基于 CAS 的门户单点登录方案①

刘 峰[1]，王 峥[2]，曹华平[1]，罗守山[3]

[1](国家计算机网络应急技术处理协调中心，北京 100029);
[2](中油财务有限责任公司，北京 100007)
[3](北京邮电大学 计算机学院，北京 100876)

摘 要：门户是一种个性化、单点登录、内容聚合的 Web 应用系统。为了实现门户的单点登录，分析了不同的实现方法，选取基于集中式认证的 CAS 协议来实现，并针对 CAS 在单点退出的缺陷进行改进，提高系统安全性和用户访问便利性。

关键词：门户；单点登录；集中式认证；CAS；单点退出

Portal Single Sign-on Scheme Based on CAS

LIU Feng[1], WANG Zheng[2], CAO Hua-Ping[1], LUO Shou-Shan[3]

[1](National Computer Network Emgergency Response Technical Team/Coordination Center of China, Beijing 100029, China)
[2](China Petroleum Finance Co. Ltd, Beijing 100007, China)
[3](School of Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing 100029, China)

Abstract: A portal is a Web based application that commonly provides personalization, single sign-on and content aggregation. In order to achieve a single sign-on on portal, after analyzing different implementations, the paper finally implement it with the centralized authentication CAS protocol, and improves the CAS defect of single logout, so it progress much on system securities and user access conveniences.

Keywords: portal; single sign-on; centralized authentication; CAS; single logout

随着网络技术的发展和多年的信息化建设积累，一个企业或组织内部存在着多个独立、分散的面向不同业务的 Web 应用系统。然而，由于缺少统一规划，各系统基于自身视角，采用不同的技术架构，数据标准不统一且存在不一致和冗余，相互间难于共享资源，形成了“信息孤岛”。

为了提高数据的综合利用价值，往往会在此基础上建立一个整合了各个应用系统内容和服务的门户（Portal），使得不同用户角色（员工、客户、供应商等）都可以在其中方便的找到自己需要的所有内容。

JSR-168 规范提到，Portal[1]是一个基于 Web 的应用系统，通常会提供内容聚合、个性化、单点登录的信息系统表现层的平台。Portal 提供复杂的个人化设置供用户定制页面，使得不同用户角色登录 Portal 后，Portal 会根据用户的角色和个性化设置显示不同的信息内容。单点登录（Single Sign On，简称 SSO）[2]是指用户只需登录 Portal 一次就可以访问所有的应用系统，由 Portal 和应用系统协调合作，匹配出 Portal 用户和各个应用系统用户的对应关系，而要运行跨多个应用系统协同合作的工作流时，无需用户逐个输入各应用系统的用户名密码。

根据保留各个应用系统登录入口还是只信任 Portal 的登录入口，用户认证授权数据存放位置的不同，认证和授权过程分别在哪个系统里实现，存在多种 SSO 的实现方案。

1 常见的门户单点登录方法分析

1.1 基于认证信息数据库的二次登录

Portal 端有一个包含所有应用系统认证信息的数据库。用户登录 Portal 后，对于应用的访问首先会从

① 基金项目:国家自然科学基金(60803157);北京市自然科学基金(4092029)
收稿时间:2010-10-06;收到修改稿时间:2010-11-20

数据库中取得对应应用系统中该用户名、密码等，包装成一个请求，发送到应用系统的登录地址，自动进行二次登录。

此方法的本质是密码集中存储和跳转链接，对原有应用系统无需改动，但需要集中保存多套完整的认证信息数据库，既加重了 Portal 的负担，又增加了安全隐患。且各应用保留了自己的认证模块和用户管理模块，还得考虑认证信息在 Portal 和应用之间的同步问题。另外，这种单点登录是显式的，对用户不透明。仅从 Portal 提供的链接访问才是单点登录的，直接在应用原地址上访问只限于应用自身范围。

1.2 基于唯一认证中心的集中式管理

屏蔽掉原应用系统的登录入口，整个域（门户和集成的应用系统）内只信任唯一的认证中心。集中式单点登录要经过以下几个步骤：①所有应用共享一个认证中心。应用不处理登录请求，而是转发到认证中心。②认证中心验证用户身份（Credential），如用户名、密码、证书等。③认证成功后，认证中心生成加密的认证标志（Ticket），返还给浏览器。④以后用户访问其他从未访问过的应用时，将携带着这个 Ticket。⑤应用应该能对 Ticket 进行提取，通过与认证中心的通讯，请求对 Ticket 进行校验。⑥认证中心对 Ticket 检验的判断结果必须被应用信任，然后认证中心向应用返回用户信息（Principal）。⑦应用就会利用 Principal 建立与浏览器之间的会话（Session）。⑧当用户再次访问之前访问过的应用时，应用会通过 Session 判断用户已登录过。

