基于角色的用户权限系统设计*

傅元宏 牛德雄 李 燕
(南昌大学信息工程学院计算机系 南昌 330031)

摘 要 权限控制设计是系统设计中很重要的一部分,介绍基于角色控制的原理。对于权限系统的关键部分进行分析,给出关键部分的设计,对权限系统其它要考虑的因素进行说明。

关键词 权限 用户权限系统 角色 用户
中图分类号 TP317.4

1 引言

信息技术的应用无所不在,随之而来的问题是"信息安全"问题。针对OSI(Open System Interconnection)模型人们提出实现系统安全应该提供的5种服务:身份鉴别、访问控制、数据完整性、数据机密性、抗抵赖。其中身份鉴别是信息安全中的第一道防线,对信息系统的安全有着重要的意义。每个系统都要求用户遵循一定的安全策略,比如要求输入用户ID和口令。用户为了避免忘记口令而带来麻烦,用户一般会简化密码,或者在多个系统中使用相同的口令,或者创建一个口令"列表"——这些都是会危及信息保密性的几种做法。不过使用用户ID和口令这方法也有它的优点:如简单,容易实现等。因此现在的大多数用户权限系统是使用了用户ID和口令来识别合法用户。[7]

本设计的系统是在基于角色控制基础上提取改进而来的,对使用用户ID和口令来识别合法用户进行了改进。

2 角色控制基本说明

用户:是一个可以独立访问计算机系统中的数据或者用数据表示的其它资源的主体,我们用user表示一个用户。

角色:是指一个组织或任务中的工作或位置,它代表一种资格、权利和责任。用ROLE表示角色。

资源:指的是受保护的信息资源。

权利:是指对资源的各种操作,如读,写等[1]。

使用角色的概念,将权限三元组分成role - user和role = resource + privilege)的关系。权限操作即判断"who对what(which)进行how的操作"的逻辑表达式是否为真。是一个三元组:user + resource + privilege[5]。user包括group和user。group是user的一个集合,group与user是多对多的关系。group与group当然之间也是多对多的关系。实现时可考虑将group与user等同,没有子的group即为user。不过这样较为抽象,不为一般用户所接受,所以一般分开实现。此处不应将组织关系与group混为一谈。

用户与用户组组成的应该是一个图,而不应是一棵树。在具体应用中可考虑将其关系变成一个树的关系。resource包括resource和package。package是resource的集合,关系同user和group之间的关系。在一般系统中可不实现package。在具体实现中可将resource和package等同实现,没有子package即为resource。privilege包括privilege和privileges.两者之间的关系如同user和group之间的关系。在此不作太多讨论。role = (resource + privilege)而在具体实现上为了降低系统的复杂性,可以将group与role合并实现。上述为权限系统的核心部分。下面是处理多种类型资源的权限系统扩展。它只是为应用系统提供服务的,是一种约束条件,不出现在权限三元组之中。

为了考虑到不同资源类型的resource拥有不同的Privilege,应引入resource type和Operator. resource type是resource的一个type信息。当然resource type也属于resource。每个resource有一个resource type属性。resource type与resource是多

* 收到本文时间:2006年11月14日
作者简介:傅元宏,男,硕士研究生,研究方向:软件工程。牛德雄,男,硕士,副教授,研究方向:信息交换、物流。李燕,女,研究方向:信息交换。

对多的关系。resource type 与 resource type 之间是多对多关系。当然为了实现简便可以实现为一对多的关系。Operator 表示 resource type 与 privilege 的关系,只是一种限定条件,即表示在某类资源上有某种权限或权限集合。是为应用系统服务的,并不是权限系统的核心。再往外层走可能还包括某类资源(resource type)限定在某些 user(用户或用户组)内。但它也只是权限系统的一些附加服务而已。

3 系统分析

3.1 用户与角色

角色的定义来源于用户,也就是可以依据现实中的不同岗位定义出不同的角色[3]。具体应用中的角色由用户根据自己设想的组织机构进行添加设置,提供一个专门模块用来设置组织机构,用户通过组织机构进行角色管理。例如:用户可以通过部门机构来进行角色的管理,部门采用编号分层的方式。这类数据仅为方便用户管理角色而存在,在系统的其他方面不存在任何意义。在系统中每个角色有个唯一的角色 ID 来标识。同样在系统中每个用户有个唯一的用户 ID 来标识。用户的授权基础是角色。当一个角色授予一个用户时,此用户就拥有该角色的权限。用户可以扮演多个角色。

