计算机科学2001Vol.28№.10

基于PKI的Kerberos跨域认证协议的实现与分析

One PKI Based Kerberos Cross-Realm Authentication Protocol

顾文刚　程朝辉　荆金华　张根度

（复旦大学计算机科学系　上海200437）

Abstract This document describes one PKI based Kerberos cross-realm authentication protocol to provide a method for using public key cryptography in cross-realm authentication. The protocol described here specifies a way in which message exchanges are to be used to transport inter-realm secret keys protected by encryption under public keys certified as belonging to Kerberos KDCs.

Keywords PKI, Kerberos protocol, Cross-Realm Authentication

一、引言

认证是一种用于确认实际通信双方是否是对方所声称的人的一种技术。随着网络应用日益普及、电子商务的日渐发展，网上交易、网上支付等的安全性越来越受到人们的重视。而在这些过程中，确认通信双方的身份是十分重要的。

在传统的Kerberos协议中，用户根据职能、单位、需求等可以划分为不同的域，每个域中有各自的Kerberos服务器，域中用户通过该Kerberos服务器认证，以访问域中服务。Kerberos协议本身就支持跨域认证：在实现过程中，首先通过交换域间密钥将两个域中的票证发放服务器（TGS）互相注册为对方域中的用户（两个方向上的域间密钥往往不同），然后，当一个域中的注册用户希望访问外域服务时，他向本地TGS服务器发送一个请求，申请外域TGT，本地TGS服务器用域间密钥加密外域TGT发放给用户；用户向外域TGS服务器提交请求时，外域TGS服务器用域间密钥解密该TGT，确信该TGT是由用户本地TGS服务器签发后，发放服务票证，并在票证中指明用户是在其他域中认证的。在本文将描述的协议中，不需要将TGS服务器互相注册为其他域用户，而是利用PKI公钥证书来建立域间信任关系。

二、协议描述

第一阶段（步骤1、步骤2）是客户同本地认证服务器（AS）通信，获得TKT(C,L)（即TGT）的过程。

步骤1　C→AS：

IdC, RealmLTGS, IdLTGS, Times, Nonce(1)

步骤2　AS→C：

RealmC, IdC, TKT(C,L), E{Key_1, Nonce(1), Times, RealmLTGS, IdLTGS}Key_C

其中 TKT(C,L)＝RealmAS, IdAS, E{Key_1, RealmC, IdC, Temes}Key_LTGS

第二阶段（步骤3）是客户向本地票证发放服务器（LTGS）提交请求，申请到外域票证发放服务器（RTGS）的票证TKT(C,R)的过程。

步骤3　C→LTGS：

IdC, RealmRTGS, IdRTGS, Times, Nonce(2), TKT(C,L), Authenticator(1)

其中 Authenticator(1)＝E{RealmC, IdC, TS(1)}Key_1

第三阶段（步骤4、步骤5）是本地票证发放服务器同外域票证发放服务器通信，获得TKT(L,R)的过程。如果本地票证发放服务器已经拥有了票证TKT(L,R)且该票证还有效，那么步骤4、步骤5是可以省略的。

步骤4　LTGS→RTGS：

IdC, RealmRTGS, IdRTGS, Times, Nonce(3), Authenticator(2),

S{Authenticator(2)}PKeyPrivLTGS, Certificates(LTGS)

其中 Authenticator(2)＝TS(2), Checksum

步骤5　RTGS→LTGS：

RealmLTGS, IdLTGS, TKT(L,R), E{Key_Temp}PKeyPubLTGS,

顾文刚　硕士研究生，研究方向：计算机网络与分布式计算。程朝辉　硕士研究生，研究方向：计算机网络与分布式计算。荆金华　副教授，研究方向：计算机网络与分布式系统、网络安全和工程。张根度　教授，研究方向：计算机网络与分布式系统、网络安全和工程。

