西南交通大学

硕士学位论文

统一身份认证系统的研究与实现

姓名：陈小云

申请学位级别：硕士

专业：密码学

指导教师：楼新远

20070601

摘 要

随着基于企业应用系统的普及，越来越多的应用系统被开发出来，但由于各种客观条件的因素，这些应用系统开发时期和使用的技术各不相同，且每个系统都有各自独立的身份验证机制，这就造成了在实际应用中，用户将花费大量的时间来输入验证信息(用户名和密码)，影响了工作效率，而且当企业员工进行调整时候，需要对所有系统的身份信息进行人工调整，这无疑增加了系统管理员的负担，且容易出错，所以研究这一问题的解决方案具有很强的现实意义，单点登录（Single Sign On），简称为 SSO，是目前解决上述问题的流行解决方案，SSO 的简单定义是在多个应用系统中，用户只需要登录一次就可以访问所有相互信任的应用系统。它把实际用户映射成一个电子票据，当用户登录时，获得其电子票据，用户登录其它站点时，不需要重复登录。

本文主要围绕建立目前企业应用系统需要单点登录功能而展开，首先，简要介绍了单点登录需要用到的信息安全方面的理论进行了一个简要的介绍，并对现在流行的单点登录模型 Kerberos、Microsoft Passport 和 SAML 模型的特点和原理作了详细的分析和比较。在这基础上，分析了现在企业应用的实际，设计了较为灵活的基于 SAML 协议和 RBAC 协议原理的单点登录模型，该模型充分考虑了企业的业务实际要求，对 RBAC 协议进行了一定的改进，以保证权限控制的灵活性，最后，根据设计出的模型，结合 ASP.NET 和数据库相关技术给出了一个企业单点登录系统的具体实现，该系统充分考虑了功能和安全方面的要求，对企业的单点登录改造有一定的积极意义。

关键词：SSO（单点登录）；SAML；XML；身份认证；RBAC。

Abstract

As the software applications of the enterprise are more usual, a lot of applications are develop out, these application are work out by different technique and have their own authentication mechanism because of the actual factor. So if the users want use these application, they should log on one by one, they should spend a lot of time in typing the authentication information (username and password). It is not efficiency. If the workers were adjusted, all their information in the application systems must be adjusted too. It is a hard work for the administrator and very easy to cause error, so it has a strong meaning to research the problem and work out a solution, Single Sign on (SSO) is a popular solution for these problem. The definition of the SSO: User can visit these applications which trust each other by log on for once. Actually, the function of SSO is map the user identity to a kind of sign present by data. Once user log on, he can get the electron ticket from the SSO, after that, the user don' t need to log on any more.

The paper' s main task is to build a SSO software function for the present enterprise, First, it introduces the basic information security theory concerning the SSO, then contrast and anlyse the popular modle of implemention SSO(Kerberos、Microsoft Passport、SAML) detailedly . Based on these, anlyase the enterprise' s actual requirement, work out a flexible modle based on SAML and RBAC, and improve on RBAC to fit the flexibility. Finally, according the designed model, use the ASP.NET and Database technique to implemention a SSO system for enterprise , it give a good pattern for implemention SSO.

Key Words: SSO(Single Sign On);SAML; XML; Identity Authentication; RBAC。

第一章 绪论

1.1 背景与意义

随着 Internet 的飞速发展，基于 B/S（浏览器/服务器）结构的企业应用软件也得到了快速发展，各种应用系统如财务系统、人事系统、生产管理系统等已经应用到很多企业的生产管理活动中去，为企业提高工作效率和管理水平做出了巨大的贡献，但由于企业应业务、自身条件、和当时软件技术的影响，这些不同的系统往往是在不同的时期建设起来的，运行在不同的平台上；也许是由不同厂商开发，使用了各种不同的技术和标准，每个应用系统都有自己独立的一套身份验证机制，各应用系统采取分散登录、分散管理。又由于针对企业应用各个系统的联系又十分紧密，各用户需的要使用大多数的系统，如图 1-1 所示：

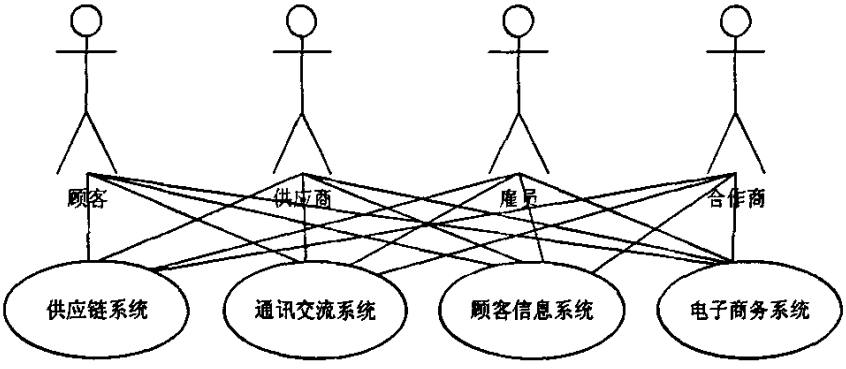


图 1-1 各系统应用情况示意图

如果不使用单点登录（SSO），势必造成了企业的工作人员在实际工作中要频繁的在各个系统登录和注销，严重的影响了生产效率，同时很多系统都要维持一个用户身份信息是很麻烦的，例如：如果企业一个员工离职，则管理员需要把他在各个系统的身份信息一一删除，相当麻烦，有研究表明，用户在其工作时间登录系统占了大约 20 分钟左右的时间，这无疑是人力资源的一种浪费，并且每个系统都要考虑安全方面的问题，这些问题得到了广泛的认识，因此信息系统急需建立一个统一的身份认证系统，以保证用户操作的方便和应用

系统的安全。所谓统一身份认证就是用户基于最初访问的一次身份认证，就能对其被授权的资源进行无缝访问。单点登录（Single Sign On），简称为 SSO，是目前比较流行的企业业务整合的解决方案之一。SSO 的定义是在多个应用系统中，用户只需要登录一次就可以访问所有相互信任的应用系统，如图 1-2 所示：

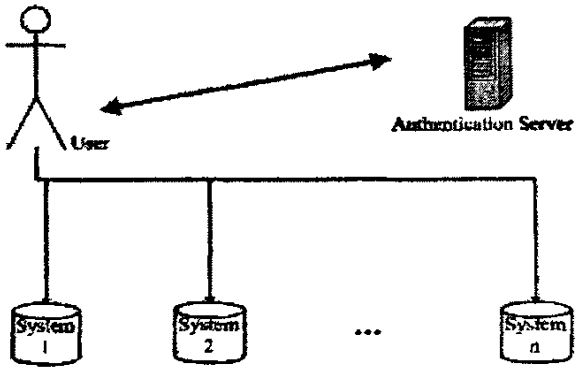


图 1-2 单点登录示意图

建立了单点登录系统，可以大大提高工作效率，用户不需要为记住每个系统的登录方式了，可以大大提高用户的工作效率和更好的用户体验，同时安全人员有更多的精力投入到身份认证服务上，对提高系统的安全性有很大的好处，所以有很强的实用价值。

1.2 国内外现状

国外统一身份认证的研究比较活跃,目前最流行的技术主要有微软的 Passport 技术和自由联盟规范，其中微软的 Passport 技术实际上是一种 Web Service，它是由微软公司控制的中央统筹式的单一登录服务。

自由联盟（Liberty Alliance）是由 SUN、IBM、HP、Intel、Oracle、Novell 等 150 多家公司和机构组成的一个围绕身份认证的技术,该联盟的宗旨是创建一个具有开放性、联合的、单一身份识别的解决方案。与微软的 Passport 集中认证不同，自由联盟采用的是一种称为联邦认证（Federated Identity）的机制，即自由联盟中的身份提供者不唯一，而是可以相互独立存在的，使用联邦认证的好处是在有可以避免因单点故障而导致的系统瘫痪。自由联盟相关协议的基

础和核心是 SAML（Security Assertion Markup Language,安全断言标记语言）。

同时随着统一身份认证研究的深入和较高的商业意义,目前已经出现了很多商用软件和产品[2]，专门的 SSO 商业软件主要有：Netgrity 的 Siteminder（已经被 CA 收购）、Novell 公司的 iChain、RSA 公司的 ClearTrust 等。还有门户产品供应商自己的 SSO 产品如：BEA 的 WLES、IBM 的 Tivoli Access Manager、Sun 公司的 identity Server、Oracle 公司的 OID 等。这些商业软件一般适用于客户对 SSO 的需求很高，并且企业内部采用 COTS 软件 omino、SAP、Sieble 的系统。

采用这些商用软件一般都要对要集成的系统做些改造，如在要集成的系统上安装代理，首先统一这些系统的认证方式（一般采用 LDAP 或数据库），然后才能实现 SSO，相对比较麻烦。

另外，也有很多开源组织从事 SSO 的研究如 JOSSO、OPENSSO、SOURCEID 等。

国内的很多机构和系统正在进行统一身份认证的改造,如电力、银行、政府、高校等信息化比较先近的企业。

1.3 论文研究内容与工作目标

本文主要对现有的成熟模型基础上进行研究,首先对统一认证需要用到的基本理论知识进行了一定的阐述，然后对现在流行的统一身份认证模型进行了详细的分析和比较，最后结合身份认证模型的基础理论和企业应用的实际情况，构造了适合现阶段企业的身份认证模型，最后根据设计，给出了一个详细的实现。

1.4 论文的组织结构

本文的内容是这样安排的：第一章介绍了问题产生的背景，分析了国内外的研究技术现状，提出本文的研究目标；第二章介绍了统一身份认证需要用到的基础理论；第三章分析和比较了现在流行的统一身份认证模型；第四章进行了企业统一身份认证需求分析并设计了具有一定实用性的统一身份认证模型；第五章根据模型设计的要求给出了一个具体的实现；最后是全文的总结。

第二章 身份认证的基础理论

2.1 信息安全的基本概念

网络信息系统安全的内容包括了系统安全和信息安全两个部分。系统安全主要是网络设备的硬件、操作系统和应用软件的安全（这不在本文的讨论范围内）；而信息安全主要是指信息的存储、传输的安全。一般从以下五个方面定义信息系统的安全[33][40]：

保密性（confidentiality）——保密性是指信息不泄露给非授权用户、实体和过程，不被非法利用。

完整性（Integrity）——完整性是指数据未经授权不能进行改变的特性，即信息在存储或传输过程中保持不被非法修改、破坏和丢失，并且能够判别出数据是否已被改变。

可用性（availability）——可用性是指可被授权实体访问并按需求使用的特性，即当需要时授权者总能使用到数据，而不受其他因素影响。

可控性（controllability）——可控性是指可以控制授权范围内的信息流向及行为方式，对信息的传输及内容有控制能力。

不可否认性（non-repudiation）——不可否认性是指行为人要对自己的信息行为负责，不能抵赖自己曾有过的行为，也不能否认曾经接到对方的信息。

2.2 密码学理论

2.2.1 基本概念

密码学一直都是信息安全的最基础的理论，密码学的理论基础来源于基本的数学理论—数论。它通过数学运算把真实的信息隐藏，从而达到保密的目的，加密技术有很悠久的历史，当前，由于网络的高速发展和各个行业基于互联网的应用不断的深入，密码技术更成为了保障信息的完整性和可靠性，防止信息被篡改、假冒和伪造的有力手段，密码技术已经成为保障国家的信息安全的必须研究的技术，所以各国对密码技术的研究都非常重视。

加密包括使用密钥对数据进行编码[3, 5, 6]，从而使偷听者无法方便的阅读这些数据。经过加密的数据称为密文，原始的数据成为明文。从密文到明文的转换过程称为解密，评估一种加密算法安全性的最常用的方法是判断该算法是否是计算安全的。即在有意义的时间范围内，如果利用可用资源进行系统分析后无法攻破系统，那么这种加密算法就是计算安全的。一个密钥系统采用的基本工作方式称为密钥体制，在一个密钥体制中，密钥算法和密钥是保密的关键，按照密钥算法的特点，密钥体制可分为对称密钥体制和非对称密钥体制。

2.2.2 对称加密

对称加密指的是加密和解密算法都使用相同密钥的加密算法。

具体如下：

E(p,k)=C ； D(C,k)=p

其中：

E= 加密算法， D= 解密算法， p= 明文，

k= 加密密钥， C= 密文。

对称密钥加密算法根据对明文消息加密方式的不同分为两大类，即流密码和分组密码。流密码每次处理的是数据的一个位；分组密码每次对一个数据块进行处理。最常用的私钥加密算法——DES（读作 DEZ），该算法被国家标准技术研究所（NIST）1 在 1977 年定为美国标准[NIST77]。其他的私钥加密算法包括 IDEA[Lai92]和 Skipjack（用在 Clipper chip 中）[NIST94a]。

2.2.3 非对称加密

非对称加密包括两个密钥——一个公钥和一个私钥，也称为公钥加密。如果信息使用公钥进行加密，那么通过使用相对应的私钥可以解密这些信息，过程如下：

E(p,ku)=C ； D(C, kr) =p

其中：

E= 加密算法， D= 解密算法， p= 明文，

ku= 公钥， kr= 私钥， C= 密文。

如果信息使用私钥进行加密，那么通过使用其相对应的公钥可以解密这些信息，过程如下：

E(p,kr)=C ； D(C, ku) =p

其中：

E= 加密算法， D= 解密算法， p= 明文（原始数据），

ku= 公钥， kr= 私钥，C= 密文。

公钥加密的想法最早由 Diffie 和 Hellman 于 1976 年[DH76]在密钥交换中提出的,现在最常用的公钥加密方法是 RSA[RSA78]，RSA 由 Rivest、Shamir 和 Adleman 在 1978 年开发。使用 RSA 有三个阶段。阶段 1 包括确定公钥和私钥。阶段 2 包括加密消息。最后，阶段 3 包括解密消息。

2.2.4 数字签名

数字签名是以密码学的方法对数据文件产生的一组代表签名者身份和数据完整性的数据信息[4]。数字签名必须保证以下三点：

（1）能够核实发送者对报文的签名；

（2）发送者不能否认对报文的签名；

（3）接受者不能伪造对报文的签名。

数字签名也称为消息摘要，它使用[NIST93B]中所描述的安全哈希函数。这个哈希函数被称为摘要函数，通常长度为 128 位。确定性的摘要函数应用到整个文档中，并根据消息中的每一位信息产生一个值。有两种方法可以利用共享的私钥来计算摘要。

第一种方法是计算消息的哈希值，然后通过私钥对这个数值进行加密。然后消息和已加密的摘要一起发送。接收者可以再次计算消息摘要，对摘要进行加密，并与接收到的加密摘要进行比较。如果这两个加密摘要相同，那就说明该文档没有被改动。

第二种方法将私钥应用到消息上，然后计算哈希值。这种方法的结果如下：计算 D(M,K) 其中：D 是摘要函数、M 是消息、K 是共享的私钥然后可以发布或分发这个文档。由于第三方并不知道私钥，而计算正确的摘要值恰恰需要它，因此消息摘要能够避免对摘要值自身的伪造。在这两种情况下，只有那些了解秘密密钥的用户才能验证其完整性，所有欺骗性的文档都可以很容易的检验出来。用于数字签名的公钥加密使用 RSA 算法。 在这种方法中，发送者利用私钥通过摘要函数对整个数据文件（代价昂贵），或文件的签名进行加密。私钥匹配的最主要优点就是不存在密钥分发问题。这种方法假定你信任发布公

钥的来源。然后接收者可以利用公钥来解密签名或文件，并验证它的来源和/或内容。只有正确的公钥才能够解密信息或摘要。最后，如果你要将消息发送给拥有已知公钥的用户，那么你就可以使用接收者的公钥来加密消息或摘要，这样只有接收者才能够通过他们自己的私钥来验证其中的内容。

2.3 身份认证

2.3.1 身份认证定义

一个身份是一个指定的参与者，一个参与者是一个惟一确定的实体[4, 6]。就计算机系统而言，身份是实体的一种计算机表达，通常，一个用户就是约束为一个独立实体的身份，特定的系统也有不同的约束条件。系统使用许多不同的方法来表达用户身份。认证就是验证用户身份的过程，证实用户身份与其宣称的身份是否相符，认证技术源于认证系统中不断出现的身份欺诈问题，如非授权的非法问题，身份认证是对用户身份的证实[33]，用以识别合法或者非法的用户，阻止未授权用户访问网络资源。

2.3.2 身份认证的方法

（1）通过一个用户所知道的某些信息对其进行验证，例如用户密码和用户名，这是目前软件系统最常用的方法，虽然不一定是最安全的方法。

（2）通过用户所拥有的某些东西对其进行验证，例如钥匙、身份证等，这是日常生活中常见的验证身份方法，在计算机系统中，主要用于 CA 的身份识别。

（3）通过用户本身所具备的生理特性对其进行验证，如用户的指纹或者视网膜，这是最安全也是最昂贵的方式，而且就目前的技术而言，其本身识别率没有达到 100%，现在更多利用其作为辅助验证的手段。

2.3.3 身份认证的意义

（1）用户身份是访问控制决策的一个参数。通过认证，可以将一个系统的参与者绑定为计算机内部的身份表达。不同的系统有不同的表达方法，但所有的访问决策和资源分配操作都要假定这种绑定是正确的。

