模块

映射策略的配置表在映射过程中起决定作用，因此映射策略配置表的制定必须通过安全的方式来管理，映射策略配置管理模块向管理员提供一个 Web 界面用于制定用户属性的映射策略。需要进行跨域属性映射双方的一方管理员登陆本系统生成一个属性映射策略并对其进行签名，只有在对方管理员检查并签名后，本映射策略文件方可生效。系统在进行属性映射的时候，必须检查映射策略文件的签名信息，只有数字签名验证通过，映射策略才会被启用。

（7） 映射提供者（Mapping Provider）

映射提供者是跨域授权中介系统的关键组件，它负责完成相关的属性映射和转换功能。当中介服务模块调用映射模块进行属性转换时，它将根据源域和目的域信息查找相关的属性映射文件并对用户属性进行相应转换。下图所示为映射提供者结构图。

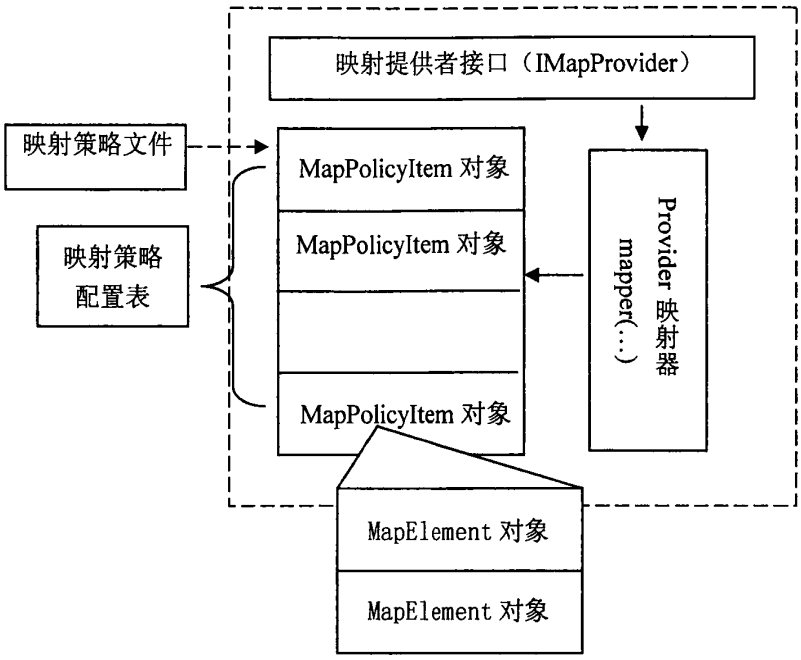

图 4.16 映射提供者结构图

映射提供者在启动时将根据属性映射策略配置表把相关的映射策略加载进内存进行缓存，在往后的调用中将直接查找缓存，提高了查询效率。

映射提供者通过如下 Java 接口 IMapProvider 对外提供映射服务：

```
public interface IMapProvider {
    public boolean domainsMatch(String orgnDomain,
                                String trgtDomain,
                                String mappingType);
    public Map<String, List<String>> mapper(String orgnDomain,
                                            String trgtDomain,
                                        Map<String, List<String>> orgnAttrs);
}
```

方法 domainsMatch(String orgnDomain, String trgtDomain, String mappingType)判断是否存与源授权域 orgnDomain，目标授权域 trgtDomain 和映射类型对应的映射策略，若有，返回 TRUE，否则，返回 FALSE。若返回 FALSE 则跨域中介系统直接向目标授权域返回空值。

方法 mapper(String orgnDomain, String trgtDomain, Map<String,

List<String>> orgnAttrs)进行用户属性映射，其输入、输出参数意义如下：

输入参数 orgnDomain, trgtDomain 分别存放源授权域名与目标授权域名；

输入参数 orgnAttrs 表示用户在源授权域中的用户属性集合。

本跨域授权系统中提供了两个 IMapProvider 的实现：

GroupRoleMapProvider：负责执行用户角色（组）交叉映射（参见 3.2 节）

AttributeMapProvider：负责执行属性映射（参见 3.3 节）

为支持自定义的映射策略，用户可以自定义 IMapProvider 的实现，通过配置到属性映射策略配置表即可生效。

映射提供者采用面向对象的方法进行设计。在 MapProvider 实现类中包含一个映射策略列表：mapPolicyItemList。符合 domainsMatch 方法的映射策略被存储在 mapPolicyItemList 中。每个映射策略将被初始化为 MapPolicyItem 类，MapPolicyItem 中包含了多个 MapElement（映射元素）。当调用 MapProvider 的 mapper 方法时，方法首先循环调用 mapPolicyItemList 的每个 MapPolicyItem 的 mapper 方法，MapPolicyItem 又调用映射策略中各个映射元素 MapElement 的 mapper 方法。最后将各个映射元素的映射结果进行整合返回中介服务模块。

映射模块的执行流程为：

（1） 映射提供者在启动时将根据属性映射策略配置表把相关的映射策略加载进内存进行缓存。在此过程中映射提供者根据源域与授权域双方的公钥验证各个映射策略文件的签名信息。经过签名验证的映射策略才会被加载。

