doi:10.11887/j.cn.201403022 http://journal.nudt.edu.cn

面向大型机构的统一身份管理方法*

王 超[1],郭长国[1],刘东红[1],李安琪[2],王怀民[3]
(1.中国电子设备系统工程公司研究所,北京 100141;
2.解放军后勤学院,北京 100858;
3.国防科技大学,湖南 长沙 410073)

摘 要:通过借鉴自由联盟组织提出的Liberty框架,并针对该框架的结构以及身份提供者之间信任关系的建立模式进行改造,提出一种面向大型机构的新型身份管理联盟。新型的身份管理联盟比Liberty框架更加适用于具有分布性、自治性、全局性和协同性的大型机构。新型的身份管理联盟在物理结构上可看作由多个身份提供者节点组成的一棵树,其中每个身份提供者节点必须并且只能与其父节点和子节点建立信任关系,这与现实中各个大型机构的树状层级结构是完全相符的。在系统实现过程中,依托新型身份管理联盟的树状结构,并采用LDAP实现了用户认证数据的分级存储。此外还通过安全认证网关在网络层对用户的接入进行控制,从而能够同时支持B/S和C/S两类应用系统的单点登录。

关键词:大型机构;统一身份管理;身份管理联盟;身份提供者;单点登录
中图分类号:TP391 **文献标志码**:A **文章编号**:1001-2486(2014)03-0122-07

Unified identity management method for large organizations

WANG Chao[1], *GUO Changguo*[1], *LIU Donghong*[1], *LI Anqi*[2], *WANG Huaimin*[3]
(1. Institute of Chinese Electronic Equipment System Engineering Corporation, Beijing 100141, China;
2. Liberation Army Logistics College, Beijing 100858, China;
3. National University of Defense Technology, Changsha 410073, China)

Abstract: In order to solve the problems of unified identity management in large organizations' information systems, a new identity management alliance for large organizations is proposed. Through consulting the Liberty Framework raised by the Liberty Alliance Organization, as well as reforming its structure and its pattern of building trust relationships among IDPs, the new identity management alliance is more suitable than the Liberty Framework for large organizations, which are distributed, autonomous, globally unified, and coordinated. In terms of physical structure, the new identity management alliance can be regarded as a tree consisting of multiple IDP nodes while each node must and can only develop trust relationships with its father node and child nodes. This is totally in line with the tree-like hierarchy of every large organization in the real world. In the system realization, based on the tree-like structure of the new identity management alliance, a hierarchical storage of the authentication data is achieved by adopting LDAP. In addition, the user access control was conducted by a security authentication gateway at the network layer, which consequently makes it possible to support B/S and C/S application systems at the same time.

Key words: large organization; unified identity management; identity management alliance; IDP; single sign-on

随着互联网技术的发展和网络空间的延伸,以政府和跨国公司为典型代表的大型机构试图利用互联网拓展新的业务渠道和服务模式。注册和登录几乎是用户访问所有网站的必需步骤,用户的用户名、口令或证书等认证数据则是各类网站重点保护的敏感信息,因此大型机构在开展互联网业务之前必须首先建立安全可靠的身份管理机制。所谓身份管理,就是为用户提供统一的身份标识和认证机制,使合法用户能够登录应用系统并访问系统资源,同时运用一系列技术手段实现对用户认证数据的全生命周期管理[1]。互联网上的大部分网站都独立维护着各自的用户认证数据,并且采用了不同的身份标识和认证方法,因此用户不得不保留他们在每一个网站上使用的用户名和口令。统计结果显示,每个用户一生平均需要记忆40对用户名和口令[2],这显然已经成为了一种负担。尽管有些人选择在每个网站设置相同的用户名和口令,但是这种做法增加了用户身份

* **收稿日期**:2013-08-26
基金项目:国家自然科学基金资助项目(91118004)
作者简介:王超(1985—),男,山东聊城人,工程师,硕士,E-mail:chrace.wang@hotmail.com

被盗用的危险。此外,一个网站维护的用户认证数据已经足够庞大,而用户名或口令被用户遗忘之后的重置或找回处理则进一步增加了网站的运维成本。因此,在基于互联网构建身份管理系统时,大型机构面临的主要问题不在于具体的认证协议,而是如何降低用户认证数据的维护成本,同时改善用户的登录体验。理想条件下,大型机构的身份管理系统应该对各个业务系统的用户认证数据进行集中管理,并在此基础上提供统一的身份认证功能,使得所有的业务系统能够共享统一的用户群体;同时它还应该支持单点登录,使用户能够获得"一次登录、随处访问"的操作体验。如果能够达到上述目标,即实现了统一身份管理。

