基于证书的单点访问模型

谢鸿波，冯 军，周明天

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】以授权管理基础设施和公钥基础设施为基础，研究了单点访问系统的实现模型。这两种技术在单点访问系统中实现身份认证和访问授权控制功能。通过中间件的方式来实现身份认证和授权管理模型，并设计了相关的安全协议，对整个体系结构的安全做了简单分析。该技术可使现有应用做较少的修改就能实现一个安全、透明的单点访问系统。

关 键 词 公共密钥基础设施; 特权管理基础设施; 单点访问; 单点登录
中图分类号 TP393.08 文献标识码 A

Single Access System Based on Certificate

XIE Hong-bo，FENG Jun，ZHOU Ming-tian

(School of Computer Science and Engineering, UEST of China Chengdu 610054)

Abstract SAS(Single Access System) is studied based on the PKI (Public Key Infrastructure) model and PMI (Privilege Management Infrastructure) model. The PKI and PMI can provide functions of authentication and authority in the SAS. The way to implement the authority and authentication using middleware as well as related secure protocol is discussed. The security of the system's architecture is also analyzed. These models enable a secure, transparent SAS with least modification to current applications.

Key words public key infrastructure; privilege management infrastructure; single-access system; single-sign on

单点登录是为解决用户要记忆多个口令和用户名的烦恼而提出的。它为用户提供认证信息的集中管理及灵活、强健的主认证。然而，该系统只是帮助用户完成了身份认证，而资源的访问控制则完全是由各应用自己管理，有的甚至没有控制。因此，目前的单点登录系统应该说只是解决了用户在访问各应用系统时要输入多个口令和用户名的麻烦，它并没有给安全域中的用户提供统一的安全平台。

单点登录系统不应该仅仅是帮助用户记忆多个用户名和口令。它应该为安全域中的用户提供完整的安全服务，包括身份认证和资源的授权管理。文献[1]发布了X509V3版本，提出了公共密钥基础设施(Public Key Infrastructure，PKI)模型。它提供了信任服务框架。文献[2]发布了X509V4版本，提出了特权管理基础设施(Privilege Management Infrastructure，PMI)模型，它提供了授权服务框架。PKI、PMI结合起来，可以实现对资源的安全访问控制。

为此，在PKI和PMI的基础上本文提出了一种基于中间件的实现模型，其中PKI实现统一的身份认证，而PMI完成统一的用户授权管理。这样的系统称为单点访问系统(Single Access System，SAS)，因为它不仅实现了单一的身份认证，还完成了用户对资源的访问控制。

1 单点访问的系统模型

以公共密钥体制为基础，结合了PKI/PMI的思想，本文提出了以下基于证书的单点访问系统模型。在模型中身份认证和授权管理是完全分离的，这是一种双证书对等系统，是比较理想的体系结构。

在这个模型中，基础设施中的PKI系统为用户(包括客户端和服务器端)发布身份证书(Public Key Certificate，PKC)，PMI系统根据资源为用户发布授权属性证书(Attribute Certificate，AC)。客户端要访问资

收稿日期：2003 – 05 – 14
基金项目：国家863计划资助项目(863-104-03-01)
作者简介：谢鸿波(1973 –)，男，博士生，主要从事计算机信息安全、安全协议分析方面的研究.

源必须经过安全服务层的验证。而安全服务器层的实现是通过中间件的方式来完成的。

1.1 基于PKC的身份验证模型

在基于证书的单点访问模型中，如图1所示。图中每个实体都拥有一个身份证书。用户*A*的证书可以表示为：$CertA=\{V, Sn, A_I, ID_A, Issuer, K_p, T_A, other, [hash(V, Sn, A_I, ID_A, Issuer, K_p, T_A, other)]SignCA\}$。

