Kerberos 协议在单点登录系统中的改进及应用①

陈 丽，于东海

(黄河科技学院 计算机科学系，郑州 450063)

摘 要：深入分析了目前国内外流行的单点登录模型，探讨了对称密钥下 Kerberos 协议的局限性，然后结合公钥技术，对 Kerberos 认证协议进行改进。设计并实现了基于代理的单点登录子系统，使用二次票据方法保证了单点登录系统的安全性。该方案在可实施性、运行可靠性以及管理等方面都有较好的性能。

关键词：单点登录；Kerberos 协议；身份认证；代理；票据

Improvement and Application of Kerberos Protocol to Single Sign-On System

CHEN Li, YU Dong-Hai

(Department of Computer Science, HUANGHE Science and Technology College, Zhengzhou 450063, China)

Abstract: This paper researches existing popular Single Sign-On model and Kerberos authentication protocol, and discusses the limits of Kerberos protocol in symmetric key technology, improves Kerberos protocol with the public key technology. Then it designs and realizes the agent-based single sign-on system, and by the second ticket method, ensures the safety of the SSO system. Meanwhile, the program has better performance in implementation, operating and management.

Keywords: SSO; Kerberos protocol; identify authentication protocols; agent; ticket

随着计算机和网络应用的快速发展，计算机用户每天都要登录到多个不同的应用系统中[1]。用户将不得不按照各个应用系统的要求分别登录进入相应的系统。这需要管理员针对不同的应用系统和用户设置登录凭证、维护管理各个系统的用户信息库，不但增加了工作量同时也存在安全隐患，将导致以下问题：

(1) 如每个系统都开发各自的身份认证系统将造成资源的浪费，消耗开发成本，并延缓开发进度；

(2) 多个身份认证系统会增加整个系统的管理工作成本；

(3) 用户需要记忆多个帐户和口令，使用极为不便。

(4) 无法实现统一认证和授权，多个身份认证系统使安全策略必须逐个在不同的系统内进行设置，因此修改策略的速度可能跟不上策略的变化；

(5) 无法统一分析用户的应用行为。

解决上述问题的方案是采用单点登录技术(single sign-on，SSO)[2]。用户只需一次身份验证，就可对所有被授权的应用系统或资源进行无缝的访问，从而提高工作效率，降低操作成本，提高系统的整体安全性。

1 SSO模型

当前流行的 SSO 模型主要有三种[3]：基于 Broker 的 SSO 模型；基于 Agent 的 SSO 模型；基于 Gateway 的 SSO 模型。三种模型中，通过分析基于 Agent 的 SSO 模型系统扩展性好，方便系统增加新的服务。在此模型中，应用服务器在认证代理服务器注册后，即可为客户提供新服务。与基于认证网关的模型相比，此模型减小了认证服务器的负担，经过身份认证后，客户机和服务器交互时不再经过认证服务器，而是二者直

① 收稿时间:2010-05-03;收到修改稿时间:2010-06-21

接通信，没有了认证网关模型中由于认证网关而带来的瓶颈问题。配置相对灵活，此模型中认证代理服务器和应用服务器可分布在互联网中的不同地方，而不必集中在一起。

因此，选择基于 Agent 的 SSO 模型构建单点登录认证系统。

2 SSO协议的设计

现有的单点登录系统多基于 Kerberos 协议实现。Kerberos 采用对称密钥技术给协议的应用带来一定局限性。本文对 SSO 系统认证协议的设计，充分吸取了 Kerberos 协议的设计思想，并通过分析该协议的缺点，设计了适用于本系统的认证协议。

Kerberos 协议在用户登录时，利用 AS 验证用户身份并发放票据。用户使用该票据获得 TGS 的授权访问其它任何一个服务，这样，在访问各应用服务时，用户只需进行一次身份认证，实现了单点登录，即 SSO。