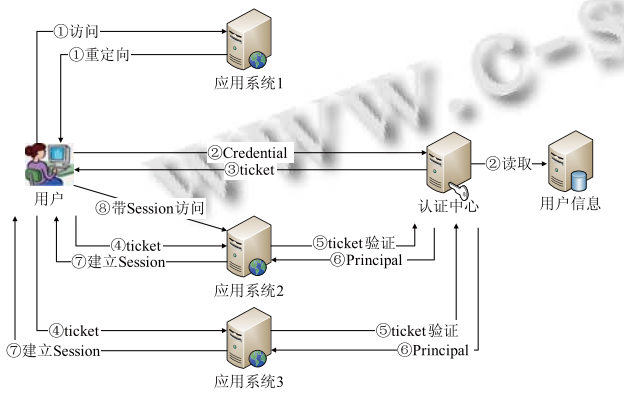

图 1　集中式单点登录的模型

这种方法需要在应用上配置过滤器以屏蔽掉登录入口，来处理与认证中心的通信。对原系统进行一定的改动，如果原系统的认证、授权机制是组件化的，容易剥离和替换，就不难实现。

另外，这种方法在用户认证授权数据存放位置、认证和授权过程的交接两个方面非常灵活。认证中心处可以不保留任何应用的认证授权数据，在返回给应用的 Principal 中只包含 Portal 的用户，由应用自己处理 Portal 用户与自身用户的对应关系；也可以直接将对应应用的用户信息打包到 Principal 返回给应用，或者更进一步将授权数据也打包，即原应用接管了授权过程。但是不用保存各个应用用户的密码和证书，减少了数据量。一般情况下采用集中的用户认证数据存储，来简化管理、避免同步问题，授权是否集中可根据情况而定。在此基础上开发新应用时，可以省去独立的身份认证模块，避免了编写功能同质化、内容相似的代码，提高了效率。因此，大多数门户选取此种方法来实现单点登录。

2　CAS协议

CAS（Central Authentication Service）[3]是耶鲁大学发起的一个实现 Web 应用单点登录的开源项目。它通过独立的认证中心来实现集中式的认证；不断开发不同语言的客户端认证模块，能够适合任何语言编写的应用；使用 HTTPs 作为通讯协议保证安全性；在认证中心缓存了身份票据，无需在浏览器端共享 cookie，解决了以往集中式单点登录不能跨域的问题[4]。

2.1 组成要素

服务（Service）：前端服务是指浏览器可直接访问的 Web 应用；后端应用不能直接被浏览器访问，由前端应用调用，获取数据间接返回给浏览器。

CAS Server：认证中心的角色，需要在服务之外独立部署。CAS Server 采用何种认证数据库的存储和查询技术，跟 CAS 协议是分离的。

CAS Client：部署在服务上，用来屏蔽和转发访问请求。

身份票据（Ticket Grant Cookie/Proxy Grant Ticket）：登录后由 CAS Server 为每个用户生成唯一一份，返回给访问者作为身份的唯一依据，以后访问从未访问过的新服务时要凭此得到授权。身份票据有两种，分别是用于基本协议的 TGC 和代理协议的 PGT。

TGC 保存在浏览器 cookie 中，PGT 保存在服务的会话（Session）中。

服务票据（Service Ticket/Proxy Ticket）：有两种，分别是用于基本协议的 ST 和代理协议的 PT。通过了授权后，转发服务票据给授权服务以告知其访问权。服务向 CAS Server 验证服务票据的真实性后，允许用户访问应用系统。

2.2 协议

基础思想是通过唯一的身份票据来获取对应不同服务使用请求的服务票据，通过服务票据来访问服务[3,5]。

2.2.1 基本协议

基本协议以浏览器为访问者。在未建立起服务与浏览器的 Session 前，CAS Client 过滤浏览器访问请求，如果请求中不包含 ST，CAS Client 则重定向请求到 CAS Server 对用户进行认证，将 TGC 返回给浏览器；CAS Server 还会产生一个随机的 ST，然后写入请求重定向到 CAS Client。最后，CAS Client 和 CAS Server 之间完成了一个对 ST 的验证，CAS Server 返回给 CAS Client 用户信息（NetID），CAS Client 建立服务与浏览器间的 Session，允许访问。

