# 企业信息门户单点登录系统的设计与实现①

## Design and Implementation of Single Sign – on in Enterprise Information Portal

陈观林　张　泳　（浙江大学城市学院 计算机科学与工程学系 浙江 杭州 310015）

**摘　要：**本文介绍了一个企业信息门户中单点登录系统的设计与实现。系统实现了一个基于 Java EE 架构的结合凭证加密和 Web Services 的单点登录系统，对门户用户进行统一认证和访问控制。论文详细阐述了该系统的总体结构、设计思想、工作原理和具体实现方案，目前系统已在部分省市的广电行业信息门户平台中得到了良好的应用。

**关键词：**单点登录 企业信息门户 Web 服务 认证 访问控制

## 1　引言

随着计算机信息技术的迅猛发展，Web 技术应用已在全球范围内普及。信息技术不仅实现了企业内和企业间信息的共享，而且改变着企业的经营运作方式。企业信息门户（Enterprise Information Portal，简称 EIP）就是近年 IT 领域一项重要的新技术，也是企业信息化的重要发展方向。

EIP 将企业的所有应用和数据集成到一个信息管理平台之上，并以统一的用户界面提供给用户，使企业可以快速地建立企业对部门、企业对员工的信息门户。EIP 是一个基于 Web 的系统，它通过唯一的访问入口，将企业各种不同的管理系统、数据库系统以及网络共享资源集成在一起，为用户提供形式多样的信息和统一安全管理机制。EIP 的构建涉及 Portal、内容管理、数据集成、单点登录等多方面的内容和技术，单点登录则是其中尤为重要的功能。

## 2　单点登录技术

单点登录（Single Sign On，简称 SSO）是一种认证和授权机制，主要目的是为了方便用户访问多个系统。用户只需在登录时进行一次注册，就可以在多个系统间自由穿梭，不必重复输入用户名和密码来确定身份，从而实现“一次登录，全网访问”。单点登录的实质是安全上下文（Security Context）或凭证（Credential）在多个业务系统之间的传递或共享。当用户登录系统时，客户端软件根据用户的凭证为用户建立一个安全上下文，安全上下文包含用于验证用户的安全信息，系统用这个安全上下文和安全策略来判断用户是否具有访问系统资源的权限。

SSO 登录方式减少了用户在不同系统中登录耗费的时间，避免了处理和保存多套系统用户的认证信息，缩短了系统管理员管理用户权限的时间，增加了管理的便利性。通过 SSO 功能，使得操作人员在企业信息门户的统一界面中，通过一次登录，可以了解到各个不同系统的信息，而且可以随意切换到相关的功能系统进行专业的业务处理。另外，SSO 系统从根本上抛弃了传统认证中用户名密码以明文传输的方式，而是采用了结合密码学技术的新的认证机制，从而大大提高了整个系统的安全性。

