第34卷 第14期 计 算 机 工 程 2008年7月
Vol.34 No.14 Computer Engineering July 2008
·安全技术· 文章编号：1000—3428(2008)14—0152—03 文献标识码：A 中图分类号：TP319

一种单点登录协议的设计

李继勇，陶 然
(北京理工大学信息科学技术学院，北京 100081)

摘 要：Kerberos 单点登录协议存在口令猜测、重放攻击、缺乏认证等安全问题，该文以 Kerberos 协议为基础，设计一种新的单点登录协议，该协议修改了 Kerberos 协议的框架，引入一次性口令和授权服务机制，解决了 Kerberos 协议存在的问题，提供一种更安全、且扩展性强的单点登录协议。

关键词：单点登录；Kerberos 协议；一次性口令；协议安全性

Design of Single Sign-On Protocol

LI Ji-yong, TAO Ran
(Information Science Technology Institute, Beijing Institute of Technology, Beijing 100081)

【Abstract】 Kerberos protocol has some security problems, such as password guess, replay attack, and absent authentication. This paper designs a new Single Sign-On(SSO) protocol based on Kerberos. The SSO protocol modifies Kerberos's framework, which solves Kerberos problem by using one time password and authorization. The new SSO protocol is a more secure and expansibile protocol.

【Key words】 Single Sign-On(SSO); Kerberos protocol; one time password; protocol security

1 概述

单点登录(SSO)协议是实现多个应用统一认证的安全协议。典型的SSO系统采用精心设定的可信传递协议保证用户认证成功，并能够将这一信号传达到后续的应用中。Kerberos协议[1-2]是实现SSO的一个基本协议。文献[3]总结了Kerberos协议的一些安全问题。文献[4-5]引入公开密钥算法解决了部分协议的安全问题。

通过分析 Kerberos 协议的重要安全性问题及使用上的局限性问题，本文设计了一个新的单点登录协议，新协议采用一次性口令等技术成功解决了 Kerberos 协议的不足，为单点登录系统提供了一种可靠的实现手段。

2 Kerberos 协议

2.1 协议介绍

Kerberos 协议是麻省理工学院为 MIT Athena 项目开发的协议，它在一个分布式工作环境中适应于对多个服务的身份鉴别及单点登录，Kerberos 协议现在最新的版本的是 V5 版，Kerberos V5 版在 RFC 1510 中有说明。

Kerberos 完整的协议流程如下：

(1)C→AS：$ID_c||ID_{tgs}||TS1$。

(2)AS→C：$ID_c||Ticket_{tgs}||E_{Kc}[K_{c,tgs}||TS2||ID_{tgs}]$

其中，$Ticket_{tgs}=E_{Ktgs}[K_{c,tgs}||ID_c||TS2||Lifetime||ID_{tgs}]$。

(3)C→TGS：$ID_v||Ticket_{tgs}||Auth_{c,tgs}$

其中，$Auth_{c,tgs}=E_{Kc,tgs}[ID_c||TS3]$。

(4)TGS→C：$ID_c||Ticket_v||E_{Kc,tgs}[K_{c,v}||TS4||ID_{tgs}]$

其中，$Ticket_v=E_{Kv}[K_{c,v}||ID_c||ID_v||TS4||Lifetime]$。

(5)C→V：$Ticket_v||Auth_{c,v}$

其中，$Auth_{c,v}=E_{Kc,v}[ID_c||TS5]$。

(6)V→C：$E_{Kc,v}[TS5]$。

协议中用到的标识如表 1 所示。

表 1 Kerberos 协议标识

标识	描述
C	客户端
AS	认证服务
TGS	票据许可服务
V	应用服务
ID_c	客户端 ID 标识
ID_{tgs}	TGS 标识
TSx	时间戳
AD_c	客户端网络地址
Lifetime	票据有效期
K_{tgs}	AS 与 TGS 共享密钥
K_v	TGS 与 V 的共享密钥
$E_{xxx}[***]$	使用密钥 xxx 对***进行加密
$K_{c,v}$	C 与 V 共享的会话密钥
$K_{c,tgs}$	C 与 TGC 共享的会话密钥

2.2 协议分析

2.2.1 存在口令猜测攻击

在Kerberos协议的第 2 个数据包中会话密钥$K_{c,tgs}$是由K_c加密的，而K_c是由用户的口令衍生的密钥，如果口令强度不够，则很容易受到基于密码字典的攻击，这种攻击实际上是通过穷举可能的口令组合实现的。使用者为了便于记忆，口令往往较为简单，这就为字典攻击制造了机会。

