基于 Acegi 框架的网络阅卷系统的设计与实现

刘振海,唐拥政
(盐城工学院 现代教育技术中心,江苏 盐城 224051)

摘要:针对现有网络阅卷系统安全性方面存在的不足,提出一种基于 Acegi 框架的网络阅卷系统(简称 BoA - NMS)的认证与授权机制。通过在现有网络阅卷系统中构建安全的 Acegi 双向认证机制、安全拦截器和设计角色访问控制授权机制,从而增强了网络阅卷的安全性和保密性,具有一定的实用推广价值。

关键词:Acegi;网络阅卷;认证;授权
中图分类号:TP393 **文献标识码**:A **文章编号**:1671 - 5322(2012)02 - 0042 - 06

考试是人才选拔的重要方式,阅卷工作是其中的重要环节。本文探讨了一种基于网络的阅卷方式,将 Acegi 安全框架认证与授权机制应用到网络阅卷系统中,并给出了实现过程。基于 Acegi 的网络阅卷系统在本文中用 BoA - NMS 表示。

传统考试阅卷一般都是采用人工手动评阅试卷的方式。这种阅卷方式,教师阅卷强度大,检查机制不完善,易出现各种阅卷差错。同时,大批量重复性的阅卷工作也使得广大教师不堪重负。此外,人工方式处理海量考试试卷的保存、管理与查阅以及分数统计都很不方便。网络阅卷模式是以计算机网络技术和电子扫描技术为依托,以控制主观评分误差、实现考试公平性原则为最终目标,把多年来人工阅卷积累起来的丰富经验和现代高新技术相结合,由阅卷老师在计算机辅助下完成试卷评阅的一种全新工作模式。

在这种网络阅卷模式中,需要将试卷的客观题和主观题分开处理。客观题由考生通过涂写专用答题卡回答问题,采用读卡器读取答题卡,从而获得考生的客观题分数;主观题由阅卷老师通过计算机评阅电子图像化的试卷,而阅卷老师获得的数据是由服务器通过网络分发的。最后,将主观题和客观题的阅卷结果通过网络合成后即为考生的最终考试成绩[1]。

采用 BoA - NMS 的网络阅卷模式,简化了评卷过程,省去了传统的试卷分发、回收、登记分数、加分环节,提高了工作效率,更重要的是避免了人工干预所带来的误差,很大程度上实现了考试的公平、公正,降低了考务管理成本。此外,在之后的考试成绩统计与分数分析方面都有固有的优势,减少了统分差错。

主观试题评阅系统和客观题的机读卡识别判卷一起,组成一个完整的网络阅卷系统。图 1 列出了网络阅卷系统的整体业务流程。

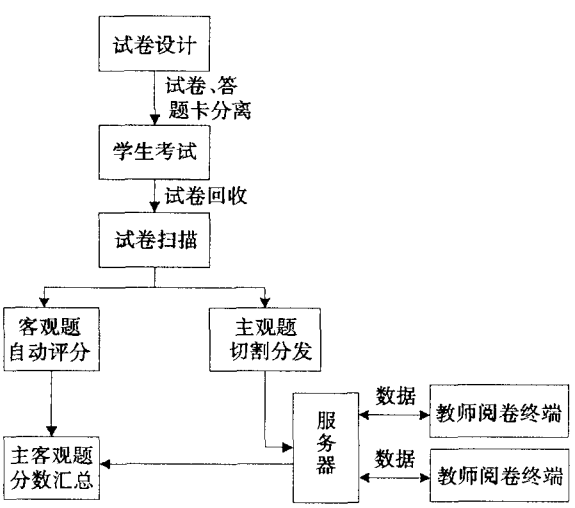

图 1 网络阅卷系统整体业务流程图
Fig. 1 Integral service flowchart of the Network Marking System

收稿日期:2011 - 09 - 08
作者简介:刘振海(1982 -),男,江苏徐州人,讲师,硕士,主要研究方向为智能算法、计算机应用。

1　Acegi 框架分析研究

1.1　认证策略

首先，Acegi 调用 AuthenticationManager（认证管理器）来对当前请求登陆系统用户进行验证，AuthenticationManager 负责委托一个或多个 Provider（认证提供者），并逐一遍历每个 Provider，直到某一个 Provider 能够成功的验证用户的身份。Provider 成功验证用户身份后，会返回一个 Authentication（证明），Authentication 中包括 Principal（用户名），Credentials（通常是密码），Authorities（该用户所拥有的权限）。这样就完成了身份验证的步骤。