E{Key_3, Nonce(3), Certificates(RTGS), S{Key_3}PKeyPrivPTGS}Key_Temp

其中 TKT(L, P) = RealmRTGS, IdRTGS, E{Key_3, RealmLTGS, IdLTGS, Times}Key_RTGS

第四阶段是本地票证发放服务器向用户C发放票证TKT(C,R)的过程，TKT(C,R)中的加密部分不用外域票证发放服务器的密钥Key_RTGS加密，而是用由RGTS生成的域间密钥Key_3加密，并在票证扩展部分附上TKT(L,R)。

步骤6 LTGS→C:

RealmC, IdC, TKT(C,R), E{Key_2, Nonce(2), Times, RealmRTGS, IdRTGS}Key_1

TKT(C,R) = RealmLTGS, IdLTGS, E{Key_2, RealmC, IdC, Times}Key_3, TKT(L,R)

第五阶段是本地用户C向外域票证服务器请求并获得TKT(C,V)的过程。

步骤7 C→RTGS:

IdC, RealmRV, IdRV, Times, Nonce(4), TKT(C,R), Authenticator(3)

其中 Authenticator(3) = E{RealmC, IdC, TS(3)}Key_2

步骤8 RTGS→C:

ReamlC, IdC, TKT(C,V), E{Key_4, Nonce(4), Times, RealmRV, IdRV}Key_2

其中 TKT(C, V) = RealmRTGS, IdRTGS, E{Key_4, RealmC, IdC, Times}Key_RV

符号说明 C:用户；AS:认证服务器；LTGS:本地票证发放服务器；RTGS:外域票证发放服务器；RV:外域应用服务器；IdAS, RealmAS:AS的身份码和所在域；IdC, RealmC:C的用户身份码和所在域；IdRV, RealmRV:服务器RV的身份码和所在域；IdLTGS, RealmLTGS:LTGS的身份码和所在域；IdRTGS, RealmRTGS:RTGS的身份码和所在域；Key_C:用户C的密钥；Key_LTGS:LTGS的密钥；Key_RTGS:RTGS的密钥；Key_RV:RV的密钥；Key_Temp:RTGS生成的临时密钥；Key_1:C和LTGS的会话密钥；Key_2:C和RTGS的会话密钥；Key_3:RTGS生成的用于加密跨域票证的密钥，即域间密钥；Key_4:C和RV的会话密钥；Times:所请求的票证的开始、终止及续签等票证生存期信息；TS(1):时间戳1，用于抗重发攻击；Nonce(1):随机数1，用于抗重发攻击；TKT(C,L):AS颁发给用户C的到LTGS的票证，即TGT；TKT(L,R):RTGS颁发给LTGS的到RTGS的票证；TKT(C,R):LTGS颁发给用户C的到RTGS的票证；TKT(C,V):RV颁给C的票证；Authenticator(1):签别码1；PKeyPubLTGS, PKeyPrivLTGS:LTGS的PKI公钥和PKI私钥；PKeyPubRTGS, PKeyPrivRTGS:RTGS的PKI公钥和PKI私钥；E{Data}Key:表示用Key对Data加密；S{Data}Key:表示用Key对Data作数字签名；Certificates(LTGS):LTGS的PKI公钥证书；Certificates(RTGS):RTGS的PKI公钥证书；

三、协议分析

步骤1、2是用户通过Kerberos AS请求和应答获取TKT(C,L)的过程。用户C向本地KDC提交一个票证请求；AS在收到请求后，生成一个供用户C和LTGS共享的会话密钥Key_1，并从数据库中取出LTGS和用户C的密钥Key_LTGS和Key_C，用LTGS的密钥Key_LTGS加密Key_1及用户C的标识和票证生存期等其他信息产生TKT(C,L)，用用户C的密钥Key_C加Key_1及LTGS的标识，票证生存期等其他信息，同TKT(C,L)一起发给用户C。