（2）在将安全相关事件记入审计日志时，用户身份要被记录下来。实现可追查要依赖这些审计日志，可追查性要求参与者有明确的身份表达。

2.3.4 证书的概念

证书就是一种标签，它将身份与密钥绑定[6]。证书的一般形式是颁发者CA(Certificate Authority)用自己的私钥对主体(对于颁发证书对象)身份的Hash值、公钥以及某些信息(比如颁发时间或有效期等)进行签名。为了验证证书，用户使用颁发者的公钥解密出 hash 值并验证证书中的数据。证书中通常不仅包括证书所有者的名字和它的公开密钥，还可以包括其它很多信息，这样的证书称为扩展证书。用户拥有的一个证书及其所对应的私钥，就可以通过证书来证实自己的身份。

X.509 证书是一个国际标准，现今的数字证书的格式一般采用这个标准，该标准已经成为当今很多网络安全应用，如 PKI、SET、SSL、PGP 等的基础。

X.509 提供了三种使用证书列表的认证过程：单向认证、双向认证和三向认证。

（1）单向认证只保护消息的完整性和原创性。当有人使用用户签名（私钥）签署了时间戳、现时值和目标身份时，就实行单向认证。接收者可以通过使用发送者的公钥来"逆向签署"信息，从而验证信息的真实性，可以从证书列表获得发送者的公钥。由于只对目标进行认证，因此称为单向认证。

（2）双向认证允许发送者或发送者对接收者或目标进行验证。除了进行单向认证以外，目标会发送给发送者一个回复。回复中包括新的时间戳、原始的现时和新的现时。该回复使用发送者的公钥进行签名。在公钥密码学中，只有相对应的私钥或发送者的私钥才能解密这个回复。而且现时必须是原始的现时，否则这个消息就不可信。

（3）当目标和发送者没有同步的时钟或不希望信任时钟时，则采用三向认证。除了进行双向认证以外，发送者将对目标的回复再发送一个回复，包括在原有回复中的新现时，这里只需要验证相匹配的现时值，不再需要验证时间戳。

2.4 XML 签名

XML 签名是一种 XML 传输过程中的数字签名，XML 签名可以应用于任何数字的内容上，而不仅限于 XML。XML 签名支持检测数据的完整性，消息认证和加密者的身份，这些服务适用于任何数据类型，不管是包含签名的 XML

文档还是其他数据[41]。

一个 XML 签名可以应用于一个或多个资源内容。根据签名的数据相对于签名元素的位置，签名可以分为三类：被封装式签名（Enveloped signature，指 signature 元素位于被签名数据块之内）、封装式签名（Enveloping signatures,指 signature 元素包含了被签名数据块）、分离式签名（Detached signatrures，指 Signature 元素和被签名数据相互独立）。

元素 Signature 标志了整个 XML 数字签名，它包含了签名信息元素（SignedInfo）、签名值（SignatureValue）、密钥信息元素（KeyInfo）和客体元素（Object）四个关键子元素。

1. 签名信息元素（SignedInfo）：包含与签名相关的所有信息，如果数据对象信息、数据对象的处理方式以及签名算法等。
2. 签名信息元素（SignatureValue）：Base64 编码过的签名值。
3. 密钥信息元素（KeyInfo）：密钥信息，以使验证方得到验证密钥。
4. 客体元素（Object）：可包含任何元素（如数据对象，时间戳等）。

2.5 SOAP 协议

2.5.1 SOAP 的提出

SOAP 作为一种异质软件对象在网络上沟通的方法，最初是由微软提出的，Microsoft 在 1977 年考虑基于 XML 的分布式计算。目标是使应用程序使用 HTTP 上层的远程过程调用（RPC）相互通信。不过,它并没有和微软的任何技术捆绑在一起,而是作为一个开放的标准提议。但是, 上在最初的 1998 年的方案(由 Microsoft , UserLand 和 Devel2-opMentor Inc. 共同制定) 中,强调了支持 BizTalk 即微软 SOAP 策略的方法。当最初排斥该提议的 IBM 加入以后,SOAP 协议才开始脱离最初的微软“倾向”,变得更加开放。Sun 公司最初也排斥该提议,一直到 2000 年的 6 月才有所改变,低调的表示支持 World Wide Web Consortium (W3C) 在 2000 年 5 月确认的版本。其它一些 B-B 公司,如 Ariba ,CommerceOne Corp 和 Lotus ,也表示支持由 W3C 提出的提议[7, 10] 。

2.5.2 SOAP 构成

SOAP 是一个基于 HTTP 和 XML 的请求/ 响应 RPC 协议(虽然 SOAP1.

1 规范中只定义了 SOAP 与 HTTP 绑定,但 SOAP 亦可与其它协议结合) 。它采用了已经广泛使用的两个协议:HTTP 和 XML。SOAP 把 XML 的使用代码化为请求和响应参数编码模式,并用 HTTP 作传输。采用几行代码和一个 XML 解析器,HTTP 服务器(如 MS 的 IIS 或 Apache) 立刻成为了 SOAP 的 ORBS(SOAP 对象请求代理) 。

SOAP 本身并没有定义任何应用程序语义,如编程模型或特定语义的实现,实际上它通过提供一个有标准组件的包模型和在模块中编码数据的机制,定义了一个简单的表示应用程序语义的机制。这使 SOAP 能够被用于从消息传递到 RPC(远程进程调用) 的各种系统。

2.5.3 SOAP 协议规范

SOAP 的主要设计目标是简单性、可扩展性, SOAP 规范主要由三部分组成[11]:

(1) SOAP 信封(Envelop) 它构造定义了一个整体的 SOAP 消息表示框架,可用于表示消息中的内容是什么,是谁发送的,谁应当接受并处理它,以及这些处理操作是可选的还是必须的等。

(2) SOAP 编码规则(Encoding Rules) 定义了一个数据的编码机制,通过这样一个编码机制来定义应用程序中需要使用的数据类型,并可用于交换由这些应用程序定义的数据类型所衍生的实例。

(3)SOAP RPC 表示(RPC Representation) 定义了一个用于表示远端过程调用和响应的约定,例如,如何使用 HTTP 或 SMTP 协议与 SOAP 绑定,如何传输过程调用,在具体传输协议的哪个部分传输过程响应,如我们可以在 HTTP 的响应的时候传递过程响应。这三部分在功能上是彼此独立的。特别的,信封和编码规则是被定义在不同的 XML 命名空间(Namespace) 中,这样有利于通过模块化获得定义和实现的简明性。

2.6 RBAC 的概念

基于角色的访问控制(RBAC) 的概念在 20 年前第一次提出,它是从传统的随意访问控制(Discretionary Access Control , DAC) 和强制访问控制(Mandatory AccessControl , MAC) 发展起来的。它可以实现企业的权限机制到

企业组织结构的自然映射,提高了访问控制的水平,大大提高了安全管理员的效率[12]。

NIST（The National Institute of Standards and Technology，美国国家标准与技术研究院）标准 RBAC 模型由 4 个部件模型组成，这 4 个部件模型分别是基本模型 RBAC0（Core RBAC）、角色分级模型 RBAC1（Hierarchal RBAC）、角色限制模型 RBAC2（Constraint RBAC）和统一模型 RBAC3（Combines RBAC）[13]。RBAC0 模型如图 2-1 所示。

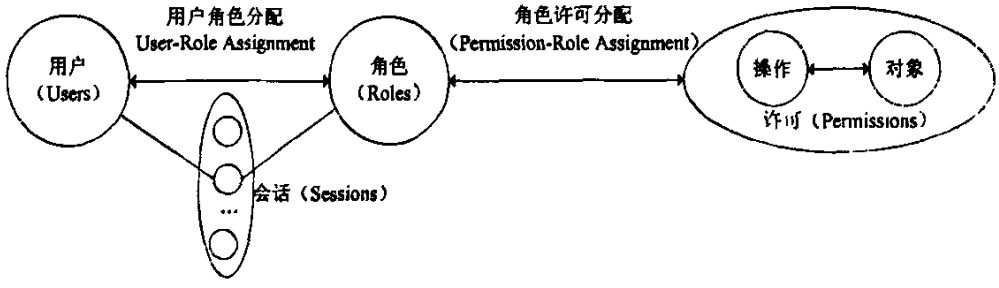


图 2-1 RBAC0 模型

RBAC0 定义了能构成一个 RBAC 控制系统的最小的元素集合。在 RBAC 之中,包含用户（users）、角色（roles）、对象（objects）、操作（operations）、许可权（permissions)五个基本数据元素，权限被赋予角色,而不是用户，当一个角色被指定给一个用户时，此用户就拥有了该角色所包含的权限。会话 sessions 是用户与激活的角色集合之间的映射。RBAC0 与传统访问控制的差别在于增加一层间接性带来了灵活性，RBAC1、RBAC2、RBAC3 都是先后在 RBAC0 基础上的扩展。

2.7 SSL 技术

Secure socket layer(SSL)协议最初由 Netscape 公司发展，现已成为网络用来鉴别网站和网页浏览者身份，以及在浏览器使用者及网页服务器之间进行加密通讯的全球化标准。由于 SSL 技术已建立到所有主要的浏览器和 WEB 服务器程序中，因此，仅需安装数字证书，或服务器证书就可以激活服务器功能了[35]。

SSL 是用来保证安全传输文件的协议，它主要使用公开密钥体制和 X.509 数字证书技术来保护信息传输的机密性和完整性，但它不能保证信息的不可抵赖性，主要适用于点到点的传输，SSL 中有连接和会话这两个重要概念：一个 SSL 连接提供一种合适类型服务的传输，它是点对点的关系，连接时暂时的，每一次连接只和一个会话关联；一个 SSL 会话是在客户机与服务器的一个关联，会话定义了一组可供多个连接共享的加密安全参数，避免为每一个连接提供新的安全参数所需要的代价[36]。

SSL 提供了三种标准服务：

（1）信息加密

SSL 使用公钥体制和对称密钥体制结合到达信息保密的目的，通过公开密钥体制交换通讯双方的会话秘钥，然后通过对称加密传递数据。

（2）信息的完整性

SSL 利用机密共享和 Hash 函数提供信息完整性服务。

（3）双向认证

客户机和服务器相互识别，它们的标志号用公开密钥编码，并在 SSL 握手时交换各自的标识号。

SSL 技术为数据传输的安全性提供了保障，所以现在很多成功的商业网站都采用该技术来实现安全通信。

第三章 几种身份认证方式的分析和比较

3.1 微软 Passport

3.1.1 微软 Passport 简介

微软 Passport 单点登录身份验证技术是微软提供的一个公共网络服务，用于统一的身份认证，它是微软.NET 战略的一部分，ASP.Net 对它的实现提供了有力的支持，用户通过一次登录就可以使用户拥有访问很多网站的访问权，而不需要重复登录。微软宣称微软 Passport 的目的是使会员在使用互联网和在线购物时更方便，它得到了很多在线商业站点的支持，微软旗下的网络软件工具如 HotMail, Msn 和微软网站都加入了该机制。

Passport 主要提供两大服务：单点登录和 Wallet 服务。单点登录即为上文描述的内容，所谓的 Wallet 服务是指用户不仅可以靠单点登录多个站点获得认证，还可以添加用户属性信息如个人信息和信用卡信息等，这主要用于电子商务中。

3.1.2 微软 Passport 工作原理

Passport 协议采用微软的身份认证服务集群作为其中心验证站点，为其加盟站点提供身份认证服务，Passport 采用具有良好的安全性和优异的加/解密的三重 DES 算法对用户认证后得到的 Passport（票据）进行加密。加盟站点不直接和认证服务器发生关系，所有的认证信息将通过 Cookie 的形式保存在客户端，通过 HTTP/HTTPS 协议，把 Cookie 中的 Passport 传到加盟站点，加盟站点根据 Passport 的有效性决定客户端是否有权利访问自身资源。Passport 单点登录模型如图 3-1 所示。

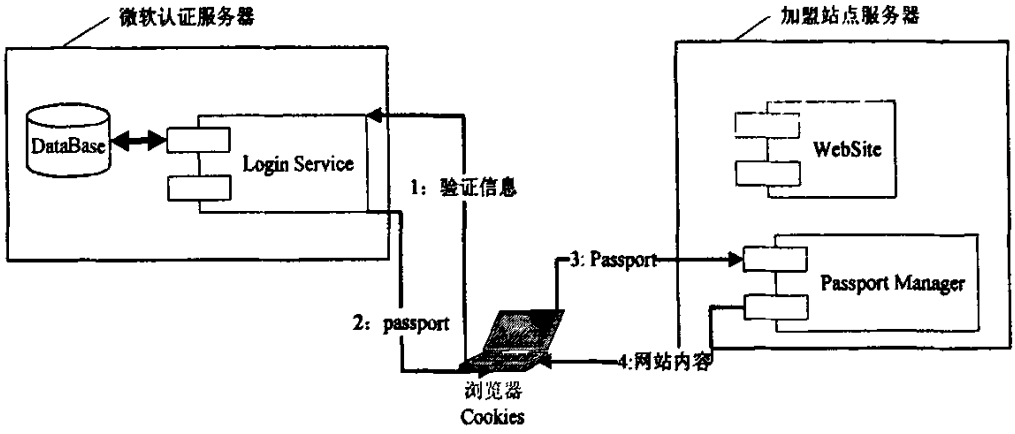


图 3-1 Passport 协议模型

在微软 Passport 单点登录认证模型中，共有三个实体：

1. 用户：在参加 Passport 的站点中使用 Passport 服务的人，用户运行客户端 Client 程序（一般是指 Web 浏览器）。
2. 参加 Passport 的站点：又称参与者，是指有 Passport 许可证的站点，向用户提供 Passport 服务，参与者运行服务方 Server 程序（包括 Passport Manager Object）。
3. Passport Server：Passport 服务的核心，提供单点登录服务和 WALLET 服务，存放用户的注册档案文件、用户名、密码，并在用户允许的前提下将用户私人信息发给参与者。

Passport 工作的具体流程如下：

（1）用户进入登录页面，输入身份认证信息如账号和口令。

（2）微软身份验证服务器验证口令，确定登录成功后，以 cookie 的形式向用户发送登录票据（passport），并将浏览器重定向到用户需要访问的站点。

（3）加盟站点收到登录票据后，确认信息的有效性和用户身份。

（4）加盟站点根据认证的结果，为用户提供相应的内容。

3.2 Kerberos 认证

3.2.1 Kerberos 简介

Kerberos 认证协议是业界的标准网络身份验证协议，被 UNIX 系统广泛的应用，其 V1~V3 是开发版本，V4 是原型 Kerberos，获得广泛的应用，V5

自 1989 开始设计，1994 成为 Internet 的标准（RFC1510）。Kerberos 身份认证系统是 MIT Athena 项目中的一部分，Kerberos 协议是在麻省理工学院起草的，旨在给计算机网络提供"身份验证"。Kerberos 协议的基础是基于信任第三方，如同一个经纪人（broker）集中的进行用户认证和发放电子身份凭证，它提供了在开放型网络中进行身份认证的方法，认证实体可以是用户或用户服务。这种认证不依赖宿主机的操作系统或主机的 IP 地址，不需要保证网络上所有主机的物理安全性，并且假定数据包在传输中可被随机窃取篡改[15]。

Kerberos 协议具有以下的一些优势[15]：与授权机制相结合； 实现了一次性签放的机制，并且签放的票据都有一个有效期； 支持双向的身份认证，即服务器可以通过身份认证确认客户方的身份，而客户如果需要也可以反向认证服务方的身份； 支持分布式网络环境下的认证机制，通过交换"跨域密钥"来实现。

Kerberos 机制的实现要求一个时钟基本同步的环境，这样需要引入时间同步机制，并且该机制也需要考虑安全性，否则攻击者可以通过调节某主机的时间实施重放攻击（Replay Attack）。

Kerberos 发布的第一个报告列出了 Kerberos 认证模型需求[5]。

（1）安全性：网络监听者不可能通过冒充其他用户获取有用的信息。通常，Kerberos 应该具有足够的坚固性，使得潜在攻击者无法找到其中的薄弱环节。

（2）可靠性：对依赖 Kerberos 访问控制的所有服务而言，Kerberos 服务缺乏可用性意味着其所支持的服务缺乏可用性。因此，Kerberos 必须具有高度的可靠性，且应使用分布式服务结构，即一个系统可以支持其他系统。