单点登录的原理如图 1 所示，具体的登录验证过程如下：

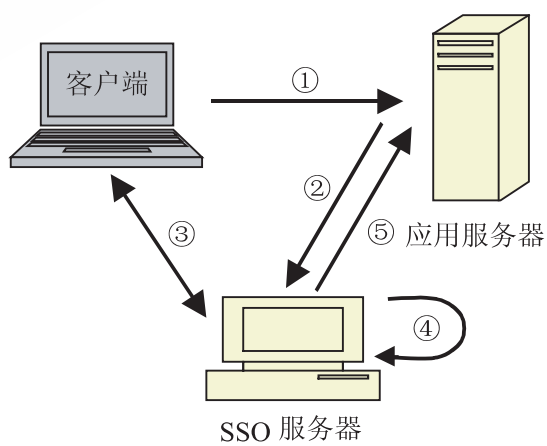


图 1　单点登录原理

① 基金项目：浙江省教育厅高校科研计划项目（20061290）

① 客户端向应用服务器请求访问某资源；

② 应用服务器重定向到 SSO 服务器请求凭证；

③ 如果用户未登录 SSO 安全域，SSO 服务器将请求重定向到身份认证服务；

④ 用户通过身份认证服务后，SSO 服务器为其生成身份标识，并签发身份凭证；

⑤ SSO 服务器重定向到应用服务器，后者验证凭证有效性，从而获得用户身份信息。

目前单点登录系统的实现有多种方法：一种是基于 ticket 凭证加密的认证，如 PKI、Kerberos 系统等；一种是建立在 Cookie 的基础上，如 IBM WebSphere、Bea WebLogic 等；另外一种是采用 Web Services 架构，将统一认证模块做成 Web 服务的方式。

本文通过凭证加密和 Web Services 相结合的方式提出一种 SSO 的解决方案，利用 Java EE 平台的 Servlet 技术提供 Web Services，将权限管理和认证授权进行抽象，集中在 Web 框架中，供所有的业务程序共享，从而提供企业信息门户中不同业务程序的单点登录功能。

## 3　系统总体设计

下面给出企业信息门户中单点登录系统的总体设计方案，该 SSO 系统通过 Web 服务传递用户身份信息，提供对已有业务系统的安全集成，可以访问门户内所有业务系统和数据，同时利用加密机制保证单点登录的安全性。图 2 为系统的总体结构图，主要包括用户注册/注销、用户登录以及与其他业务系统的验证等功能模块。

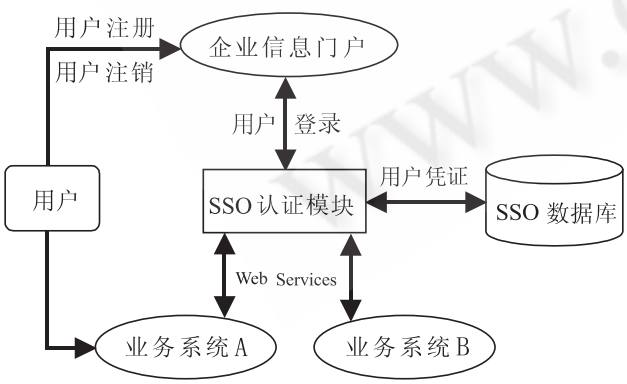


图 2　系统总体结构图

(1) 用户注册/注销流程：

①SSO 服务器管理员在 SSO 服务器上添加注册用户，保存该用户的统一登录名和密码；

②业务系统 A 的管理员在业务系统 A 的管理模块中选择所需注册用户，保存该用户在业务系统 A 的唯一识别字和用户名；

③SSO 服务器管理员可以根据用户的要求选择业务系统 A 对 SSO 的信任级别。信任级别分为三级：始终信任、验证通过时信任和始终不信任(默认为验证通过时信任)。如果选择始终不信任，则可以加强业务系统的安全性，防止其他用户利用单点登录的便利直接登录到安全性要求强的业务系统；

④用户需要在业务系统中注销时，SSO 中的相关信息一并注销。

(2) 用户登录/验证流程：

①用户登录 SSO 服务器时，会生成该用户的数字签名密钥对，并将该密钥对保存在 SSO 数据库中，以后用户登录 SSO，都会用该密钥对对用户名和密码进行签名验证；

②用户在计算机 A 上登录 SSO 服务器后，SSO 服务器会标识该用户已经通过单点登录认证，并进入门户平台。当用户要求启动业务系统 B 时，系统向 SSO 服务器发送请求用户的 SSO 用户名、待启动的业务系统 B 的标识代号，SSO 服务器判断用户是否已经通过 SSO 认证，确认通过验证后，SSO 服务器在用户计算机 A 上启动业务系统 B 的认证程序；