关于 Kerberos 协议的基本原理不再详细叙述。通过分析 Kerberos 协议存在以下局限性：

(1) 认证票据的正确性是基于网络中所有的时钟保持同步，如果主机的时间发生错误，则原来的认证票据就是可能被替换的。因为大多数网络的时间协议是不安全的，所以，分布式计算机系统中这将导致极为严重的问题。

(2) 原有的认证服务可能被存储或替换，虽然时间戳是专门用于防止重放攻击的，但在票据的有效时间内仍然可能奏效，假设在一个 Kerberos 认证域内的全部时钟均保持同步，收到消息的时间在规定的范围内(假定规定为 3 分钟)，就认为该消息是新的。而事实上，攻击者可以事先把伪造的消息准备好，一旦得到票据就马上发出伪造的票据，在这 3 分钟内是难以检查出来的。

(3) Kerberos 防止口令猜测攻击的能力很弱，攻击者通过长期监听可以收集大量的票据，经过计算和密钥分析进行口令猜测。4．随着用户数增加,密钥管理较复杂。Kerberos 拥有每个用户的口令字的散列值,AS 与 TGS 负责用户间通信密钥的分配。当 N 个用户想同时通信时,仍需 N*(N-1)/2 个密钥。

从对 Kerberos 的局限性分析可以看出，其很多缺陷均是由于采用对称密钥技术造成的。如果能将公钥技术有机地融合到 Kerberos 中去，便能提高 Kerberos 协议的安全性。从将来的发展来看，将公钥系统结合进现有的系统中去也是一种趋势。基于这种考虑，并结合系统的实际情况，文中设计了一个适合于本系统的认证协议。在此协议中使用代表用户访问次数的“共享变量”N 来防止重放，图 1 是对本协议消息流程的描述。

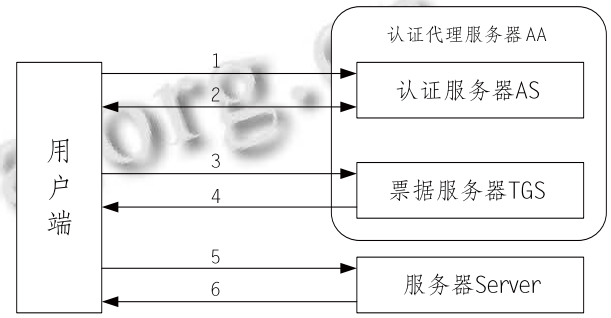

图 1 本系统协议消息流程图

改进后的协议如下：

①票据请求（Ticket_req），C->AS:C,S,N,CertC

② 票 据 发 放 (Ticket_rep) ， AS->C:TGT,((Kct,N) SignAS)EncCp

TGT=((C,AS,Kct,N,lifetime)SignAS)EncTGTp

③TGS 票据请求（Ticket_req），C->TGS:TGT,（N, seq）Kct

④TGS 票据发放(Ticket_rep)，TGS->C:TS,((Kcs,N) SignTGT)EncCp

TS=((C，S，Kcs，N，lifetime) SignTGT) EncCs

⑤服务请求(S_req)，C->S:Ts,(N,ra,seq,opinion)Kcs

⑥服务器身份认证(S_rep)， S->C:(ra)Kcs(可选由 opinion 决定)