（3）透明性：理想情况下，用户除了输入口令外，不需要知道认证的发生。

（4）可伸缩性：系统应能支持大量的客户端和服务器，这需要采用模块化、分布式的体系结构。

3.2.2 Kerberos 工作原理

Kerberos 有很多版本的标准，下面通过版本 4 来说明 Kerberos 工作原理。Kerberos4 认证模型如图 3-2 所示：

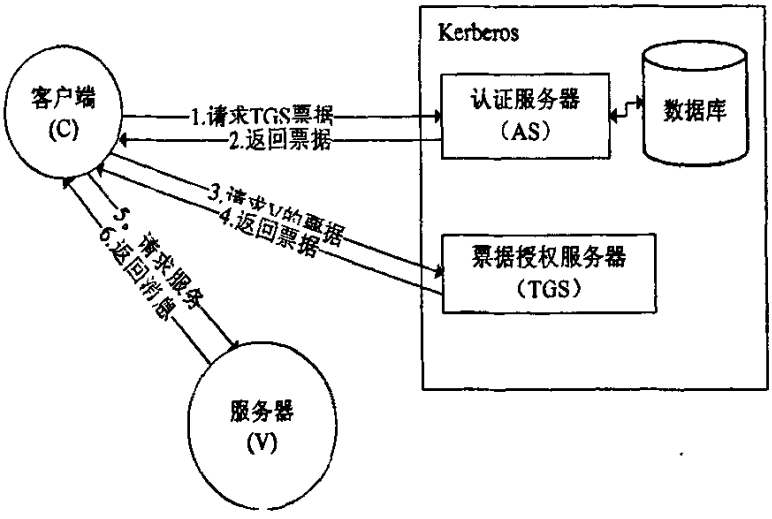


图 3-2 Kerberos 工作原理图

整个会话过程说明如下：

1. $C \rightarrow AS$

消息：$ID_C \| ID_{tgs} \| TS_1$，其中 $\|$ 表示连接。

表示：客户端申请票据授权票据的过程。

其中 ID_C 为用户标志，ID_{tgs} 表示用户要访问的 TGS 标识，TS_1 请求验证的时间戳。

2. $AS \rightarrow C$

消息：$E_{Kc}\left[K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}\right]$

$$Ticket_{tgs} = E_{Ktgs}\left[K_{c,tgs} \| IDc \| ADc \| ID_{tgs} \| TS_2 \| Lifetime_2\right]$$

AS 返回应答消息，其中包括 AS 产生的票据，用户口令派生的密钥 K_C 将其加密，加密的消息中还有一份 C 和 TGS 的会话密钥 $K_{c,tgs}$。

返回消息中包括 $K_{c,tgs}$ 表示客户端和 TGS 在当次会话的会话密钥，ID_{tgs} 为 TGS 标识，TS_2 为发送票据的时间戳，$Lifetime_2$ 表示票据的生命期，$Ticket_{tgs}$ 是客户端用于访问 TGS 的票据，ADc 为客户的网络地址。

3. $C \rightarrow TGS$

消息：$ID_v \| Ticket_{tgs} \| Authenticator_c$

$$Authenticator_c = E_{Kc,tgs}[IDc \| ADc \| TS_3]$$

客户端申请服务授权票据。其中ID_v为服务器标识（客户需要访问的服务器标识），$Authenticator_c$客户端生成的合法票据，其中TS_3为客户端请求票据时间戳。

4. $TGS \rightarrow C$

消息：$E_{Kc,tgs}[K_{c,v} \| ID_v \| TS_4 \| Ticket_v]$

$$Ticket_v = E_{Kv}[K_{c,v} \| IDc \| ADc \| ID_v \| TS_4 \| Lifetime_4]$$

TGS返回消息，其中包括服务授权的票据。其中$K_{c,v}$为客户C和V的会话密钥，ID_v为V的标识，TS_4票据发送的时间戳，$Ticket_v$为客户C访问V的票据，$Lifetime_4$为票据的生命期。

5. $C \rightarrow V$

消息：$Ticket_v \| Authenticator_c$

客户端申请服务。

6. $V \rightarrow C$

消息：$E_{Kc,v}[TS_5 + 1]$ （如果需要双向认证）

如果需要双向认证，服务器给予应答消息，消息为认证消息时间戳的值加1，并用会话密钥加密。C解密后可得到时间戳，由于消息由会话密钥加密，所以可以判定应答消息是来自于V的。

3.2.3 Kerberos跨区域认证原理

属于不同行政机构的客户/服务器网络通常构成了不同域，在一个Kerberos服务器中注册的客户与服务器属于同一个域。由于域内用户可能需要访问其它区域的服务，所以Kerberos提供了域间认证的机制，通过这种域间认证使建

立分布式系统的统一身份认证成为可能，Kerberos 区域认证的原理由图 3-3 所示。

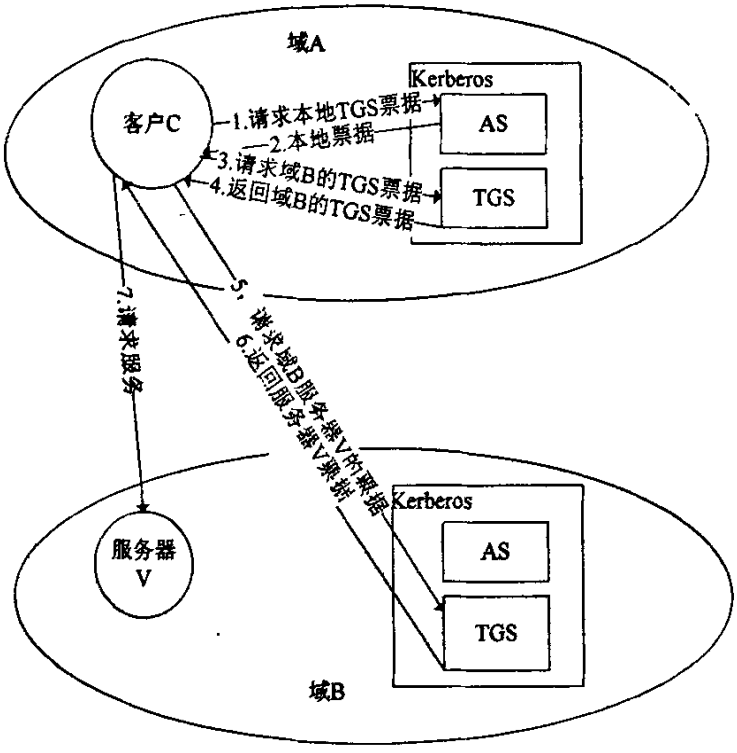


图 3-3 Kerberos 域间认证工作原理图

当用户访问其它域时，用户按照通常的方式请求本地 TGS 票据 ，然后申请远程 TGS 的票据，然后通过远程的 TGS 取得远程的服务。

由于用户的认证是各个区域独立完成的，因此要建立区域间认证，区域之间必须相互信任各区域对用户的认证，因此 Kerberos 协议让每个互相操作域共享一个密钥，双方的 Kerberos 服务器相互注册，从而达到了跨区域认证的目的。

3.3 SAML 协议

3.3.1 SAML 简介

SAML(Security Assertion Markup Language) 标准定义了一个在线商业伙伴之间交换安全信息的框架，更准确的说，SAML 定义了一个基于 XML 标准

的框架用于实体之间交换安全信息，SAML 1.1 是由 OASIS (the Organization for the Advancement of Structured Information Standards)标准组织的 SSTC（Security Services Technical Committee，安全服务技术委员会）制定的[20]。

SAML 可以实现不同安全服务系统之间的互操作，SSTC 通过了很多用例（需求）来推动 SAML 的需求。SAML1.x 中最重要解决的问题就是单点登录的问题，用户使用用户名和口令登录源站点。然后，用户希望无需再次验证即可访问目标站点。图 3-4 显示了认证点和目标站点之间能使用户通过单点登录访问双方站点的交互。

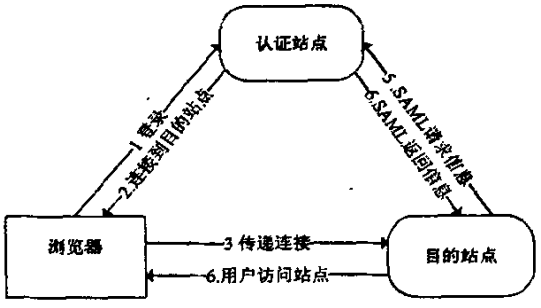


图 3-4 SAML 使用场景图

如图 3-4 所示,用户通过在认证站点进行认证登录，认证站点和目的站点通过交换用户认证信息，从而到达认证的目的,在这里认证站点即为一个认证区域的中心认证服务站,一个认证区域一般只有一个这样的认证站点,全面担任用户身份认证的责任,而目的站点,就是需要认证站点提供服务的应用系统,一个区域一般都有多个这样的目的站点,由于把认证的工作都交给了认证中心,所以目的站点一般不会考虑身份验证的问题。

SAML V1.0 在 2002 年 9 月成为 OASIS 的标准。2003 年 9 月 SAML V1.1 成为标准,后面陆续产生了自由联盟 ID-FF 1.1 和 1.2 以及 Shibboleth。OASIS、自由联盟和 Shibboleth 最初从三个角度实现联邦身份：OASIS SAML 主要关注企业对企业互动（企业之间的单一登录），自由联盟关注需要保密的消费者（企业对消费者）互动，而 Shibboleth 则将重点放在要求匿名性的教育环境。因此，他们修改和扩展了最初的 SAML 1.0 规范来支持不同的用户。这些联邦协议不能互操作，或者说是不向后兼容[21, 22]。不兼容给消费者的应用带来了麻烦，很多机构必须通过协议映射和转换技术来支持多个协议，而这些技术造成关键特性或功能的支持空隙,延缓了发展速度,增加了联邦身份部署的费用。

所以为了解决这个问题，结构信息标准推进组织（OASIS）、自由联盟和Shibboleth 于 2005 年联手开发单一标准，替换掉他们以前的工作成果。合作的结果就是 SAML 2.0。

3.3.2 SAML 工作原理

由于笔者项目是基于 SAML1.1 标准构造的，所以下面介绍的主要是 SAML1.1 协议的内容，SAML 规范体系主要由三个部分构成：即断言（Assertion）、请求/响应协议（Request and Response Protocol）、绑定和配置（Bindings/ Profiles），各个部分紧密的联合在一起，构成了整个 SAML 协议的实现，如图 3-5 所示。

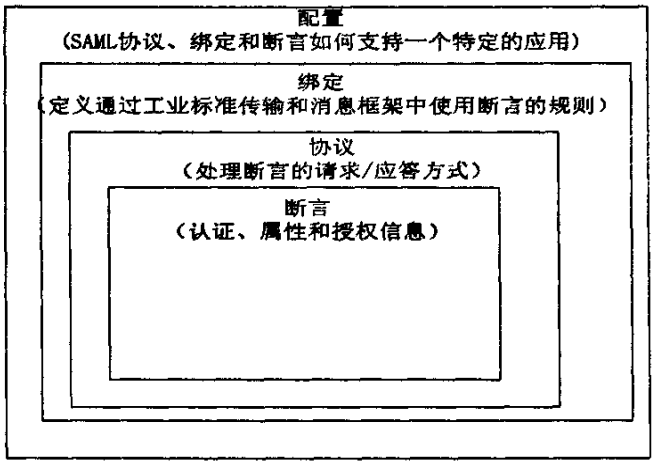


图 3-8 SAML 各部分关系图

下面对各部分的构成作进一步的详细描述如下：

（1）断言(Assertions)

断言(Assertions)是 SAML 的基本数据对象，由一个或多个语句构成的信息包构成，是对主体（用户、计算机）的安全信息（身份、权限等）的 XML 描述形式，SAML 断言包括三种形式：认证断言（Authentication Assertion）、属性断言（Attribute Assertion）和授权决议断言（Authorization Assertion），其中认证断言描述与认证成功事件相关的信息（如认证机构、方式和有效期等）；授权决议断言描述许可权查询和检查结果，此结果可以是接收或拒绝主体对资源的访问请求，属性断言描述与主体的认证和授权相关的信息，如主体的标志、所属的用户、角色、可访问的资源及权限等。

所有的声明都由下列四类要素组成[28]：

1. 基本信息（Basic Information）:与声明构造相关的信息，SAML规范的版本号（Major Version、Minor Version），断言唯一标识（AssertionID），断言发行者的唯一标识（Issuer），断言的发行时间（IssueInstant）及断言的有效期等。
2. 主张(Claims)：一个或多个由断言发行者生成的陈述（Statement）。陈述有如下几种：主题陈述（SubjectStatement），认证陈述（AuthenticationStatement），属性陈述(AttributeStatement)，授权决议陈述（AuthorizationDecisionStatement）。一个声明标记中可以包含多个陈述。
3. 条件（Conditions）：断言的状态可能受某些条件的限制,如声明的有效性可能依赖于来自某种确认服务的信息。
4. 建议（Advice）：断言中可以包含一些额外的，由发行者提供的用于认证或授权的辅助信息。建议可为另一断言的标识，或是一个完整的声明。

（2）请求/响应协议（Request/Response Protocols）

SAML的请求/响应协议，它规定了两点间共享SAML数据所需交换的消息种类和格式。它通过<Request>和<Response>两个元素来实现，SAML请求者发送<Request>元素到SAML响应者，响应者产生<Response>元素，如图所3-6所示：

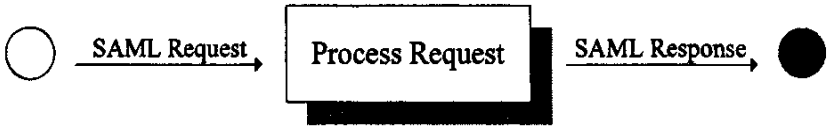


图3-6 请求/响应协议示意图

<Request>元素中主要包含下列元素：<Query>(用于定义新类型的查询)、<SubjectQuery>（用于定义对SAML主体的查询）、<AuthenticationQuery>（用于对认证的查询）、<AttributeQuery>（用于对属性的查询）、<AuthorizationDecisionQuery>（用于对授权的信息进行查询）、<AssertionIDReference>（通过AssertionID的值查询一个断言）、<AssertionArtifact>（通过一个断言辅件查询一个断言）。

（3）绑定（Bindings）

绑定（Bindings）详细的描述 SAML 的协议到底层通信协议的映射，如 HTTP 上的 SOAP 消息交换之类的传输协议，在 SAML1.1 中，目前采用的是基于 HTTP 的 SOAP 绑定，SAML 请求和返回通过 HTTP 作为从目标站点到源站点的 SOAP 消息发送，SAML 信息就是一个 SOAP 信息，存放在 SOAP 消息体中，当 SOAP 消息用于从 SAML 机构来回地发送断言的请求和响应时，SOAP 消息体只能有 SAML 消息。

（4）配置（Profiles）

配置（Profiles）描述了控制在底层通信协议中嵌入和提取和集成 SAML 信息的一组规则。SAML 定义了两个支持单点登录（SSO）的基于 WEB 浏览器方式。Browser/Artifact 方式和浏览器/POST 方式。

下面对这两种方式简要的进行说明：

SAML 应用的实现由三个部分组成：①主体(Principals)，即用户。②服务提供者(Service Providers，SP)，即各种应用系统。③身份提供者(Identity Providers，IDP)，即身份认证服务器[30]。

1．Browser/Artifact 方式

又被称为 Identity Provider 推方式，登录时序流程如图 3-7 所示：

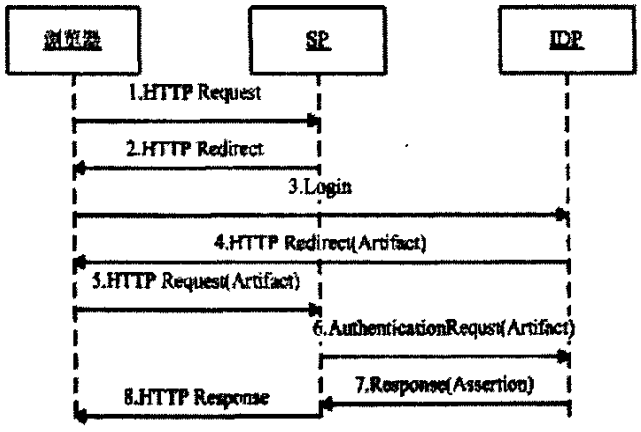


图 3-7 Browser/Artifact 方式示意图

①．用户访问 SP；

②．SP 检查自身的用户 Session，如果 Session 存在，表明用户已登录，直接跳转到 8，否则重定向到 IDP 的登录界面；

③. IDP 提示用户登录；

④. 如果验证成功，IDP 即生成用户的身份认证断言和 Artifact，并建立断言与 Artifact 的对应关系，然后将此 Artifact 作为参数向用户发送 HTTP 重定向指令；