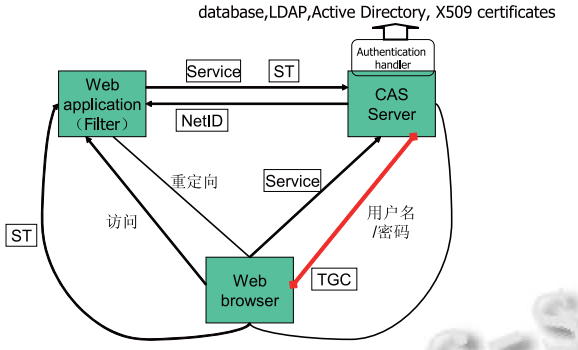

图 2 CAS 基本协议

2.2.2 代理协议

代理协议[6,7]以前端服务为访问者，可理解为基本协议的嵌套调用。①第一层调用在基本协议的基础上，附属了 PGT 和 PGTURL（PGTIOU 用于关联作用）。PGT 被前端应用（图 4 中的 Web Application）用来作为代理访问后端应用的凭据。②嵌套调用时，前端应用向 CAS Server 提交 PGT 和 Service（应用的标识），CAS Server 给前端应用发放 PT，PT 跟 ST 的作用一模一样，前端应用传 PT 给后端应用(图 4 中的 Back-end application)，后端应用就可以根据 PT 从 CAS Server 处获得被代理的用户的 NetID。

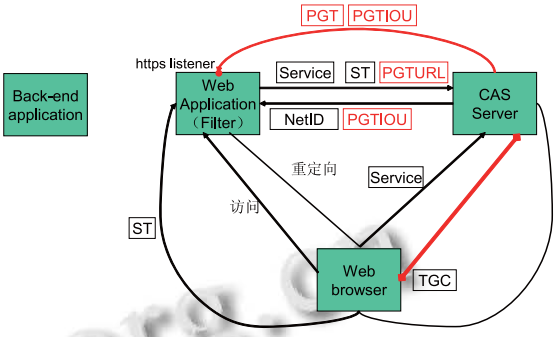

图 3 CAS 代理协议的第一层调用

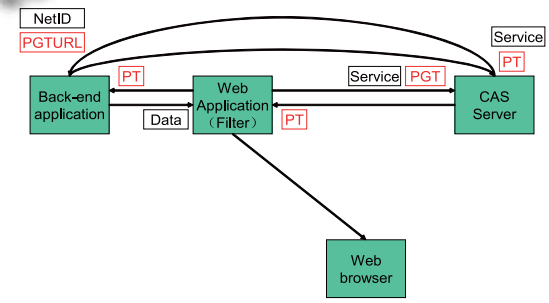

图 4 CAS 代理协议的嵌套调用

2.3 安全机制

TGC/PGT 通过 HTTPs 协议传送，且有自己的存活周期，超时则无效。ST 基于随机数生成，只能使用一次，无论 ST 验证是否成功，CAS Server 都会将服务端的缓存中清除；假设用户拿到 ST 之后，他请求服务的过程又被中断了，ST 就被空置了。此时为了防止 ST 被截取盗用，给 ST 设定了比 TGC 短得多的存活周期。

3 系统分析与设计

3.1 系统设计

IAAA 用户数据库：一个统一的用户信息数据库，整合或共享所有应用系统的用户，它存储了用户的身份信息、授权信息、操作日志等等，是实现统一认证服务的基础。

IAAA 系统：它提供用户的身份管理（Identity）、认证（Authentication）、授权（Authorization）、审计（Audit）这四大公共服务的独立 Web 应用，屏蔽和替代了各应用系统内部的用户身份管理模块。它在 CAS