3.2 对象和资源控制

这里的对象指的是应用系统中可视的对象,如:按钮,菜单选项等。对象控制是对对象的属性进行控制,其是通过角色与用户授权来实现。程序员可以事先设定也可以在事后加入。

资源粒度的大小确定是一定很重要也是很麻烦的事。粒度太大,那么不利于系统的精确用户权限控制。如果粒度太小就会使的用户权限控制很复杂,不利于设计和维护。在以往系统控制中只使用可视的对象的控制,增加了 web 页面保护并且粒度大小设定为了类中的方法。至于为什么要设定到方法是因为方法是基本的操作元素。对要求很高的资源保护甚至可以在代码中来实现。

3.3 权限的非集中管理

在以往系统中权限是集中管理的,由系统管理员统一的处理,这样做有其好处,比如简单容易实现等。不过也有其不足的地方:系统管理员的负担过大,并且难对所有的岗位有全面的准确了解。对于大型的管理系统,在对系统的权限进行标准细致的划分,把一部分权限设置交给高级用户来进行,这样有利于解决集中控制所带来的问题。这里要控制好高级用户的权限平衡问题,不要让某一高级用户拥有过大的权限。当然在实际应用中可以综合以上两种管理方式。

4 系统设计

4.1 角色定义

角色是用户和权限的中介。这样用户与色之间、角色与权限之间形成的是多对多的关系。角色定义的依据与实际的岗位是有关系的。角色之间是有继承关系的,这继承是多继承关系。当一个角色继承了另外一个角色那么该角色就拥有了被继承的角色权限。角色的继承关系反映了实际的岗位间关系。角色定义示意图如图 1 所示。[1]

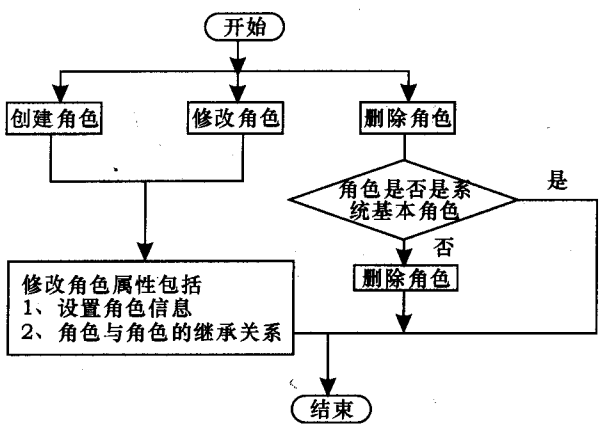

图 1 角色定义图

在具体的代码实现时,可以考虑使用 XML 配置文件来实现。详细的配置与具体的环境有关,例如 J2EE 技术中 EJB 的配置文件 ejb - jar. xml 就可以实现类中的方法与角色进行关联,此外可以在 Web 的配置文件 web. xml 中设定特定的页面只能由哪些角色有权限访问,后在 J2EE 服务器的配置文件(一般也是 XML 文件)实现角色与用户或者用户组的关联这样就可以实现表现方式和业务逻辑的保护[3]。对角色权限创建、修改和删除只要修改相应的配置文件就可以了,对用户授予角色也是修改相应的配置文件。这样做是比较的方便,也易与维护,性能有保障,因为优秀的 J2EE 服务器对此进行了优化。

4.2 用户和用户组的定义

用户是系统的最终用户。用户定义指的要建立用户的信息,最起码要有用户名和密码。用户信息也是验证用户合法性的关键。在一般系统中,本文还是提倡使用用户名和密码作为验证合法用户的依据。不过对这要加点限制,比如规定密码必须是 6 位以上还要求包含数字。此外用户名和密码