其中，*V*是证书的版本号，*Sn*是该证书的序列号，A_I是对证书进行签名的算法的标识，ID_A是用户*A*的唯一标识符，K_p是用户*A*的公钥，*Issuer*是证书发布者的唯一标识符，T_A指出证书有效期，*other*是指其他信息，如证书扩展项等，*hash*()是单向散列函数，如MD5、SHA等，[]*SignCA*是(Certificate Authority，CA)对[·]中的数据的签名值，可以使用CA的公钥进行验证，而CA的公钥算法可以使用RSA或椭圆曲线算法，采用RSA算法作为公钥算法[3]。CA产生自签名证书，成为安全域中的信任根。

用户证书的申请过程为：(1) 用户产生自己的RSA公钥、私钥对，将公钥$P=(e, n)$以及其他身份信息提交给注册机构(Registration Authority，RA)；(2) 注册机构审核用户的身份信息，通过后继续，否则停止；(3) CA根据用户身份信息和其公钥*P*生成用户证书，并将该证书返回给RA。同时将该证书发布到目录服务器上；(4) RA通过在线或离线的方式将证书返回给用户。

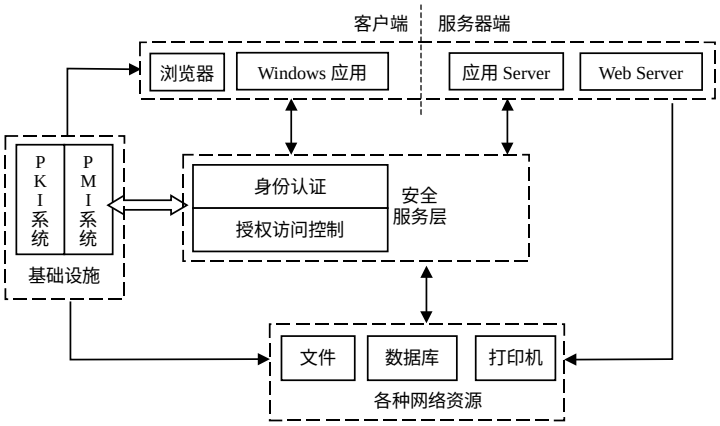

图1 基于证书的单点访问模型

1.2 基于AC的授权访问模型

公钥证书只是将用户身份和公钥绑定在一起，但并不能得出该用户拥有那些权限。这样的工作由属性证书AC来完成。属性证书的目的是将实体与其享有的权限绑定，主要用途是访问控制。属性证书的签发和管理与公钥证书相似，主要区别是证书格式。其证书格式为：$CertA=\{V, Sn, A_I, ID_A, Issuer, Authorization, T_A, other, [hash(V, Sn, A_I, ID_A, Issuer, Authorization, T_A, other)]SignAA\}$。与CA公钥证书相比，它不包含公钥，而包含一个授权属性(Authorization)，且证书由授权机构(Attribute Authority，AA)(或Source of Authority，SOA)签名。在授权模型中，采用基本的PMI模型[2]，如图2所示。

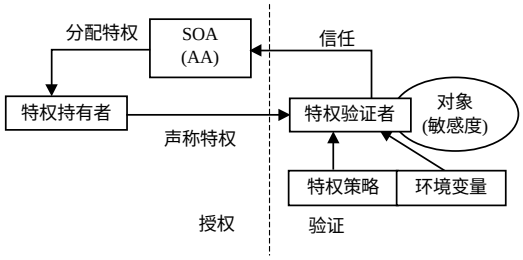

图2 授权模型及验证

特权验证者在得到用户属性证书后根据以下4点来判断用户是否有权访问某一对象：(1) 用户权限：包含在属性证书中，它决定了授权机构给用户授予的权限；(2) 特权策略：即对象的访问策略；(3) 环境变量：验证者进行权限判断时所需要的环境变量，如时间等；(4) 对象及其操作方法的敏感度：反应了资源的属性，如文档密级等。