⑤. 用户被重定向到 SP；

⑥. SP 根据此 Artifact 向 IDP 发送 Authentication Request 请求；

⑦. IDP 查询 Artifact 与断言的对应表后，将签名发送给 SP；

⑧. SP 收到断言,如果断言表示验证成功，则生成用户登录 Session，并将系统界面返回给用户，用户登录成功。

2. Browser/POST 方式

又被称为 Identity Provider 推方式，登录流程如图 3-8 所示：

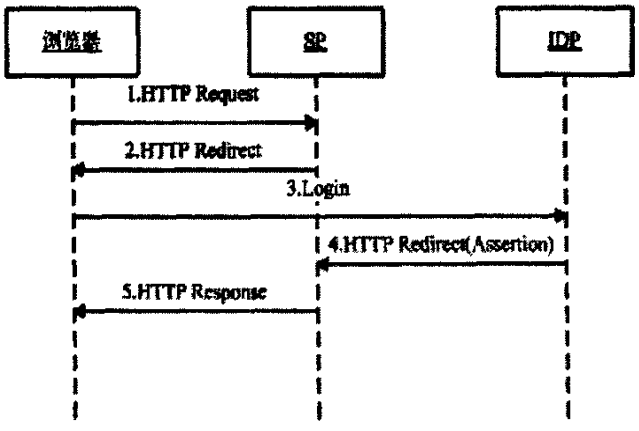


图 3-8 Browser/ POST 方式示意图

①. 用户访问 SP；

②. SP 检查自身的用户 Session，如果 Session 存在，表明用户已登录，直接跳转到 5，否则重定向到 IDP 的登录界面；

③. IDP 提示用户登录；

④. 如果验证成功，IDP 生成一个 HTML 的 FORM，其中包含了断言信息，然后重定向到浏览器。

⑤. SP 收到断言,如果断言表示验证成功，则生成用户登录 Session，并将系统界面返回给用户，用户登录成功。

这两种方式各有优点，Browser/POST 方式可减少 SP 与 IDP 的交互，节省

带宽，而 Browser/Artifact 方式安全性更高：SP 实时查询 IDP，可确保断言的时效性，另外 SP 查询过 Artifact 后，IDP 即删除 Artifact 与断言的对应关系，可防止重放攻击。

3.3.4 SAML 跨域认证原理

SAML 可以实现跨区域认证，用户的身份信息被放在不同的区域各自的身份认证服务器上，它在原有各个区域的认证模型的基础上，通过建立各区域间的认证服务器的信任关系，从而使跨域认证能在本认证域进行，它和 Kerberos 跨区域认证很类似，但它相对于 Kerberos 要简单一些，其原理图如图 3-9 所示。

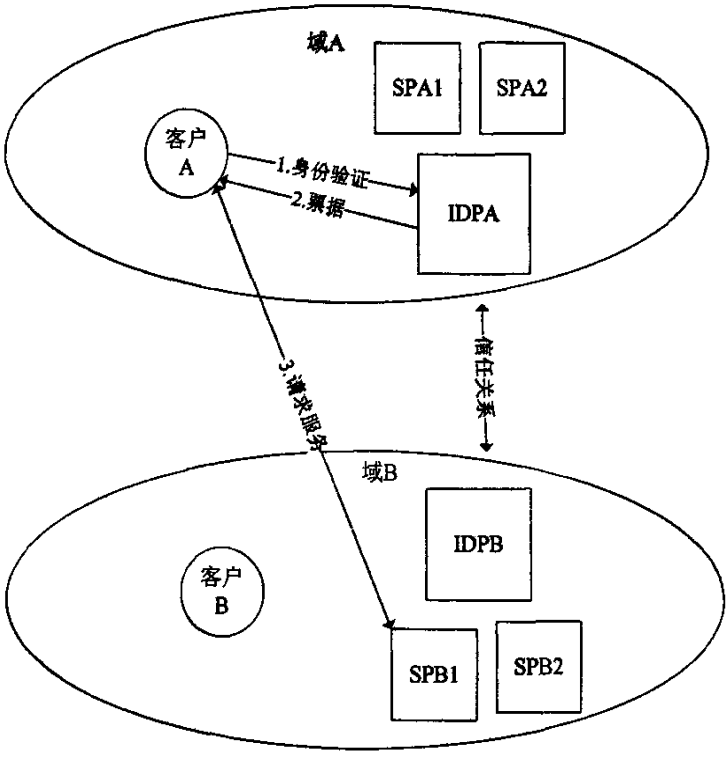


图 3-9 SAML 统一身份认证模型

如果用户 A 想访问 B 中的服务 SPB1，由于域 A 和域 B 建立了信任关系，所以用户 A 只需要在本区域认证服务器 IDPA 中认证，获得访问票据，则可以对域 B 中的服务进行访问，在 SAML1.1 中，没有给出建立信任关系的具体方式，在 SAML2.0 中给出了相应的详细定义，在这里就不详细描述了。

3.4 身份认证模型的安全分析和比较

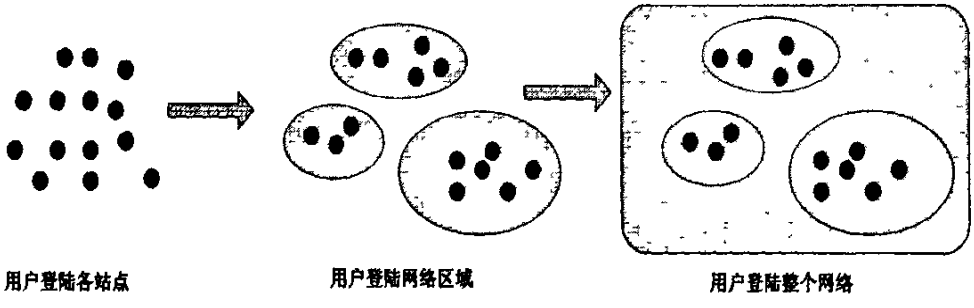
3.4.1 网络身份认证趋势

在对各身份认证模型的分析之前，现对互联网的身份认证发展趋势做一个简要的说明，如图3－10所示：

在最初的网站应用系统中，各个应用系统相互没有关联，属于一种离散的状态，用户单独的登录网站，所以对于用户必需记住所有网站其对应的用户名和密码，如何减轻用户记忆用户名密码和重复登录的负担成为这阶段主要需要解决的问题。

后面发展成为单个用户登录某个按一定逻辑和规则划分的一个网络区域，这个时期产生的技术以微软的Passport为代表，只要用户登陆其网域的任何一个站点，用户就拥有访问其它站点的权限，但是当用户要访问其它网络区域的站点，仍然需要重新登录，如何建立一种机制可以使不同网域的身份认证机制能够合作，让用户的身份验证更加简单，成为了这阶段的主要问题。

目前如何建立一种标准，该标准能够把不同网域的身份认证联合起来，让用户一次登录就能访问整个网络，SAML协议的提出为解决这一问题提供了可能，该协议意图建立一种标准，可以让不同网域的认证系统建立身份验证合作，目前该协议越来越受到很多软硬件厂商的支持，使实现理想认证方式（一次登录访问整个网络）成为了可能。



3-10 网络身份认证趋势示意图

3.4.2 身份认证模型安全性分析

3.4.2.1 微软Passport安全性

微软Passport协议使用了SSL协议，HTTP重定向、Cookie技术和3DES

加密等技术来实现单点登录。现在安全性进行简单的分析[14,15]。

（1）密钥的安全性

Passport Server 和参加 Passport 的站点唯一地共享一把 3-DES 密钥，这些密钥必须被安全生成并安全分配。然而，产生钥匙的安全度很大程度上依赖于钥匙产生的随机性，要做到好的随机性并不容易，现在许多系统被攻破就是因为随机性太差。分配密钥的最理想途径是通过物理信件或电话，但这在参加 Passport 的站点急剧增多时并不现实。在 Passport Server 中，只用了一把密钥来加密所有的 Passport Cookie，一旦该密钥暴露，则所有的 Cookie 将处于危险之中。

（2）Cookie 的安全性

由单点登录的工作原理可知，Cookie 是用户取得 Passport 服务器的通行证，谁获得 Cookie 谁就将获得 Passport 服务，若没有了 Cookie 就将失去 Passport 服务。如果 Cookie 不幸被恶意破坏者得到，则这个破坏者就可以在用户原有登录的基础上存取用户的信息，冒充用户而获得服务。如果 Cookie 被一恶意破坏者删除，则 Passport 服务会莫名其妙地中断，可能给用户造成损失。

（3）重定向的安全性

SSL 协议只在 Passport Server 向用户提交表单和用户完成表单返回给 Passport Server 时起作用，在各次重定向时，SSL 并不参与。这样恶意站点就可以修改用户通信中的重定向（为伪装的可信起见，可事先申请一个真实的证书，但当然不是合法 Passport Server 的证书），利用用户的疏忽（如：用户很少检查 Passport Server 的证书和 Passport Server 的 URL），将其指向恶意站点。用户以为自己是在和合法的 Passport Server 交互，实际上是在和恶意站点交互，从而使用户陷入危险当中。

（4）Passport Server 的安全性

Passport Server 存放着所有用户的注册档案文件、用户名、用户密码等信息，如果受到 DOS 攻击，则 Passport Server 里面的所有信息将变得不可用，用户得不到单点登录的认证，导致那些安装了 Passport 服务且丢弃传统登录服务的站点将变得不可操作，即无法对用户完成认证。

3.4.2.2 Kerberos 的安全性

（1） 口令猜测攻击。用户联机时通过用户口令获得与认证服务器共享

的相同的私钥。尽管使用 TGT 减少了联机次数，避免口令的经常使用，但毕竟有口令的键入，这是极不安全的。口令很容易被窃听和截取，入侵者可以记录下联机会话，通过计算密钥分析记录下的会话进行口令猜测攻击，攻击者即获得客户端口令，进而使用该口令获得 Kerberos 票据。

（2）Kerberos 认证协议中依靠认证者信息中的时间标记来抗重放攻击。它假定在一个区域内的所有用户的时间同步,受到的时间标记在规定的时间范围内（要做到真正的时间同步很困难），就认为不是重发的。实际上,攻击者可以先把伪造的消息准备好,一旦得到认证者信息马上发出去。

（3） 密钥的存储问题。Kerberos 认证中心拥有该域上所有用户实体的密钥，如果被攻破，那么整个域的安全性也将遭到损害，后果将是灾难性的。

（4） 系统程序的安全性、完整性问题。实际上，最严重的攻击是恶意软件攻击。Kerberos 认证协议依赖于 Kerberos 软件的绝对可信，而攻击者可以用执行 Kerberos 协议和记录用户口令的软件来代替所有用户的 Kerberos 软件，达到攻击目的。如果装有 Kerberos 的平台的安全遭到损害，Kerberos 的安全性也将部分或全部丧失。

3.4.2.3 SAML 协议的安全性

SAML 单点登录系统在消息传递过程中主要有以下风险[31]:

（1）拒绝服务攻击

由于认证服务器处理一个 SAML 请求时， 首先需要从请求信息的 XML 文档中提取相关信息， 然后从数据库或文件中查询相应请求数据的断言信息用于构造回复消息。这一般需要占用一定的系统资源，而构造请求消息却简单得多。攻击者可重复发送请求消息，从而造成服务器拥塞、崩溃，使正常的请求服务无法进行。解决方案有以下几种：

1. 会话层用户认证。发送端在建立传输层连接后，必须提供能惟一标识用户身份的 Certificate 进行认证。接收端记录认证结果及 IP 地址等信息，将用户标识与地址信息绑定。如果受到拒绝服务攻击，则可以通过用户标识和地址信息绑定及记录的认证结果等信息来简单地判断。未通过认证的用户发送的数据包可以丢弃。此外，也可以通过这些信息进行追踪，找出攻击者。

2. 对请求消息签名。由于计算数字签名信息较难，而验证签名相对简单，因此可以要求对发送的 SAML 请求消息进行签名。这样发送请求消息的难度

增加，从而减少发送与接收消息时发送端与接收端工作量的不对称性，减少受到拒绝服务攻击的可能。同时可在每个SAML(请求消息中加时间戳标志，以阻止重放的拒绝服务攻击。

3. 限制可以发送请求信息的用户。将可以发送请求消息的用户限制在特定范围， 也可以在一定程度上减少服务拒绝攻击。

（2）防止窃听

为防止SAML(消息在传输中被他人窃听，所有SAML(请求/ 回复消息在应用层都应有一定的加密措施。SAML 中推荐使用的加密技术是,XML Encryption., XML Encryption 可以对XML文档中的部分或全部数据内容进行加密。对于同一个文档中的不同部分还可以用不同的密钥进行加密， 然后把同一个 XML(文件发给不同的接收者，而接收者只能看见和他相关的文件内容。如果需要更高的保密性，则可以在会话层加密。SSL 或 TSL 均可以对会话层数据加密。

（3）抗重放攻击

由于SAML(的请求+ 回复消息都带有时间戳和惟一的ID，因此可以防止重放攻击。

（4）防止消息的篡改

SAML (的请求/回复消息中使用了数字签名， 因此可以防止消息的篡改，保证应用层重要数据的完整性。但要获得更高的安全性，必须对全部会话内容进行签名。SSL、TLS、IPSec 均可满足此要求。

（5）抗中间人攻击

中间人攻击是指攻击者截获通信双方的报文并用自己的报文替代原始报文。使用 SSL、TLS、IPSec 对传输数据进行签名和加密，可以防止中间人读取会话内容，从而阻止传输数据的中间人攻击， 同时也可以保证会话密钥的安全性。

3.4.3 身份认证模型的比较

3.4.3.1 三种模型总体比较

首先对三种模型进行一个总体的比较，模型比较如表3-1所示：

表 3-1 三种模型的总体比较表

项目	Kerberos	Passport	SAML
统一认证方式	分布式	集中	分布式
安全性	高	低	中
跨平台	是	否	是
操作系统限制	无	有	无
实现复杂性	高	低	中
网站适用性	不适用	适用	适用
跨网域	是	否	是

认证方式：Kerberos 和 SAML 都采用了分布式联合的方式来实现统一身份认证，而微软 Passport 则采用了集中认证的方式实现。

安全性：Kerberos 的安全机制更复杂，所以更加具有安全性，SAML 协议采用了证书对断言加密，也有很好的安全性，而 Passport 的安全机制更简单一些，所以相对安全性更低一些。

跨平台：由于 SAML 协议采用了受到厂商广泛支持 XML 数据，所以它有更好的开发性和跨越平台性；Kerberos 只需要不同的网域拥有由于相同的认证机制，所以可以跨平台实现；微软 passport 必须构建在微软的.NET 技术之上，因此不具有跨平台性。

实现复杂性：Kerberos 涉及了大量的票据操作和加/解密操作和其它一些安全机制，所以实现较复杂；而 SAML 只涉及 XML 格式的身份数据传递所以相对简单一些，微软的 passport 可以通过.NET 技术很方便的实现。

网站适用性：Kerberos 需要客户端用于强大的计算能力，而以浏览器为客户端的网站并不适用，因为浏览器的计算资源相对有限。

跨网域：SAML 和 Kerberos 都有机制支持跨网域实现，而微软的 passport 只适用于其控制的网域。

3.4.3.2 Kerberos 与 SAML 比较

SAML 协议和 Kerberos 协议都采用的是跨域的统一认证方式实现统一身份认证，它们都可以定义多个认证域，各认证域是按一定规则划分的独立的身份认证区域，各个区域通过信息交换来实现统一身份认证。，但它们也有一定的区别：

Kerberos 使用认证服务器和票据服务器，实现客户端和服务器端的双向认证，保证了消息传递的安全性。相对于 SAML 协议，它的实现显得过于复杂，特别在进行跨域认证，客户和 Kerberos 服务器的交互过多，这增加了网络的负担，而 SAML 通过传递 Artifact 的方式在安全性较低的浏览器 Web 服务器之间传递用户认证信息(SAML 声明) 的引用，它对网络的压力要小的多，而且由于客户端需要频繁的加密/解密操作，Kerberos 要求客户端有较强的计算能力，然而对计算资源有限的浏览器，Kerberos 不会是一种很适宜的方式，从这方面看，SAML 更适合 B-S 结构的系统。

Kerberos 和 SAML 协议都可以实现跨区域认证，然而它们的实现机制并不一样，Kerberos 要求需要不同的区域拥有一对密钥对来建立信任关系，而 SAML 协议是通过跨域查询身份信息实现，比较起来，SAML 协议更有开发性，而 Kerberos 的方式具有更强的耦合性。

3.4.3.3 微软 Passport 和 SAML 比较

微软 Passport 和 SAML 的最大不同，就在于微软采用中央集权式的控制方式，而 SAML 采用联合的方式，各个认证区域的用户可以拥有和管理自己的用户身份信息，各区域之间通过加盟的协议联系在一起，若一个区域内的用户想访问其它区域的系统，该系统可以向本地的认证服务器发出身份查询请求，由认证服务器传回认证结果。

使用集权式的方式可以很简单的把各个应用系统通过中心认证服务器紧密联系起来，所有的认证工作交由中心认证服务器完成，从而可以很简单实现统一认证的功能要求，但 Microsoft Passpor 机制存在一个明的缺陷,即单点失效(SinglePointFailure)：当中心认证服务器瘫痪时，所有应用系统的认证将陷于瘫痪。虽然微软承诺用户 Passport.com 的信息的可靠性和安全会得到充分的保障,但于 Passport.com 在整个模型中的核心地位,很有可能成为黑客攻击的目标。

微软 Passport 需要一个 Email 地址和密码创建 Passport 帐号，它保存了其它 14 个字段的信息，包括名字、Email 地址、地址、国家、邮政编码、语言、时区、性别、生日、职业等。为了保护上述私人信息，微软采用了 SSL 和 3DES 加密技术。相对于 SAML 协议微软 Passport 更适合 B-C（企业一客户）的商业模式，它可以方便快捷的为个人用户提供身份认证服务，但对需要

管理自己员工身份信息的企业它不太适合，因为它的权力过于集中，而SAML的设计更多的考虑到了企业用户的需求，更适合B-B(企业－企业)商业模式，但并不表示它不适应B-C模式，实际上，用户只要在交起始点进行过一次登录，在后续交易过程中，用户的认证和交易有关的权限等信息就可以在具有信任关系的企业间和业伙伴间进行分发，从而实现用户的跨站点访问。