（2） 映射模块被中介服务系统调用时，映射模块首先根据传入的源授权域，目标授权域和映射类型查找属性映射策略配置表（调用 domainsMatch 方法）。

（3） 若不存在相关映射策略，映射提供者将返回空值，若存在相关映射策略则映射提供者将各映射策略的返回结果进行合并，返回中介服务模块。

5.4 应用场景

本节将通过实例来展示本跨域访问控制系统。不同授权域间的安全互操作包含两种方式：基于全局属性和基于源域局部属性的映射。

本应用场景包含如下几个参与者：

Dihin：来自 HUST 的用户

HUST.EDU.CN：源域，用户所在域

WHUT.EDU.CN：目标域，资源提供者

Resource.jsp：目标域中对外提供服务的一个页面。

本应用场景图如下图 4.17 所示：

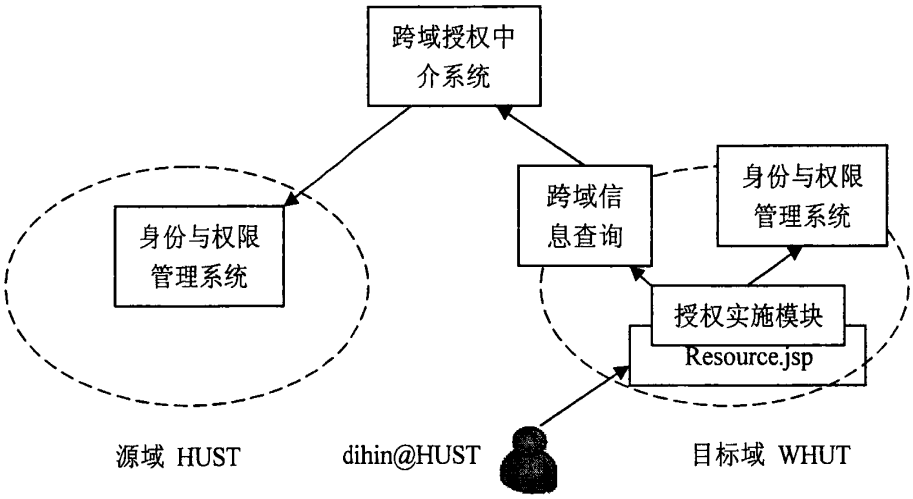

图 4.17 应用场景流程图

在 WHUT.EDU.CN 的授权管理中管理员为 resource.jsp 设置了如下策略：用户组 guest 和全局角色 student@global 读取权限。如下图 4.18 和图 4.19 所示。

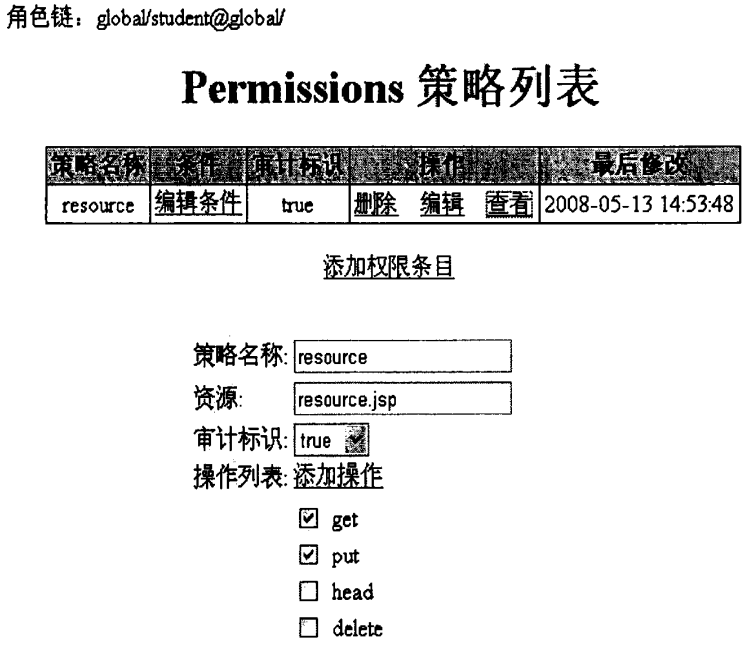

图 4.18 全局角色的赋权

资源位置：Resources/resource.jsp/

ACL 策略列表

策略名称	条件	继承标识	审计标识	操作	最后修改
acl	编辑条件	true	true	删除 编辑 查看	2008-05-14 11:08:23

添加策略条目

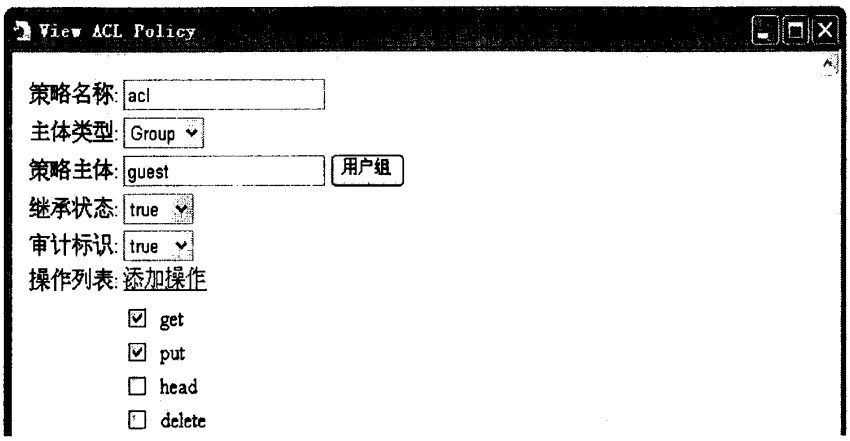

图 4.19 ACL 策略设置图

本跨域访问控制系统在用户经过身份鉴别后被调用，因此用户 Dihin 在访问 WHUT.EDU.CN 的 resource.jsp 的时候必须先经过身份鉴别，系统将用户重定向到源域进行身份鉴别，如图 4.20 所示。

