计算机密码学理论与应用

非公钥类身份认证协议：Kerberos

*Stallings*教程15.2~15.3

$$ed = 1 \bmod \varphi(N)$$

$$Y = M^e \bmod N$$

$$M = Y^d \bmod N$$

非公钥类身份认证协议：Kerberos V

Stallings 15.3节

- 因特网标准
- RFC#4120 (2005)
- RFC#1510 (1997)
- *Stallings*教程：15.3节

Network Working Group
Request for Comments: 4120
Obsoletes: 1510
Category: Standards Track

C. Neuman
USC-ISI
T. Yu
S. Hartman
K. Raeburn
MIT
July 2005

The Kerberos Network Authentication Service (V5)

Status of This Memo

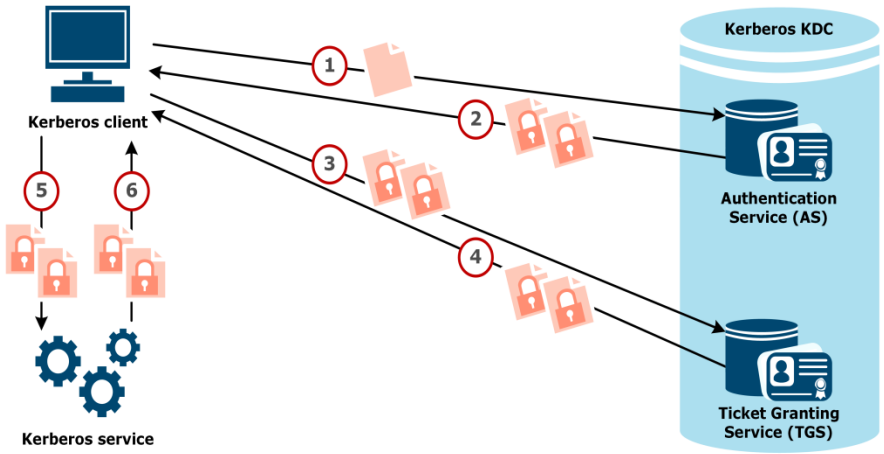
This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

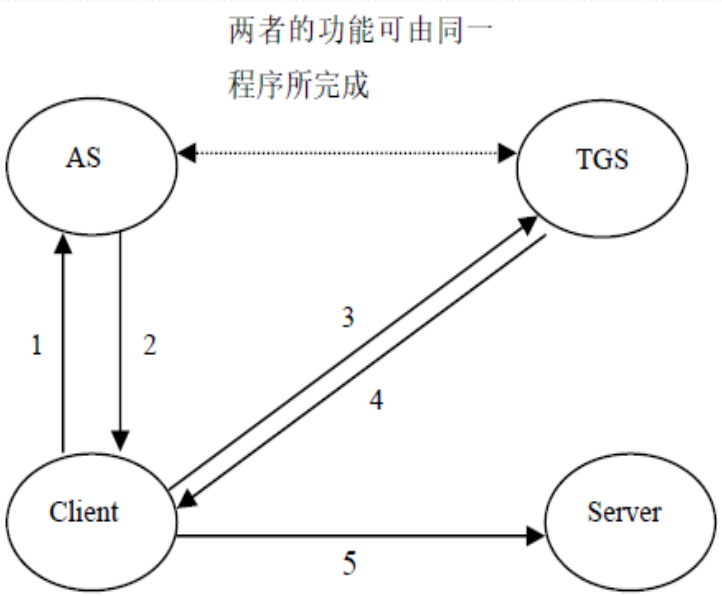
This document provides an overview and specification of Version 5 of the Kerberos protocol, and it obsoletes RFC 1510 to clarify aspects of the protocol and its intended use that require more detailed or clearer explanation than was provided in RFC 1510. This document is intended to provide a detailed description of the protocol, suitable for implementation, together with descriptions of the appropriate use of protocol messages and fields within those messages.