相对于微软Passport而言，SAML更有优势，SAML的最大优点就是因为它是一个开放式的标准，与平台、消息机制和传输协议无关的解决方案。与安全系统的体系结构和实现相独立，企业可以在不改变其原有安全解决方案的基础上，通过实现基于SAML的接口，就可以安全地交关于用户、交易等的安全信息，各个系统可以通过标准的SOAP协议进行交互而不需要考虑各个系统的具体实现技术，因为基于XML结构的SOAP协议得到了很多软件厂商的支持，任何安全服务引擎都可以实现SAML，从而促进异构安全系统间的互操作。而微软Passport必须使用Microsoft平台提供服务，对其它系统目前还无法支持，如Linux、UNIX等。

从上面的分析来看，用SAML协议实现统一身份认证更加符合企业应用系统的需要。

第四章 统一身份认证系统功能分析和结构设计

4.1 系统需求

本系统的创建背景假定为对一个企业进行统一身份认证改造，该企业的应用系统均采用 B-S（浏览器-服务器）结构构建，系统进行统一身份认证改造后，需要能够实现单点登录功能，并且需要有灵活的访问控制，保证和原来系统的紧密结合，实现用户的统一身份认证。

4.2 设计的总体目标

4.2.1 功能目标

（1） 统一用户管理

以用户（员工、客户）ID 为身份主体进行管理，替代其在原来各个应用系统中的各自主体（账户）的管理，使各个系统通过这个唯一的主体进行管理，多个应用系统的用户管理只有一个管理入口；从而减少了当用户发生变化的情况下，减轻系统管理员对用户维护的工作量，这里实现该功能的主要目的就是要把用户的身份信息统一化，使用户的身份表达口径一致，避免原来多个系统对用户身份的理解二义性，统一身份认证首先要考虑到“统一身份”。

（2） 统一认证

原来的各系统都有一套自己的认证机制，这样有很大的弊端，如果一个系统被攻击，由于很多系统的用户名和密码都一样，所以可能造成其它系统被攻破，如果只有一个权威的认证中心来进行应用系统访问主体身份的鉴别，企业可以集中精力保证认证中心的安全性，可以降低企业系统的运营成本，减少员工用于系统登录所花费的时间，提高企业的生产效率。

（3） 统一访问控制

在统一用户管理、统一认证的基础上，对用户访问的资源进行有效控制。由于现在的业务系统都很复杂，使用的用户可能包含所有人员，且每个人员所负责的工作和工作中的权利范围都是有限的，所以他们访问系统应该受到一定

的限制，并且有些企业机密信息只有企业的高层才能够使用，所以必须根据其自身的职权分配其访问系统的权限，系统需要设计一种统一的访问控制方式来解决该问题。

4.2.2 性能目标

（1）安全性

身份验证机制设计合理，充分考虑系统的安全性，能够防止目前常见网络攻击（监听、重发、篡改断言等）的机制，防止攻击者的非法访问，使系统安全得到保障。

（2） 方便性

系统需要减少用户多次的登录认证，减少用户使用系统的负担，减轻系统管理员验证基础维护的工作量，企业在进行统一身份认证改造后，用户使用系统应该更加方便。

（3）灵活性

系统可以灵活满足企业用户的实际需要，从用户的需求出发，使用户的身份和访问控制信息都能灵活的进行配置，并且能够适应将来的扩展需要，满足企业实际的业务需要。

4.3 总体构架

该系统采用 B-S 结构，整个统一身份认证体系由一个认证服务器和应用系统服务器（该系统主要用于实验目的，因此这里的应用服务器只有一个，在实际应用中应用系统应该有多个）构成，其中认证服务器负责对用户进行身份验证，并把认证结果返回给应用服务器，应用服务器根据认证的结果，结合其自身的 RBAC 访问控制系统，对用户访问系统进行授权，其体系结构如图 4-3 所示：

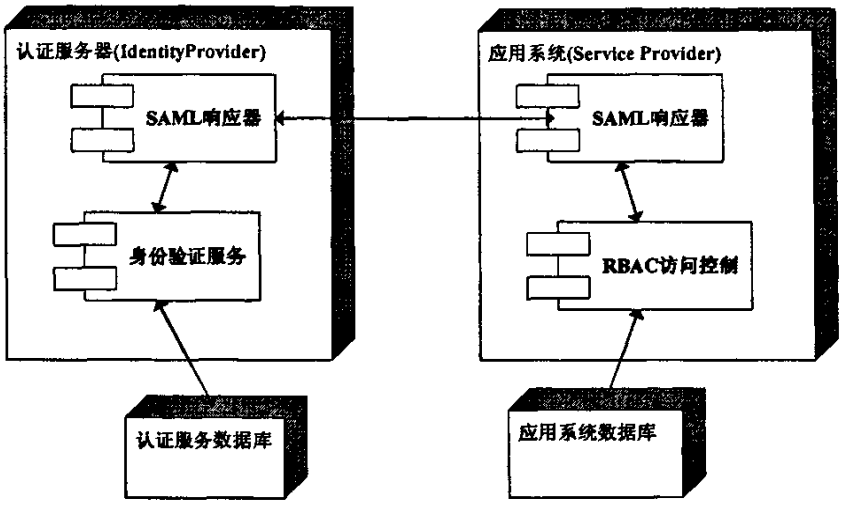


图 4-1 统一认证体系结构

认证服务端由 SAML 响应器、身份验证服务和认证服务数据库构成，SAML 响应器用于与应用服务端进行 SAML 协议信息，用于完成认证服务器和应用系统的身份信息处理和通信，身份验证服务主要用于对登录用户进行验证，而认证服务数据库用于保存用户的身份相关的信息。

应用服务端由 SAML 响应器、RBAC 访问控制和应用系统数据库组成，其中 SAML 响应器用于与认证服务端进行 SAML 协议信息，用于完成认证服务器和应用系统的身份信息处理和通信，RBAC 访问控制组件用于根据用户在该系统中的角色对用户访问系统的权限进行有效的控制，而应用系统端数据库主要包括应用系统的账户和对应权限信息，当然还应包括用于保存应用系统业务数据的信息，由于本论文的关注点主要在认证方面，故在本系统中不会表达。

4.4 功能模型分析与设计

4.4.1 用户管理

该系统以用户 ID 为主体进行管理，替代其在原来各个应用系统中的主体（账户）管理。

按照统一用户管理的定义，在理想情况下，所有应用系统的访问主体信息皆为认证中心管理的用户信息，实际的应用系统不保留任何主体（用户）信息，只保留应用系统所需的业务数据。当需要用户相关身份信息时可以从认证中心直接读取，由认证中心平台对用户信息进行统一的存储和管理，这样地话，

能够使所有用户的身份信息能够按照统一的标准进行管理和维护，从而最大程度的保证用户信息的唯一性和标准性以及用户权限的一致性。应用系统端不需再考虑用户身份信息管理、用户身份认证和资源访问控制等内容，降低了应用系统的开发难度，这也减轻了系统管理员的负担，如当系统中某个员工离职，管理员没有必要再把每个应用系统的用户删除，这是非常理想的一种方式。

但是在实际的应用中，各应用系统一般是在不同时间阶段进行开发，并且每个应用系统都有自己的访问控制方式，而且各业务系统对用户身份信息理解不同，从而造成用户身份信息的不一致。如果要采用理想的方式，则必须对现有的应用系统进行改造，需要对应用系统本身的用户、权限等模块进行大规模的改造，工作量和难度都会很大，对遗留系统、产品等系统，由于用户信息和业务逻辑关系紧耦合，或者向统一认证理想的用户管理模式改造的代价太高，并且目前很多企业的业务系统都需要不间断使用，所以这样直接改造肯定会给企业使用系统造成麻烦，而且所需要的改造资金也很大。所以，笔者根据上述实际情况，提出了采用映射的方式把各个系统的账户映射到身份认证服务端的唯一的用户 ID 方式，并把原来在各系统进行的身份验证工作转移到认证服务端来，这样既保留了原来的系统的账户信息，降低了改造难度和成本又可以把身份认证工作集中到一个认证中心来解决，如图如图 4-2 所示：

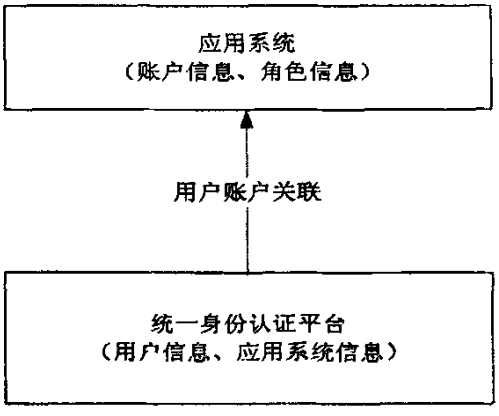


图 4-2 统一用户管理示意图

统一身份认证平台通过唯一的用户身份标志，通过用户和应用系统的原有账户进行关联，就完成了这种映射。

4.4.2 统一认证

按照统一认证的功能要求即系统要求只有一个权威的认证中心来进行应用

系统访问主体的鉴别，在统一用户管理的基础上，对用户身份进行确认和鉴别，由于统一用户管理模型笔者采用了用户 ID 和各信息系统本身账户信息关联的方式，所以统一认证时，由统一认证中心为登录用户提供与之对应的票据，由用户 ID 和各信息系统的账号关联关系，然后把票据分发给用户需要登录的应用系统，如图 4-3 所示：

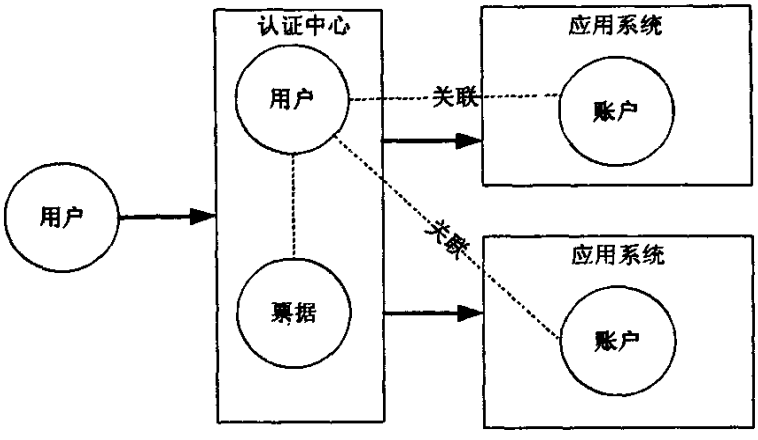


图 4-3 统一身份认证示意图

根据第三章中介绍的模型，Microsoft 的 Passport、Kerberos 和自由联盟的 SAML 都可以作为认证中心使用的技术方式，但由于 Passport 有其固有的缺点，就是权利过于集中，这和各组织没有办法管理自己的用户，当认证中心服务器出问题，它会使所有的应用系统陷于瘫痪，并且它使用 3-DES 加密的对称加密手段使得其密钥的保护和分发非常麻烦。

而使用 Kerberos 认证都需要客户端计算处理能力有要求，而现在的浏览器却是无法胜任的，这就需要对客户端进行设计，增加了系统实现和部署的复杂度，而且目前大多数的企业应用系统都采用了 B/S 结构（采用浏览器做为客户端），所以采用 Kerberos 的方式显然不合适。

而 SAML 协议没有上述限制，它使用 XML 格式的数据表达身份信息，具有开发性、标准性的特点，使实现单点登录的一个理想选择。我们这里采用了亦采用 SAML 认证体系来完成统一认证的功能，该模型采用传统的用户名/密码的方式对用户的身份进行鉴别，用 SAML 断言表示认证结果，应用系统根据认证结果，给用户提供服务，本系统将采用 SAML 的 Browser/Artifact 配置方式（参见 3.3.2 节），实现用户-认证中心-应用服务器之间的绑定，其间还要

考虑一定的安全措施保证系统的安全。

4.4.3 访问控制

根据系统功能的要求，在理想情况下系统要求只有一个权威的访问控制中心来进行权限的分配和鉴别，在统一用户管理、统一认证的基础上，对用户访问的资源进行控制，即身份的验证和授权在认证中心完成，这样的话各系统不需要考虑访问控制的问题，所有的访问控制权交给认证中心进行处理，所以应用系统的功能将更加单一不必考虑用户身份的管理和授权。

但由于各业务系统之间角色划分基本独立，无法对一个角色信息进行多次的复用，这样会造成一个用户下可能会存在数量很多的角色，并且此种模型需要由专职部门和人员对用户与角色进行维护，因此当系统中角色种类较多时，认证中心的角色维护会相当的复杂，会加大操作员的操作难度和工作量，并且容易造成逻辑错误和漏洞，在这里，本系统根据需求按照用户和业务系统中账号的对应，通过账号去访问系统本身的权限控制系统，本文采用目前比较常用的 RBAC 模型进行访问控制，实际上，目前很多应用系统都采用这样的访问控制方式，为了减轻改造的成本，尽量的采用或较少的原来的访问控制方式，访问控制方式如图 4-4 所示：

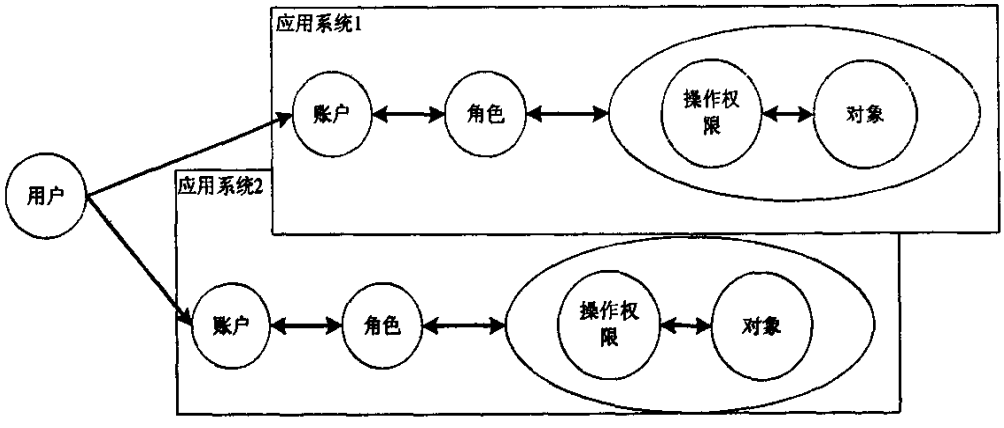


图 4-4 统一访问控制示意图

在这里每个应用系统都有自己独立的一套 RBAC 控制规则，这样就把认证中心不需要关心应用系统的访问控制表达，由各个应用系统根据自己系统的特点定义自己的访问控制规则。降低了认证端和应用端之间的耦和度。

目前企业应用系统划分角色的规则一般采用根据职位的方式划分，但由于

企业管理的实际情况很复杂，有很企业划分职位时候不可能做到很细致，而可以完全用于定义信息系统中的角色，所以根据职位不能满足一些特殊情况，所以有必要对这种特殊情况给予系统考虑，实际上，笔者在从事企业应用系统的开发中，客户经常说某个职位的某个人其访问系统的权限与该职位大多数人员的权限不一样（多或少）。所以笔者为了满足系统灵活性的要求，对原有的 RBAC 模型进行了一定的修改，基于 RBAC 的修改是在 RBAC0 基础上进行的（参见 2.6 节），修改后的 RBAC 模型如图 4-5 所示：

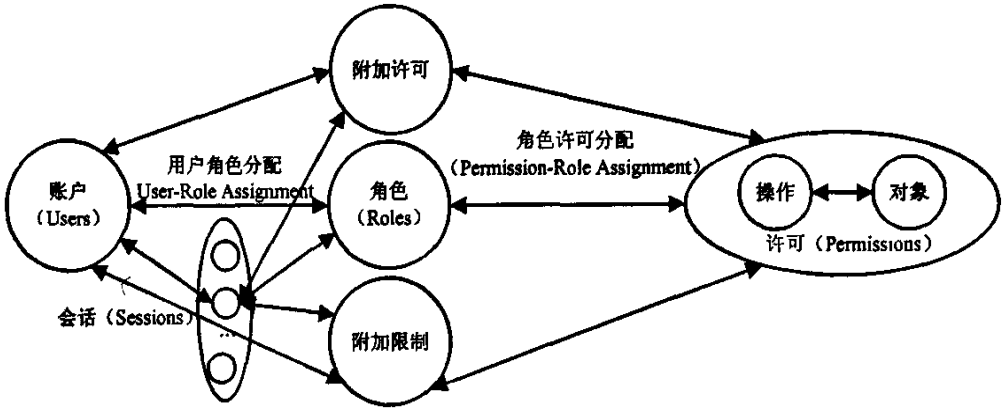


图 4-5 修改后的 RBAC0 模型

修改后的访问控制模型，增加了附加权限许可和限制许可，相当于给一些特殊用户添加了用户到许可的直接配置通道，通过它可以灵活的配置该类用户的许可（即权限），更好的满足业务系统的需求。

