# 基于数字证书的通用权限管理的设计与实现①

曹　望　尤志强　(湖南大学 软件学院 湖南 长沙 410082)

**摘　要：** 在基于数字证书的应用系统开发中，针对不同的业务系统的权限管理，都要重新编写权限控制代码，造成人力资源浪费的问题。分析并比较了基于数字证书的权限管理与传统用户名/密码方式认证的权限管理区别，提出利用功能向量码的方法实现一种通用的基于数字证书的权限管理模块。利用该模块，只需替换验证签名模块，无需再编写代码，就可以应用到不同的基于数字证书认证的系统中，从而实现权限控制和业务的分离。本系统已经开发并在多个系统中使用，应用效果良好。

**关键词：** 数字证书;权限控制;角色;身份认证;通用性

## Design and Implementation of Universal Permission Management Based on Digital Certificate

CAO Wang, YOU Zhi-Qiang (Software School, Hunan University, Changsha 410082, China)

**Abstract:** For the trivial design and the complexity of the permission management in digital certificate-based system development, human resources are being used unwisely in writing codes of various business functions. To solve this problem, a kind of general and digital certificate-based permission management module, using function vector code, is successfully designed through analyzing and comparing the differences between digital certificate-based systems and the traditional username/password system. This kind of module can be embedded in various digital certificate-based systems by replacing the verifying module without rewriting the code, which makes the separation of permission control and transaction possible. This strategy has been tested and verified through different systems and has achieved good performance.

**Keywords:** digital certificate; access control; role; identity authentication; universal

随着信息化的发展，数字证书的应用越来越普遍，基于数字证书认证的应用系统也越来越多。在应用系统开发中，权限管理模块是系统不可缺少的一个模块。目前的通用权限管理模块都是针对用户名/密码认证方式设计，由于基于数字证书的认证方式与传统的用户名/密码认证方式不同，因此那些传统的通用权限管理模块不能直接应用到基于数字证书的应用系统中。而在基于数字证书的应用系统开发中，针对不同业务系统通常都要重新编写权限控制代码，这样容易造成人力资源浪费。因此，本文分析了基于数字证书的权限管理与传统用户名/密码方式认证的权限管理的区别，并利用用户、角色、资源、功能属性四者之间的关系和数据库的设置，设计了一个通用的基于数字证书的权限管理模型。利用此模型，可以方便地分配角色和设置用户权限，实现对不同业务系统的权限管理。

## 1　基于数字证书的权限控制系统思想

### 1.1 术语解释

(1)用户：可以独立访问系统中的数据的主体。

(2)角色：指一个组织或任务中的工作或岗位。用户拥有自己所属角色权限的并集[1]。

(3)资源：用户所能访问的窗口和数据的通称，例如一个窗口、某个页面，都是一种资源。

(4)功能属性：对资源可进行的操作，比如新增、

① 收稿时间:2009-12-17;收到修改稿时间:2010-01-29

删除、修改、打印、查询等。

(5)功能掩码[2]：一个数字，该数字是由一个或多个特定规律的数字经过特定运算后得到的结果。其中每个数字都代表特定的功能。

(6)功能向量码：一个二进制格式的字符串，每一个二进制位代表特定的功能，其对应的功能项与功能属性集合中功能属性的顺序有关。

(7)数字证书：由权威机构 —— CA 证书授权中心发行的，能提供在 Internet 上进行身份验证的一种权威性电子文档，人们可以在互联网交往中用它来证明自己的身份和识别对方的身份。

(8)USBKEY：数字证书的载体，外形跟普通 U 盘一样，但它的内部结构比 U 盘复杂，它内置了 CPU、存储器、芯片操作系统(COS)。其内置的 CPU 可以实现加解密和签名的各种算法，并且加解密运算是在 USBKEY 内进行的，保证了密钥不会出现在计算机内存中，从而杜绝了用户密钥被黑客截取的可能性[3]。