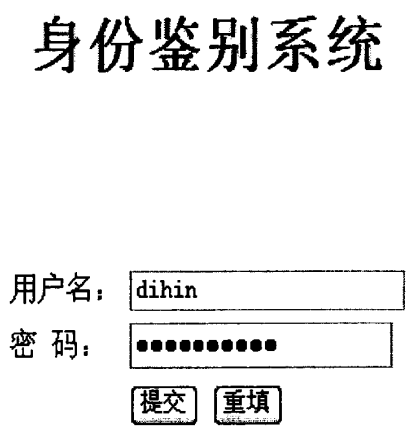

图 4.20 用户身份鉴别

完成跨域鉴别后，WHUT.EDU.CN 的授权实施模块拦截到用户请求并通过用户会话中已鉴别的身份信息判断用户是否来自其它授权域，若来自其它授权域，授权实施模块弹出页面要求用户选择所在授权域，如图 4.21 所示：

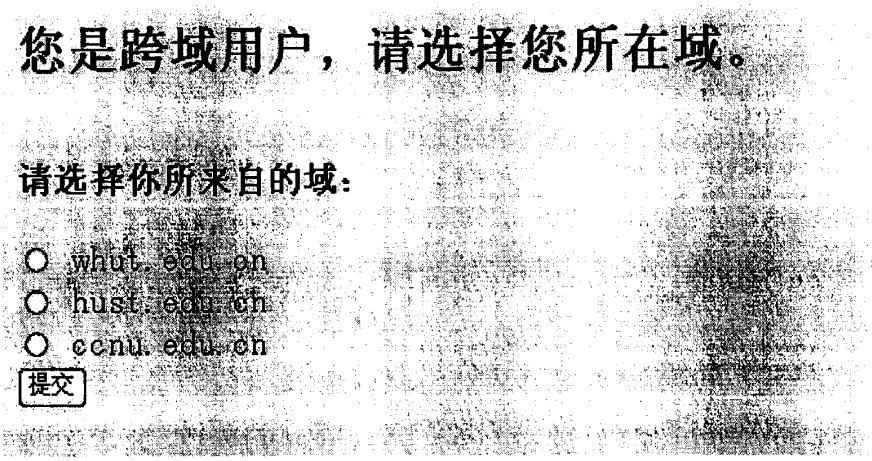

图 4.21　授权域选择页面

选择 HUST.EDU.CN 域后，跨域信息查询模块将发送请求到跨域授权中介系统请求来自 HUST.EDU.CN 的用户属性信息。

根据映射方式的不同，跨域授权中介系统将进行不同的处理，本节将通过两个典型的实例来展示本跨域访问控制系统：

第一种方式是通过全局角色进行授权。跨域授权中介系统直接查询用户的全局角色属性并响应目标域，目标域根据全局角色对用户进行授权，无需进行角色间的映射。在中介的授权管理中可以看到 dihin 用户被 HUST.EDU.CN 的管理员设置为全局用户，中介服务模块查询中介 LDAP 中用户 dihin 的全局角色属性，该用户拥有全局角色 student@global 和 teacher@global，如图 4.22 所示。然后将全局角色属性通过 SAML 响应返回 WHUT.EDU.CN 的跨域信息查询模块。全局角色的权限由各授权域赋予，假设 student@global 和 teacher@global 在 WHUT.EDU.CN 中被设置为冲突角色（即违背了动态职责分离），则将弹出角色选择页面如图 4.23 所示。用户 dihin 选择了 student@global 角色，WHUT.EDU.CN 的授权管理为全局角色 student@global 赋予了 resource.jsp 的访问权限，如图 4.18 中所示。