1　国内外研究现状

早在上个世纪末,微软公司就开始基于. NET 平台推广"Passport"项目,该项目以微软的 Passport. com 站点作为身份管理的核心枢纽,对所有用户的认证数据进行集中存储和维护,并为微软的全部授权站点统一提供身份认证服务[3]。用户只需要在任意一个授权站点登录并获取通行证,就可以自由地访问其他授权站点,而不需要重复登录。尽管"Passport"项目实现了单点登录,但是业界普遍担心微软会借机收集大量用户的个人信息,因此该项目并未得到广泛支持,最终以失败告终[4]。随后微软又启动了"Windows CardSpace"项目,致力于为异构身份管理系统提供抽象身份表示层的"身份元系统"研究。该项目在以"用户名 + 口令"为主的传统认证方式基础上扩展了基于身份标识卡的新型认证方式,并能够有效地防止钓鱼等网络犯罪行为的发生[5]。

2001 年,为了在身份管理方面与微软抗衡,SUN 公司牵头成立了由 150 多家 IT 公司组成的自由联盟组织,该组织致力于创建一种兼具架构开放性和实践指导意义的身份管理解决方案[6]。通过采用分布式的身份管理技术,自由联盟提出了 Liberty 框架,该框架的核心设计理念是建立"身份联盟",即应用系统在各自保留原有身份管理机制的前提下,围绕某个身份管理中心签署信任协议,并分别与其建立身份关联关系,从而形成一个身份管理信任圈[7]。由于存在身份关联关系,当用户在身份管理中心通过登录认证之后,就可以直接访问该信任圈内任意的应用系统,即实现了单点登录。根据身份管理中心之间的信任关系,多个信任圈又可以自发地组建规模更大的信任域,从而扩大单点登录的覆盖范围[8]。这种由信任圈、信任域两级形成的身份联盟架构一方面为单个应用系统扩展了用户群体,另一方面由于减少了可能的登录次数而改善了用户的使用体验。与集中统筹式的 Passport 相比,Liberty 最根本的改变则在于身份管理中心不再唯一,用户的认证数据也不再被集中存储和维护。

为了具备更好地开放性和兼容性以便于推广应用,自由联盟将 Liberty 框架构建在一些已被广泛接受的国际标准和规范之上,其中最基础和核心的部分是由结构化信息标准促进组织(Organization for the Advancement of Structured Information Standards, OASIS)在 2002 年 11 月制定的安全断言标记语言(Security Assertion Markup Language, SAML)规范,目前该规范已经发展到了 2.0 版本[9]。顾名思义,SAML 规范其实是一种标记安全断言的 XML 扩展,它能够支持异构安全服务系统之间的互操作,因此通常被用来解决开放网络环境下 Web 应用系统单点登录以及 Web 服务安全等问题。作为一种平台无关且易于扩展的安全信息交换机制,SAML 规范目前已经得到了众多大公司的支持。

除上述公司和组织所提供的解决方案和标准规范之外,各大开源社区也有许多与身份管理相关的协议及其开源实现不断地涌现出来。目前运用较为广泛的是由耶鲁大学发布的中央认证服务(Central Authentication Service, CAS)协议[10],以及由 Live Journal 社区创办人 Brad Fitzpatrick 推出的 OpenID 协议[11]。它们都能够在开放的网络环境下解决 Web 应用系统的单点登录问题,但是两者的实现机制有所不同,CAS 通过一个中央认证服务器为多个 Web 应用系统提供集中式的身份认证服务[12],而 OpenID 则完全采用去中心化的架构,支持用户自由选择甚至搭建自己的认证服务器[13]。

2　需求分析与技术路线选择

2.1　大型机构信息系统需求分析

大型机构通常都具有以下四个显著特征:一是分布性,大型机构通常由多个部门或单位组成,其地理覆盖范围很有可能是整个国家,有些跨国公司甚至可以覆盖全球;二是自治性,各个部门(单位)都在自身覆盖范围内行使独立的管理职能;三是全局性,各个部门(单位)在自治的同时,必须接受上级的统一协调,即"局部服从全局";四是协同性,大型机构往往具有领域相关的远景目标,其下属的所有部门(单位)都会为这个目标而共同努力。以