采用PRBAC的访问控制模式[4]，对于属性证书中的*Authorization*属性，定义为：用户的安全标签，包含3个子项，安全策略、安全级别、安全类型。该字段可以表示为*Authorization* = {*Sec-policy -ID*，*Sec-level*，*Sec-type*}，其中，*Sec-policy-ID*表示安全策略号，对应于安全策略文件中的某个安全策略；*Sec-level*表示信

息的敏感程度，如机密、秘密、绝密等；*Sec-type*表示敏感信息的访问权限类型，如限制性、许可型等。

1.3　模型的讨论

采用基于证书的单点访问模型。在这个模型中使用PKI和PMI技术为用户提供统一的身份认证和授权管理服务。在系统实现上，通过在应用和资源之间增加一个安全服务中间件来实现。首先，在用户和服务器之间完成身份认证，并建立一个安全关联(如会话密钥、登录凭证等)，安全关联用来保证后续的安全通信；其次，在进行权限判定时，采用“拉”模式，即：用户访问资源时，只需提交在完成身份认证后得到的安全关联就可以，特权验证者根据安全关联中的用户信息到AA上去找到用户的属性证书(即将属性证书从AA“拉”到特权验证者)。采用“拉”方式对现有协议和应用不需要多大的修改，便于系统的使用和部署。

安全域中所有客户端和服务器之间的身份验证和授权验证的管理完全由安全层中间件来完成，从而实现了单点访问。

对于名字空间，在该模型中，不同的用户用*UserID*来区分，对于同一用户的属性证书和公钥证书中的主题名(Subject Name)分别由*UserID*+“*A*”和*UserID*+“*PK*”表示，而两种证书中的证书序列号(Certificate Serial Number)是由各自的发证机构在自己的范围内唯一定义的。因此，通过*UserID*+“*A*”和*Sn*可以唯一确定一个属性证书，定义一个凭证为：*Cred*={*UserID*+“*A*”、*Sn*}。在用户申请属性证书的同时，也产生一个与之对应的凭证(credential)，因而凭证可以看成是属性证书的缩影。属性证书、公钥证书和属性证书凭证的关系如图3所示。

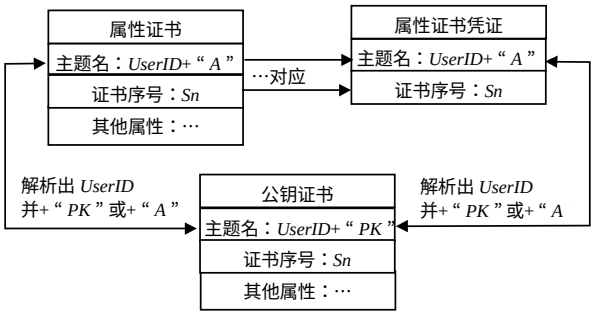

图3　属性证书及其凭证与公钥证书之间的关系

2　模型的通信协议

在身份认证阶段，采用了SSL的思想[5]。即，客户端和服务器通过公钥证书完成双向身份认证，并采用Diffie_Hellman密钥交换协议来产生会话密钥[6]，以后的通信就建立在这个会话密钥上。通信过程如图4所示。