2.2.2 客户端存储数据多、运算复杂

在Kerberos协议中，客户端必须保存 2 个重要的数据：一个是客户端与TGS的会话密钥$K_{c,tgs}$，另一个是客户端与应

作者简介：李继勇(1974－)，男，讲师、博士研究生，主研方向：信息安全与对抗；陶 然，教授、博士生导师

收稿日期：2007-08-20 **E-mail：**tigerlijiyong@263.net

用服务的会话密钥$K_{c,v}$，如果某个用户允许访问的应用服务有N个，那么客户端需要保存的会话密钥将有N+1 个。若要访问N个应用服务，客户端至少需要与认证服务及票据服务进行 2(N+1)次交互，过多的客户端与服务器的交互容易给攻击者获得更多可分析的数据。

2.2.3 容易受到重放攻击

在现有的一些关于Kerberos协议的分析资料中，很多都指出该协议容易受到重放攻击，但并没有详细分析是哪个流程容易受到攻击[3-5]。通过分析，Kerberos协议存在重放攻击的是步骤(1)、步骤(3)、步骤(5)3 个步骤。攻击者可以对步骤(1)中的内容进行重放，这时攻击者获得一个新的票据许可票据，但是由于攻击者不知道真实用户的口令，因此无法解密并获得客户与TGS之间的会话密钥，这时，攻击者后续的步骤无法进行，该重放攻击没有意义。同理，步骤(3)的重放攻击也没有意义。

如果攻击者重放步骤(5)的数据包，在时间戳的有效范围内，应用服务鉴别通过，这时攻击者忽略第 6 个数据包，攻击者将获得对应用的访问权限，重放攻击成功。因此，必须采取措施防止步骤(5)的重放攻击。

2.2.4 不能对客户及服务进行认证

Kerberos 协议还有一个突出的问题是在用户和认证服务器交互的过程中，服务器没有对用户进行认证，用户也没有确认服务器的真实性，作为一个身份认证协议是不可取的。

2.2.5 缺乏授权及访问控制机制

在现有的 Kerberos 协议框架下扩充用户授权方面的信息是非常困难的，如果强制在服务许可票据中加入用户的授权信息，一方面要求 TGS 服务具有维护服务权限信息的能力；另一方面由于权限的频繁变更，要求用户频繁获取服务许可票据。这种协议的变化已经违背了 Kerberos 的最初设计原则，并且在现有框架下使用也不方便。

3 一种新的单点登录协议

3.1 协议设计

3.1.1 架构设计

在新的 SSO 协议中，取消了 TGS 的概念，取代的是授权服务(ARS)。新协议结构如图 1 所示。

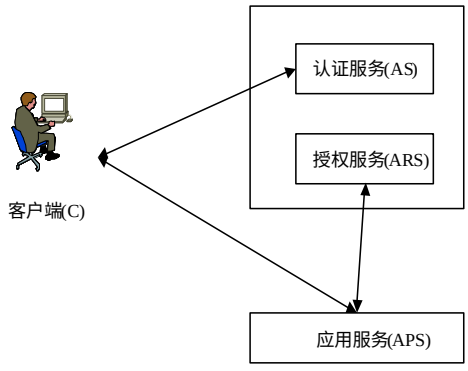

图 1 新协议架构

3.1.2 初始化阶段

初始化阶段在认证服务上产生一对公开密钥(p, r)，其中，p 为公钥，r 为私钥，算法可以是 RSA 或其他符合要求的公开密钥算法。

3.1.3 注册阶段

新的 SSO 协议注册过程分为以下 3 步：

(1)客户端注册

每一个客户端需要在认证服务中注册标识ID_c，并在认证服务中注册一个动态口令K_{c1}，口令K_{c1}由认证服务随机生成，在客户端K_{c1}由用户的静态口令pwd加密保存。认证服务器中需要存储一对动态口令，称为本次动态口令及上次动态口令，在注册时本次动态口令记为K_{c1}，上次动态口令为空。认证服务器同时将公钥p分发给每一个注册的客户端。

(2)授权服务注册

认证服务为授权服务分发一个密钥K_{ars}，该密钥只有认证服务和授权服务知道，用来保护认证令牌。

(3)应用服务注册

每个应用服务需要在认证服务中注册应用的标识ID_v，并获得认证服务分发的密钥K_{aps}，该密钥同时由认证服务分发给授权服务，用来确认应用服务的身份及建立授权服务与认证服务之间的安全会话。另外，授权服务需要注册每个应用用户的授权信息，并建立用户授权表。