认证功能的两个核心接口是：Authentication 和 Authentication Manager。

（1）Authentication 接口

Authentication 包含用户基本信息的类，其中包括：用户名、密码和该用户的具体权限以及下列方法：getPrincipal（ ）、getCredential（ ）和 getAuthorities（ ），如图 2 所示。

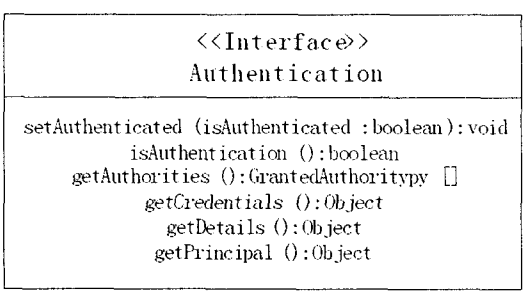

图 2　Authentication 接口

Fig. 2　Authentication interface

（2）AuthenticationManager 接口

AuthenticationManager 接口定义了验证 Authentication 对象合法性的方法，其实现无非是数据库搜索一下是否存在这个用户、密码是否匹配等。真正查询数据库的那个类是 DaoAuthenticationProvider。这个接口执行后返回的结果是一个 Authentication，而不是用一个布尔值来表示验证成功或失败，如图 3 所示。AuthenticationManager 能够设置授权给用户 Principal 的权限列表信息。Authenticat ion Manager 接口仅有 authenticate 方法。

Acegi 为 Authentication Manager 提供了 Provider Manager 实现。其中，Provider Manager 仅是对 Authentication Provider（认证提供者）列表的包装。Authentication Manager 用于管理 Authentication Provider。它的作用是使你能够通过多个不同的认证提供者（JAAS 认证、DAO 以及 LDAP）来对用户进行认证。认证管理器将依次调用认证提供者的认证方法，直到认证通过。

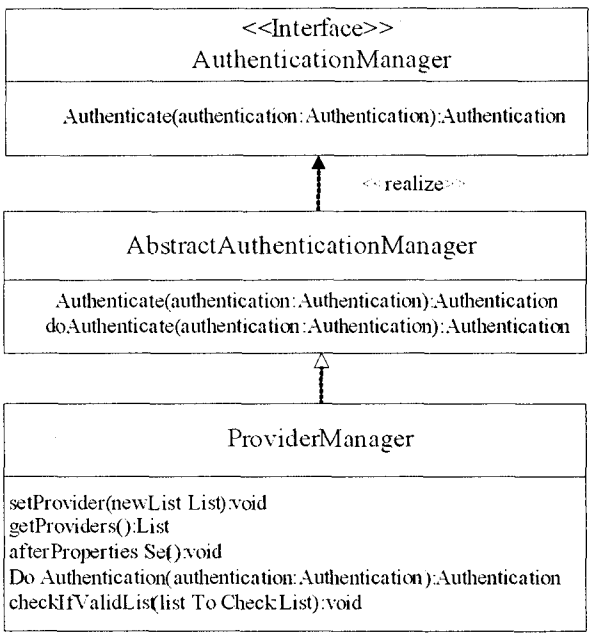

图 3　AuthenticationManager 接口及其子类

Fig. 3　AuthenticationManager interface and its subclass

1.2　授权策略

授权模块是建立在认证模块的基础之上，当 Acegi 获得 Authentication 后，也就确定了用户的身份。每当用户请求访问某受保护资源时，Acegi 会调用 AccessDecisionManager（访问决策管理器）来决定用户的 Authentication 是否有恰当的权限来访问当前的受保护资源，有则授权用户通过，无则抛出错误信息，以达到访问控制的目的。授权功能的核心接口是：AccessDecissionManager。

授权过程中最关键的接口是 AccessDecissionManager，通过它来实现对已认证用户的授权。此外还借助于安全拦截器实现了对受保护资源的受限访问，如图 4 所示。