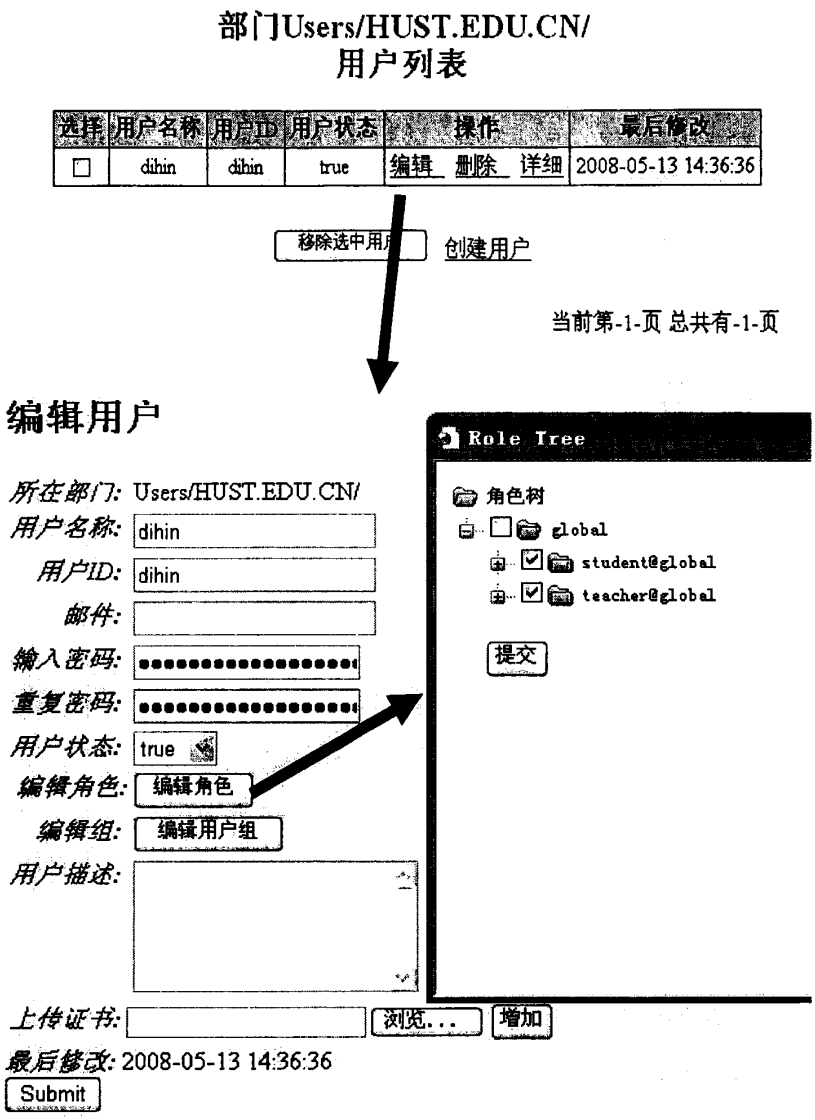

图 4.22 中介管理系统中的全局用户

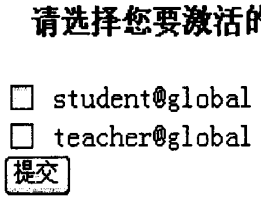

图 4.23 角色激活页面

因此用户 dihin 将获得查看 resource.jsp 的权限，如下图 4.24 所示：

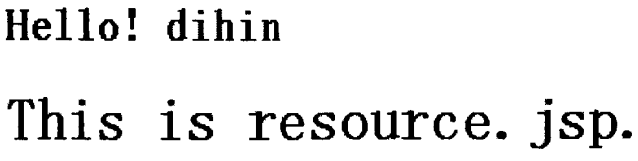

图 4.24 授权成功图

若删除上图 4.18 中全局角色的权限，则返回拒绝页面，如图 4.25 所示：

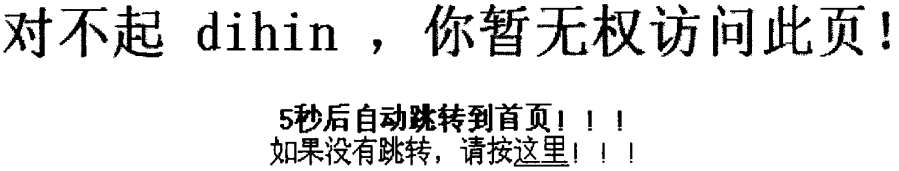

图 4.25 授权失败图

第二种方式是查询用户所在域（源域）的用户属性，然后调用映射提供者模块将源域的用户属性转换为目标域可识别的用户属性（包括全局属性和目标域属性），本例以用户组，角色交叉映射进行说明。dihin 在 HUST.EDU.CN 中的用户信息如图 4.26 所示。

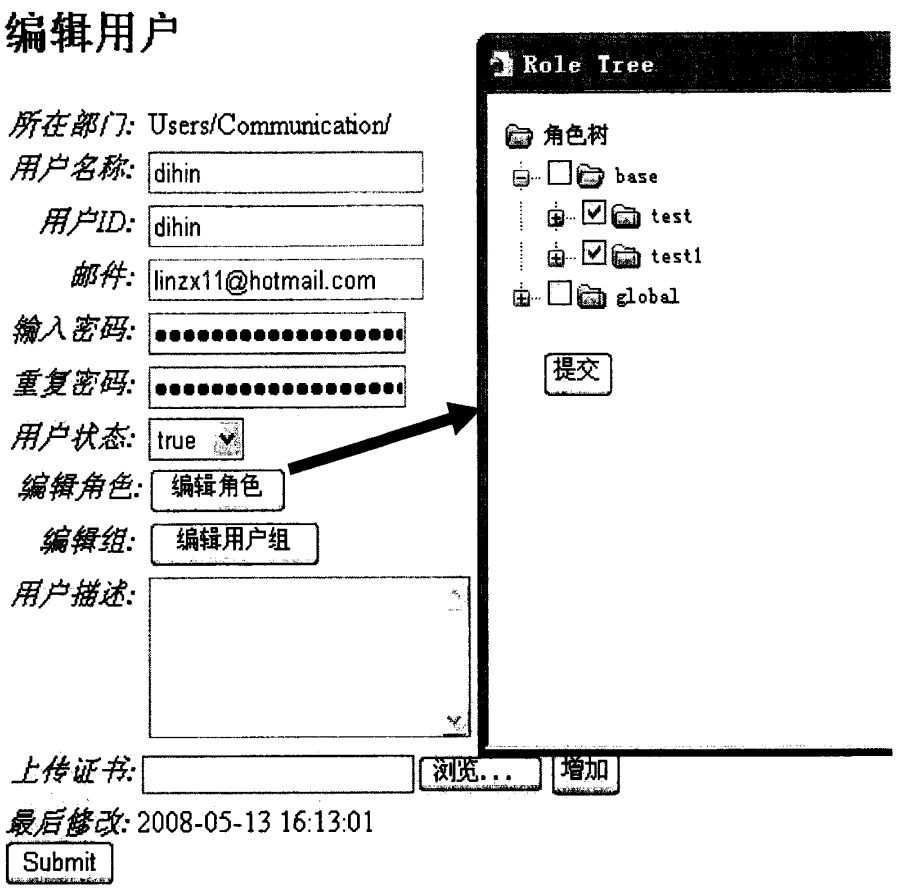

图 4.26 用户信息页面

用户 dihin 拥有 test 和 test1 两个角色，HUST.EDU.CN 身份与权限服务将包含用户属性的 SAML 断言响应跨域授权中介系统。跨域授权中介系统的映射提供者将对收到的角色属性进行转换。映射提供者查找属性映射策略配置表（XML 格式，如图 4.27 所示）

```
<map-item>
   <source-domain>HUST.EDU.CN</source-domain>
   <response-domain>HUST.EDU.CN</response-domain>
   <mapping-mode>Global_Local</mapping-mode>
   <mapper>GroupRoleMapperProvider</mapper>
   <map-policy>mappolicy/HUST2WHUT.xml</map-policy>
</map-item>
```