3.1.4 协议流程

新 SSO 协议用到的标识说明见表 2。

表 2 新协议标识表

标识	描述
C	客户端
AS	认证服务
ARS	授权服务
APS	应用服务
ID_c	客户端 ID 标识
ID_{ars}	ARS 服务 ID 标识
K_{c1}	本次动态口令
K_{c2}	上次动态口令
ST	认证令牌有效起讫时间
ET	认证令牌有效终止时间
K_{ars}	AS 与 ARS 共享密钥
K_{aps}	ARS 与 APS 共享密钥
E_{xxx}[***]	使用密钥 xxx 对***进行加密
RS1, RS2	随机数
FLAGS	ARS 返回给 APS 的授权信息

新 SSO 协议的流程如下：

第 1 部分：认证服务鉴别

(1)C→AS：$ID_c||ID_{ars}||E_p[ID_c||K_{c1}]$。

(2)AS→C：ID_c $||Token_{ars}||E_{Kc1}[E_{Kc2}||K_{c,ars}||ID_{ars}]$

其中，$Token_{ars}=E_{Kars}[K_{c,ars}||ID_c||ID_{tgs}||ST||ET]$。

第 2 部分：应用服务鉴别、授权

(3)APS→C：$ID_{aps}||$ RS1。

(4)C→APS：ID_c || ID_{aps} || RS2||$Token_{ars}$ ||$Auth_{c,aps}$

其中，$Auth_{c,aps}=E_{Kc,ars}[ID_c||$ ID_{aps} ||RS1||RS2]。

(5)APS→ARS：ID_c || ID_{aps} || $Token_{ars}$ ||$Auth_{c,aps}||E_{Kaps}$ [RS1|| RS2||ID_{aps}]。

(6)ARS→APS：$E_{Kc,ars}$[RS2]||E_{Kaps}[FLAGS||RS]。

(7)APS→C：$E_{Kc,ars}$[RS2]。

协议说明：

第 1 部分客户端向认证服务发送请求信息，$E_p[ID_c||K_{c1}]$为认证服务公钥加密的客户端标识及本次动态口令，为了获得K_{c1}，在发送步骤(1)中数据包之前需要客户端用户输入正确的静态口令pwd来解密并获得K_{c1}。

认证服务收到步骤(1)中数据包后，使用私钥解密$E_p[ID_c||K_{c1}]$，获得ID_c及K_{c1}，利用认证服务存储的本次动态口

令和上次动态口令进行比较，如果K_{c1}与本次动态口令和上次动态口令都不相同，则认证失败。如果K_{c1}与本次动态口令相同，认证服务将产生新的动态口令K_{c2}，并将上次动态口令置为K_{c1}，本次动态口令置为K_{c2}。如果K_{c1}与上次动态口令相同，认证服务将产生新的动态口令K_{c2}，并将本次动态口令置为K_{c2}，上次动态口令保持不变。这样做是为了防止步骤(2)中数据包在发送中丢失。事实上，如果步骤(2)中数据包丢失，最终结果是客户端不能获得K_{c2}，那么在下次认证时客户端发送的依然是上次的K_{c1}，通过上面的描述，可以知道该K_{c1}能够匹配服务器中的上次动态口令，因此认证也能够成功。而一旦步骤(2)中数据包被客户端成功接收，客户端将会用K_{c2}代替K_{c1}，并将K_{c2}用pwd加密保存。这时，客户端再次认证时将会与服务器的本次动态口令匹配。也就是说，在认证失败的情况下，协议能够自动调整而保证下一次的认证同步。

认证通过后，认证服务还产生会话密钥$K_{c,ars}$，由K_{c1}加密并返回，同时在认证令牌中也包含$K_{c,ars}$，并用E_{Kars}加密，保证只有授权服务能够解密。

当客户访问应用服务时，应用服务将发送步骤(3)中数据包给客户，其中 RS1 是为了防止重放攻击。

客户收到应用服务的随机数，将认证令牌及$Auth_{c,aps}$发送给应用服务(见步骤(4)中数据包)，$Auth_{c,aps}$中包含RS1 是为了防止步骤(4)中数据包的重放。$Auth_{c,aps}$中的ID_c及 ID_{aps}是为了让授权服务确认是哪个客户端访问哪个应用服务。

当应用服务收到步骤(4)中数据包后，将发送步骤(5)中数据包给授权服务，其中ID_c 及ID_{aps}是为了授权服务与$Auth_{c,aps}$中数据进行比较，确认出示令牌的客户就是当初获得令牌的客户。$E_{Kaps}[RS1||RS2||ID_{aps}]$是用来确认应用服务身份的，其中的RS2 可防止应用服务冒充。事实上，一个冒充的应用服务可能试图伪造步骤(3)中数据包并发给用户，并将客户端发送的步骤(4)中数据包转发给授权服务，但是由于冒充的应用服务不知道K_{aps}，因此无法构造$E_{Kaps}[RS1||RS2||ID_{aps}]$，从而避免应用服务的假冒。