③业务系统 B 的认证程序通过 SSO 服务器的 Web Services 生成此次会话的密钥对，并将该密钥对保存到 SSO 数据库中，然后用其中一把密钥对用户对应的业务系统 B 的登录用户名进行加密，接着 Web Services 将解密密钥、密文和此次会话的 SessionID 发送至请求端。请求端程序用解密密钥将密文进行解密，获取用户对应的业务系统 B 的登录用户名，最后请求端程序验证该登录用户名，如果验证通过，则根据该用户的权限启动业务系统 B，并返回 Web Services 一个 Success 信息；如果验证不通过，则返回一个 Failure 信息。

系统具体认证流程如图 3 所示。

## 4　SSO 系统的数据库设计

SSO 系统的数据库负责存储 SSO 平台关于单点登录帐号、用户系统绑定、登录信息记录以及帐号加密等数据，主要包括 SSO 用户表、SSO 业务系统表、用户系

统关联表、业务系统登录信息表等数据表。

①SSO 用户表(SSO_USER):用来设置 SSO 平台的登录帐户信息,包括帐号、密码、密钥等内容。

②SSO 业务系统表(SSO_APPLICATION):用来设置与 SSO 平台系统相关联的业务系统信息,包括安全级别、业务系统代号等。

③用户系统关联表(SSO_USER_APP):用来设置系统和相应系统登录用户的关联信息,包括系统安全级别、系统登录用户帐号、SSO 登录 ID 等。

④业务系统登录信息(SSO_APP_LOGININFO):用来设置 SSO 帐号与业务系统进行交互生成密钥,相应业务系统通过解密获取正确信息从而顺利登录业务系统的数据,包括 SessionID,密钥,密文,业务系统 ID 等。

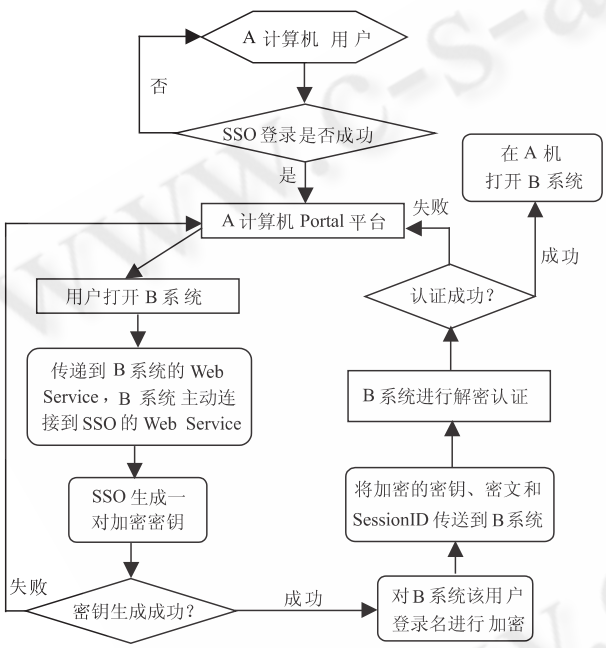


图 3　单点登录系统验证流程

表 1 列举了业务系统登录信息表的详细设置。

## 5　认证服务器的具体实现

认证服务器的实现是整个系统的核心,下面以办公业务系统的认证登录为例,通过以下步骤具体实现企业信息门户的单点登录功能。

(1)用户登录企业信息门户后,通过欢迎主界面 welcome. jsp 访问办公业务系统的页面链接,利用 userLogonAction 的 validate 方法进行调用;

表 1　业务系统登录信息表

| SSO_APP_LOGININFO业务系统登录信息 | | |
|---|---|---|
| UUID | VARCHAR(60) | <pk> |
| SESSIONID 登陆 SessionID | VARCHAR(128) | |
| APP_ID 应用系统 ID | VARCHAR(32) | |
| USER_ID 用户 ID | VARCHAR(10) | |
| PRIKEY 私钥 | VARCHAR(128) | |
| PUBKEY 公钥 | VARCHAR(128) | |
| DESKEY 密钥 | VARCHAR(128) | |
| CRYPTOGRAPH 密文 | VARCHAR(1024) | |
| ISIGNATURE 数字签名 | VARCHAR(128) | |
| CA 数字证书 | BLOB | |
| ASK_TIME 请求时间 | TIMESTAMP | |
| ASK_RESULT 请求结果(0\1\2) | CHAR(1) | |
| ASK_TIMEOUT 失效时间 | TIMESTAMP | |