身份认证协议(8)：

Stallings 15.3节

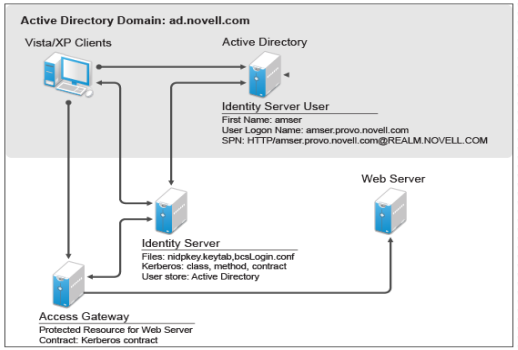
- *Kerberos* V身份认证协议(1)



1. Client → AS : c, tgs, n
2. AS → Client: $\{K_{c,tgs}, n\}_{Kc}, \{T_{c,tgs}\}_{K\ tgs}$
3. Client → TGS: $\{A_c\}_{Kc,tgs}, \{T_{c,tgs}\}_{K,tgs}$, s, n
4. TGS → Client: $\{K_{c,s}, n\}_{Kc,tgs}, \{T_{c,s}\}_{Ks}$
5. Client→Server: $\{A_c\}_{Kc,s}, \{T_{c,s}\}_{Ks}$

c	客户端 Principal
s	服务器的 Principal
addr	客户端的 IP 地址
life	该 Ticket 的生存期
n	nonce，一个随机串
TGS	Ticket Granting Server
AS	Authentication Server
K_x	x的密钥
$K_{x,y}$	x和y的会话密钥
$T_{x,y}$	x向y申请服务所用的Ticket
$\{M\}_{K_x}$	用 x 的密钥加密明文 M 后的密文
A_x	x发出的authenticator

Figure 4-4 Example Kerberos Configuration

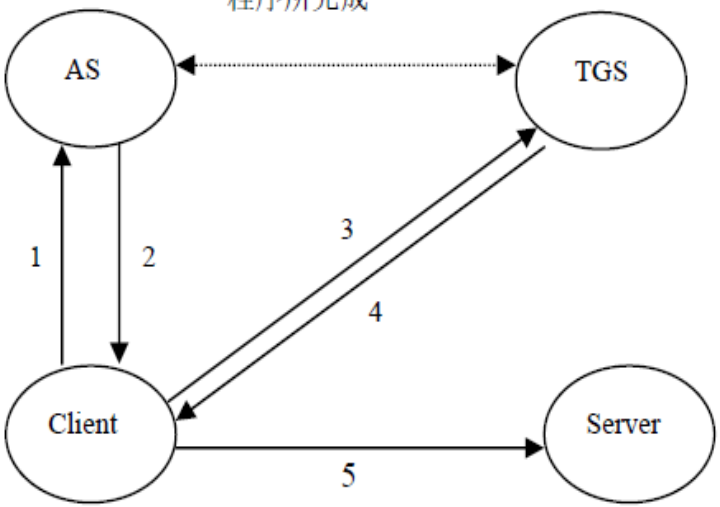


身份认证协议(8)：

Stallings 15.3节

- *Kerberos* V身份认证协议(2)

两者的功能可由同一程序所完成



1. Client → AS : c, tgs, n
2. AS → Client: $\{K_{c,tgs},n\}_{Kc},\{T_{c,tgs}\}_{K,tgs}$
3. Client → TGS: $\{A_c\}_{Kc,tgs},\{T_{c,tgs}\}_{K,tgs}$, s, n
4. TGS → Client: $\{K_{c,s},n\}_{Kc,tgs},\{T_{c,s}\}_{Ks}$
5. Client→Server: $\{A_c\}_{Kc,s},\{T_{c,s}\}_{Ks}$

c	客户端 Principal
s	服务器的 Principal
addr	客户端的 IP 地址
life	该 Ticket 的生存期
n	nonce，一个随机串
TGS	Ticket Granting Server
AS	Authentication Server
K_x	x的密钥
$K_{x,y}$	x和y的会话密钥
$T_{x,y}$	x向y申请服务所用的Ticket
$\{M\}_{K_x}$	用 x 的密钥加密明文 M 后的密文
A_x	x发出的authenticator