### 1.2 RBAC 权限控制模型的基本原理

2003 年 4 月，美国国家标准与技术研究所(NIST)在总结前人成果的基础上制定出了 RBAC[4]的最新草案，给出了 RBAC 参考模型的详细描述和功能规范，其管理模型如图 1 所示。

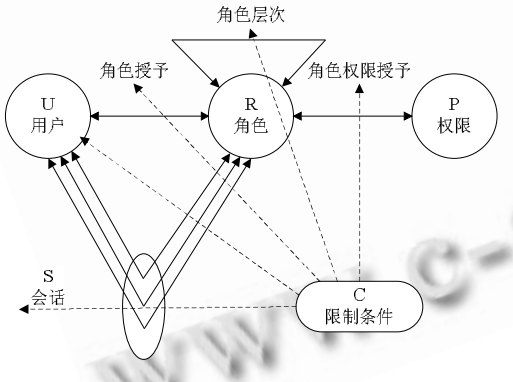


图 1 RBAC 模型[5]

该 RBAC 模型的核心思想是将访问权限同角色关联起来，通过给用户分配合适的角色，每个角色再分配相应的权限，这样角色就把用户与权限联系起来了，并且大大减少了权限分配时的工作量，增加了授权的灵活性。在 RBAC 模型中，角色是其核心，系统根据管理中相对稳定的职权和责任来划分角色，每种角色可以完成一定的职能。用户通过饰演不同的角色获得角色所拥有的权限，一旦某个用户成为某角色的成员，则此用户可以完成该角色所具有的职能。采用 RBAC 模型来构建信息管理系统的用户权限管理模块，通过对用户表、角色表、角色权限分配表、用户角色关联表进行合理的设置，再加上应用程序的控制，可以完成多用户、多级别的权限管理[6]。

### 1.3 基于 PKI 的数字证书认证模式

PKI(public key infrastructure)，即"公钥基础设施"，是一种遵循既定标准的密钥管理平台，它能够为所有网络应用提供加密和数字签名等密码服务及所必需的密钥和证书管理体系[7]。基于 PKI 的“数字证书认证模式”可以有效保证用户的身份安全和数据传输安全。数字证书是由可信任的第三方认证机构 —— 数字证书认证中心(CA)颁发的一组包含用户身份信息的数据结构，PKI 体系通过采用加密算法构建了一套完善的流程，保证数字证书持有人的身份安全。与传统的用户名/密码认证模式相比，基于 PKI 数字证书的认证模式主要有如下优点：(1)基于 PKI 数字证书认证模式不必在网上传递用户口令，因此安全性比用户名/密码认证模式要高。(2) 用户可用自己的签名私钥向本地或远程环境的实体证明自己的身份，实现了网络环境身份鉴别。(3)签名和私钥可用于数据来源认证。即用户用自己的私钥对数据进行签名，并且该签名数据可以作为不可否认的证据。

### 1.4 与传统用户名/密码认证权限系统的区别

根据 RBAC 控制模型原理，传统的基于用户名/密码认证模式的权限管理系统一般包括用户身份认证，用户管理，角色管理，资源管理，角色权限管理，用户授权等模块。基于数字证书的权限管理也同样包含上述模块，但与传统权限管理系统相比还是有比较大的区别，特别是在用户身份认证、用户管理等模块，具体如下：

(1)用户身份认证，基于用户名/密码认证模式，验证身份只是简单的比对用户名和密码。而采用基于 PKI 的数字证书认证模式，是由系统产生随机码发送给用户，用户使用 USBKEY 对该随机码签名，并将用户名、随机码、签名值提交到服务器，服务器验证用户签名来判断用户是否合法。

(2)对于用户的管理，传统认证模式的权限管理系统，用户注册时只需提交基本信息，系统验证用户信息格式即可。而在基于数字证书的权限管理系统中，用户注册时，除了输入用户信息外，还要录入用户的