都有一定的使用期限。用户的信息存储与数据库中,其表字段含有用户名、用户 ID 和密码等。如:

userID	username	password
000001	java	javajava

用户组可以方便管理同一类用户,用户组中的用户拥有组所拥有权限。当然用户组中的用户成员可以单独拥有自己特有的权限[7]。

4.3 访问控制机制

合法用户只能访问自己所拥有权限的资源,比如普通员工只能看到自己的工资详细表单。如何来实现就是通过访问控制机制来实现的。系统可以根据授予该用户的角色来生成该用户的权限表,该表可以是 XML 文件,因为 XML 表示数据能力很强且容易被服务器所解析。当用户要请求执行某个操作的时候,系统就检查这表来确定该用户是否有权执行该操作。

5 系统的其他问题和技术说明

当前计算机环境的主要问题是,计算机系统是可以信任的,但事实却是当前的计算机系统不能被信任,有严重的安全漏洞和错误。用这些不可靠的部件构筑安全平台,风险很大。当新的系统实现了认证和访问控制后,与旧有的认证和访问控制机制无互操作性可言。另外,系统管理员的水平和素质也影响系统安全。

与组织结构相关的问题也是要考虑的。访问授权的规则需要规定哪些资源是哪些用户拥有什么权限(读,写等)。为简化,用户中类似的需要和权利被划分成组,而组的划分往往是基于现实的组织结构。这样管理不同组的用户是件费力的事情,比如用户转移到别的部门,那么他的访问控制的权限也应得到及时更动。

对于新老系统的互操作性问题的解决方法,一种对应用系统本身做出改动,另外一个选择的方法是提供一个标准的安全 API。这二种方法各有特点,具体问题具体的分析。本文建议尽量采用了要求对应用系统本身做出改动,对其进行标准化,为之后的扩展提供方便。

因为传输要使用了加密技术,所以需要一个消息构件。消息构件也是模型中的关键构件之一,主要实现消息服务需要提供的功能。负责在整个平台中接收和发送信息、请求操作或操作的返回结果。由于整个系统有部分是在外网上运行的,传递消息的速度、消息的合法性、消息的完整性等都是消息服务必须考虑的,所以消息服务包括解/压缩消息、签署与加密/解密消息以及验证消息的有效性等功能,如图 2 所示。

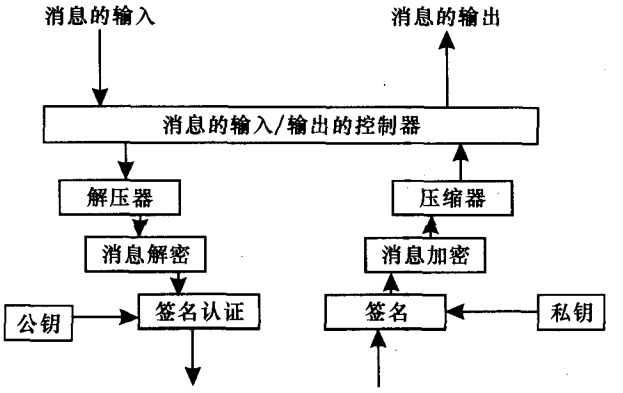

图2 消息服务

6 结束语

用户权限控制是系统中很重要的一部分,要考虑的方面很多,本文仅谈了一部分。例如由于现实世界的复杂性,对权限系统而言,最主要的问题是无法找到一个通用的识别资源的方法。本文是假定所有资源都以一个统一的方式被系统识别的前期下做了以上设计。用户权限控制子系统设计好后要不断的改进,因为安全总有新的挑战。

参考文献

[1]唐成华,陈新度. 管理信息系统中多用户权限管理的研究及实现[J]. 计算机应用研究,2004,3

[2]陈传波,夏义兵. 基于 ASP. NET 技术及三层网络架构的权限管理系统模型[J]. 计算机工程,2003,29(12)

[3]万海东,王红卫. 基于 Web 的信息系统的权限设置方式[J],计算机系统应用,2002,7

[4]任善全,吕强等. 基于角色的权限分配和管理中的方法[J]. 微机发展,2004,14(12)

[5]陈金玉,刘东荣,李卓伟,吴德垠. 基于角色控制的教学权限访问系统的设计与实现[J]. 重庆大学学报(自然科学版),2005. 12

[6] Tim Bray, Jean Paoli. Extensible Markup Language (XML) 1. 0 (Third Edition), W3C Recommendation. Feb 2004. http://www. w3. org/TR/REC - xml/.

[7]Mark Artiges. BEA Weblogic Server 8. 1 大全[M]. 机械工业出版社,2005