符号表示说明：

AA：认证代理服务器(Authentication Agent)；

C：用户端(Client)唯一标识；

S：应用服务器(Server)唯一标识；

Xs：X 的私钥；

Xp：X 的公钥；

CertC：C 的证书序列号；

(Info)SignXs：用 X 的私钥对信息 Info 签名；

(Info)EncXp：用 X 的公钥对 Info 加密；

(Info)K：用对称密钥 K 对 Info 加密；

TGT：认证代理 AS 为用户 C 访问服务器 TGS 而颁发的票据；

TS: TGS 为用户 C 颁发的访问服务器 S 的票据；

Nc：用户端 C 保存的一个整数，记录用户访问次数；

Ns：应用服务器 S 所保存的用户访问次数，用于防止重放；

N：协议消息中使用的用于防止重放的整数，其值由 Nc 决定，在此称之为“同步变量”。

Lifetime：票据生存时间，(从服务器第一次接收服务请求后开始计时)；

ra：随机会话密钥；

seq：服务请求序列号；

opinion：用户是否要求对服务器验证，1：要求，0：不要求。

在 Kerberos 中，采用了发放票据的方法实现了权限的集中管理，本系统中也采用用户向认证代理 AA 证实自己的身份来取得票据。此时，认证代理便可以根据用户的权限的情况来决定是否发给用户票据。当应用服务器 S 取得服务请求后，S 应该可以验证票据的发放者和票据的持有者的身份。

由于采用了公钥体制，用户的身份已经包含在证书中了；利用签名，则可以保证消息来源的真实性、完整性和不可否认性；利用接收方的公钥加密，即可以保证只有接收方才能将数据解开，同时也能保证消息的完整性和机密性；使用各方所拥有的本地所保存的同步变量 N，就可以保证消息的新鲜性。

在规模较大，网络中各实体比较分散的网络环境下，实现系统各部分的时间同步可以说非常困难。本系统利用了同步变量 N 来替换 Kerberos 中的时间戳。对于同步变量 N，其应为位数足够大的整数，在初始使用时，由于客户端 C、服务器 S 本地均没有变量 N，则认为其值为 null，null 小于任何一个整数。当用户端第一次进行传输时需要取 N=Nc+1，即 N=null+1，这时可以取一个适当范围内的随机数，做为第一次会话中使用的 N。当应用服务器 S 收到 N 后，便可以为此用户设置本地变量 N 的值。

在上述协议中可以引入票据生存期(lifetime)参量。

在第二和第四步中，认证代理 AA 在票据中加入了票据的生存期 lifetime，说明此票据的生存时间。在生存期内用户能够重复使用票据，而不必再向认证代理 AA 提出申请。对于票据生存时间，由认证代理 AA 在发放票据的时候给出。

在第五步中，用户 C 除了向应用服务器 S 提交最初协议中的内容外，还加入了 ra 和 seq，ra 为 C 生成的一个随机数。在最初的协议中，使用 Kcs 做为会话密钥，在这里使用 ra 做为二者会话密钥，因为用户 C 在票据的生存期内可以重复访问服务器 S，出于安全性考虑，每次会话都使用不同的会话密钥 ra。为了防止在票据的生存期内攻击者对服务器请求消息的重放,协议中引入了请求的序列号 seq。seq 从 1 开始，每次加一。

对于服务器 S，除了保留同步变量 N 外，还要保存用户票据的到期时间 Texp，以及用户最后一次服务请求的序号 seq。

当服务器 S 收到用户的请求服务消息后，取得票据中的 N 有可能出现以下三种情况：

(1) N>Ns，说明此请求使用的是新票据，在对请求的所有验证通过后，则使 Ns=N，设置过期时间 Texp=Tnow+lifetime，同时使本地所保存的 seq 为此次请求的中的 seq；

(2) N=Ns，说明此请求为重新登录请求，此时检查 Tnow>Texp，看票据是否过期，在验证票据的发放者和持有者后，还要检查请求中的序列号 seq 是否大于本地所保存的上次请求的序列号，如果成立，则将 seq 更新为此次的请求的序列号，否则认为此消息为重放消息。

(3) 若 N<Ns，则认为此消息为重放消息，将其抛弃。

若 opinion=1，则用户要求对服务器验证，此时服务器 S 需要返回消息 6，若 opinion=0 则服务器不需要返回用户对自己的认证消息 6。

在第六步中，服务器将 ra 加密后返回给用户 C，由于 ra 的随机性，所以攻击者无法冒充，也无法进行重放，这样就验证了此消息的真实性和新鲜性。

3 安全性分析

分析此协议的安全性时，我们主要从协议的逻辑性方面来分析，因为像拒绝服务等攻击方式，应该由防火墙或入侵检测系统来阻止。这里我们主要采用了 BAN 逻辑分析协议的安全性，BAN 逻辑是基于知识