(2) validate 方法接收用户的 SessionID,从而调用 SSOWebServiceBean 类中的 validate 方法来进行单点登录判断;

(3) SSOWebServiceBean 类进行认证处理后,如果验证正确,则将数据处理结果返回至办公业务系统 userLogonAction 的 validate 方法中;

(4)验证成功后,用户通过办公业务系统 userLogonAction 的 validate 方法登录到该业务系统。

SSO 系统通过 Servlet 技术实现认证应用 userLogonAction,采用 Web Services 对认证服务进行封装,在现有各异构平台的基础上实现了通用的与平台、语言无关的单点登录功能。

userLogonAction 的 validate 方法中的关键代码如下:

```
public ActionForward validate(ActionMapping ac-
tionMa –
pping, ActionForm actionForm, HttpServletRequest
servletReq –
uest, HttpServletResponse servletResponse) {
//获得 SessionID
String sessionid = servletRequest. getParameter("
sessionid");
try {
String[] result = SSOWebServiceDBO. getInstance
(). vali –
date(sessionid, CommonConstants. APPLICATION-
ID). split(",");//调用 SSOWebServiceDBO 方法返回
的结果,以逗号为分隔符号
if (!result[0]. equalsIgnoreCase("1000")) {
```

```
//判断单点登录的安全类型
servletRequest. setAttribute (CommonConstants.
ERROR_MESSAGE," 单点认证信息操作失败！\n" +
SSOWebServiceDBO. getInstance (). getMessages
(result[0]));
/登录失败
return actionMapping. findForward(" failure");/
} else {
String logonid = DESEncryptPlugin. getInstance().
decrypt(
result[2], result[1]);//解密用户名
List logonList = (List) UserLogonDBO. getInstance
().
doLogon(tableData);//通过用户名获取该系统
的用户相关信息
DBOperateMessage message = (DBOperateMes-
sage) logonList. get(0);//返回结果
if (message. isSuccessful()) {
//登录成功则更新 SSO 信息
SSOWebServiceDBO. getInstance (). validate (ses-
sionid,
CommonConstants. APPLICATIONID, result[1]);
return actionMapping. findForward(" success");
//成功进入首页
...
}}}
```

以上是 SSO 认证服务器对用户访问办公业务系统进行单点登录验证的详细开发的介绍，整个 SSO 系统采用了 Java EE 三层体系结构，集成了 MVC 设计模式、Struts 框架和 EJB 技术，以 BEA WebLogic 8.1 作为中间件服务器顺利实现。

## 6 结束语

本文介绍了一个企业信息门户中单点登录系统的设计与实现，系统通过凭证加密和 Web Services 相结合的方式，基于 Java EE 架构，提供了企业信息门户中不同业务系统的单点登录功能，应用端的认证由系统自动完成，方便对门户用户进行统一的管理和访问控制，具有较好的安全性。目前系统已在部分省市的广电行业信息门户平台中得到了良好的应用。

## 参考文献


1 皮晓东. 单点登录的研究与实现. 计算机应用与软件，2007，24(6)：156 – 158.

2 刘润达，诸云强，宋佳，冯敏. 一种简单跨域单点登录系统的实现. 计算机应用，2007，27(2)：288 – 291.

3 徐福仓，蔡玲玲，吴敏. 电子政务内门户单点登录系统的实现. 计算机系统应用，2007，(1)：99 – 101.

4 谭立球，费耀平，李建华. 企业信息门户单点登录系统的实现. 计算机工程，2005，31(17)：102 – 104.

5 邱航，杜向辉. 单点登录原型系统 KSSO 的设计与实现. 计算机工程与设计，2006，27(9)：1645 – 1648.