Kerberos的静态配置/初始化：

(1)每个c持有一个对称密钥K_c，K_c为c和AS间共享；

(2)每个TGS持有一个对称密钥K_{tgs}，由TGS和AS共享；

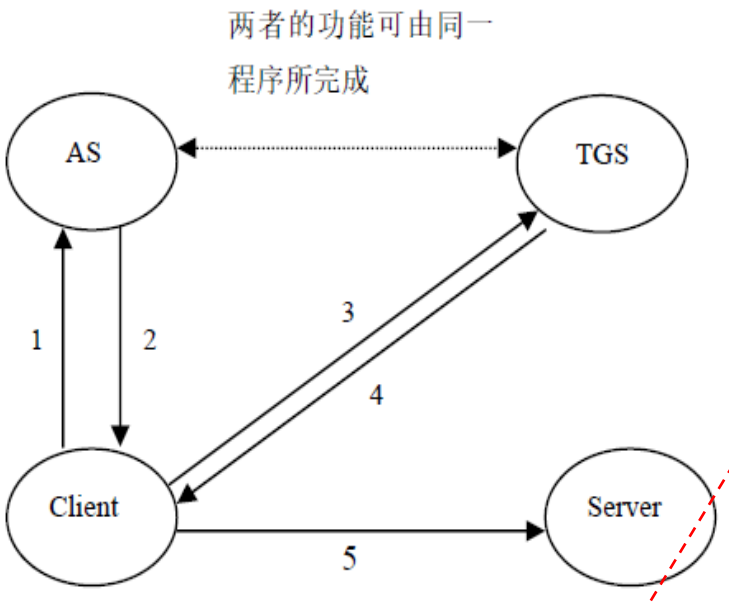
(3)每个Server持有一个K_s，由TGS和Server共享；

(4)AS上存储合法用户c的身份标识及其K_c的列表；

(5)AS和TGS上配置实时访问控制的安全策略库。

身份认证协议(9)： Stallings 15.3节

- *Kerberos* V身份认证协议(3A)



1. Client → AS : c, tgs, n
2. AS → Client: $\{K_{c,tgs},n\}_{Kc},\{T_{c,tgs}\}_{K\ tgs}$
3. Client → TGS: $\{A_c\}_{Kc,tgs},\{T_{c,tgs}\}_{K,tgs}$, s, n
4. TGS → Client: $\{K_{c,s},n\}_{Kc,tgs},\{T_{c,s}\}_{Ks}$
5. Client→Server: $\{A_c\}_{Kc,s},\{T_{c,s}\}_{Ks}$

第二步：AS→Client: $\{K_{c,tgs},n\}_{Kc},\{T_{c,tgs}\}_{Ktgs}$，即AS发送TGT($T_{c,tgs}$)和会话密钥$K_{c,tgs}$给Client。

AS收到Client第一步发来的消息后，依据Principal的名字tgs知道Client拟申请TGT。在验证自身数据库中存在 c所表示的Principal后，AS首先生成能够证明Client身份的Ticket $T_{c,tgs}$和一个随机会话密钥$K_{c,tgs}$。$T_{c,tgs}$的结构如下：

$T_{c,tgs}=\{c, tgs, addr, realm, timestamp, life, K_{c,tgs}\}$

然后AS向Client发送下列消息：用c的密钥K_c加密的$K_{c,tgs}$和n以及用TGS的密钥K_{tgs}加密的$T_{c,tgs}$。将Client发送过来的n加密后再发回的作用是向client证明自己确实就是AS，以防止第三方冒充AS(因为K_c只有Client和AS两者共享，故$\{K_{c,tgs},n\}_{Kc}$只能由AS正确生成)。