和信仰的形式逻辑分析方法，它是通过认证协议运行过程中消息的接收和发送，来从最初的信仰逐渐发展为协议运行要达到的目的主体的最终信仰。由于协议证明篇幅较长辑方法来分析协议的安全性，由于篇幅较长，这里不再给出具体的证明过程。

4 SSO系统的实现

上述对 SSO 系统的设计主要是在高速公路工程信息管理系统(HighWay Project Information Management System，简称 HWPIMS)中应用。在本系统中包括多个应用子系统，如：支付系统、信息上报系统、审批系统、信息查询系统等，同时用户众多，人员结构复杂，这就给运行在网络环境下的系统提出了较高安全性要求。因此在 HWPIMS 系统中提供单点登录功能，用户可以无缝地访问信息平台中所有授权的应用系统，同时提高了系统的整体安全性。整个系统的运作主要是基于 Windows 平台，认证代理服务器和应用服务器程序的运行平台为 Windows 2000，而客户端主要使用的是 Pocket PC，操作系统为 Windows 2000。开发工具主要是 VC，密码函数库选用的是微软的 CryptoAPI。

单点登录子系统主要负责用户的登录与认证，也为应用服务器提供用户的一些基本权限信息。系统结构如图 2 所示。主要包括认证代理 AA 及证书颁发机构 CA。

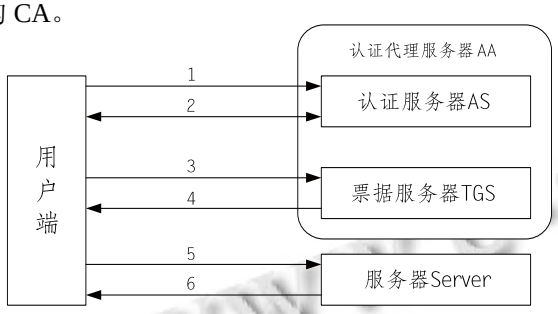

图 2 认证系统结构图

证书分颁发机构(CA)：或称认证中心，主要负责对整个系统中各实体身份进行统一管理，能够生成证书与制定证书安全策略，同时还要负责对证书库的管理和维护，从而完成整个证书的生命周期管理。另外，证书中心还需要提供对私钥的备份与恢复功能。

认证代理(AA)：认证代理作为专门用于提供认证服务的服务器，主要完成用户访问应用服务的授权，负责对用户的合法性及其级别进行检查，看其是否满足所要申请的服务，然后向合法用户发放票据，并规定票据的生命期等。

在本系统中使用二次票据法来增强系统安全性。用户端申请服务票据时，用户需要分别与 AS 和 TGS 进行交互，在这其中用户使用了两张票据，一张用来访问 TGS，一张用来访问所请求的服务，而且每个票据都有其生命期。将认证代理服务器分为两部分，一部分为 AS，另一部分为 TGS，如图 3 所示。当用户申请服务时，先与 AS 交互，得到访问 TGS 的票据，用户可以在较长一段时间内，利用此票据来向 TGS 提出申请其它服务。这样，用户输入用户口令的操作只在向 AS 提出申请访问 TGS 的票据时使用，在此票据生存期内，用户申请其它服务，用 AS 颁发的票据向 TGS 提出申请即可，而不用再利用自己的私钥。