上特征决定了大型机构的信息系统具有多域协同特性:一方面,每个部门(单位)分别对各自的用户和资源实施运维管理和安全保障,从而形成多个相互独立的自治域;另一方面,为了合作完成任务,部门(单位)之间必须实现跨自治域的系统互连互通,如跨域服务访问和数据传输等。随着业务规模的扩大以及部门(单位)之间依赖程度的加深,这些跨域的交互行为逐渐常态化。

大型机构下属的各个部门(单位)通常位于不同的地理区域。虽然在理论上可以通过单一的数据中心为各个部门(单位)的业务信息系统提供统一保障,但是这种"一对多"的保障模式在实施效果上很容易受到网络状况和并发访问量的影响,而且该数据中心一旦发生故障必然导致整个大型机构陷入瘫痪。因此大型机构不可能将其数据中心部署为单一的物理节点,而应该使其具备"物理分布、逻辑统一"的分布式特征。此外,大型机构一般具有较为清晰和固定的树状层级划分,在层级树中处于上级的部门(单位)通常对其下级部门(单位)具有管辖权,而且一个部门(单位)所处的层级与其覆盖的地理区域也有直接关系,即上级部门(单位)的覆盖区域会包含其下级部门(单位)的覆盖区域。因此,大型机构在数据中心建设中也会考虑层级划分的问题,将各个物理节点按照实际的层级关系组织成一颗树,该树中的每一个物理节点都对应到处于相应层级的某个部门(单位),从而为该部门(单位)覆盖区域内的业务信息系统提供保障。

2.2 统一身份管理技术路线选择

统一身份管理系统的体系架构可以分为独立架构、集中架构和联盟架构三种[14]。其中独立架构的实质是各个应用系统独立实施身份管理,这在技术上是最容易实现的一种,但是它无法支持单点登录,因而不适用于大型机构开展大规模系统集成的应用场景;采用集中架构的典型技术是CAS协议,它虽然解决了单点登录的问题,但是中央认证服务器作为该协议的核心,需要为所有的应用系统和用户实体提供认证服务,因而不适用于具有广阔地理覆盖范围的大型机构;以Liberty框架为代表的联盟架构由于采用了SAML规范,能够在开放网络环境下通过身份断言机制实现单点登录,而且还可以通过在不同的身份提供者之间建立信任关系来扩展单点登录的支持范围。考虑到大型机构的特征属性,以及其信息系统所处的多自治域环境,目前大型机构在统一身份管理的具体实现上应该借鉴联盟架构模式。然而由于"自由联盟"组织在基于互联网的商业合作模式之上所构建的Liberty框架并不一定完全满足大型机构的统一管理和维护等实际需求,还需要根据大型机构的层级化保障模式来设计身份提供者的具体实现,使得它们之间建立的信任关系能够与现实中的树状层级关系相匹配,从而将原来完全自发形成的身份管理联盟改造成一种在大型机构统一管理维护之下的"受控联盟"。

统一身份管理必须实现统一身份认证和用户单点登录这两个主要目标,其中统一身份认证的实现必须对用户的认证数据进行集中管理。当前用户认证数据的存储和管理方式主要有数据库和目录服务两种,考虑到认证数据的数据量较大,而且在大多数情况下面临着频繁的读取操作,对认证数据的新增、修改和删除等操作的频率则相对较小,因此目录服务比数据库更加适应现实需求。但是还需要根据大型机构各部门(单位)之间的层级关系,对目录服务的存储结构进行分布式设计,从而在集中管理的前提下实现用户认证数据的分级存储,以适应大型机构广阔的地理覆盖范围和大量用户的并发访问。在用户单点登录方面,单点登录的实现模型可以分为经纪人模型、代理人模型和网关模型三种[15]。其中经纪人模型必须对原有的应用系统进行改造,才能够使它们识别认证服务器发放的电子身份标识;代理人模型虽然不需要进行大规模改造,但是代理软件的适用范围有限,因此同样会增大应用系统集成的难度。网关模型与以上两种模型不同,它的本质是一种利用安全认证网关在网络层对用户进行接入控制的方法。在通过安全认证网关对用户和应用系统进行隔离的前提下,无论用户在其计算机终端上使用浏览器还是各种客户端程序,都必须首先通过安全认证网关的登录认证,否则发出的访问请求无法透过安全认证网关而到达应用系统。这种将接入控制实现在网络层的方法一方面减少了对应用系统集成的影响,另一方面有助于在单点登录方面同时支持C/S和B/S等多种软件体系架构,从而使终端用户获得更好的操作体验。