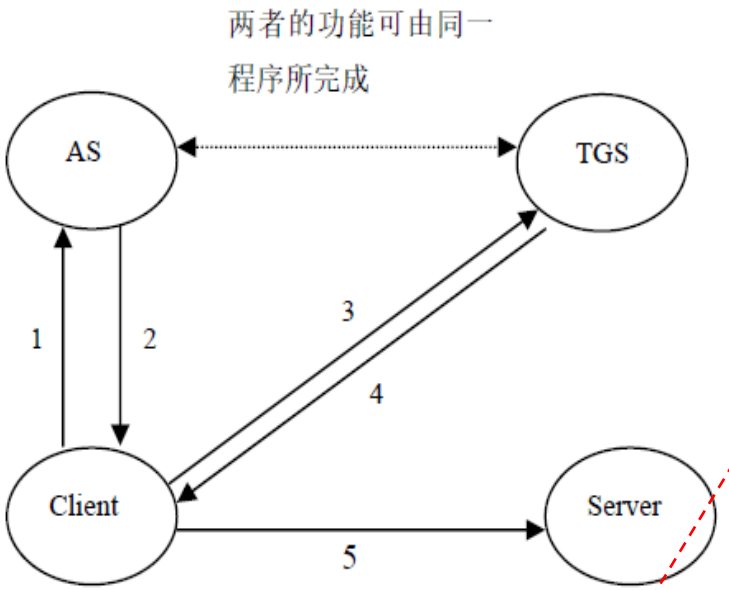
第三步：Client→TGS: $\{A_c\}_{Kc,tgs},\{T_{c,tgs}\}_{Ktgs}$, s, n 即Client向TGS申请用以访问Server的Ticket（图中的$T_{c,s}$）。

最后Client向TGS发送消息，消息的内容有：原封不动地照搬第二步收到的$\{T_{c,tgs}\}_{Ktgs}$、用$K_{c,tgs}$加密的A_c、代表要申请的服务的Principal的名字s和一个新的随机串n（作用与第一步中的n相同)。

第四步：TGS→Client: $\{K_{c,s},n\}_{Kc,tgs},\{T_{c,s}\}_{Ks}$，即 TGS 发送$\{T_{c,s}\}_{Ks}$和会话密钥$K_{c,s}$给Client。

身份认证协议(9)： Stallings 15.3节

- *Kerberos* V身份认证协议(3B)



1. Client → AS : c, tgs, n
2. AS → Client: $\{K_{c,tgs},n\}_{Kc},\{T_{c,tgs}\}_{K\ tgs}$
3. Client → TGS: $\{A_c\}_{Kc,tgs},\{T_{c,tgs}\}_{K,tgs}$, s, n
4. TGS → Client: $\{K_{c,s},n\}_{Kc,tgs},\{T_{c,s}\}_{Ks}$
5. Client→Server: $\{A_c\}_{Kc,s},\{T_{c,s}\}_{Ks}$

第二步：AS→Client: $\{K_{c,tgs},n\}_{Kc},\{T_{c,tgs}\}_{Ktgs}$，即AS发送TGT($T_{c,tgs}$)和会话密钥$K_{c,tgs}$给Client。

AS收到Client第一步发来的消息后，依据Principal的名字tgs知道Client拟申请TGT。在验证自身数据库中存在 c所表示的Principal后，AS首先生成能够证明Client身份的Ticket $T_{c,tgs}$和一个随机会话密钥$K_{c,tgs}$。$T_{c,tgs}$的结构如下：

$T_{c,tgs}=\{c, tgs, addr, realm, timestamp, life, K_{c,tgs}\}$

然后AS向Client发送下列消息：用c的密钥K_c加密的$K_{c,tgs}$和n以及用TGS的密钥K_{tgs}加密的$T_{c,tgs}$。将Client发送过来的n加密后再发回的作用是向client证明自己确实就是AS，以防止第三方冒充AS(因为K_c只有Client和AS两者共享，故$\{K_{c,tgs},n\}_{Kc}$只能由AS正确生成)。