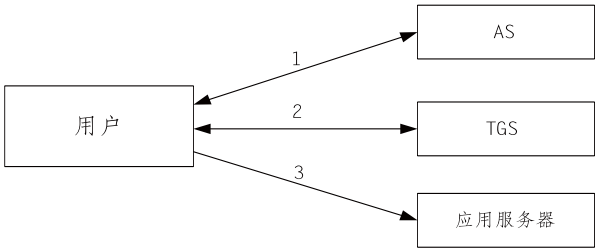

图 3 “二次票据”方法

在会话期间，非法攻击者有可能会盗取用于向 TGS 申请服务的票据，但由于此票据有生命期，当票据过期后，攻击者再进行进一步的破坏活动，因此可减小由此带来的损失。

4.1 SSO 流程

下图描述了未认证的用户访问应用程序的过程，认证服务器分为 AS 与 TGS 两部分。具体描述如图 4 所示。

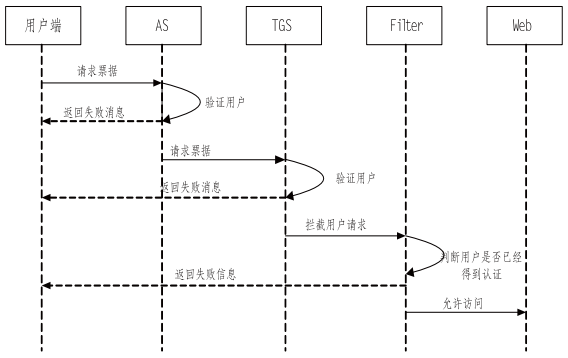

图 4 SSO 流程序列图

$R_n(u, j, t)$

(1) 用户首先到第一个认证代理服务器 AS 请求访问 TGS 的票据，如用户验证通过，则到第二个认证代理服务器请求访问应用服务所需要的票据 TS，否则，返回失败信息。

(2) 用户通过认证则将请求转发到 Filter，Filter 拦截此请求，判断是否已经通过认证。如果用户未通过认证，返回失败信息。

(3) 如果用户已经通过认证，Filter 判断用户请求中有没有希望访问的业务应用程序地址信息，将请求转发到系统选择页面。

4.2 SSO 代理服务器

认证代理服务器主要负责用户访问的授权，为用户对应用服务器的请求颁发票据。图 5 是认证代理服务器的总体结构。

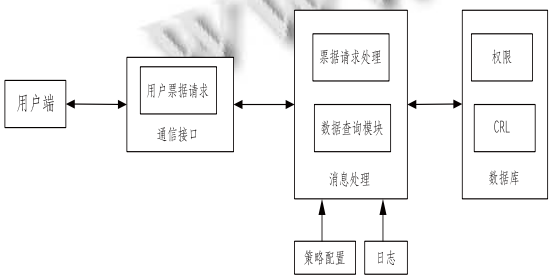

图 5 认证代理总体结构

认证代理主要功能为完成对用户的票据请求处理，根据用户等级及权限进行访问授权。通信接口部分负责接收发送与用户进行通信的消息；消息处理部分主要负责对用户消息进行具体处理；数据库部分主要负责用户 CRL 列表以及用户的权限的存放；策略配置模块主要负责系统运行时一些可变参数的指定；日志部分负责记录各种事件，并提供管理员查询接口。

5 总结

本文结合公钥技术，对 Kerberos 认证协议进行改进。设计并实现了基于代理的单点登录系统，使用二次票据方法保证了单点登录系统的安全性。在本系统中减少了用户多次登录认证；减轻了维护人员数据维护的困难；第三方应用容易实现单点登录。同时该系统在开发时就充分考虑了扩展性。Server 使用了平台无关的 Java 语言开发，整个系统的数据通信过程采用 SSL 协议。下一步的工作重点是将该系统进一步完善，由于在系统中使用加解密算法增加了系统的时耗，怎样降低系统负担，提高效率也是一个要解决重要问题。

参考文献

1 孙宝林,杨球,吴长海.RSA 公开密钥密码算法及其在信息交换中的应用.武汉理工大学学报(交通科学与工程版),2000,24(2):169－172.

2 李腊元,李春林.计算机网络技术.北京:国防工业出版社,2001.300.

3 肖攸安,李腊元.一类高效密钥协商方案的研究.武汉理工大学学报(交通科学与工程版),2003,27(6):758－761.