为更好的描述该模型，现在假定：

角色的对应权限集合为 **PR**；

附加许可对应的权限集合为 **PA**；

附加限制对应的权限集合为 **PL**；

用户时访问系统的应有权限 **PF**；

则用户访问系统的权限，其角色对应的权限加附加许可表示对应的权限，再减去限制许可表示中对应的权限。所以系统给用户最终的授权，用集合公式表示为：$PF = PR \cup PA \cap \overline{PL}$

通过在原来对角色授权的基础上，增加了用户附加许可和附加限制的表示，

可以根据一些特殊人员进行特殊的配置，这样就能灵活配置用户的访问控制权限，更好的满足业务系统的要求，对实现而言，只需要在原来系统的数据库中增加两张关系表表达，实现很方便，这样既实现灵活性的要求，又可以降低改造成本。

第五章 统一身份认证系统详细设计和实现

5.1 数据库设计

按照模型设计的要求，统一认证系统包括两部分数据库，一部分属于认证服务器的 IdentityInformation 数据库，另一部分是应用系统的数据库 ServiceInformation 数据库（因为只为了表示统一身份认证，在该系统设计只有访问控制部分的设计）。

首先介绍 IdentityInformation 数据库的设计，设计该数据库的目的，主要是为了表达同一用户管理模型中的映射关系，把各应用系统的账户和统一的用户 ID 联系起来，该数据库由 Application、User、User_Account_Application 三个表构成，各表之间通过外键相关联，数据库设计如图 5-1 所示：

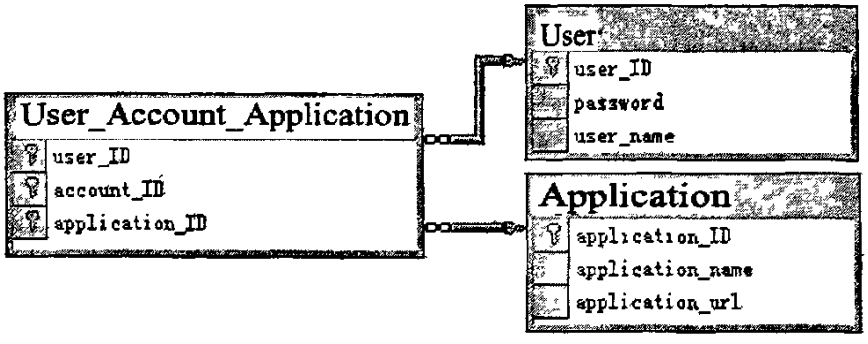


图 5-1 IdentityInformation 数据库

根据上图的表设计，现在分别对各表的设计进行介绍：

表 User 的主要作用是为了表达统一身份认证的统一用户的目的，在这把有应用系统的各个账号由 User 表中的 user_ID 表示，这样用户的身份就被统一起来了，用户的验证主要根据这张表的 user_ID 和 password 判断，它由用户唯一标志（user_ID）、密码（password）和用户名（user_name）构成。

表 User_Account_Application 的主要作用是为了建立用户到各应用系统中的账号和账号对应访问的系统建立对应关系，由于根据登录模型设计的要求，访问控制由系统自行决定，所以建立这样的关联后，用户访问各系统就可以根

据系统自身的访问控制表，进行用户系统授权。它由用户唯一标志（user_ID）、账号标志（account_ID）和应用系统唯一标志（application_ID）构成，这三个字段联合构成一个主键，用于表示三者的唯一关系。

表 Application 的主要作用是为了描述应用系统的信息，包括应用系统唯一标志（application_ID）、应用系统名称（application_name）,和应用系统的访问地址（application_url）,由于应用系统都采用的 B-S 结构，所以可以用 URL 地址表示应用系统资源的位置。

下面介绍应用系统的数据库 ServiceInformation，设计该数据库的目的是表达设计的访问控制模型，设计的表只包含完成访问控制的部分，该数据库包括四个基本信息表即 User(用户表)、Role(角色表)、Privilege(权限表)、object(操作对象)它们分别定义了用户、角色、权限、操作对象各个实体的基本信息；另外有五个个关系表即 User_To_Role（用户-角色关系表）、User_To_SubPrivilege(用户-附加权限关系表)、User_To_SubPrivilege(用户-限制权限关系表)、 Privilege_To_Object(权限-操作对象关系表)。数据库中各表的关系设计如图 5-2 所示：

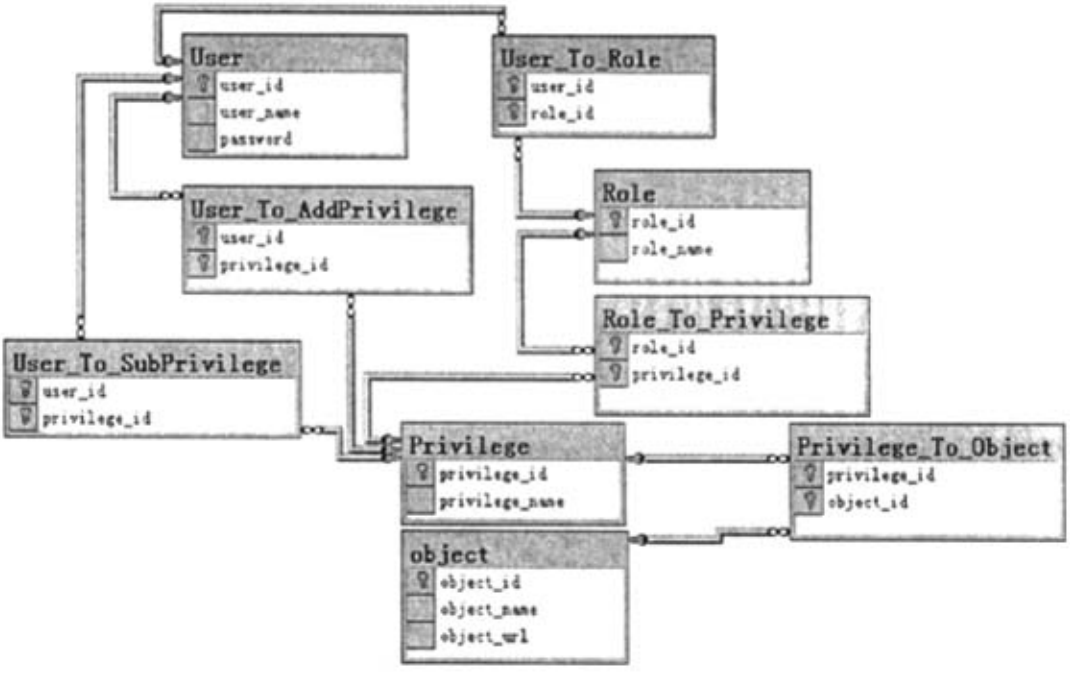


图 5-2 ServiceInformation 数据库

根据上图的表设计，现在分别对各表的设计进行介绍：

User 表的作用主要是用户描述应用系统的账户信息，它是访问控制权限的载体，它对应于认证端的账户的概念。

Role 表用于定义角色，一个角色代表使用享有系统权限相似的一部分人。

Privilege 表用于定义系统的权限。

Object 表用于表达系统的功能实体，它和权限关联通过 Privilege_To_Object(权限-操作对象关系表)关联起来，用于表达访问控制的结果。

User_To_Role（用户-角色关系表）的作用主要描述用户到角色的对应，而 Role_To_Privilege（角色-权限关系表）用于表达角色拥有的权限，这样通过用户就能够找到用户到权限的一个对应，同样 User_To_SubPrivilege(用户-附加权限关系表)、User_To_SubPrivilege(用户-限制权限关系表)也可以分别得到一个用户到权限的对应，通过上述对应关系可以根据模型设计的要求找到一个用户到权限的映射，从而达到访问控制的目的，笔者通过 SQL 语言中各表的连接查询可以表达出模型设计的要求，具体的实现方式将在下面的章节作详细介绍。

5.2 SAML 组件介绍

该系统中采用了 ComponentSpace 公司的 SAML 组件，该组件很好的实现了 SAML1.1 协议中所要求的功能，现在对该组件的组成和功能进行一个简要的介绍，该组件按 SAML 的协议层次，把组件分成了 Assertions、Protocol、Bindings、Profile 四部分（名称空间）。

（1） Assertion

通过 Assertion 可以创建、修改和访问 SAML 断言；完成了针对断言 XML 数据信息序列化或反序列，并创建和验证 SAML 断言的 XML 签名。其中包括 Action、Attribute、Subject、Assertion、Attribute Statement 类等。

（2） Protocol

通过 Protocol 可以创建、修改和访问 SAML 请求和返回信息；完成该 XML 信息的序列化和反序列化；并创建和验证请求/返回消息的 XML 签名。

AssertionArtifact、AttributeQuery 、SubjectQuery、Request : RequestAbstract、Response 、Status 等。

（3）Bindings

通过 Bindings 可以通过基于 HTTP 的 SOAP 协议接收和发送 Protocol 信息。主要包括类 SOAP 和 SOAPBinding。

（4）Profile

Profile 提供了对 browser/artifact 和 browser/post 的配置方式提供支持，主要包括了 BrowserArtifactProfile 类。

5.3 系统实现

5.3.1 系统详细模块结构

该系统基于 B-S 结构创建，将采用微软的 ASP.NET 技术作为实现基础，数据库则采用了微软的 SQL2000 数据库，消息的传递主要采用网页传值的方式实现，根据模型设计的要求，系统由认证服务器端（Identity Provider）网站和应用服务器端（Service Provider）网站构成，详细模块的结构如图 5-3 所示：

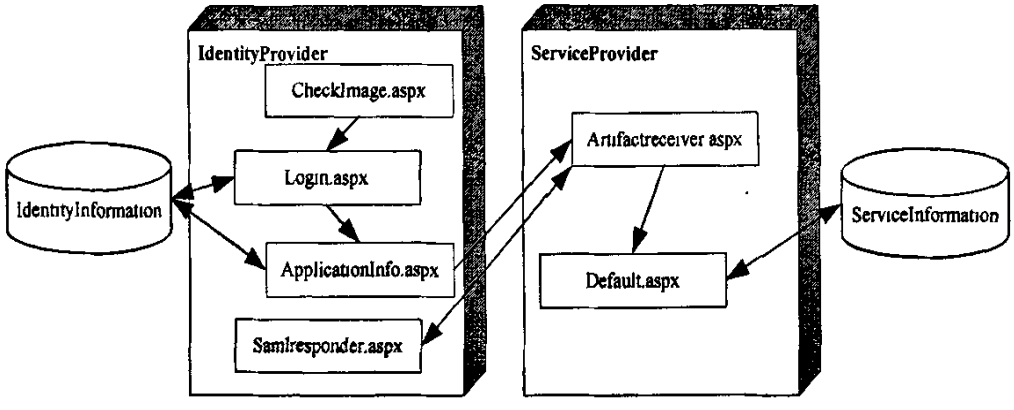


图 5-3 SSO 详细模块结构设计

如上图所示，每个页面（aspx 文件）都实现模块中的部分功能，通过各个页面之间的身份信息的传递 ，最终达到实现模型设计的要求，满足统一身份认证的要求，现针对各个页面的功能作一个简要的功能介绍：

（1）认证端 (IdentityProvider)网页

认证端网站中包含页面和其对应的功能描述如表 5-1 所示：

表 5-1 IdentityProvider 各模块

CheckImage.aspx（验证码页面）	主要用于随机生成图片码，用户每次登录必须输入不同的图片验证码，防止攻击者采用自动攻击工具不断的尝试用户名和密码，增加了系统的安全性，增加攻击者尝试猜测密码的难度，也是目前网站比较常用的安全方式之一。
Login.aspx（登录页面）	主要通过访问认证中心数据库中的用户名和密码信息来判断用户的身份是否合法，如果认证成功，则转道 ApplicationInfo.aspx(应用系统信息页面)。
ApplicationInfo.aspx（应用系统信息页面）	主要根据已经通过验证的登录用户，依据数据库中该用户对应的可访问的应用系统信息，在页面上显示，供用户选择登录，用户选择登录后将生成用户认证断言和票据并保存在应用中，然后把票据传递给应用服务器的 Artifactreceiver.aspx（票据接收处理页面），应用系统端作进一步处理。
SamlresponderAspx（断言查询）	主要根据应用系统端传递过来的请求信息，找到对应该请求的断言，并根据该断言，把断言结果返回给应用系统端，应用系统端作进一步处理。
IdentityInformation（认证端数据库）	认证端数据库，该数据库保存了用户的身份验证数据和应用系统信息的相关信息。

（2）应用系统端（ServiceProvider）

应用系统端网站中包含页面和其对应的功能如表 5-2 所示：

表 5-2 ServiceProvider 各模块

名称	功能描述
Artifactreceiver.aspx（票据接收页面）	根据认证服务器端的票据，构造 SAML 请求对象，并把查询断言传递到认证服务器，认证服务器根据其保存的断言信息，返回查询结果（SAML 返回对象）给该页面，应用系统端根据该结果决定是否给用户访

	问系统的权利。
Defalut.aspx （系统默认页面）	该页面为应用系统的总控制台，在这里可以访问应用系统的具体功能，如果用户认证成功，则表示用户有权进入了该应用系统，该页面将根据对用户的访问控制权利，对用户访问系统的功能作限定。
IdentityInformation （应用系统端数据库）	应用系统端数据库，保存了原来系统的用户的身份（账户）信息和对应访问控制信息。

5.3.2 统一验证流程

该系统通过 SAML 协议的 Browser/Artifact 方式进行实现，登陆流程大体为用户在验证服务器端验证通过后，则生成断言和票据（Artifact）并把断言和 Artifact 绑定在一起并保存到认证服务器端的全局变量（在该系统的实现中为 ASP.NET 的 Application 全局变量），认证服务器把票据传递给认证服务器，应用服务器根据票据（Artifact）查询认证服务器所对应的断言，并把查询结果（认证结果）返回给应用服务器，应用服务器判断结果的有效性和真伪性，如果真实有效则对用户访问系统授权。具体的登录时序图，如图 5-4 所示：

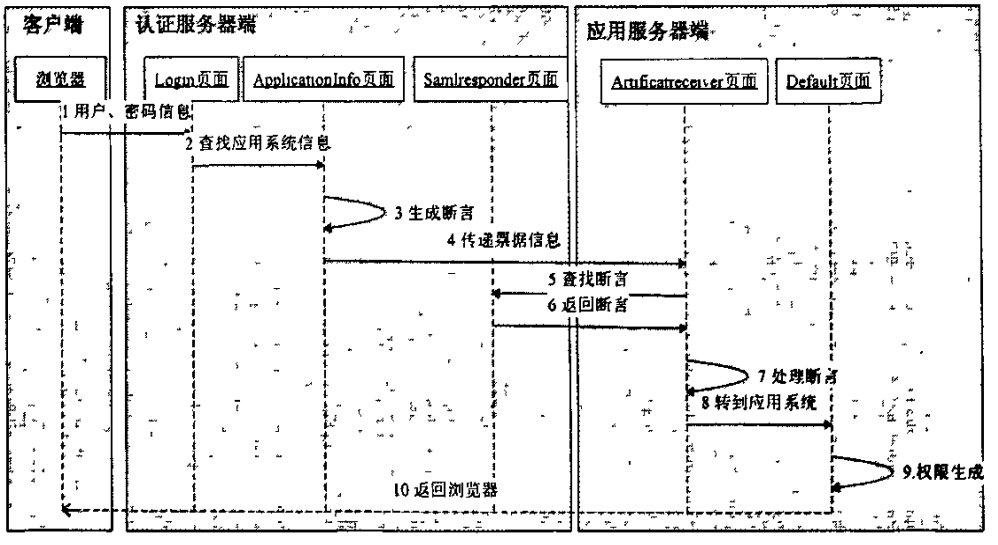


图 5-4 登录流程时间序列图

下面根据登录流程的时间序列图，对各部分的功能和实现给出一个详细的

描述如下：

（1） 用户输入用户名、密码、验证码到认证服务器进行验证，本系统采用传统的用户名和密码的方式进行验证，登录处理流程如下：

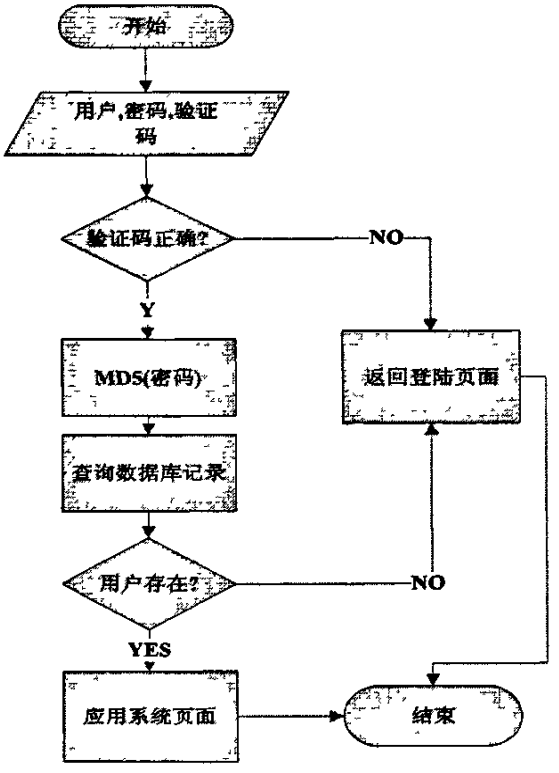


图 5-5 认证端登录处理流程图

首先，系统验证图片码，如果成功，对输入的密码加密（由于数据库保存的密码是一个 MD5 的 Hash 值，所以需要对用户输入密码进行 MD5 加密），然后再和数据库中的记录进行比对，如果用户验证成功，系统把通过验证的用户 ID 写入 Session（会话），页面将跳转到 ApplicationInfo 页面作进一步处理，源代码如下：

```
If IsPassValidate(UsernameText.Value, PasswordText.Value) Then
    Session("USERID")= UsernameText.Value
    Response.Redirect("ApplicationInfo.aspx")
End If
```