Acegi 安全系统中，使用投票策略的 AccessDecisionManager 共有 3 个具体实现类：AmrmativeBased、ConsensusBased 和 UnanimousBased。它们的投票策略是，AfirmativeBased 类只需有一个投票赞成即可通过；ConsensusBased 类需要大多数投票赞成即可通过；而 UnanimousBased 类需要所有的投票赞成才能通过。

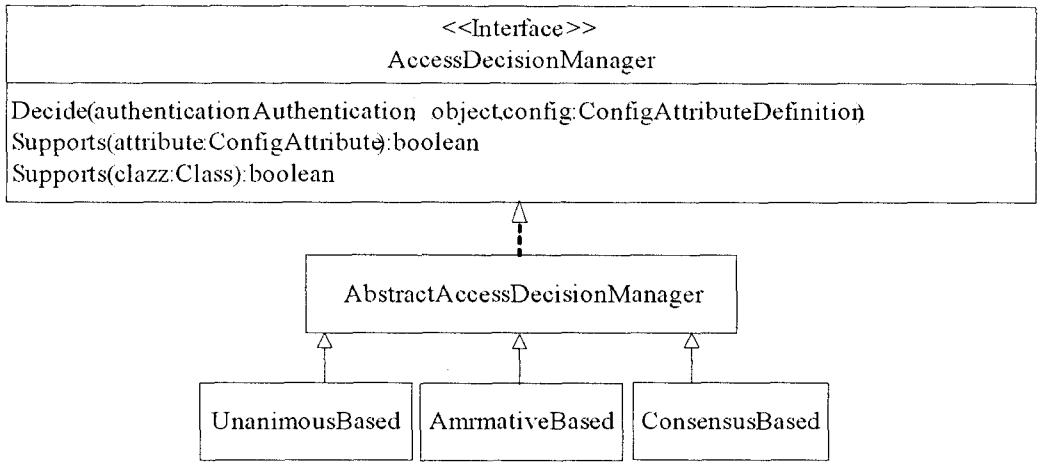

图4 AccessDecissionManager 接口及其子类

Fig. 4 AccessDecissionManager interface and its subclass

1.3 Web 服务中 Acegi 框架的实现

这里以 Web 服务中的 HTTPForm 认证为例，说明 Acegi 认证和授权原理[2]：

(1)首先通过 HttpSessionContextIntegrationFilter 初始化 ContextHolder 中的内容。如果 session 中已经存在 Context，则直接初始化到 ContextHolder 中，否则创建一个新的 Context，此时 url 为/j_secuirty_check。

(2)进入用户登录系统，Acegi 从 AuthenticationProcessingFilter 中得到用户的登录信息(包括 Principal 和 Credential)然后调用 AuthenticationManager 进行认证。若认证通过，返回带有 Principal 授权信息的 Authentication 对象，此时 ContextHolder 中 Authentication 对象已拥有 Principal 的详细信息；若认证失败，则转到“authenticationFailureUrl”。用户从 AuthenticationProcessingFilter 中得到用户的登录信息，其配置如下：

< bean id = " authenticationProcessingFilter" class = " net. sf. acegisecurity. ui. webapp. AuthenticationProcessingFilter" >

< property name = " authenticationManager > < ref local = " authenticationManager"/ > </property >

< propertyname = " authenticationFailureUrl" > < value > /login. jsp </value > </property >

< property name = " defaultTargetUrl" > < value > / </value > </property >

< property name = " filterProcessesUrl" > < value > /j_security_check </value > </property > </bean >

(3)用户登录成功后，继续进行业务操作。

(4)安全拦截器(FilterInvocationInterceptor)收到客户端操作请求后，将操作请求的数据包装成安全管理对象(FilterInvocation 对象)。然后，从配置文件(ObjectDefinitionSource)中读出相关的安全配置参数 ConfigAttributeDefinition：

< bean id = " filterInvocationInterceptor" class = " net. sf. acegisecurity. intercept. web. FilterSecuirtyInterceptor" >

< property name = " authenticationManager" > < ref local = " authenticationManager"/ > </property >

< property name = " accessDecisionManager" > < ref local = " accessDecisionManager"/ > </property >

< propertyname = " objectDefintionSource" > < value > CONVERT_URL_TO_LOW ERCASE _BEFLRE_CONPARISON