图 4.27 属性映射策略配置表

通过匹配源域和目标域查找相应的映射方式，映射提供者和映射策略。符合本请求的映射策略是 HUST2WHUT.xml，映射策略的格式如图 4.28 所示。

```
<mapPolicy>
  <policyElement>
    <request-attribute>
          <attribute>role=test</attribute>
          <attribute>role=test1</attribute>
     </request-attribute>
     <response-attribute>
          <attribute>groupID=guest</attribute>
     </response-attribute>
  </policyElement>
</mapPolicy>
```

图 4.28 映射策略文件

如图所示，来自源域的 test1, test 角色被映射为目标域的 guest 用户组，由于在目标域中给 resource.jsp 中包含一条 ACL 策略，允许 guest 用户组访问本资源，如下图 4.19 所示。

因此，来自 HUST.EDU.CN 的用户 dihin 将拥有访问 WHUT.EDU.CN 域中 resource.jsp 的权限。授权成功返回如图 4.24 所示，授权失败如图 4.25 所示。

5.5 本章小结

本章在第三章介绍的理论模型的基础上对基于映射方式的跨域授权模型进行具体实现。本系统通过引入跨域授权中介系统来实现跨域访问控制系统。本跨域访问控制系统支持两种类型的跨域授权，一是基于全局用户和全局属性的方式，另一种则是基于映射的跨域授权方式包括用户角色（组）交叉映射和基于属性的映射方式。

第 6 章　总结与展望

6.1 研究工作总结

随着分布式技术的不断发展，异构系统间的资源共享的需求也在不断提升，这就给我们带来了很多安全方面的问题，而访问控制作为 5 大安全服务之一是分布式网络中资源共享的关键问题。现有的信息网络是一个由多种访问控制机制和多种安全策略的异构授权域组成的环境，不同的域可能有不同的访问控制目标和要求。资源共享要求用户能够跨越授权域对资源进行访问，因此如何在不改变本地的访问控制机制，安全策略和保持本地自主性原则的基础上实施跨域授权就变得极为重要。分布式环境下的身份鉴别和访问控制一直都是分布式安全研究的重点，由于跨域访问控制必须建立在身份鉴别的基础上，且由于访问控制机制和权限格式的差异，这给跨域授权带来了更大的困难。由于 RBAC 的众多优点和广泛使用，目前的跨域授权大部分集中在通过角色映射来实现 RBAC 访问控制系统间的安全互操作，但是并不能解决在采用不同访问控制机制系统之间的安全互操作。本文的研究成果主要包含如下两点：

（1）　提出了三种跨域授权理论模型，一是在前人的研究基础上给出一种基于角色映射的跨域授权模型，该模型适用于 RBAC 访问控制系统间的安全互操作；二是提出了一种用户角色（组）交叉映射的跨域授权模型，该模型解决了 ACL 与 RBAC 访问控制系统之间的安全互操作；三是作者在上述两种模型的基础上进行抽象，并首次提出了一种基于属性映射的跨域授权理论模型，该模型适用于所有基于主体属性授权的访问控制系统之间的安全互操作。

（2）　提出通过引入跨域授权中介系统研究开发了一套实现上述跨域授权理论模型的跨域访问控制系统。本访问控制系统为一定范围内的资源提供集中的授权管理和服务，系统支持多种访问控制机制（ACL，RBAC，ABAC）并具有扩展性。跨域授权中介系统是实现跨域访问控制的桥梁，它为各授权域提供权限信息的转换服务。

6.2 进一步研究方向

虽然目前跨域访问控制系统在功能上已经基本实现了，但是作为一个成熟的访问控制系统在细节上还远远不够，并且在一些方面上还存在着一些不足，主要包含如下几方面：

（1） 管理员分区域分层次的管理仅仅依靠LDAP的访问控制机制粒度比较粗，并且由于不同的LDAP数据库产品具有不同的访问控制策略语法，在进行移植的时候，就必须重设管理员权限。因此引入一种与LDAP数据库松散耦合的分区域分层次的管理机制十分必要。

（2） 目前系统的策略条件只支持时间限制，在往后的工作中将支持其他的限制条件，并且增强限制条件的扩展性，允许用户自定义限制条件。

（3） 加入XACML策略。OASIS XACML是为增强分布式系统的访问控制互操作性而提出的。在后续的开发中系统将支持XACML策略，基于XML的XACML为我们提供一种通用的，可扩展的访问控制策略语言，其天生就支持基于属性的访问控制机制的特性和XML的描述格式在分布式环境的访问控制中将起到重要的作用。

（4） 优化授权决策服务器，跨域中介系统映射模块的执行效率。

（5） 提供更友好的管理员操作界面。

参考文献

[1] IBM Tivoli Access Manager WebSEAL 概述 [EB/OL].
http://publib.boulder.ibm.com/tividd/td/ITAME/SC32-1134-01/zh_CN/HTML/amweb41_admin04.htm