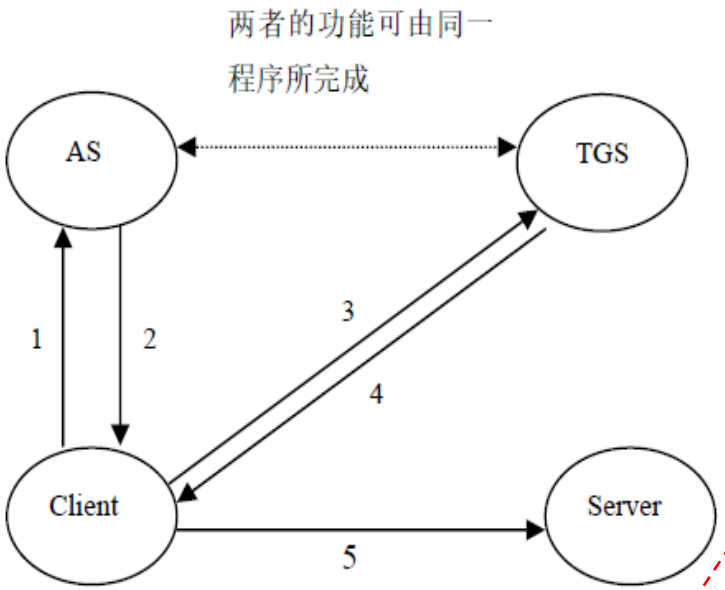
第三步：Client→TGS: $\{A_c\}_{Kc,tgs},\{T_{c,tgs}\}_{Ktgs}$, s, n 即Client向TGS申请用以访问Server的Ticket（图中的$T_{c,s}$）。

最后Client向TGS发送消息，消息的内容有：原封不动地照搬第二步收到的$\{T_{c,tgs}\}_{Ktgs}$、用$K_{c,tgs}$加密的A_c、代表要申请的服务的Principal的名字s和一个新的随机串n（作用与第一步中的n相同)。

第四步：TGS→Client: $\{K_{c,s},n\}_{Kc,tgs},\{T_{c,s}\}_{Ks}$，即 TGS 发送$\{T_{c,s}\}_{Ks}$和会话密钥$K_{c,s}$给Client。

身份认证协议(9)

- *Kerberos* V身份认证协议(4)



1. Client → AS : c, tgs, n
2. AS → Client: $\{K_{c,tgs},n\}_{Kc},\{T_{c,tgs}\}_{K\ tgs}$
3. Client → TGS: $\{A_c\}_{Kc,tgs},\{T_{c,tgs}\}_{K,tgs}$, s, n
4. TGS → Client: $\{K_{c,s},n\}_{Kc,tgs},\{T_{c,s}\}_{Ks}$
5. Client→Server: $\{A_c\}_{Kc,s},\{T_{c,s}\}_{Ks}$

第二步：AS→Client: $\{K_{c,tgs},n\}_{Kc},\{T_{c,tgs}\}_{Ktgs}$，即AS发送TGT($T_{c,tgs}$)和会话密钥$K_{c,tgs}$给Client。

AS收到Client第一步发来的消息后，依据Principal的名字tgs知道Client拟申请TGT。在验证自身数据库中存在 c所表示的Principal后，AS首先生成能够证明Client身份的Ticket $T_{c,tgs}$和一个随机会话密钥$K_{c,tgs}$。$T_{c,tgs}$的结构如下：

$T_{c,tgs}$={ c , tgs, addr, realm, timestamp, life, $K_{c,tgs}$}

然后AS向Client发送下列消息：用c的密钥K_c加密的$K_{c,tgs}$和n以及用TGS的密钥K_{tgs}加密的$T_{c,tgs}$。将Client发送过来的n加密后再发回的作用是向client证明自己确实就是AS，以防止第三方冒充AS(因为K_c只有Client和AS两者共享，故$\{K_{c,tgs},n\}_{Kc}$只能由AS正确生成)。

第三步：Client→TGS: $\{A_c\}_{Kc,tgs},\{T_{c,tgs}\}_{Ktgs}$, s, n 即Client向TGS申请用以访问Server的Ticket（图中的$T_{c,s}$）。