3 统一身份管理系统架构设计

3.1 面向大型机构的身份管理联盟

目前,联盟架构在商业领域的典型应用是自由联盟组织基于SAML规范建立的Liberty框架。在Liberty框架中,通常由一些影响力较大的网站(如Google等)充当身份提供者。其他网站作为身份依赖者,可以自由地选择可信任的身份提供

者,并与它们建立起身份关联关系。当用户访问某个网站时,该网站能够以重定向的方式请求其依赖的身份提供者对该用户进行认证。身份提供者在完成对用户的认证之后,通常将身份认证信息以 SAML 断言的形式发送到目标网站,而目标网站则把收到的认证断言与本地的用户账户进行匹配,从而以模拟的方式实现用户的自动登录[16]。此外,身份提供者之间还能够自发地建立信任关系,使得身份认证信息能够在它们之间相互传递,从而每一个身份提供者不仅可以直接处理身份认证请求,还可以通过其他可信的身份提供者进行间接的身份认证。相互信任的身份提供者将它们各自的身份依赖者联合在一起,形成一个身份管理联盟,用户无论登录到哪个网站,都可以自由地访问同一联盟内的其他网站,而不需要再次登录[17]。

在上述身份管理联盟中,无论是身份依赖者对身份提供者的选择,还是各个身份提供者之间信任关系的建立,都是完全自发的行为,这属于典型的“自由联盟”模式。如果把“自由联盟”模式直接应用于大型机构的统一身份管理系统,大型机构将无法掌握全局范围内的依赖和信任关系,更不可能对这些关系进行调整。因此必须根据大型机构的特征对“自由联盟”模式进行改造,使它能够满足大型机构的统一管理和维护需求。考虑到大型机构的数据中心在物理部署结构上完全符合大型机构的树状层级划分,可以在数据中心的每一个物理节点部署一个身份提供者,由它负责为该节点覆盖范围内的身份依赖者统一提供身份认证功能。在为各自的身份依赖者处理身份认证请求的同时,每个节点的身份提供者必须并且只能与其父节点和子节点的身份提供者建立起信任关系,即它只能将身份认证请求转发给其父节点或者某个子节点的身份提供者,这使得身份提供者之间的信任关系仅仅并且必然存在于具有父子关系的节点之间。图 1 由此给出了一种具有树状结构的身份管理联盟,这种联盟架构能够与现实中各部门(单位)之间的树状层级关系相匹配。

在上述身份管理联盟中,当某个用户访问某个网站时,该网站将身份认证请求发送到其所在节点的身份提供者之后,存在 3 种不同的情形:如果用户与网站处于同一个节点,网站所在节点的身份提供者应该直接处理该身份认证请求;如果用户所在节点属于网站所在节点的一棵子树,则网站所在节点的身份提供者应该将身份认证请求沿着两个节点之间的路径转发到用户所在节点的

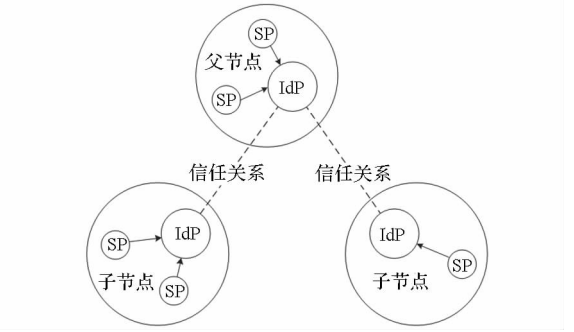

图 1　具有树状结构的身份管理联盟

Fig. 1　Identity management alliance with a tree structure

身份提供者;否则网站所在节点的身份提供者应该将身份认证请求转发给其父节点的身份提供者。上述过程将一直持续下去,直至身份认证请求被用户所在节点的身份提供者接收为止。待用户所在节点的身份提供者完成认证之后,身份认证断言才被发送到目标网站,最终完成模拟登录。

综上所述,由于身份提供者完全依托大型机构的数据中心来构建,并且能够与数据中心的各个物理节点一一对应,此外身份提供者之间的信任关系也能够与各个物理节点之间的拓扑关系相匹配,即与大型机构各个部门(单位)之间的树状层级划分相匹配,最终构建的身份管理联盟将不再是自发形成的“自由联盟”,而是一种在大型机构统一管理之下的“受控联盟”。