数字证书，并且数字证书需要经过证书策略模块的验证，如验证用户证书是否存在，证书是否过期，以及该证书是否由系统根证书所颁发等。

(3)在数字证书的应用中，管理员或用户在进行一些重要操作时(如关键信息的提交)，系统会用用户的签名私钥对所提交的信息进行数字签名，并将签名值存入数据库中，以保证用户此次操作的不可否认性。而在传统的权限管理模块中不会有此功能。

### 1.5 设计思路

文献[2]提出用 2 的 N 次幂来标识每个功能属性，再将所有的功能属性建立一个功能集合列表。假若有新增、删除、修改、打印、浏览等功能，那么对每一个功能都给予一个 2 的 N 次幂的值，如新增=1，删除=2，修改=4，打印=8，浏览=16，当给角色赋权限的时候，按照这些值的和来赋予操作权限(文献[2]中将这些值的和，称为功能掩码)。若某个角色只有新增和删除用户的权限，那么将该角色对资源的功能掩码设成新增和删除的和，即为 3，系统在判断功能掩码时，就知道这个角色拥有新增和删除的权限。如果功能掩码是 4，那么只有修改的权限。如果要知道某角色对某个资源有哪些操作权限，则取出该资源的功能掩码值 K，分别用 K&1，K&2，K&4，K&16… ，如果为真，则表示有值等于“&”右边整数的权限。

但是这样做有一个问题：当功能个数比较多，如 N=64 的时候或更大的时候，那么 2 的 N 次幂值将是个巨大的数字。

如果将功能掩码不用数字来表示，而是用二进制字符串来表示，假设功能属性集合中的操作项中：新增=1，删除=2，修改=4，打印=8，浏览=16，则功能掩码 29(1+4+8+16)用二进制可表示为“11101”，表示有新增、修改、打印、浏览的功能。根据功能集合中功能项编号的规律，可以发现二进制字符串中的每一位表示一个功能属性，从右往左依次表示功能集合列表中的功能属性项的顺序，如图所示。则二进制字符最右边的那一位表示第一个功能新增，右边第二位表示第二个功能删除，右边第三位表示修改……依次类推。若要判断该功能掩码是否具有某项功能操作，则只要看对应的功能序列位是 1 还是 0，若是 1 表示具有该操作，否则表示不具有该项操作。如图 2 所示，表示功能掩码的二进制字符，删除线表示不具有此项功能。

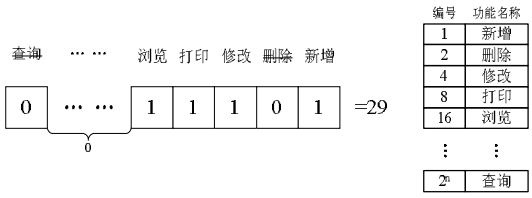


图 2　二进制字符表示的功能掩码

如果把二进制字符串的功能掩码字符的顺序反转过来，形成新的字符串，这样只要将操作项依次排列，且分配序号，如新增=1，删除=2，修改=3，打印=4，浏览=5。判断一个功能向量码是否具有某一功能，只需判断该功能向量码中对应的功能序号位上的字符是 1，还是 0。由于该字符串表示功能属性的集合，且要求功能属性项顺序排列，因此把这个反转后的字符串叫做功能向量码。如图 3 所示，要判断该功能向量码是否有修改(序号为 3)的功能，只要判断该序号位(第 3 位)的字符是 1，还是 0，显然图中的功能向量码是有修改功能的。这样以字符串的形式来存储功能掩码，功能项的个数将不受限制，而且检查每个操作也十分方便。因此用该功能向量码来表示每条权限的权限值极为方便，并且不受功能属性个数的限制。