（2）在 ApplicationInfo 页面，系统将根据系统中用户 ID 的 Session 值，对

数据库进行查询，找出关联到改用户 ID 的用户相关系统信息并展示到页面上供用户选择，其源代码如下：

```
strSql = "SELECT C.application_name, C.application_url, B.account_ID FROM
  [User] A INNER JOIN" + " User_Account_Application B ON A.user_ID = B.user_ID
  INNER JOIN " + "Application C ON B.application_ID = C.application_ID" +
  " WHERE (A.user_ID = '" + Session("USERID").ToString() + "')"
Dim dataAdapter As New SqlDataAdapter(strSql, dbConn)
Dim ds As New DataSet
dataAdapter.Fill(ds)
  '绑定数据到 GridView 控件
gvApplication.DataSource = ds
gvApplication.DataBind()
dbConn.Close()
```

（3）当用户选择了需要进入的应用系统并确认进入后，系统将根据用户ID对应于该应用系统的用户帐户产生断言和票据，并把它保存在全局的Application变量中，其源代码如下：

```
' 创建断言
Dim assertion  As  Assertion
= CreateAssertion(ConfirmationMethod.Methods.Artifact, userCode)
' 创建票据(artifact)
Dim artifact As BrowserArtifactProfile.ArtifactType1 = CreateArtifact()
Dim artifactString As String = artifact.ToString()
' 建立断言和票据的关联并存入全局变量 Application
Application.Add(artifactString, assertion)
```

（4）生成断言后，系统将把票据(aritficate)以 URL 传值得方式传到ArtifactReceiver 页面，源代码如下：

```
' 把票据传递到应用系统端
stringBullder.AppendFormat("http://localhost/SAMLServiceProvider2/SAML/ArtifactRec
eiver.aspx?TARGET={0}&SAMLart={1}",HttpUtility.UrlEncode(target),HttpUtility.UrlE
ncode(artifactString))
Response.Redirect(stringBullder.ToString(), False)
```

（5-8）由于第5步到第8步联系比较紧密，是SAML的主要身份信息交换机制，所以在这里把它们放在一起进行说明，首先ArtifactReceiver页面在装载时，根据传回来的票据值，创建SAML的Request对象，并通过它向应用服务器页面（在这里是SamlResponder页面）查询断言；Samlresoponder页面负责处理SAML 请求，并根据票据(artifact)查找Application中相应的断言，并构造SAML的Response对象，通过认证服务器的私钥进行加密，并返回；ArtifactReceiver页面处理SAML的Response对象，并根据断言的结果，其中断言中包含了统一认证用户的ID对应到应用系统的账号，给用户访问系统授权（这里采用了ASP.NET的窗口认证方式，该方式可以保留用户的登录信息到Cookie中）；根据返回的断言结果，赋予访问用户访问应用系统的权利，页面跳转到应用系统页面，这里采用了.NET的窗口认证方式，可以通过Cookie保存用户的登录信息，用户在信息有效期内不需要在登录就能进入系统，具体的流程如图5-6所示。

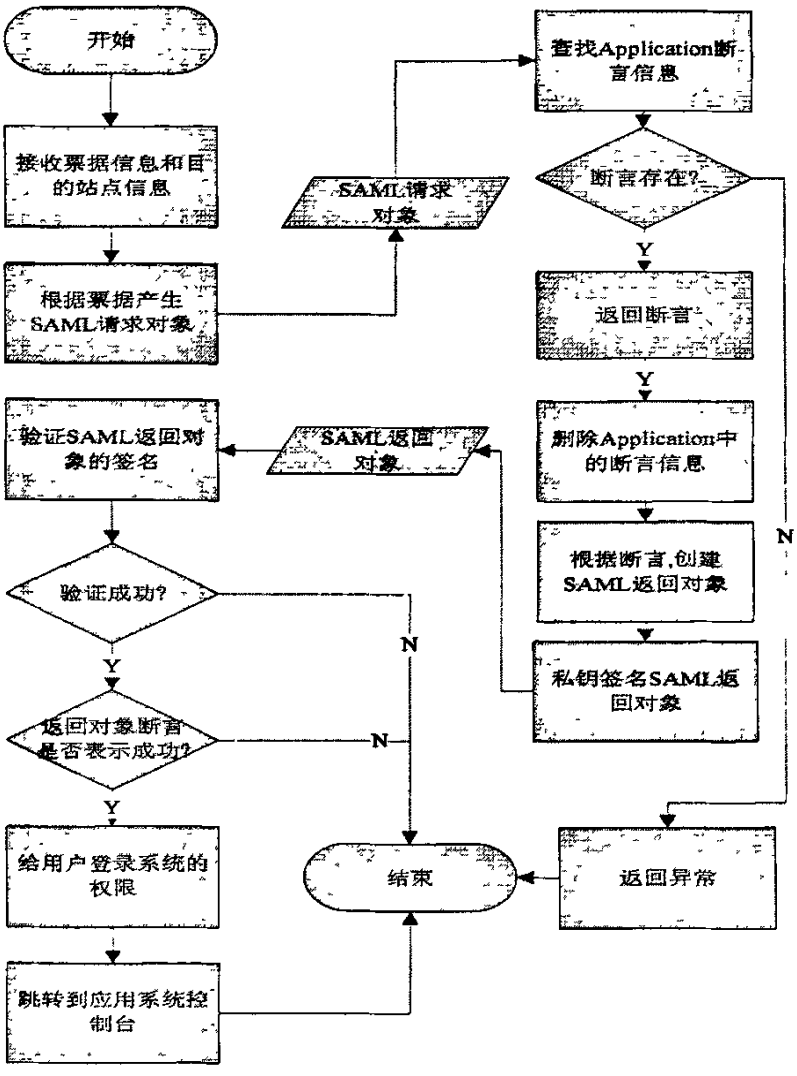


图5-6 SAML身份信息交换流程图

（9）在应用系统端 Default 页面，根据认证用户对应的账号和应用系统数据库中关于该账号的访问控制信息，对用户进行授权，在这里通过从程序中调用 SQL2000 中数据库中的存储过程，来完成这一复杂的 SELECT 查询，其中存储过程的源代码如下：

```
CREATE PROCEDURE
[dbo].[ GetPrivileges]
(@user_id varchar(50))
AS
--找出用户-角色-权限关系中的权限，并去除用户-权限限制-权限关系中的权限
```

```
SELECT A.[privilege_id],A.[privilege_name]
FROM   [USER] B
INNER JOIN [User_To_Role] C ON B.[user_id]=C.[user_id]
INNER JOIN [Role_To_Privilege] D ON C.[role_id]=D.[role_id]
INNER JOIN [Privilege] A ON D.[privilege_id]=A.[privilege_id]
WHERE B.[user_id]= @user_id AND A.privilege_id NOT IN
(
 SELECT A.privilege_id FROM [USER] B
 INNER JOIN [User_To_SubPrivilege] E ON B.[user_id]=E.[user_id]
 INNER JOIN [Privilege] A ON A.privilege_id=E.privilege_id
 WHERE B.[user_id]= @user_id
)
UNION
--找出用户-附加权限-权限关系中的权限，并去除用户-权限限制-权限关系中的权限
 SELECT A.privilege_id,A.privilege_name FROM [USER] B
 INNER JOIN [User_To_AddPrivilege] E ON B.[user_id]=E.[user_id]
 INNER JOIN [Privilege] A ON A.privilege_id=E.privilege_id
 WHERE B.[user_id]= @user_id AND A.privilege_id NOT IN
(
 SELECT A.privilege_id FROM [USER] B
 INNER JOIN [User_To_SubPrivilege] E ON B.[user_id]=E.[user_id]
 INNER JOIN [Privilege] A ON A.privilege_id=E.privilege_id
 WHERE B.[user_id]= @user_id
)
GO
```

上面的存储过程主要根据模型设计的要求，通过一个复杂的 SELECT 语句完成对用户账户对应的系统权限的查找，查找完成后，系统根据权限和操作对象的对应关系，把它们绑定在一起，在该系统中，操作对象对应于一个菜单项，如果用户拥有该权限，才能看到对应的功能菜单项，其实现的源代码如下：

```
'获得权限表对象
Dim   dt As New DataTable
```

```
dt = GetPrivilegeTable("[dbo].[ GetPrivileges]", para)
'根据权限表，拼接查询操作对象的 SQL 字符串
Dim strSQL = " SELECT object_name,object_url  FROM object A INNER JOIN
Privilege_To_Object B ON A.object_id=B.object_id  WHERE "
Dim i As Integer
For i = 0 To dt.Rows.Count - 1 Step 1
  If  i = 0  Then
    strSQL += " B.privilege_id='" + dt.Rows(i)("privilege_id") + "'"
  ElseIf  i <> 0  Then
    strSQL += " OR B.privilege_id='" + dt.Rows(i)("privilege_id") + "'"
  End If
Next
  '查找操作对象并绑定到功能菜单
Dim sqlAdapter As New SqlDataAdapter(strSQL, conn)
Dim dt1 As New DataTable
sqlAdapter.Fill(dt1)
For i = 0 To dt.Rows.Count Step 1
  Dim tmpNode As New TreeNode(dt1.Rows(i)("object_name").ToString(),
  dt1.Rows(i)("object_url").ToString())
  tmpNode.NavigateUrl="javascript:Navigator('" + dt1.Rows(i)("object_url").ToString()
  + "');"
  TreeView1.Nodes(0).ChildNodes.Add(tmpNode)
 Next
```

10.统一身份认证流程完成，用户可以进入系统和使用其所具有的系统功能，下次用户登录可直接访问系统的 URL。

5.3.3 系统安全性的保证

（1）由于认证服务器采用用户名-密码的方式对系统进行认证，所以用户和密码等信息在传输过程中可能被窃听，笔者采用 SSL 对通信信道加密的方式，来保证数据传输的安全性，通过 Windows2003 操作系统可以很方便通过对 IIS

服务器的配置的实现 SSL。

（2）对于篡改断言的问题，笔者通过认证服务器对发送到应用系统端的断言通过自己的数字证书中的私钥签名后在发出，应用服务器端通过公钥验证签名，可以确保断言的真实性和不可抵赖性。

（3）重发和中间人攻击，笔者采用 Browser/Artifact 方式时，每次生成的 Artifact 均不相同，而且在使用过 Artifact 后会删除与断言之间的映射，从而使重播无效。

（4）为了保护用户密码的私隐性，所有的密码都用 MD5 算法进行了 Hash，这样数据库管理员就无法看到用户的明文密码。

（5）采用了图片验证码，防止攻击者对密码进行试探性攻击，使攻击者猜测用户密码的难度加大。

5.4 系统功能测试

5.4.1 测试数据构造

本该测试主要针对功能进行测试，认证端主要实现统一用户管理和身份认证的功能，应用服务器端主要验证访问控制功能。为了验证功能，首先构建系统所需要的测试数据如表 5-3、5-4 所示：

表 5-3 认证服务器端数据表

用户表		
user_ID	password	user_name
Jerry	6d5990ec39501bcd	Jerry
Tom	6d5990ec39501bcd	Tom
应用系统表		
application_ID	application_name	application_url
App001	测试应用系统	http://192.168.1.7/ServiceProvider/Default.aspx
App002	客户管理系统	http://192.168.1.7/ServiceProviderx/Default.aspx
App003	资源管理系统	http://192.168.1.7/ServiceProviderxx/Default.aspx
用户-应用系统-帐户关系表		
user_ID	account_ID	application_ID

Jerry	123	App002
Jerry	GH001	App001
Tom	007	App002
Tom	dd	App003
Tom	GH002	App001

表 5-4 应用服务器端数据表

用户名

user_id	user_name	password
GH001	Jerry	ca07899fbd010cbd
GH002	Tom	ca07899fbd010cbd

角色表

role_id	role_name
Manager	经理
Worker	工人

操作对象表

object_id	object_name	object_url
o001	财务管理	FanancialManage.aspx
o002	库房管理	PeopleManage.aspx
o003	客户管理	CustomerManage.aspx
o004	制度管理	PrincipleManage.aspx
o005	机密管理	SecretManage.aspx
o006	设备管理	DeviceManage.aspx

权限表

privilege_id	privilege_name
001	管理财务信息
002	管理库房信息
003	管理客户信息
004	管理制度信息
005	管理机密信息
006	管理设备信息

角色-权限关系表

role_id	privilege_id
Manager	001
Manager	002
Manager	003
Manager	004
Worker	003
Worker	006

用户-附加权限关系表

user_id	privilege_id
GH001	002
GH002	005
GH002	006

用户-限制权限关系表

user_id	privilege_id
GH002	002
GH002	005

用户-角色关系表

user_id	role_id

权限-操作对象关系表

privilege_id	object_id

GH001	Worker	001	o001
GH002	Manager	002	o002
		003	o003
		004	o004
		005	o005
		006	o006

上面构造了 Jerry 和 Tom 的用户数据，它们在测试应用系统中分别对应于帐号 GH001 和 GH002，并且在应用系统端的数据库，根据他们对应于系统的帐号分别在各种关系表中配置了不同的值，主要用于测试访问控制模型的正确性，为了测试系统的功能，分别通过 Jerry 和 Tom 通过认证端服务器登录到测试应用系统作为测试用例。

5.4.2 测试结果分析

根据测试用例，按照模型设计功能的要求，结合数据库中的测试数据数据，用户 Tom 能访问的应用系统为客户管理系统、资源管理系统、和测试应用系统（该系统用于测试访问控制模块的功能），而用户 Jerry 能访问的应用系统为测试应用系统和客户管理系统，用户 Tom 进入测试应用系统可以访问财务管理、客户管理、制度管理和设备管理的功能，而用户 Jerry 可以访问库房管理、客户管理和设备管理的功能。测试的结果对照图如下图 5-7 所示：

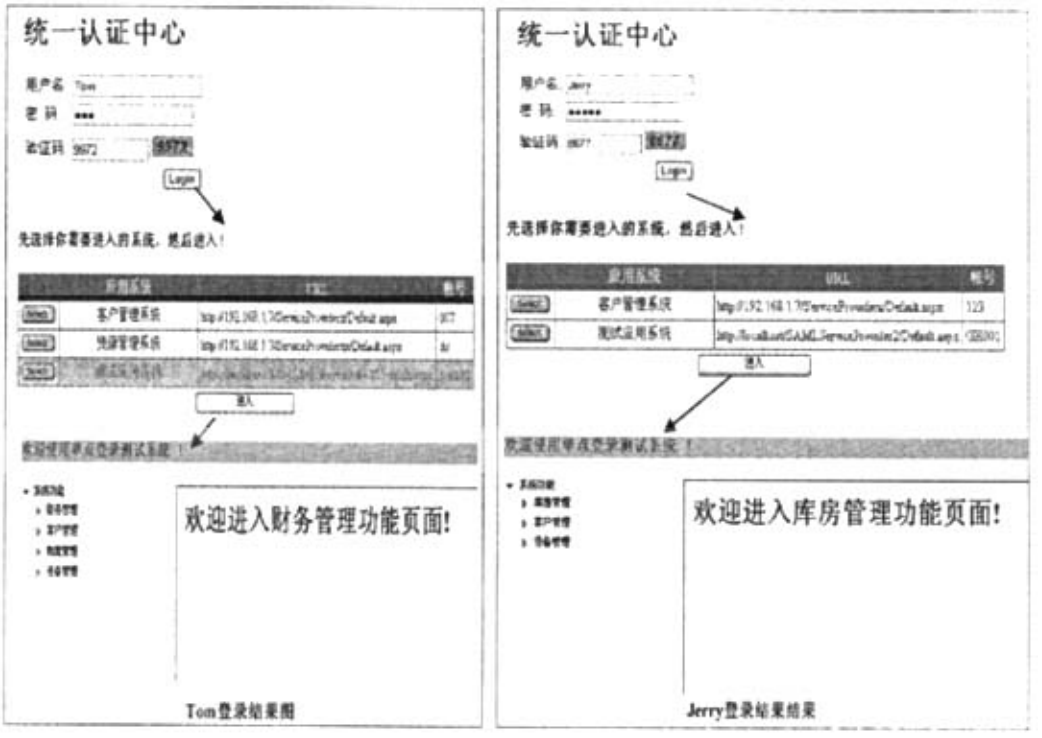


图 5-7 测试结果对照图

对统一用户管理模块而言：用户 ID 为 Tom 和 Jerry 分别对应于测试应用系统的 GH001 和 GH002，这样就建立了用户 ID 到应用系统账号的映射。