3.2　基于目录服务的认证数据管理

所谓目录,就是按照某种顺序排列起来的对象信息列表,这些信息用于详细记录与对象相关的各种属性[18]。目录在计算机领域的应用通常称为目录服务,它可以被看成一种用于存储描述性信息的特殊数据库。与普通数据库不同,目录服务针对数据的读取和搜索操作进行了专门优化,能够保证在数据量较大的情况下快速响应这些操作请求;此外目录服务还提供大范围复制信息的功能,从而在缩短响应时间的同时提高了可用性和可靠性;然而目录服务不具备事务处理能力,因而不支持批量的数据更新,只能执行简单的数据更新操作[19]。考虑到大型机构在实现统一身份管理的过程中需要为各个业务信息系统提供统一的身份认证功能,因此必须对全局所有用户的认证数据(包括用户名、口令或证书,以及一些基本属性)进行集中存储和维护。这些认证数据的数据量很大,并且需要面临频繁的读取和复杂的搜索操作,但是发生数据更新的可能性相比之下却小得多,因此可以将目录服务技术应用到大型机构的认证数据管理中。目录服务技术主要包

括 X.500 和 LDAP 两个国际标准协议,其中目前应用较为广泛的是 LDAP[20]。

LDAP 目录服务通常按照现实世界的地理位置和组织结构对 LDAP 目录服务中的所有条目进行组织,最终形成一棵目录信息树(Directory Information Tree,DIT),对目录信息树的设计会直接影响到系统的扩展性、适应性和动态性[21]。常见的目录信息树设计方式有两种,一种是按照地理位置设计,另外一种是按照组织结构设计。在设计过程中需要遵循的原则是尽量减少目录信息树的层次,这是因为层次越少,条目的标识名就越短,受其他情况变化的影响就越小,同时管理起来也越方便。考虑到大型机构信息系统的区域化与层级化保障模式,在采用 LDAP 目录服务对大型机构的认证数据实施统一存储和管理的过程中,将按照组织结构来设计目录信息树,即从代表大型机构的树根向下,为总部、区域和地区等各级部门(单位)分别设置相应的组织单元,最终到达存储个人认证数据的叶子节点。然而大型机构的用户数量往往较多,这将使得用于组织认证数据的目录信息树结构非常庞大。由于 LDAP 服务器负责实际执行对认证数据的查询和验证操作,这时候如果仅仅使用单一的 LDAP 服务器将很容易造成负载过大或者网络堵塞,并使得依赖 LDAP 服务器的身份认证服务无法正常工作。因此必须根据大型机构各个部门(单位)之间的树状层级关系,构建跨越多个物理节点的分布式目录服务系统,实现认证数据的分级存储。

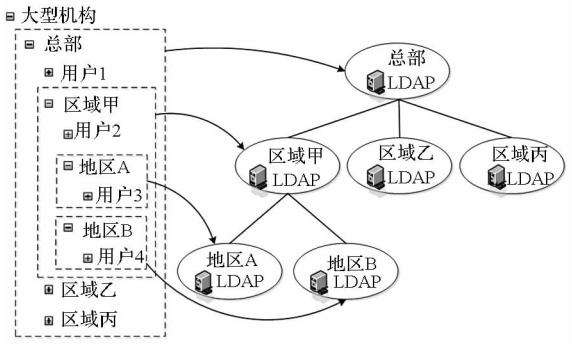

图 2 基于 LDAP 的认证数据分级存储

Fig. 2 Hierarchical authentication data storage based on LDAP

图 2 给出了一种基于 LDAP 的认证数据分级存储架构。用于组织认证数据的整个目录信息树将依托大型机构的数据中心,分布在各个物理节点部署的 LDAP 服务器中。每个 LDAP 服务器都按照本节点所对应的部门(单位),从整个目录信息树中复制相应的分支,并定期与其父节点进行数据同步。在执行身份认证的过程中,无论待验证的用户来自哪个节点所对应的部门(单位),身份认证服务都只需访问本节点部署的 LDAP 服务器即可。如果本节点的 LDAP 服务器没有查询到待验证用户的信息,该 LDAP 服务器将向其父节点的 LDAP 服务器发出请求。如果父节点的 LDAP 服务器仍然查询不到,还将继续向更上一级节点的 LDAP 服务器发出请求,直至最终查询到待验证用户的信息。