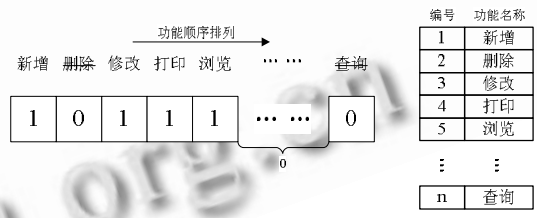


图 3　功能向量码结构

## 2　系统实现

### 2.1 数据库设计

将角色、角色和用户的关系、角色的权限、资源，用户证书等信息保存到数据库中，主要的数据表如下(数据表中仅包含主要字段)。

(1)资源表 TB_Resource，用来保存系统所有的资源，后面附加该资源的功能向量码，如表 1 所示。

表 1　资源表(TB_Resource)

| 字段名 | 字段描述 | 数据类型 | 说明 |
|---|---|---|---|
| ResourceCode | 资源编码 | VARCHAR2(20) | 资源编号 |
| ResourceName | 资源名称 | VARCHAR2(50) | |
| FunctionCode | 功能向量码 | VARCHAR2(255) | “1101”格式 |

(2)角色对应的权限表 TB_Privilege，用来保存角色和资源的对应关系,如表 2 所示。

表 2　角色对应的权限表(TB_Privilege)

| 字段名 | 字段描述 | 数据类型 | 说明 |
|---|---|---|---|
| RoleID | 角色编号 | NUMBER(6) | 自动编号 |
| ResourceCode | 资源编码 | VARCHAR2(20) | |
| FunctionCode | 功能向量码 | VARCHAR2(255) | “1101”格式 |

(3)资源的操作属性表 TB_Operation，用来描述资源的操作如新增，修改，删除，打印等功能，如表 3 所示。

表 3　资源的操作属性表(TB_Operation)

| 字段名 | 字段描述 | 数据类型 | 说明 |
|---|---|---|---|
| OperationId | 操作项编号 | NUMBER(6) | 顺序编号 |
| OperationName | 操作名称 | VARCHAR2(20) | |

(4)用户证书表 TB_UserCert，存储用户的签名证书和加密证书，如表 4 所示。

表 4　用户证书表(TB_UserCert)

| 字段名 | 字段描述 | 数据类型 | 说明 |
|---|---|---|---|
| UserId | 用户编号 | VARCHAR2(20) | |
| SignCertSeq | 签名证书序列 | VARCHAR2(50) | |
| SignCertData | 签名证书数据 | BLOB | 二进制数据 |
| EncryptCertSeq | 加密证书序列 | VARCHAR2(50) | |

## 2.2 管理员添加新用户

与传统的用户名/密码模式相比，该系统添加新用户时主要验证新用户数字证书的有效性。管理员添加新用户时，先插入需要注册的 USBKEY，填写用户信息，提交到服务器，服务器验证用户信息，并检查证书的有效性，具体流程如图 4 所示。

## 2.3 资源添加

添加新的资源时，系统从数据库或缓存中读取顺序功能属性列表，根据功能属性依次动态构建操作复选框，将复选框 checkbox 的 value 属性设置为功能属性项的编号。如图 5 所示，系统的功能属性集合共有 5 项，根据上文中功能向量码的定义，将新增、删除、修改、打印、查询等功能属性顺序排列。那么可以设置第一个(新增)checkbox 的 value=1，第二个(删除)checkbox 的 value=2……。“用户管理”为即将添加的一个资源项，该资源有进行 3 个功能操作，分别为删除，修改，查询，保存后，该资源的功能向量码为 “01101”。