[2] http://www.microsoft.com/windowsserver2003/gpmc. Windows Server 2003 Active Directory 技术概述白皮书

[3] 赵铭 黄慧山 基于活动目录的集成验证授权中心[J] 广东电力 2006 年 7 月 第 7 期 19 卷

[4] SUN Java System Access Manager [EB/OL].
http://www.sun.com/software/products/access_mgr/index.jsp

[5] WebLogic 安全服务体系结构[EB/OL].
http://edocs.bea.com.cn/wls/docs92/secintro/archtect.html

[6] Trusted Computer System Evaluation Criteria (TCSEC). Department of Defense. December 1985
http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html

[7] D. F. Ferraiolo, D. R. Kuhn. Role Based Access Control. 15th National Computer Security Conference. 1992

[8] R.S.Sandhu, E.J.Coyne, H.L.Feinstein, C.E.Youman. Role-Based Access Control Models. IEEE Computer 29(2): 38-47, IEEE Press. 1996
http://csrc.nist.gov/rbac/sandhu96.pdf

[9] 隋韦韦，企业环境下基于角色与任务的访问控制研究，青岛大学硕士学文论文，2007

[10] PEARLMAN L, WELCH V, FOSTER I, et al. A community authorization service for group collaboration[C]. Proceedings of the 3rd IEEE International Workshop on Policies for Distributed Systems and Networks. Los Alamitos, Cal., USA: IEEE Computer Society Press, 2002: 50-59.

[11] CHADWICK D, OTENKO A, BALL E. Role-based access control with X.509 attribute certificates [J]. Internet Computing, 2003, 7(2):62-69.

[12] JOHNSTON W, MUDUMBAI S, THOMPSON M. Authorization and attribute certificates for widely distributed access control[C] Proceedings of the 3rd IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. Los Alamitos, Cal., USA: IEEE Computer Society Press, 1998: 340-345.

[13] CHADWICK D W, OTENKO A. The PERMIS X.509 role based privilege management infrastructure [J]. Future Generation Computer Systems, 2002, 19(2):277-289.

[14] Apu Kapadia, Jalal Al-Muhtadi, IRBAC 2000: Secure Interoperability Using Dynamic Role Translation, University of Illinois, 2000

[15] Liang Chen and Jason Crampton, "Inter-domain Role Mapping and Least Privilege", Proceedings of the Symposium on Access Control Models and Technologies (SACMAT), 2007

[16] T. L. Prasanna Venkatesan. A Ranking Based Cross Domain Role Mapping and Authorization Architecture for Grid Computing Systems [EB/OL].
http://hipc.org/hipc2007/posters/role-mapping.pdf

[17] 徐 云，肖田元. 基于角色映射的跨平台授权研究[J]. 计算机集成制造系统. 2007.9

[18] eXtensible Access Control Markup Language, Version 2.0, OASIS Standard, February 2005.
http://docs.oasisopen.org/xacml/2.0/access control-xacml-2.0-core-spec-os.pdf

[19] SAML 2.0 profile of XACML v2.0 OASIS Standard, 1 February 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf

[20] 访问控制安全机制及相关模型[EB/OL].
http://www.infosecurity.org.cn/article/pki/accessctrl/24070.html

[21] Department of Defense, Trusted Computer System Evaluation Criteria, DOD 5200.28-STD Dec.1985

[22] Butler W. Lampson. Protection. Proc. 5th Princeton Conf. on Information
http://research.microsoft.com/~lamnson/09-Protection/Acrobat.pdf

[23] D.Bell and L.LaPadula, Secure Computer Systems: Unified Exposition and Multics Interpretation, Technical Report MTR-2977 Rev.1, MITRE Corporation, Bedford, Mar.1975.
http://csrc.nist.gov/rbac/Role_Based_ccess_ontrol-1992.htm1

[24] David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli. Role-Based Access Control. [M].

[25] 沈海波，洪 帆. Web 服务中结合 XACML 的基于属性的访问控制模型[J]. 计算机应用，2005.12.

[26] 沈海波，洪　帆. 面向 Web 服务的基于属性的访问控制研究[J]. 计算机应用，2005.12

[27] 沈海波，洪 帆. 基于属性的授权和访问控制研究[J]. 计算机应用，2007 年 1 月第 27 卷第 1 期

[28] 刘晓妮，多域访问控制模型研究，山东大学硕士学位论文，2007

[29] Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0 [EB/OL]. OASIS Standard, 15 March 2005
http://docs.oasis-open.org/security/saml/v2.0/

[30] Liberty Alliance ID-FF 1.2 Specifications [EB/OL]. 2005.5
https://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_ff_1_2_specifications

[31] Web Services Federation Language [EB/OL]. 2006.12
http://www.ibm.com/developerworks/library/specification/ws-fed/

[32] OASIS Web Services Trust Language [EB/OL]. 2007.3
http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.doc

[33] John Hughes, Eve Miler. Security Assertion Markup Language (SAML) 2.0 Technical Overview [EB/OL]. 2006.10
http://www.oasis-open.org/committees/documents.php?wg-abbrev=security

[34] 陈婕，跨与授权管理系统的研究与实现，硕士学位论文，2007

[35] Internet Engineering Task Force (IETF) RFC4510 [EB/OL].
http://tools.ietf.org/html/rfc4510

[36] Remote Method Invocation [EB/OL].
http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp

[37] 架构 Web Service: 什么是 Web 服务[EB/OL].
http://www.ibm.com/developerworks/cn/webservices/ws-wsar/part2/