Server 基础上扩展，只有它可以访问、修改 IAAA 用户数据库。当认证通过后，将授权数据以 XML 格式附在消息中返回给应用。

综合门户和 Web 应用：综合门户采用 Oracle Portal 技术，以 Portlet1 整合 Web 应用的内容形成 HTTP 代码片段，多个 Portlet 合成一个页面。实质上还是一个独立部署的 Web 应用。为了屏蔽和转发认证请求，实现集中式认证，在综合门户和 Web 应用上配置 CAS Client。Portlet 使用 PortletRequest 和 PortalResponse 来处理本 Portlet 范围内的认证请求。为了使 Portlet 支持 CAS 协议，扩展 Portlet 以便在 PortletSession 中接收 PT 并发送验证 PT 请求。

公共服务：如上次登录信息、证件照片、捡拾卡信息、密码重置等。这些服务被抽取出来，采用 WebService 技术封装并单独部署，可以被其他应用、公共服务调用，但浏览器不能直接访问。这些服务只能对登录用户开放，对公共服务的调用采用 CAS 代理协议来完成用户身份认证，因此在发布公共服务的服务器上也要配置 CAS Client。

3.2 CAS 在单点退出（Single Logout）问题上的不足

表 1　CAS 协议中与认证有关的实体

实体	保存位置	参与阶段和发挥作用
TGC	1、CAS Server 的 Ticket cache 中； 2、浏览器 cookie 中	从 CAS Server 登录以后，访问从未访问过的某个服务，用来申请 ST
PGT	1、CAS Server 的 Ticket cache 中； 2、已访问过，并且具有代理授权的服务的 Session 中	已访问过的服务以前段服务身份代理访问它本身从未访问过的另一个服务，用来申请 PT
ST/PT	1、CAS Server 的 Ticket cache 中； 2、用户欲直接或间接访问某个服务的请求中	CAS Client 获得浏览器转发的 ST/PT，再发回 CAS Server 验证通过，即可获得服务的访问权，得到 NetID，建立 Session
Principal	已访问过的服务的 Session 中	继续访问已获访问权的某个服务

表 1 列举了在 CAS 协议的不同步骤，参与认证过程的各种实体的保存位置和发挥的作用。在 CAS 协议定义中，在 CAS Server 开放了一个 logout 接口，用于接受退出请求。但是 logout 接口仅仅清理掉了浏览器 cookie 中的 TGC 和 CAS Server 中的 Ticket cache，仅能阻止继续使用已经失效的 TGC/PGT 来申请访问从未访问过的服务。然而，对于已获访问权的服务，浏览器与应用服务器已经建立起 Session，而 CAS Server 没有销毁该 Session，使得这些服务仍能继续访问。而本文也通过实验证明了这一漏洞。

3.3 改进方案

对 CAS 协议进行扩展，增加单点退出协议。单点退出协议有两个触发时机，一个为被动，即用户发出退出请求时；一个为主动，当某一 TGC/PGT 超时失效时，CAS Server 除了清理 CAS Server 中的 Ticket cache，还要向所有已获得访问权的服务发出销毁 Session 的请求。

所有请求都由服务主动重定向到 CAS Server，因此 CAS Server 本身并不知道服务的地址。CAS Server 若想主动向服务发送请求来清除 Session，必须事先在 CAS Client 上配置的一个监听接口 LogoutListener。监听接口接收包含 Session 的唯一标示的退出请求，在应用服务器上销毁 Session。为了防止假造请求和截获信息，相互通信使用 HTTPs 协议。

然后，要对 CAS 基本协议和代理协议做简单的修改。在应用系统向 CAS Server 验证 ST/PT 有效后，CAS Client 还将向 CAS Server 回送一个请求。这个请求包含两个参数：LogoutListener 接口的地址 LougoutURL 和认证成功后应用系统建立的 Session 的标识 clientSession。

4　结束语

本文提出了基于 CAS 协议实现集中式认证的门户单点登录，并增加了单点退出协议，提高了用户访问便利性和系统安全性，具有一定的实用和推广价值。由于 CAS 协议是基于 HTTPs 的，下阶段将考虑配置密钥数据库，对 SSL 密钥的生成、存储、访问和集中管理进行深入的研究。