3.3 基于网关模型的单点登录方法

网关模型的实现原理是以一个安全认证网关作为“关卡”,将所有对外提供服务的应用系统与外界隔离开来,即应用系统被部署在安全认证网关一侧的受控区域内,而处于另一侧的用户在未通过安全认证网关认证的情况下,其所发出的访问请求无法到达应用系统。安全认证网关是一种专用的、部署在网络边界或者网络域之间的网络安全防御设备,通常以防火墙、加密服务器和认证服务器相结合的方式来搭建,它通过修改标准的 IP 层处理程序,能够在对网络上传输的 IP 数据包进行处理的同时,对其中携带的信息进行加密和过滤等操作[22]。安全认证网关在技术形态上可以分为软件网关和硬件网关两种,最初的安全认证网关主要通过软件的方式来实现,然而当前软件形态的网关正在逐步被硬件形态的产品取代[23]。如果使用安全认证网关,用户还需要在计算机终端上安装与安全认证网关相对应的接入控制客户端。用户在访问应用系统之前,必须通过接入控制客户端登录到安全认证网关,从而与安全认证网关之间建立起一条加密的传输通道,使得用户的访问请求能够透过安全认证网关而到达应用系统。而对于应用系统而言,则可以从安全认证网关获取来访用户的身份认证信息,从而无须用户再次登录。

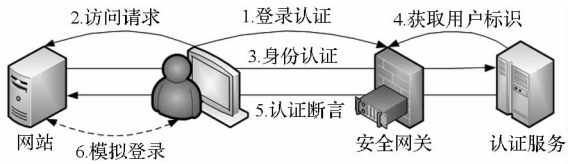

图 3 基于网关模型的单点登录流程

Fig. 3 Single sign-on process based on gateway model

在基于网关模型实现单点登录的过程中,本文提出一种需要由接入控制客户端、安全认证网关、目标服务、身份认证服务和 LDAP 目录服务五个实体相互配合实现的单点登录方法,该方法的实现过程如图 3 所示。其中,接入控制客户端是

安全认证网关在用户终端的代理程序,它需要安装在用户使用的计算机终端上。用户开机后必须在首次访问目标服务之前运行该客户端并输入用户名和口令,接入控制客户端将向安全认证网关提交该用户名和口令并完成登录。安全认证网关通常以防火墙、加密服务器和认证服务器相结合的方式来搭建,它在验证完用户名和口令之后,通过接入控制客户端与用户终端建立起一条加密的传输通道,使得用户终端发出的访问请求能够透过它的防火墙而到达目标服务。目标服务在收到用户终端发出的访问请求之后,将向身份认证服务请求对来访的用户进行认证。认证的过程将与目标服务的具体技术形态有关,如果目标服务是网站,那么它将会把首次来访的用户重定向到身份认证服务,待身份认证服务向它返回认证断言后再为用户进行模拟登录;如果目标服务是 SOAP、REST 等类型的 Web Service,或者是任何通过私有协议访问的服务,它将首先根据用户的链接向安全认证网关获取该用户的 ID,然后再向身份认证服务请求与该用户 ID 相关的属性断言。身份认证服务其实是一个采用 SAML 规范的身份提供者,它接收目标服务以重定向方式发出的身份认证请求,或者以直接调用方式发出的属性查询请求,然后从存储着用户认证数据的 LDAP 目录服务中读取信息,并以 SAML 断言的形式返回给目标服务。

4　系统关键功能设计与实现

4.1　身份提供者设计与实现

在统一身份管理系统中,身份提供者一方面通过安全认证网关获取合法用户的认证信息,另一方面将这些信息以符合 SAML 规范的断言形式传递给提出身份认证请求的身份依赖者,从而支持用户的单点登录。SAML 规范支持“推”和“拉”两种不同的断言传播模式,而本文所实现的身份提供者则采用“推”模式向身份依赖者传递认证断言,主要实现了身份认证和断言生成两个功能。身份认证主要是指对用户的身份进行验证,以便确定用户的合法性。由于身份认证请求通常由身份依赖者以浏览器重定向的方式发送到身份提供者,身份认证服务必须首先对该请求进行解析,然后再利用安全认证网关来验证用户的合法性,因此该功能在具体实现上主要包括请求解析和身份认证两个方法。断言生成主要是指根据对用户身份的认证结果,为用户生成身份断言和属性断言,并以符合 SAML 规范的形式返回响应消息。