[38] OpenLDAP 2.2 Administrator's Guide [EB/OL].
http://www.openldap.org/doc/admin22/

[39] Ben Alex. Acegi Security Reference Documentation [EB/OL].
http://www.acegisecurity.org/guide/springsecurity.html

[40] Mattias Arthursson, Ulrik Sandberg, Eric Dalquist. Spring LDAP Reference Documentation [EB/OL].
http://static.springframework.org/spring-ldap/docs/1.2.1/reference/

[41] XFire User Guide [EB/OL]. http://xfire.codehaus.org/User%27s+Guide

[42] Web Services Security: SOAP Message Security 1.1 (WS-Security 2004) OASIS Standard Specification, 1 February 2006

http://docs.oasis-open.org/wss/v1.1/

[43] Saul Qunming Yuan. Using Acegi With XFire For Web Service Authentication & Authorization [EB/OL].

http://www.jroller.com/sqyuan/entry/using_acegi_for_authentication_authorization

[44] Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) V1.1 OASIS Standard [EB/OL], 2 September 2003

http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

[45] OpenSAML User Manual [EB/OL].

https://spaces.internet2.edu/display/OpenSAML/OSTwoUserManual

[46] Mark Bartel, John Boyer, Barb Fox. XML-Signature Syntax and Processing W3C Recommendation 12 February 2002 [EB/OL].

http://www.w3.org/TR/xmldsig-core/

致 谢

在本文完成之际，首先要感谢我的导师龙毅宏教授，是他给我提供了进入国家大型项目的机会，并目在课题的分析、设计以及论文的撰写过程中给了我悉心的帮助和指导。龙老师渊博的知识、开阔的思路、严谨的治学态度和忘我的工作精神使我学到了许多书本上学不到的东西，让我终生受益。在此，表示最衷心的感谢。

感谢在课题研究和系统开发过程中，与我朝夕相处的同学们，他们是：高超，黄朝，吴硕，汪克炎，潘丹。大家在工作上和生活上互相帮助，互相支持。我非常怀念与他们工作生活的这段日子。

感谢父母多年来的养育之恩，感谢他们对我每一次选择的全力支持，正是由于他们无微不至的关怀与体谅使我能够在这三年来静心学习，潜心研究，感谢他们给予我最强大的精神支柱。

感谢我的室友高庆，赵春雷，桂树同学，七年的同学经历让我们情同手足，感谢他们七年来在学习和生活中给予我的巨大帮助，希望将来他们都能在各自岗位上干出自己的一番事业。感谢荣于谦同学，谢谢你陪我度过这三年的美好时光。

衷心感谢所有关心和帮助过我的人！

衷心感谢各位专家百忙之中对本文的审阅和赐教！

攻读硕士学位期间发表的论文和参与的项目

[1] 林智鑫，基于 XACML 的 RBAC 授权机制，中国科技论文在线，2008 年 4 月

[2] 参与科技部 “国家科技支撑计划” （2006BAH02A03），“现代服务业共性服务集成化技术” 项目

附 录

发往跨域中介系统的SAML请求格式如下：

```
<Request
    RequestID="…. "
    MajorVersion="1"
    MinorVersion="1"
    IssueInstant="…"
    xmlns="urn:oasis:names:tc:SAML:1.0:protocol"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        ……//签名信息
    </ds:Signature>
    <AttributeQuery resource="目标域（应用系统所在域）">
        <Subject xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
            <NameIdentifier
                Format="……"
                NameQualifier="源域（用户域）">
                用户名称
            </NameIdentifier>
            <saml:SubjectConfirmation>
                    <saml:ConfirmationMethod>
                    urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
                </saml:ConfirmationMethod>
                <ds:KeyInfo>
                        <ds:X509Data>
                        <!—X.509证书-->
                        <ds:X509Certificate>
```

```
                    ...
                    </ds:X509Certificate>
                    </ds:X509Data>
                </ds:KeyInfo>
            </saml:SubjectConfirmation>
        </Subject>
        <AttributeDesignator
            xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
            AttributeName="用户属性名称"
            AttributeNamespace="源域（用户域）"
            mappingType="映射类型">
        </AttributeDesignator>……
    </AttributeQuery>
</Request>
```

1）Request/ @RequestID

该属性表示SAML请求ID号，是一个具有xsd:ID类型的标识，具体用法见SAML1.1规范[19]。

2）Request/@IssueInstant

该属性表示SAML请求发送的时间，是一个具有xsd:dateTime类型的UTC时间，具体用法见SAML1.1规范。

3）Request/ Signature

SAML信息的签名信息，由SAML请求发出者签名，参照XML Signature[46]。

4）Request/AttributeQuery/@Resource

在本发明中，其值为SAML请求发送方所在授权域的域名，如whut.edu.cn。

5）Request/AttributeQuery/Subject

通过NameIdentifier或SubjectConfirmation标识被查询用户。

6）Request/AttributeQuery/Subject/NameIdentifier

NameIdentifier元素表示被查询用户，当用户采用用户ID/口令进行身份标识时，该元素的值采用Domain\UserID的形式，其中Domain是用户所在授权域的域名，UserID是用户标识（用户名），即Format是Windows Domain Qualified Name; 当用户采用数字证书作身份标识时，其值是用户数字证书的主体甄别名

（Distinguished Name），即Format是X509 Subject Name。

7）Request/AttributeQuery/Subject/NameIdentifier/@NameQualifier

当用户采用用户ID/口令进行身份标识时，该属性可以不用，若采用的话，可用来表示被查询用户所在的域，如itrus.com.cn，这样若NameIdentifier元素的值（内容）不包含Domain（域）信息，则可以用该属性设置用户所在授权域的域名；若二者皆指定域信息，则必须一致。