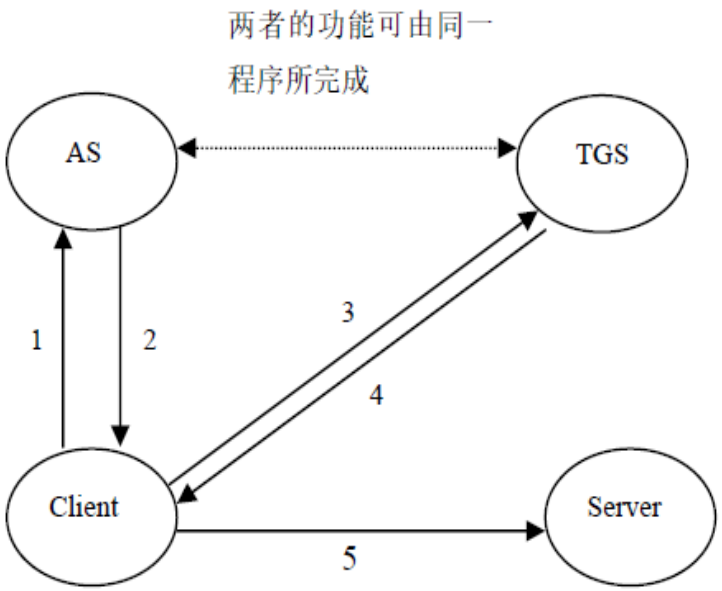
最后Client向TGS发送消息，消息的内容有：原封不动地照搬第二步收到的$\{T_{c,tgs}\}_{Ktgs}$、用$K_{c,tgs}$加密的A_c、代表要申请的服务的Principal的名字s和一个新的随机串n（作用与第一步中的n相同)。

第四步：TGS→Client: $\{K_{c,s},n\}_{Kc,tgs},\{T_{c,s}\}_{Ks}$，即 TGS 发送$\{T_{c,s}\}_{Ks}$和会话密钥$K_{c,s}$给Client。

A_c：client用来表达访问操作和内容的数据块。

身份认证协议(9)

- *Kerberos* V身份认证协议(5)



1. Client → AS : c, tgs, n
2. AS → Client: $\{K_{c,tgs},n\}_{Kc},\{T_{c,tgs}\}_{K\ tgs}$
3. Client → TGS: $\{A_c\}_{Kc,tgs},\{T_{c,tgs}\}_{K,tgs}$, s, n
4. TGS → Client: $\{K_{c,s},n\}_{Kc,tgs},\{T_{c,s}\}_{Ks}$
5. Client→Server: $\{A_c\}_{Kc,s},\{T_{c,s}\}_{Ks}$

第二步：AS→Client: $\{K_{c,tgs},n\}_{Kc},\{T_{c,tgs}\}_{Ktgs}$，即AS发送TGT($T_{c,tgs}$)和会话密钥$K_{c,tgs}$给Client。

AS收到Client第一步发来的消息后，依据Principal的名字tgs知道Client拟申请TGT。在验证自身数据库中存在 c所表示的Principal后，AS首先生成能够证明Client身份的Ticket $T_{c,tgs}$和一个随机会话密钥$K_{c,tgs}$。$T_{c,tgs}$的结构如下：

$$T_{c,tgs}=\{c, tgs, addr, realm, timestamp, life, K_{c,tgs}\}$$

然后AS向Client发送下列消息：用c的密钥K_c加密的$K_{c,tgs}$和n以及用TGS的密钥K_{tgs}加密的$T_{c,tgs}$。将Client发送过来的n加密后再发回的作用是向client证明自己确实就是AS，以防止第三方冒充AS(因为K_c只有Client和AS两者共享，故$\{K_{c,tgs},n\}_{Kc}$只能由AS正确生成)。

第三步：Client→TGS: $\{A_c\}_{Kc,tgs},\{T_{c,tgs}\}_{Ktgs}$, s, n 即Client向TGS申请用以访问Server的Ticket（图中的$T_{c,s}$）。

最后Client向TGS发送消息，消息的内容有：原封不动地照搬第二步收到的$\{T_{c,tgs}\}_{Ktgs}$、用$K_{c,tgs}$加密的A_c、代表要申请的服务的Principal的名字s和一个新的随机串n（作用与第一步中的n相同)。

第四步：TGS→Client: $\{K_{c,s},n\}_{Kc,tgs},\{T_{c,s}\}_{Ks}$，即 TGS 发送$\{T_{c,s}\}_{Ks}$和会话密钥$K_{c,s}$给Client。

A_c：client用来表达访问操作和内容的数据块。

非公钥类身份认证类协议

Kerberos 协议

Figure 4-4 Example Kerberos Configuration