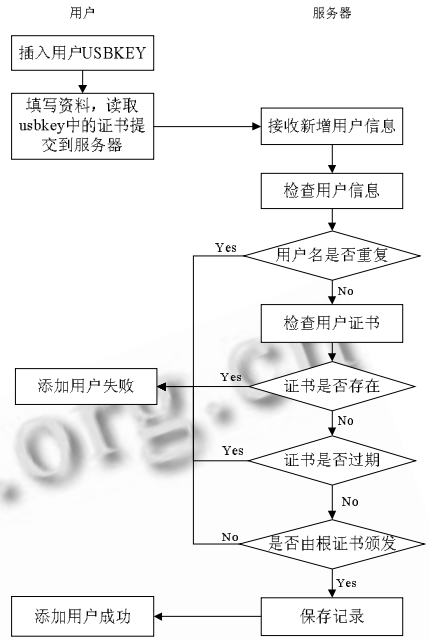


图 4　添加用户流程

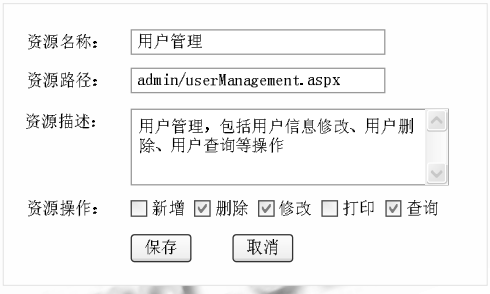


图 5　资源添加界面

## 2.4 角色权限分配

系统根据管理员选择的资源和资源的功能属性，生成权限数据，每条权限数据包含资源和功能向量码，操作界面如图 6 所示。具体操如下。


图 6　角色权限分配界面

(1)选择要分配权限的角色，系统加载资源列表；

(2)在加载资源的同时，根据每个资源的功能向量码，分别显示相应的操作复选框，显示复选框流程如图 7 所示；

(3)管理员选择该角色所能访问的资源以及每个资源所能进行的功能操作；

(4)系统再根据所选的操作计算每条权限的功能向量码，并将其存入数据库中(计算每单资源的功能向量码算法流程如图 8 所示)。

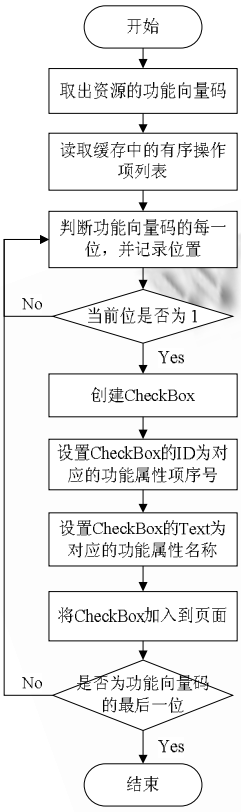


图 7 显示功能属性复选框

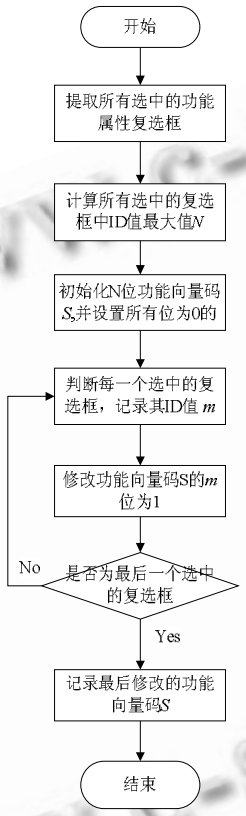


图 8 计算单个资源的功能向量码

## 2.5 用户身份验证

基于数字证书的用户验证跟传统的用户名/密码方式的用户验证不同，系统不是通过比对用户名和密码来判断用户是否合法；而是发送一个随机码给用户，用户输入其 ID 号,并用 USBKEY 中的签名私钥对随机码进行签名，然后将用户名、随机码、签名值提交给服务器，服务器将随机码、签名值、用户签名证书组成 XML 串发给认证中心(CA)，进行验证，最后根据验证结果来判断用户是否合法。具体流程如图 9 所示。