身份提供者的类设计如图 4 所示,其中

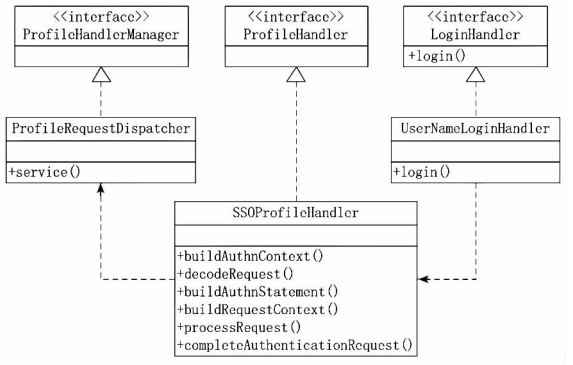

图 4　身份提供者的类设计

Fig. 4　Class diagram of IDP

ProfileRequestDispatcher 类用来接收由服务提供者发出的身份认证请求,并判断该请求是否为 POST 请求。如果是 POST 请求则实例化 SSOProfileHandler 类,并通过调用其中的 processRequest 方法来获取身份认证请求中的 SAMLRequest 对象,对其中的信息进行解码和验证,之后再调用 completeAuthenticationRequest 方法。该方法将实例化 UserNameLoginHandler 类并调用其中的 login 方法。该 login 方法首先调用 LDAP 目录服务的接口来获取用户的认证数据并进行验证,如果验证成功则生成断言信息,否则返回错误。

4.2　身份依赖者设计与实现

身份依赖者的功能是通过在现有的 Web 应用系统中扩展单点登录组件的方式来实现的,本文设计的单点登录组件主要实现了请求拦截和断言解析两个功能。请求拦截主要是指拦截来自用户的访问请求并对其进行分析,该功能可以通过 J2EE Filter 或者 Web Service Handler 等技术来实现,并以读取 Session 或者浏览器 Cookie 的方式判断用户的登录状态。在截获用户请求之后,单点登录组件首先查看用户的 Session 或者浏览器 Cookie 中是否已经存在登录凭证,如果发现当前用户尚未登录,则以 HTTP 重定向的方式向特定的身份提供者提交身份认证请求。断言解析主要是指对从身份提供者返回的认证断言进行校验和分析,并根据通过校验的认证断言内容实施用户的模拟登录。为了保证断言信息在传输过程中的安全性,单点登录组件需要利用身份提供者的公钥对断言信息进行签名验证,这样可以有效地防止认证断言在“推”的过程中被非法篡改。如果签名验证通过,单点登录组件还将把断言中包含的认证信息与本地账户进行匹配,并向用户的 Session 或者浏览器 Cookie 中写入登录凭证,从而

完成用户的模拟登录。

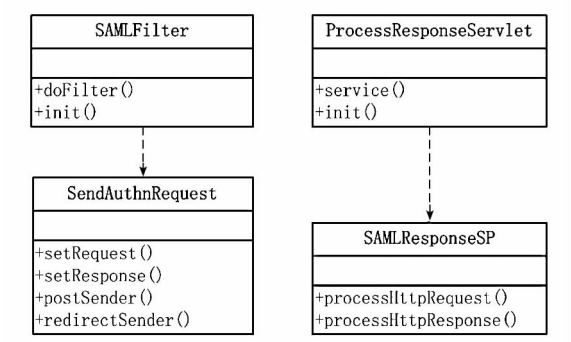

图5 单点登录组件的类设计

Fig. 5 Class diagram of single sign-on component

单点登录组件的类设计如图5所示,其中SAMLFilter类主要负责拦截Http请求并对Session进行检查,然后生成SAML断言请求并将Http请求重定向到身份认证服务。ProcessResponsServlet类主要负责处理从身份认证服务返回的SAML断言响应,如果身份认证成功则从SAML断言中获得当前用户的唯一身份标识,并将该唯一身份标识存放在当前用户的Session中,然后重定向到用户请求的目标页面。如果身份认证失败则重定向到失败页面或错误页面。SendAuthrRequest类主要负责生成SAML断言请求,并将请求发送到配置文件中指定的身份提供者。SAMLProcesserSP类主要负责解析身份提供者返回的SAML断言响应,并从中得到用户的唯一身份标识。