对于统一身份认证模块而言：用户 Tom 和 Jerry 都是在认证服务器进行身份认证工作，并且都通过了认证。

对访问控制模块而言：由于在测试应用系统中账户 GH001 和 GH002 配置的权限不一样，所以他们能访问的功能模块也不一样。

当用户登录后，用户关掉浏览器，用户不需要再次登录就可以直接进入需要登录的应用系统，这样就减少了用户的重复登录，实现了单点登录。

分析测试结果表明，系统实现达到了系统设计的要求，对于企业统一身份认证改造具有一定的参考意义，由于采用标准的 SAML 协议传递信息为将来系统的扩展提供了方便。

测试的过程中通过写文件的方式对测试过程中产生的 SAML 身份认证信息进行了记录，见附录。

总结与展望

随着企业信息化的加剧，企业应用系统越来越趋于多样化和复杂化。身份认证成为各系统必不可少的一部份，然而由于各系统的身份认证各自为政，迫切需要对统一身份认证进行研究，也是现阶段企业应用系统研究中的重要组成部分。

1、本文主要完成的工作：

(1)本身份认证所需要的理论知识进行了详细的梳理和充分的说明。

(2)通过对目前流行的统一认证机制的原理分析和比较，设计了比较适合企业应用，具有较高灵活性的身份认证机制模型。

(3)根据模型设计的要求，结合现有的技术手段对模型进行了实现，该系统充分满足了企业的实际要求，达到了模型设计的功能要求。

(4)对完成的系统进行了验证性测试，并对测试结果进行了深入的研究。

2、今后研究工作的展望：

本文通过对统一身份认证应用的研究，实现了基于.Net 技术的企业统一认证系统的原型。但该系统还有待于进一步的研究和改善。

(1)对于跨企业间的身份认证问题需要深入的研究，因为联合身份认证对于企业越来越重要。

(2) 基于 JAVA 技术的应用系统平台如何实现统一身份认证需要深入的研究。

致 谢

首先，我要诚挚地感谢我的导师楼新远副教授。本文从选题到定稿，都是在楼老师的悉心指导下完成的。在三年的研究生学习期间，无论是在学业上还是在生活上，我都得到了他无微不致的关怀和照顾。他严谨的治学态度和敬业精神，以及对我的谆谆教诲、严格要求，都将使我终生受益。

其次，在我论文的收集和研究过程中，软件工程实验室的同学们给予我很大的帮助，在此一并表示感谢。

最后，我还要感谢我的家人和朋友，是他们在物质上、精神上给了我全力的支持，可以说没有他们就没有今天的我。

参考文献

[1] http://www.projectliberty.org/

[2] http://bbs.chinaitlab.com/redirect.php?tid=178831&goto=lastpost

[3] 分布式安全,http://www.microsoft.com/china/security/bestprac/ch11ce.asp

[4] 李兵,用户统一身份认证系统的设计与实现[硕士学位论文],哈尔滨理工大学,2005.

[5] William Stallings.密码编码学与网络安全（第三版）.电子工业出版社,2005

[6] Matt Bishop.计算机安全学.电子工业出版社,2005

[7] Steve Graham. 用 JAVA 构建 Web 服务. 机械工业出版社,2003

[8] W3C.Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation. http://www.w3.org/TR/REC-XML.Feb.2004

[9] 史东平. 基于 XML 和 WebService 的异构信息集成研究[硕士学位论文],山东大学, 2006.4.

[10]The Origins of SOAP from IBM Rejection to W3C Recognition[EB/ OL] . http :/ / www. intelligentent erprise. com.

[11]曾铮,吴明晖,应晶,简单对象访问协议 SOAP 综述,计算机应用研究,2002,02

[12]耿晖,王海波,基于 XML 的角色访问控制(RBAC),计算机应用研究,2002,12

[13]段云所（著）.信息安全概论.北京:高等教育出版社. ISBN:704012314,2003: 119-120.

[14]陈晓东.基于微软护照协议的单点登录系统的研究[硕士学位论文]华中科技大学, 2004,11.

[15]马亚娜,钱焕延,Passport 单一登录协议及其安全性分析,计算机工程,2000,10

[16]金辉,Sigle signon,http://www-128.ibm.com/developerworks/cn/ security/ se-sso/index.html

[17]邓永江,程转流,一个改进的 Kerberos 认证协议设计与分析,福建电脑,2006,06

[18]Manish Verma,使用 SAML 确保可移植的信任,http://www.ibm.com/developerworks/cn/xml/x-seclay4/

[19]BethLinker,安全地共享数字身份信息,http://dev2dev.bea.com.cn/techdoc/20060919883.html

[20]OASIS,Security Assertion Markup Language(SAML) 2.0 Technical Overview, http://www.oasis-open.org/committees/download.php/11511/ sstc- saml- tech-overview-2.0-draft-03.pdf

[21]OASIS,SAML Executive Overview, http://www. oasis-open.org/ committees/ download. php/11511/sstc-saml-tech-overview-2.0-draft-03.pdf

[22]SAML 2.0 简化联邦身份,网络世界,2005.12.19 No31

[23]牛秀元,WebLogic Server9.2 中配置基于 SAML 的 SSO 详述,http://dev2dev.bea.com.cn/techdoc/20060811864.html

[24]杨青,怀进鹏,徐枞巍,基于 SAML 的协同电子商务安全服务系统,计算机工程与应用,2002,14：228-231

[25]余 荣,刘明华,基于 SAML 实现 Web Service 的单点登录,计算机与现代化,2002,12：81-85

[26]周 帆,余 堃,吴 跃,支持移动环境下信任迁移的设计,计算机应用,2005.11 Vol.25 No.11 P.2512-2514.

[27]郑芳,程颖,王林平,基于 SAML 的 Web Service 安全模型研究, 计算机与数字工程,2005 Vol.33 No.1 P.81-84.

[28]于君,基于SAML的集成身份认证机制[硕士学位论文],南京理工大学, 2004.1 .

[29]吴鹏,吉逸,基于ＳＡＭＬ的安全服务系统的设计,计算机应用研究,2004.11 No.11 P.128-130.

[30]李彭军,郭文明,陈光杰,单点登录技术在电子政务系统中的应用,信息安全与通信保密.2006.7 P.18-20

[31]王鹃,李俊娥, 安全服务语言 SAML 分析,电脑知识与技术,2003.10 P.30-32

[32]林满山,郭荷清,单点登录技术的现状及发展, 计算机应用,2004.06 Vol.24 No.11 P.248-250.

[33]龙冬阳.网络安全技术及应用.华南理工大学出版社,2006.2.

[34]XML 新手入门, http://www.ibm.com/developerworks/cn/xml/newto/

[35]SSL 技术专题,中国 IT 认证实验室,http://www.chinaitlab. com/www/special/ssl.asp

[36]董军.网络安全分析师之路.电子工业出版社,2006.8

[37]Forms Authentication in ASP.NET2.0,http://msdn2.microsoft.com/ en-us/library/aa480476.aspx

[38]A.Russell Jones.ASP.NET 与 VB.NET 从入门到精通 ,电子工业出版社,2002.9.

[39]宫恩辉,基于校园网的单点登录系统得设计与实现[硕士学位论文],江苏大学,2006.

[40]阙喜戎,孙锐,龚向阳,王纯.信息安全原理及应用[M]. 北京：清华出版社.2003 年.

[41]姜晓静.基于 XML 统一身份认证技术研究[硕士学位论文],武汉理工大学,2006,04.

[42]SAML Reference Help,http://www.ComponentSpace.com

[43]SAML QuickStart,http://www.ComponentSpace.com

[44]Yun-kyung Lee,User Authentication Mechanism Using Authentication Server in Home Network, Feb. 20-22, 2006 ICACT200G

[45]Etsuko Suzuki, A Design of Authentication System for Distributed Education , IEEE 2004, 31 May-2 June 2004 Page(s):66 - 71

[46]Andreas Pashalidis and Chris J. Mitchell, Impostor: A Single Sign-On System for Use from Untrusted Devices, 2004. GLOBECOM '04. IEEE Volume 4, 29 Nov.-3 Dec. 2004 Page(s):2191 - 2195 Vol.4

[47]Shigefusa Suzuki, An Authentication Technique Based on Distributed Security Management for the Global Mobility Network , IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 15, NO. 8, OCTOBER 1997

[48]Extensible Markup Language (XML), http://www.w3.org/XML/

[49]SOAP Specification[EB/OL]. http://www.w3.org/TR/soap/

[50]http://www.msdn.com

攻读硕士学位期间发表的论文

[1] 陈小云，黎泽良，基于 RBAC 原理的访问控制系统研究，西南交通大学学报,增刊 2007

[2] 黎泽良,楼新远，陈小云. 基于 Pocket PC 的 Socket 通信研究.四川师范大学学报(自然科学版)，增刊 2006

附 录

1. SAML 断言信息

<saml:Assertion MajorVersion="1" MinorVersion="1"
AssertionID="_e5771e4b-6463-4416-93e9-ab44b04bd4de"
Issuer="urn:source-site" IssueInstant="2007-05-06T09:24:20Z"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
<saml:Conditions NotBefore="2007-05-06T01:24:20Z"
NotOnOrAfter="2007-05-06T17:24:20Z"/>
<saml:AuthenticationStatement
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
AuthenticationInstant="2007-05-06T09:24:20Z">
<saml:Subject>
<saml:NameIdentifier NameQualifier="urn:source-site"
Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">GH002</s
aml:NameIdentifier>
<saml:SubjectConfirmation>
<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:artifact</sa
ml:ConfirmationMethod>
</saml:SubjectConfirmation>
</saml:Subject>
</saml:AuthenticationStatement>
</saml:Assertion>

2. SAML请求信息

<samlp:Request RequestID="_b2af55a1-db98-44c4-a130-89a869d5dcaf"
MajorVersion="1" MinorVersion="1" IssueInstant="2007-05-06T09:24:20Z"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
<samlp:AssertionArtifact>AAFwg5OPoSqOW2jkQgvMPXIg6wE5lq/S+Y4
MyJdIgRwFURRIgNcAAAAA</samlp:AssertionArtifact>
</samlp:Request>

3. SAML 返回信息

<samlp:Response ResponseID="_f0b446b8-59a2-43f4-a9ac-9565f63cd5dc"
InResponseTo="_b2af55a1-db98-44c4-a130-89a869d5dcaf" MajorVersion="1"
MinorVersion="1" IssueInstant="2007-05-06T09:24:20Z"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<SignedInfo>
<CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<Reference URI="#_f0b446b8-59a2-43f4-a9ac-9565f63cd5dc">
<Transforms>
<Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
<InclusiveNamespaces PrefixList="#default code ds
kind rw saml samlp typens"
xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</Transform>
</Transforms>
<DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<DigestValue>MOLIIw+HTVXEA6iSj5TJaeiSPbg=</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>hPMNLDUF8W5JtUvRSREN1P7kBIYU2vLuXndzBE0q5
DEkyj1F8Hnpqat9OH8BTl6O+aG3LcvL8zW+RI8JcCr44Glj2mlSazCwWBnWH
qq/P+if9A7EPoMHiHz56+G78jstKjv5wKfpbrHcgQP9v/sDC9gFO9+A9fl7U5/pq
oohxew=</SignatureValue>
<KeyInfo>

<X509Data>
<X509Certificate>MIIBzTCCATagAwIBAgIEQm3M6jANBgkqhkiG9w0BA
QQFADArMQswCQYDVQQGEwJVUzENMAsGA1UEChMEdGVzdDENMAsGA
1UEAxMEdGVzdDAeFw0wNTA0MjYwNTA4NThaFw0xNTA0MjQwNTA4NTh
aMCsxCzAJBgNVBAYTAlVTMQ0wCwYDVQQKEwR0ZXN0MQ0wCwYDVQQ
DEwR0ZXN0MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCgV/g3W
gSbAdu+6Tam2Nw70ucAii7h35vdLhQy2xIu3sCscCsKEjSxs3DVyWt3WSM/ovn
07rC40CMWK/9ILH9ayoiuin5YdK3lIwAcZJI1IJl9PuU4RzQ+9pqpFXKDHB3Ez
2NoV9Pvjg5RtDtIUzFhgBTnVXSLD5Ueobh0LtYK5QIDAQABMA0GCSqGSIb
3DQEBBAUAA4GBAJ8TP5c8+Hd+phlPvbhDWLuPvj/8na3oZ2Ji8ul52Yqpz2E
UwhKuszTznj1l9qWK7N3/eO+Ch3SZsiZyaZH/Cv0h4CIANJvpd3MHMmWFd+z
kylj5qsCtyuySXWxolO5Ur134PON7ULcWi1oiEmiwZZuza9qCdBwm1MbGmR7
iaPpo</X509Certificate>
</X509Data>
</KeyInfo>
</Signature>
<samlp:Status>
<samlp:StatusCode Value="samlp:Success"/>
</samlp:Status>
<saml:Assertion MajorVersion="1" MinorVersion="1"
AssertionID="_e5771e4b-6463-4416-93e9-ab44b04bd4de"
Issuer="urn:source-site" IssueInstant="2007-05-06T09:24:20Z"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
<saml:Conditions NotBefore="2007-05-06T01:24:20Z"
NotOnOrAfter="2007-05-06T17:24:20Z"/>
<saml:AuthenticationStatement
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
AuthenticationInstant="2007-05-06T09:24:20Z">
<saml:Subject>
<saml:NameIdentifier NameQualifier="urn:source-site"
Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">GH002</s
aml:NameIdentifier>

```
                <saml:SubjectConfirmation>
    <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:artifact</sa
ml:ConfirmationMethod>
                </saml:SubjectConfirmation>
            </saml:Subject>
        </saml:AuthenticationStatement>
    </saml:Assertion>
</samlp:Response>
```

统一身份认证系统的研究与实现

作者：陈小云
学位授予单位：西南交通大学
被引用次数：8次

参考文献(10条)

1. 曾铮,吴明晖,应晶 简单对象访问协议SOAP综述[期刊论文]-计算机应用研究 2002(02)

2. 耿晖,王海波 基于XML的角色访问控制（RBAC）[期刊论文]-计算机应用研究 2002(12)

3. 邓永江,程转流 一个改进的Kerberos认证协议设计与分析[期刊论文]-福建电脑 2006(06)

4. 杨青,怀进鹏,徐枞巍 基于SAML的协同电子商务安全服务系统[期刊论文]-计算机工程与应用 2002(14)

5. 余荣,刘明华 基于SAML实现Web Service的单点登录[期刊论文]-计算机与现代化 2005(12)

6. 周帆,佘堃,吴跃 支持移动环境下信任迁移的设计[期刊论文]-计算机应用 2005(11)

7. 郑芳,程颖,王林平 基于SAML的Web Service安全模型研究[期刊论文]-计算机与数字工程 2005(01)

8. 吴鹏,吉逸 基于SAML的安全服务系统的设计[期刊论文]-计算机应用研究 2004(11)

9. 李彭军,郭文明,陈光杰 单点登录技术在电子政务系统中的应用[期刊论文]-信息安全与通信保密 2006(07)

10. 林满山,郭荷清 单点登录技术的现状及发展[期刊论文]-计算机应用 2004(z1)

本文读者也读过(10条)

1. 王嘉佳 基于目录服务的统一身份认证系统的研究与实现[学位论文]2005

2. 马荣飞.MA Rong-fei 统一身份认证系统的研究与实现[期刊论文]-计算机工程与科学2009,31(2)

3. 肖琬蓉.杨生举.XIAO Wan-rong.YANG Sheng-ju 基于LDAP的统一用户认证系统设计与实现[期刊论文]-计算机科学2008,35(5)

4. 徐俊.黄传华.沈晓凡.XU Jun.HUANG Chuan-hua.SHEN Xiao-fan 基于Web Services数字校园统一身份认证系统的研究与实现[期刊论文]-计算机与现代化2007(10)

5. 李征 数据仓库统一身份认证系统的研究与实现[学位论文]2006

6. 李蔷 统一身份认证系统的研究与实现[学位论文]2008

7. 刘持莲 基于LDAP和Web服务的校园统一身份认证系统的研究与实现[学位论文]2006

8. 徐俊 基于Web Services的数字校园统一身份认证系统的研究与实现[学位论文]2007

9. 周建友 统一身份认证系统的研究与实现[学位论文]2010

10. 杨灵.邹娟.YANG Ling.ZOU Juan 基于Web Services的统一身份认证系统研究与实现[期刊论文]-现代计算机（专业版）2009(11)

引证文献(6条)

1. 唐四薪,邹赛,谢新华 基于AJAX和SAML技术的互联网单点登录系统[期刊论文]-计算机系统应用 2008(06)

2. 周浩 数字化校园中统一身份认证系统的研究与设计[学位论文]硕士 2009

3. 孙思纬,夏洪山 基于Shibboleth和SAML的跨校统一身份认证系统[期刊论文]-微型机与应用 2010(05)

4. 黄翰陞 基于SVO逻辑的身份认证安全协议形式化分析与研究[学位论文]硕士 2008

5. 王笃炎 基于SAML的单点登录系统研究与实现[学位论文]硕士 2008

6. 周广辉 企业统一用户认证平台的研究和实现[学位论文]硕士 2009