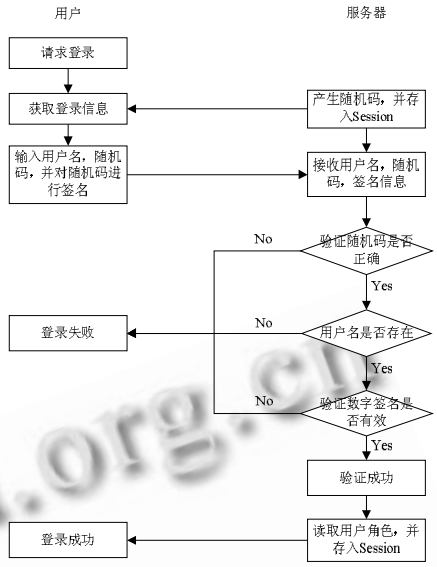


图 9 用户登录流程

一般来说，不同认证服务器会有各自的验证方式，因此系统中需预留接口，以满足不同认证设备的需要，从而实现程序的通用性。对于不同的加密设备，应用本权限系统时只需实现下面的接口函数即可。

bool VerifySignature(string signXmlString);

该函数返回验证结果，验证通过返回 true,否则返回 false。参数 signXmlString 格式为 XML 串，具体结构如下：

<MESSAGE>

<paradata>原始数据</paradata>

<certdata>签名证书数据(base64 编码)</certdata>

<signdata> 签 名 后 的 数 据 (base64 编码)</signdata>

</MESSAGE >

## 2.6 权限检查

权限检查包括资源权限检查和操作权限检查两个部分。对于资源的检查，系统在用户登录成功后，将用户的角色信息存入 Session 中。用户在每访问一个资源时，系统都会调用 CheckResourceAccess 函数，检查该用户是否具有访问该资源的权限。CheckResourceAccess 函数形式及参数如下。

public bool CheckResourceAccess(string resource,int roleId)

操作权限检查部分，用户对资源进行某项操作时，

将用户此权限的功能向量码以及此操作的功能属性编号传给 CheckOperateAccess 函数进行判断。如果返回真，表示用户有权限进行此限操作。函数的形式及参数表示如下。

public bool CheckOperateAccess(string operation Code, int operationId)

### 2.7 系统性能优化

在每次页面加载时，都要用到资源功能属性集，并且系统对访问的每个页面窗口都要进行权限检查，而一个用户又可能隶属于多个角色，因此每次都要从数据库中来检索权限会严重影响系统的性能。为解决这个问题，在系统运行时，可将这些数据缓存到服务器中，常驻内存，如果在运行中更改了这些数据，则同时更新缓存中的数据。如将资源操作属性集存入 HashTable[操作编号，操作名称]，将资源列表的数据集，权限数据集缓存到内存中，这样要用这些数据时直接从缓存中读取，节省了系统频繁访问数据库的开销，也提高了权限计算的效率。

## 3 结束语

本文分析并比较了基于数字证书的权限管理与传统用户名/密码方式认证的权限管理区别，并针对不同之处给出相应的实现方法。本文采用功能向量码的方法了设计一个基于数字证书的通用权限管理模块，使之可以在不同的基于数字证书的应用系统中很方便地设置权限。只需替换系统中验证签名的函数模块(同一加密设备无需替换)，就可以将本系统应用到其他数字证书认证的系统上。这样就大大减轻了权限管理模块的实现难度，实现了权限管理与系统业务的分离。目前该系统模块已经成功应用于国家教育电子身份认证系统中，取得了不错的效果。

### 参考文献


1 Sandhu R. Role-based Access Control Models. Computer, 1996, 29(2):38－47.

2 蔡昭权. 基于业务无关的权限管理的设计与实现. 计算机工程, 2008,34(9):183－185.

3 王权,杨林,刘伟,王桢珍.基于 USBKEY 的访问控制方法研究.计算机工程与设计,2008,29(11):27－29.

4 Ferraiolo D, Kuhn R. An Introduction to Role Based Access Control.[2007-05-10].http://csrc.nist.gov/rbac/.

5 徐斌,袁健.基于Web2.0的用户权限管理研究与实现. 计算机工程, 2008, 34(13):157－159.

6 暴志刚,胡艳军,顾新建.基于 Web 的系统权限管理实现方法.计算机工程, 2006,32(1):169－170.

7 高正宪,涂亚庆,李中学. PKI 和 RBAC 授权数字证书的设计与实现.计算机工程, 2008,34(2):117－119.