/signup. html * ROLE_ANONYM OUS, admin, tomcat

/passwordhint. html * = ROLE _ ANONYM OUS, admin, tomcat/ * * / * . html * = admin. tomcat

/clickstreams. jsp * admin </value > </property > </bean >

(5)接着，安全拦截器取出 ContextHolder 中的 Authentication 对象，把它传递给 AuthenticationManager 进行身份认证，并用返回值更新 Cont-

extHolder 的 Authentication 对象:

(6)将 Authentication 对象、ConfigAttributeDefinition 对象和安全管理对象交给 AccessDecisionManager,检查 Principal 的操作授权;

(7)如果授权检查通过,则执行客户端请求的动作,否则拒绝。

2　BoA – NMS 系统设计思想

在传统网络阅卷过程带来极大便利的同时,也面临了诸多亟待解决的安全问题。例如,在网络上传输的试卷答题和评分结果容易被人窃听或篡改;阅卷教师的身份可能被假冒;阅卷服务器容易受到黑客攻击等等。对于这些网络安全隐患,可以采取多种安全应对措施。其中,身份认证和授权访问是其中非常重要的两种措施。

针对用户认证的基本设计思路为:在服务器和客户端两边同时编写登录模块,通过采用客户端和服务器端同时进行认证,从而加大了系统的安全系数。但这样做,却加重了客户端的负担。这里采用的设计方案是:将服务器端的登录模块注册到 JNDI(JAVA Naming and Directory Interface,JAVA 命名和目录接口),客户端通过 JNDI 获取登录模块的引用,并直接进行服务器端的登录[3]。

如图5所示,首先,应用程序客户端收集用户认证信息,并通过 Spring 组件传递认证信息。客户端通过服务器端安全认证后,会提出资源访问请求,并交给 Acegi 安全拦截器。安全拦截器把请求保存在 SecurityContextHolder 中,认证管理器进行身份认证。若认证通过,则返回一个标识用户认证会话的安全会话令牌,同时把认证后产生的相关的安全会话标识和安全角色标识等用户信息保存在 SecurityContextHolder 中;若认证不通过,则要求用户重新进行身份认证。之后,Acegi 框架使用 SecurityContextHolder 安全会话标识和安全角色标识等作为参数调用访问决策管理器来实现用户的授权。访问决策管理器的投票器根据用户具有的权限来对用户是否有权访问资源进行投票,Acegi 框架根据投票结果进行判断用户是否有权访问资源,如果授权通过,则在 Spring 组件环境下建立安全上下文,从而客户端可以使用服务器端相关的资源。否则,返回无权访问资源的页面。

网络阅卷系统采用基于角色的访问控制,用户角色采用层次结构,从高向低依次为系统管理员、科目主管、题目组长、小组长和普通评卷员。高级角色只对他下属的低级角色有控制权,同级别角色之间没有控制权,如题目组长只能管理小组长,而不能管理普通评卷员。所有角色,由系统管理员管理,系统管理员通过安全服务管理控制台建立和维护角色以及角色之间的层次关系,分配各角色之间的权限,并根据用户角色的不同授予用户不同的访问权限。例如普通评卷员只有评阅试题的权限,而小组长除了评阅试题外,还有管理小组成员和分析统计结果等权限。在服务器端,不使用配置文件来表示用户角色集合信息,因为使用配置文件不能体现用户之间的层次关系以及他们的权限继承关系,而是采用前面提到的数据库技术。将用户角色放到数据库的用户信息表中,这样既增加了程序的灵活性和安全性,又弥补了修改配置文件繁琐的缺点[5]。