当用户采用数字证书作身份标识时，该属性必须使用，表示被查询用户所在的域。

8）Request/AttributeQuery/Subject/NameIdentifier/@Format

该属性用来指定NameIdentifier信息的格式，当用户采用用户ID/口令进行身份标识时，其值为

urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName

当用户采用数字证书作身份标识时，其值为

urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName。

9）Request/AttributeQuery/Subject/SubjectConfirmation

当基于数字证书标识用户时采用，该元素包含用户的数字证书。

10）Request/AttributeQuery/Subject/SubjectConfirmation/ds:KeyInfo

KeyInfo元素包含用户的数字证书（参见XML Signature）。

11）Request/AttributeQuery/AttributeDesignator

该元素代表请求方（目标域）所需的用户属性。请求中至少包含一个AttributeDesignator元素。

12）Request/AttributeQuery/AttributeDesignator/@AttributeName

该属性值表示请求方（目标域）所需的属性名称。如：“role”，“group”等。有多少个值在跨域授权API的配置信息进行定义。

13）Request/AttributeQuery/AttributeDesignator/@AttributeNamespace

该属性表示 4）中所需要获得的属性来自哪个域，一般代表源域。

14）Request/AttributeQuery/AttributeDesignator/@mappingType

该属性定义了目标域所需要的映射属性采用何种映射方式（Group_Role 或 Attribute）。该属性为自定义属性，在跨域中介系统中我们扩展了 SAML 协议的请求 schema，跨域中介系统读取该属性后，将采用相应的映射方式，对用户属性进行映射。

跨域中介系统返回的 SAML 响应信息格式如下：

```
<Response
    ResponseID="..."
    InResponseTo="..."
    MajorVersion="1"
    MinorVersion="1"
    IssueInstant="..."
    Recipient="..."
    xmlns="urn:oasis:names:tc:SAML:1.0:protocol"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        ……//签名信息
    </ds:Signature>
    <Status>
        <StatusCode Value="samlp:Success"></StatusCode>
    </Status>
    <Assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
        AssertionID="……">
        <Conditions NotOnOrAfter="……">
            <AudienceRestrictionCondition>
                <Audience>……</Audience>
            </AudienceRestrictionCondition>
        </Conditions>
        <AttributeStatement>
            <Subject>
                <NameIdentifier Format="...">
                    用户名称
                </NameIdentifier>
                <SubjectConfirmation>
                    ……
```

```
            </SubjectConfirmation>
        </Subject>
        <Attribute AttributeName="属性名称" AttributeNamespace="…">
            <AttributeValue>
            用户属性值
            </AttributeValue>
        </Attribute>……
    </AttributeStatement>
</Assertion>
</Response>
```

1）Response/ @ResponseID

该属性表示SAML响应ID号，是一个具有xsd:ID类型的标识，具体用法见SAML1.1规范。

2）Response/@InResponseTo

该属性的值为对应SAML请求中的RequestID。

3）Response/@IssueInstant

该属性表示SAML响应的时间，是一个具有xsd:dateTime类型的UTC时间，具体用法见SAML1.1规范。

4）Response/Recipient="..."

表示响应信息的接收者，可以不用，或者设置为SAML请求来自的域。

5）Response/ds:Signature

SAML信息的签名信息，SAML响应发出者签名，参见XML Signature规范。

6）Response/Status

该元素表示SAML响应的状态，成功或出错，若发生错误则该元素中包含了相关的SAML异常信息，具体用法参见SAML1.1规范。

7）Response/Assertion

该元素存放返回的断言（属性断言）。

9）Response/Assertion/IssueInstant="..."

该属性表示断言的签发时间，是一个具有xsd:dateTime类型的UTC时间，具体用法见SAML1.1规范。

10）Response/Assertion/Issuer="...."

该属性表示断言的签发者，即跨域授权中介系统，它的值可以是xsd:anyURI类型的一个URI（如跨域授权中介系统的主机名），或其他标识。

11）Response/Assertion/Conditions

该元素限定返回的断言的应用条件。

12）Response/Assertion/Conditions/@NotOnOrAfter

该属性限定断言的应用时间范围，具体用法参见SAML1.1规范。

13）Response/Assertion/Conditions/Audience

该元素设置为断言应用的授权域，即SAML请求发送者所在的域。

14）Response/Assertion/AttributeStatement

该元素存放被查询用户的属性声明。

15）Response/Assertion/AttributeStatement/Subject

该元素表示被查询的用户，其用法与SAML请求中Subject用法相同。

16）Response/Assertion/AttributeStatement/Attribute

返回与请求中AttributeDesignator元素指定的属性对应的用户属性信息。每个Attribute元素与请求中的一个AttributeDesignator元素指定的用户属性相对应。

17）Response/Assertion/AttributeStatement/Attribute/@AttributeName

该属性表示对应的用户属性名称，它与请求中对应的AttributeDesignator元素的AttributeName属性值相同。

18）.../Assertion/AttributeStatement/Attribute/@AttributeNameSpace

该属性可以不用，如果使用的话，其值与请求中对应的AttributeDesignator元素的AttributeNameSpace属性值相同。

19）.../Assertion/AttributeStatement/Attribute/AttributeValue

该元素存放用户对应属性类别的属性值，当一个用户对一个属性类别有多个值时，将有多个该元素，如当属性类别是角色而用户有多个角色时，每个AttributeValue元素对应一个角色名。