5 结束语

本文详细介绍了大型机构及其信息系统的发展现状,并对统一身份管理系统采用的典型架构和单点登录技术进行了深入分析。针对大型机构在实现统一身份管理系统的过程中面临的特殊需求,本文将互联网上普遍存在的"自由联盟"模式改造为适用于大型机构的"受控联盟"模式。此外,本文还通过目录服务和网关模型分别实现了用户认证数据的分级存储以及单点登录。本文的研究成果能够为大型机构实现跨地域、跨部门、跨业务领域的大规模系统集成提供理论指导和技术支持。

参考文献(References)

[1] 李建, 沈昌祥, 韩臻, 等. 身份管理研究综述[J]. 计算机工程与设计, 2009, 30(6): 1365 - 1370.
LI Jian, SHEN Changxiang, HAN Zhen, et al. Survey of research on identity management[J]. Computer Engineering and Design, 2009, 30(6): 1365 - 1370. (in Chinese)

[2] Steel C, Nagappan R, Lai R. Core security patterns: best practices and strategies for J2EE, web services, and identity management[M]. Englewood Cliff: Prentice Hall, 2005.

[3] Microsoft Corporation. Net passport review guide [R]. Microsoft, 2003.

[4] 刘润达, 王卷乐, 杜佳. OpenID: 一种开放的数字身份标识管理及其认证框架[J]. 计算机应用与软件, 2008, 25(12): 127 - 129.
LIU Runda, WANG Juanle, DU Jia. OpenID: An open digital identification management system and its authentication framework[J]. Computer Applications and Software, 2008, 25(12): 127 - 129. (in Chinese)

[5] 齐忠厚. Kerberos 协议原理及应用[J]. 计算机工程与科学, 2000, 22(5): 11 - 13.
QI Zhonghou. Principle and application of the kerberos protocol[J]. Computer Engineering & Science, 2000, 22(5): 11 - 13. (in Chinese)

[6] Liberty Alliance Project. Liberty architecture overview[EB/OL]. http://www.projectliberty.org/specs, 2003.

[7] 敬思远. 基于 Liberty 的统一身份管理平台设计[D]. 成都: 电子科技大学, 2008.
JING Siyuan. Design of unified identity management platform based on liberty [D]. Chengdu: University of Electronic Science and Technology, 2008. (in Chinese)

[8] Maler E. Assertions and protocols for the oasis security assertion markup language (SAML)[R]. OASIS, 2003.

[9] OASIS Standard. SAML: Security Assertion Markup Language [EB/OL]. http://www.oasis-open.org, 2005.

[10] Jasig. CAS2 architecture[EB/OL]. http://jasig.org/cas/cas2-architecture, 2009.

[11] Ellin B. About openID [EB/OL]. http://www.openidenabled.com/openid/about-openid, 2006.

[12] 刘峰, 王峥, 曹华平, 等. 基于 CAS 的门户单点登录方案[J]. 计算机系统应用, 2011, 20(6): 77 - 80.
LIU Feng, WANG Zheng, Cao Huaping, et al. Portal single sign-on scheme based on CAS [J]. Computer System &Appliance, 2011, 20(6): 77 - 80. (in Chinese)

[13] OpenID Project. What is openID[EB/OL]. http://www.openid.net, 2007.

[14] Bonatti P A, Samarati P. A uniform framework for regulating service access and information release on the web[J]. Journal of Computer Security, 2002, 10(3): 241 - 271.

[15] Groß T. Security analysis of the SAML single sign-on browser/artifact profile[C]//Proceedings of the 19th Annual Computer Security Applications Conference, 2003:298.

[16] Fielding R T. Architecture styles and the design of network-based software architecture [D]. California: University of California, 2000.

[17] Brown A, Johnston S, Kelly K. Using service-oriented architecture and component-based development to build web service application [R]. A Rational Software White Paper from IBM, 2002.

[18] Zhang L J. SOA and web services[C]//Proceedings of IEEE International Conference on Services Computing, 2006.

[19] Howes T A, Smith M C, Good G S. Understanding and deploying LDAP directory services[M]. Boston : Addison-Wesley Professional, 2003.

[20] Carter G. LDAP system administration [M]. O'Reilly Media, 2009.

[21] Pearlman L, Welch V, Foster I, et al. A community authorization service for group collaboration[C]//Proceedings of Third International Workshop on Policies for Distributed Systems and Networks, 2002:50 - 59.

[22] Bell D E, la Padula L J. Secure computer system: Unified exposition and multics interpretation [R]. Mitre Corporation, 1976.

[23] Biba K J. Integrity considerations for secure computer systems [R]. Mitre Corporation, 1977.