在这个阶段，AS服务器并不对用户的身份作任何检验，而只是简单地用用户密钥加密会话密钥发放给用户；因为在网络上只有Kerberos服务器和用户C自己知道用户密钥Key_C，所以即使有其他人冒充用户C发了该请求，在得到应答后将由于无法解密得到会话密钥Key_1而无法使用TKT(C,L)。用户C可以通过验证随机数Nonce(1)来抵抗重发攻击，确保应答是对应刚才的请求的(随机数的选取应该在一定的时间范围内不会重复)。

步骤3是用户向本地TGS提交跨域票证请求的过程。用户C在收到上面的AS应答后，保存TKT(C,L)，用自己的密钥Key_C解密得到会话密钥Key_1；用Key_1加密时间戳等信息生成鉴别码Authenticator(1)，并把鉴别码、TKT(C,L)和跨域票证请求一起提交给LTGS。

LTGS在收到请求后，解密TKT(C,L)，从中得到Key_1；用Key_1解密Authenticator(1)以检验用户C的身份。如果解密后的信息有效，说明用户确实知道Key_1，也就说明用户确实拥有该票证，即用户确实是其所声称的用户C。通过检查时间戳TS(1)，LTGS可以检测重发攻击，因为Kerberos协议假定网络上所有机器的时间是基本同步的。

步骤4、5是LTGS和RTGS通信，获得TKT(L,R)的过程。和传统的Kerberos协议中不同，这里票证发放服务器不需要预先互相注册为对方域中用户。本地票证发放服务器LTGS的请求中通过PKI公钥证书Certificates(LTGS)证明自己的身份，也就是说，RTGS对LTGS的信任是建立在对公钥证书签发机构的信任之上的。

外域票证发放服务器RTGS在收到LTGS的请

求后，查看 LTGS 所提供的 PKI 公钥证书，以确保它是由自己所信任的机构签发的。然后，从 PKI 证书中取出 LTGS 的 PKI 公共密钥 PKeyPubLTGS，用它解密 Authenticator(2)的数字签名并与 Authenticator(2)比较，因为在网络上只有 LTGS 才知道其 PKI 私有密钥，所以如果比较相同，则可以认为通信的对方确实拥有该 PKI 证书，即确实是 LTGS，而不是其他人冒充的。通过检查 Authenticator(2)中的时间戳 TS(2)可以检测重发。检查校验和可以确保数据的完整性。

在确认了请求的有效性后，RTGS 向 LTGS 签发 TKT(L,R)。首先，它生成一个临时密钥 Key_Temp 和一个会话密钥 Key_3，用 PKI 公钥证书 Certificates(RTGS)说明自己的身份，用 PKI 私有密钥 PKeyPrivRTGS 对 Key_3作数字签名来证明对 PKI 公钥证书的拥有，用 RTGS 的密钥 Key_RTGS 加密 Key_3、LTGS 标识和票证生存期等信息产生 TKT(L,R)，然后把证书、签名和 Key_3、LTGS 的请求中包含的随机数等用临时密钥 Key_Temp 加密，再用本地票证发放服务器 LTGS 的 PKI 公共密钥 PKeyPubLTGS 对 Key_Temp 加密，把它们同 TKT(L,R)一起发送给本地票证发放服务器 LTGS(这里不是直接用 PKeyPubLTGS 代替 Key_Temp 加密 Key_3等信息是从效率上考虑，因为公钥算法的效率明显比传统的对称密钥算法低得多)。

因为 Key_Temp 有 LTGS 的公共密钥 PKeyPubLTGS 加密保护，除了 LTGS 外，其他人无法解密得到 Key_Temp，这样保证了由 Key_Temp 所加密的信息只有 LTGS 可以得到。LTGS 在收到 RTGS 的应答后，用其 PKI 私有密钥解密得到 Key_Temp，然后用 Key_Temp 解密得到 Key_3、RTGS 的 PKI 公钥证书、Key_3的数字签名和随机数 Nonce(3)。通过验证随机数 Nonce(3)可以检测重发；用从 RTGS 的 PKI 公钥证书中取出的 RTGS 的公共密钥 PKeyPubRTGS 解密 Key_3的数字签名并与 Key_3比较，可以验证 RTGS 的身份。