当授权服务收到应用服务的步骤(5)中数据包后，通过$Token_{ars}$获得会话密钥，并确认认证令牌的有效性，通过ID_c, ID_{aps}, $Auth_{c,aps}$及会话密钥确认客户就是拥有该令牌的客户。通过$E_{Kaps}[RS1||RS2||ID_{aps}]$确认应用服务的真实性。授权服务从步骤(5)中数据包中也能分析出确实是客户ID_c需要访问应用服务ID_{aps}。

授权服务解析步骤(5)中数据包后将步骤(6)中数据包发送给应用服务，其中，$E_{Kc,ars}[RS2]$将转发给客户端，使客户端相信该过程确实由授权服务参与，并且应用服务是真实的。FLAGS中携带的是用户ID_c访问应用服务的授权信息，该标识保留，便于扩展。通过步骤(3)到步骤(7)的过程也使应用服务相信客户端是真实的，因为只有通过授权服务校验客户端正确后，才会发送步骤(6)中数据包给应用服务，否则将会发送失败信息。

3.2 安全性分析

新协议有效地防止了口令猜测的攻击，这由以下 2 方面保证：一是网络上传送的步骤(2)中数据包使用动态口令加密，而不是用静态口令加密，该动态口令是由认证服务生成的，其复杂度可以由算法保证；二是动态口令是随时变化的，每次认证会话将产生一个新的动态口令，下一次的认证将用新的动态口令加密数据，该方式为攻击者分析加密数据带来了难度。

在新协议中拚弃了 TGS 服务的使用，避免了客户端由于访问多个应用而需要保存大量的服务许可票据及会话密钥的问题。新协议中客户端需要保存的数据仅仅是静态口令加密的动态口令数据、授权服务的会话密钥及认证令牌 3 个信息，这些信息的量很少，既可以存储在硬盘中，也可以存储在智能卡中。

新协议在步骤(3)中加入了应用服务向客户端发送的随机数，这实际上是一种挑战/响应的认证模式，该模式比使用时间戳有更大的好处，它不需要时间同步来维护时间的一致性，每一次认证的不同随机数保证攻击者不能实施重放攻击。

新协议中使用一对公私钥解决了客户端对服务器的认证问题。这是因为只有认证服务才能解密客户端发送的步骤(1)中数据包，并获得正确的动态口令，从而保证步骤(2)中数据包的正确构造。另一方面，新协议也实现了认证服务对客户端的认证，因为只有正确的客户端能够获得动态口令，并构造步骤(1)中数据包，当认证服务解密获得动态口令时，通过比较也实现了对客户端身份的鉴别。因此，该协议是一个双向认证协议。

新协议使认证令牌校验的工作由授权服务完成，应用服务起到一个过渡作用，同时在过渡中应用服务不但获得对客户端的认证，同时也获得对授权服务的认证，并取得授权服务的授权信息，这种模式在访问控制及授权需要细化的情况下更加易于扩展。

4 结束语

现有网络应用的多样性对身份鉴别提出了单点登录的需求，本文通过分析现有的典型单点登录方式，尤其是对支持单点登录的 Kerberos 协议的分析，总结出该协议的一些安全隐患及弱点，设计了一个新的单点登录协议。该协议不仅能够解决 Kerberos 协议中的口令猜测等安全问题，还能解决 Kerberos 协议中没有的授权等扩展性问题。

参考文献

[1] Stallings W. 密码编码学与网络安全: 原理与实践[M]. 2 版. 杨 明, 译. 北京: 电子工业出版社, 2001: 256-270.

[2] Schneier B. 应用密码学: 协议、算法与 C 源程序[M]. 吴世忠, 译. 北京: 机械工业出版社, 2000: 368-370.

[3] Bellovin S M, Merritt M. Limitations of the Kerberos Authentication Systems[J]. Computer Communication Review, 1990, 20(5): 119-132.

[4] 邓永江, 程转流. 一个改进的 Kerberos 认证协议设计与分析[J]. 福建电脑, 2006, (6): 134-136.

[5] Sirbu M A, Chuang J C I. Distributed Authentication in Kerberos Using Public Key Cryptography[C]//Proceedings of the 1997 Symposium on Network and Distributed System Security. Washington, D. C., USA: IEEE Computer Society, 1997.