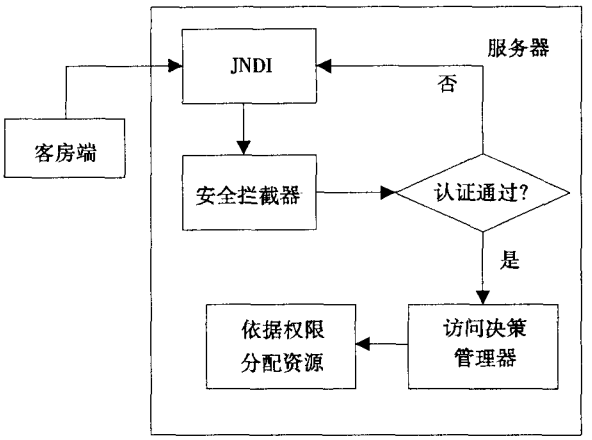

图5　网络阅卷系统访问流程图

Fig. 5　Access flowchart of the Network Marking System

3　认证与授权机制的实现

3.1　客户端认证的实现

客户端认证采用将用户提供的认证信息传递给服务器端,实现服务器端的认证。这样,客户端需要提供一种信息收集和传递的机制,用于在调用 Spring 组件之前与服务器端建立关联。

AuthLoginModule 模块用于客户端登录,主要作用是收集用户端的认证信息,并建立安全会话和服务器端的安全关联。服务器端的安全关联使用 SessionTokenManager 类来实现关联,该类的主要作用为设置和保存安全会话令牌,以供后继程序使用。AuthLoginModule 通过 JNDI 找到服务器

端的远程认证接口来完成用户的认证并建立安全会话,然后在 AuthLoginModule 的 Commit( )方法建立客户端的安全上下文。

```
//调用服务器端的认证管理器进行认证
public boolean login( ) throws javax. security. auth. login. LoginException{
Hashtable env = new Hashtable( ); env. putAll (options);
try{ InitialContext context = new InitialContext (env);
If ( options. get ( " authenticatorJNDI " ) ! = null) {
authenticatorJNDI = ( String) options. get( " authenticatorJNDI" ) ;}
authenticator = ( RemoteAuthenticator) context. lookup( authenticatorJNDI)
token = authenticator. authenticate ( principal, credential) ;
return true;} return false;}
```

3.2 认证管理器的实现

DaoAuthenticationProvider 类实现身份认证接口,提供身份认证功能并返回安全会话令牌。其主要方法 authenticator( )提供认证的具体操作的实现,其中参数 domainApplication 对象包含相关身份认证信息。

```
try{ LoginContext lc = new LoginContext ( DomainApplication. getDomainApplication ( ), new AuthCallbackHandler ( authCallback )); lc. login ( );}
Subject subject = lc. getSubject ( ) ;
AuthUserInfo authUserInfo = manager. getAuthUserInfo (username) ;
//使用会话管理器创建一个新的会话,并将相关的信息加入其中
SessionToken token = SessionTokenManager. createSessionToken (subject,
userInfo, domainApplication, authCallback) ;
return token;}
```

3.3 访问决策管理器的实现

PermissionRoleDBAdapter 类是实现一个基于角色的 PermissionAdapter 类。在其 initialize( )方法中,从数据中取得所有角色的权限信息并且组装入 Hashtable 集合类中。

在 getPermissions( )方法中收集相关主体有关的权限并且返回它们。相关的主体信息可以通过调用 ProtectedDomain 类的 getPrincipals( )方法得到。PermissionRoleDBAdapter 类的主要方法 authenticate( )实现对用户名/口令和数据库中存储数据比较工作[5]。

```
//这里定义了决策机制,需要全票才能通过
public void decide ( Authentication authentication, Object object, ConfigAttributeDefinition config) throws AccessDeniedException {
//这里取得配置好的迭代器集合
Iterator iter = this. getDecisionVoters ( ). iterator( ) ;
//依次使用各个投票器进行投票,并对投票结果进行计票
while ( iter. hasNext( )) {
AccessDecisionVoter voter = ( AccessDecisionVoter) iter. next ( ) ;
int result = voter. vote ( authentication, object, config) ;
//这是对投票结果进行处理,如果遇到其中一票通过,那就授权通过,如果是弃权或者反对,那就继续投票
switch ( result) {
case AccessDecisionVoter. ACCESS _ DENIED://这里对反对票进行计数
deny + + ; break; default: break;}//如果有反对票,抛出异常,整个授权不通过
if ( deny > 0) {
throw newAccessDeniedException ( messages. getMessage(
" AbstractAccessDecisionManager. accessDenied" ," Access is denied" )) ;}
//这里对弃权票进行处理,看看是全是弃权票的决定情况
checkAllowIfAllAbstainDecisions ( ) ;}
```