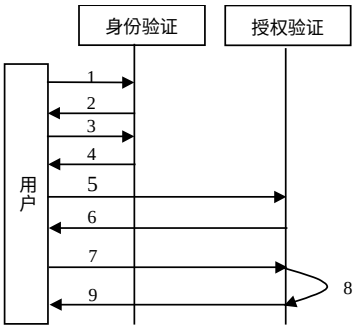

图4　身份认证和资源访问的通信协议

(1) {*Client_Hello*}

客户端向服务器发起身份认证请求。

(2) {*Server_Hello* ,*Cert_Server*, *Req_Client_Cert*}

服务器向客户端发送响应信息。

(3) {*Cert_Client*，$g^x(mod\ p)$}$_{Kps}$

客户端向服务器发送自己的公钥证书和Diffie_Hellman密钥交换协议所需要的随机秘密值。

(4) {*Cred_List*，$g^y(mod\ p)$} $_{Kpc}$

服务器验证客户的公钥证书，并向客户端发送该用户的凭证列表和Diffie_Hellman密钥交换协议所需要的随机秘密值。

(5) {*Req _Resouce*，*UserID*}$_K$

客户端向服务器请求资源，包括资源名和用户*ID*。

(6) {*Req _Cred*，*Ns*} $_K$

服务器向客户端发送用户凭证列表请求和临时值*Ns*，他们都用会话密钥加密。

(7) $\{Cred\_List, Nc, Ns\}_K$

客户端向服务器发送凭证列表，并用会话密钥加密他们。

(8) “ 拉 ” 用户属性证书

服务器根据凭证从目录服务器中 “ 拉 ” 回客户的属性证书，并验证用户的授权。

(9) $\{Nc, Resouce\}_K$

服务器验证过了用户授权后，将客户请求的资源和用户产生的临时值*Nc*返回给客户。

3 模型安全性分析

3.1 信息的保密性

本模型以公钥体制为基础。以RSA为例，密钥长度为1 024 bit时，需要1 012 MIPS(100 万次/s运算的计算机工作一年)年才能解开,而公钥证书根据PKI管理策略是要定期更新的,这个更新时间不会长于1 012 MIPS年。会话密钥的产生是基于Diffie_Hellman密钥交换协议计算的。它的安全性依赖于有限域上的离散对数问题，对于很大的素数，其求解的难度与RSA相似，同时，由于会话密钥只是在用户访问资源时产生，每次会话都会产生不同的会话密钥，要在这么短的时间内破解密码是非常困难的。

3.2 协议安全性

除了密码算法分身的安全性外，也考虑了通信协议的安全性。因为攻击者有时可以利用安全协议的漏洞在不攻破密码算法的情况下破坏安全协议。在协议中，通过设定凭证列表的有效时间，对列表信息签名等措施，防止非法使用该凭证来访问资源。在访问资源的过程中，通过在消息包中增加一个临时值以保证该消息的新鲜性，从而可以防止重放攻击。

此外，对协议过程引入了 “ 错误－停止 ” 的设计概念[7]，即，如果协议参与方发现了错误的协议消息(包括格式和内容)，就立即停止协议。这样可以比较有效地防止拒绝服务攻击(Denial of Service，DoS)。

4 结 束 语

本文针对现有单点登录系统的不足，提出了自己的单点访问模型。该模型结合PKI和PMI的先进思想，通过公钥证书和属性证书完成统一的身份认证和授权管理。其中，采用PRBAC作为访问控制模型，使用“拉”的方式获得用户属性证书。在此基础上，设计了安全通信协议，并对整个模型的安全性做了简单的安全性分析。

参 考 文 献

[1] ITU-T Recommendation X.509. Information technology-open system interconnectionthe directory: authentication sramew[S]. 1997.

[2] ITU-T Recommendation X.509, Information technology-open system interconnectionthe directory: public-key and attribute certificate frameworks[S]. 2000.

[3] Rivest R L, Shamir A, Adleman L M. A method for obtaining digital signatures and public-key cryptosystems[J]. Communications of the ACM, 1978, 21(2):120-126.

[4] CMB-04. 02. 029, 0N636216, Revision CSDN. 801: Access control concept and mechanisms[S]. 1999.

[5] Freier A O, Karlton P, Kocher P C. Version 3.0, the SSL protocol [S]. 1996.

[6] Diffie W, Hellman M E. New direction in cryptography [J]. IEEE Transcations on Information Theory, 1976, IT-22(6): 109-112.

[7] Gong L, Paul S. Fail-stop protocol: an approach to designing secure protocols[J]. Proceedings of IFIP, 1995, DCCA-5: 79-100.

编 辑 孙晓丹