步骤6是 LTGS 向用户 C 发放跨域票证的过程。LTGS 生成一个供用户 C 和 RTGS 共享的会话密钥 Key_2，用 Key_3加密 Key_2及用户 C 的标识和票证生存期等其他信息产生 TKT(C,R)，并在 TKT(C,R)的扩展部分附上 TKT(L,R)；用会话密钥 Key_1加密 Key_2及 RTGS 的标识、票证生存期等其他信息，同 TKT(C,R)一起发回给用户 C。

用户 C 在收到该应答后，用会话密钥 Key_1解密可以得到 Key_2、Nonce(2)。因为只有 LTGS 和用户 C 知道会话密钥 Key_1，只有 LTGS 和 RTGS 知道 Key_3，其他人即使截获了该信息，也会因为没有 Key_1和 Key_3而无法解密得到 Key_2。通过验证 Nonce(2)，用户可以检测重发攻击。

步骤7、8是用户 C 向 RTGS 请求获得 TKT(C,V)的过程。用户 C 用 Key_2加密时间戳 TS(3)等信息构成鉴别码 Authenticator(3)，并把鉴别码、TKT(C,R)和票证请求一起提交给 RTGS。

RTGS 在收到请求后，用 RTGS 的密钥 Key_RTGS 解密 TKT(C,R)的扩展部分(即 TKT(L,R))，得到 Key_3(也就是说，RTGS 并不需要记录下与自己建立信任关系的每个域所对应的域间密钥，因为在请求中包含了该密钥)；然后用 Key_3解密 TKT(C,R)，得到 Key_2；用 Key_2解密鉴别码 Authenticator(3)。如果 Authenticator(3)解密出来的信息有效，说明用户确实拥有该 TKT(C,V)。通过验证时间戳 TS(3)，RTGS 可以检测重发攻击。

验证通过后，RTGS 生成一个供 RV 和用户 C 共享的会话密钥 Key_4，用 RV 的密钥 Key_RV 加密 Key_4及用户 C 的标识和票证生存期等其他信息产生 TKT(C,V)，用会话密钥 Key_2加密 Key_4及 RV 的标识、票证生存期等其他信息，同 TKT(C,V)一起发给用户 C。

自此，用户 C 获得了外域应用服务器的票证 TKT(C,V)。

与传统 Kerberos 协议相比较，上述协议在继承 Kerberos 认证协议安全性的同时，简化了对域间密钥的管理。Kerberos 协议中所说的跨域认证需要把各自的票证发放服务器注册为对方域中的用户，并交换域间密钥。域间密钥的交换过程，需要引入某些 IKE 协议的支持，一旦处理不当，将降低整个协议的安全性。而在上述协议中，域间密钥的交换将在公钥保护下进行，不需要另外引入 IKE 协议。远端票证发放服务器并不需要记录下所有与这通信的域所对应的域间密钥，因为解密 TKT(C,R)所需要的解密密钥包含在了 TKT(C,R)中的 TKT(L,R)中。

在实现过程中，上述协议需要在传统 Kerberos 服务器端作少许的改动，对于客户端和应用服务器端是透明的。

结束语 在上述协议中，因为要利用公钥算法，而在有的公钥系统中，允许使用强度比较弱的密钥(如512位长度的密钥)，所以在该协议中公钥的选择是非常重要的。弱密钥不仅会消耗不必要的资源和时间，还会降低整个系统的安全性。

参 考 文 献

1 RFC1510. The Kerberos Network Authentication Service (V5)
2 RFC2459. Internet X.509 Public Key Infrastructure Certificate and CRL Profile
3 RFC2510. Internet X.509 Public Key Infrastructure Certificate Management Protocols
4 RFC1511. Common Authentication Technology Overview