4 性能测试与分析

前面我们详细分析和设计了基于 Acegi 框架的网络阅卷系统,并给出了认证与授权的实现,下面我们搭建实验环境进行性能测试。

这里采用 Log4j 日志框架来记录用户的总非法请求量以及非法访问其他试题总量,以两种典型用户来比较运用 Acegi 框架前后的情况[6]:

① 一般用户:通过截取用户登录信息数据

包，假冒阅卷员身份或通过黑客攻击阅卷服务器等手段登录阅卷服务器，在总非法请求量一定的情况下，运用 Acegi 框架前、后的访问通过量的百分率如图 6 所示。

其中“1”、“2”分别代表运用 Acegi 框架前、后的访问通过量的百分率，通过比较发现，通过采用 Acegi 框架认证机制之后，非法请求通过身份认证的百分率由 13% 降低至 6%，阅卷服务器容易受到黑客攻击以及阅卷教师的身份被他人假冒的可能性大大减少。

② 普通评卷员：只是负责试题的评阅，在一个系统中往往拥有多个普通评卷员，将普通评卷员分成不同小组，不同的评卷员只对他所负责的试题评阅，他无权去查看或评阅其他部分的试题。在非法访问其他部分试题总次数一定的情况下，运用 Acegi 框架前后的访问通过量的百分率如图 6 所示。

其中“1”、“2”分别代表运用 Acegi 框架前、后的通过率，通过比较发现，运用 Acegi 安全框架授权机制之后，非法访问其他部分试题的访问通过量由 18% 降低至 4%，有效防止了教师的越权访问及评阅，实现了考试的公平、公正，从而增强了系统的安全性能。

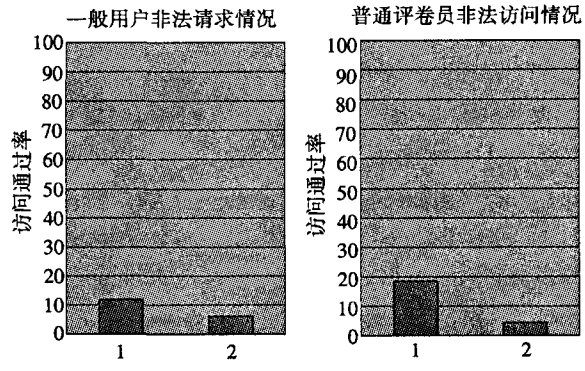

图 6　一般用户访问情况图

Fig. 6　General user access diagram

BoA－NMS 的设计降低了后续开发的难度，避免了人工干预所带来的误差，增强了网络阅卷系统的安全性，实现了考试的公平、公正，具有一定的实用推广价值。

参考文献：

[1] 朱从旭. 实现无纸考试与自动阅卷统的简易系统[J]. 计算机系统应用，2002，0215－18.

[2] 路鹏，殷兆麟. 基于 Spring 的 Acegi 安全框架认证与授权的分析及扩展[J]. 计算机工程与设计，2007，28(6)：1 313－1 316.

[3] 杨义在. 基于 J2EE 架构的网上阅卷系统的设计与实现[D]. 贵州：贵州大学，2008.

[4] 李孟珂，余祥宣. 基于角色的访问控制技术及应用[J]. 计算机应用研究，2000(10)：25 228.

[5] Sejong Oh，Ravi Sandhu. A Model for Role Administration Using Organization Structure[J]. SACMAT02，2002，6(2)：155－162.

[6] 刘渊，周世兵，孙亚民. 网上在线支付数字签名的研究与设计[J]. 计算机应用研究，2003，20(11)：110－112.

Design and Realization of Network Marking System Based on Acegi Framework

LIU Zhen-hai，TANG Yong-zheng

(Yancheng Institute of Technology，the Modern Educational and Technological Center，Yancheng Jiangsu　224051，China)

Abstract：Aiming at the existing network marking system safety deficiencies，the authentication and authorization mechanism of Network Marking System based on the security Acegi framework(BoA－NMS) was introduced. By constructing bidirectional authentication mechanism，Security Interceptor and designed role－based access control authorization mechanism of Acegi，the security and confidentiality of the Network Marking System was improved. It possessed a certain practical value.

Keywords：Acegi；Network Marking；authentication；authorization

(责任编辑：张英健)