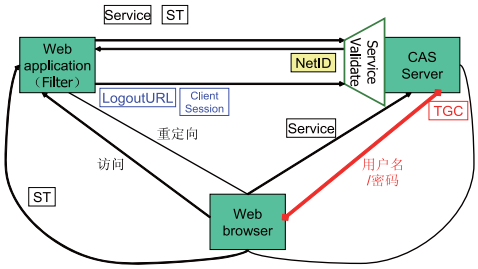

图 5　对 CAS 基本协议的修改

(下转第 102 页)

$$e(k_1,k_2)=e(xk_AQ_B,x^{-1}Q_A)$$
$$=e(k_AQ_B,Q_A)=e(Q_B,Q_A)^{k_A}$$

而对于 PKG 伪造的密文，有

$$e(k_1',k_2')=e(xk_A'Q_B,x^{-1}Q_A)$$
$$=e(k_A'Q_B,Q_A)=e(Q_B,Q_A)^{k_A'}$$

因此，当 PKG 伪造了签密者的密文时，签密者可以通过出示其秘密信息来证明该密文是 PKG 伪造的。

6 结束语

新方案克服了原方案的安全缺陷，同时实现了前向安全性、公开验证性和 PKG 的不可诬陷性；并且，新方案在解签密过程中，签名验证通过后再进行解密，避免了恶意信息的攻击，安全性得到了显著的提高，实用性更强。

参考文献

1 张申绒,张玉清,李发根,肖鸿.适于 ad hoc 网络安全通信的新签密算法.通信学报,2010,31(3):19–24.

2 Zheng Y. Digital signcryption or how to achieve cost (Signature&Encryption)<<Cost(Signature)+Cost(Encryption) .In:Crypto'971 Berlin:Springer,1997.165–179.

3 Zheng Y. Signcryption and its applications in efficient public key solution.ISW'97, LNCS 1397, Springer-Verlag, 1998.

4 Shamir A. Identity-based cryptosystems and signature schemes.Advances in Cryptology-Crypto'84, LNCS 196, New York: Springer-Verlag, 1984.47–53.

5 Ibert LB, Isquater QJJ. A new identity based signcryption schemes from pairings. Proc of IEEE Information Theory Workshop. Springer-Verlag, 2003. 155–158.

6 Boyen X. Multipurpose identity-based signcryption: A Swiss army knife for identity-based cryptography. Advances in Cryptology-CRYPT0 2003. Berlin: Springer-Verlag, 2003: 383–399.

7 Chow SSM, Yiu SM, Hui LCK, et al. Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity Information Security and Cryptology-ICISC'03. Berlin: Springer-Verlag, 2004:352–369.

8 Boneh D, Franklin M. Identity-based encryption from the Weil pairing. Advances in Cryptology-Crypro'2001, Berlin: Springer-Verlag,2001.213–229.

9 Miller V. Short Programs for Functions on Curves. [2009-02-23]. http://crypto.Stanford.edu/miller/miller.pdf

10 Zheng Y, Imai H. How to construct effcient signcryption schemes on elliptic curves. Information Processing Letters, 1998,68:227–233.

(上接第 81 页)

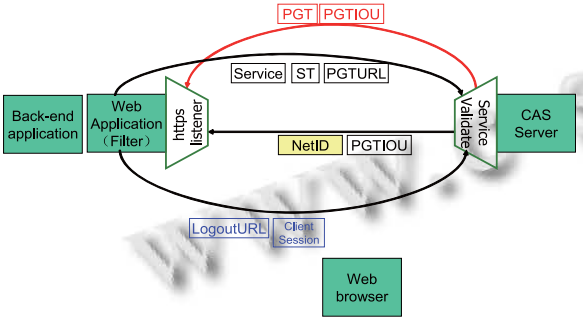

图 6 对 CAS 简单代理协议的修改

参考文献

1 Nakano H, Sugitani K, Nagai T, Kubota S, Migita, M, Musashi Y, Iriguchi N, Usagawa T, Kita T, Matsuba R. Web-based time schedule system for multiple LMSs on the SSO/portal environment. In: Education Engineering (EDUCON), 2010 IEEE. 153–158.

2 The Open Group. Single Sign-On. 1995-2010. http://www.opengroup.org/security/sso

3 JA-SIG Central Authentication Service. 2009. http://www.ja-sig.org/products/cas/

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 Petro A. CAS and JSR-168. 2007-02. http://www.ja-sig.org/wiki/display/CASC/CAS+and+JSR-168

6 Petro A. Proxy CAS Walkthrough. 2009-2. http://www.ja-sig.org/wiki/display/CAS/Proxy+CAS+Walkthrough

7 Turing D. 剖析 CAS Proxy 的设计原理. 2006-04. http://www.blogjava.net/security/archive/2006/04/26/sso_cas proxy.html