本章目录

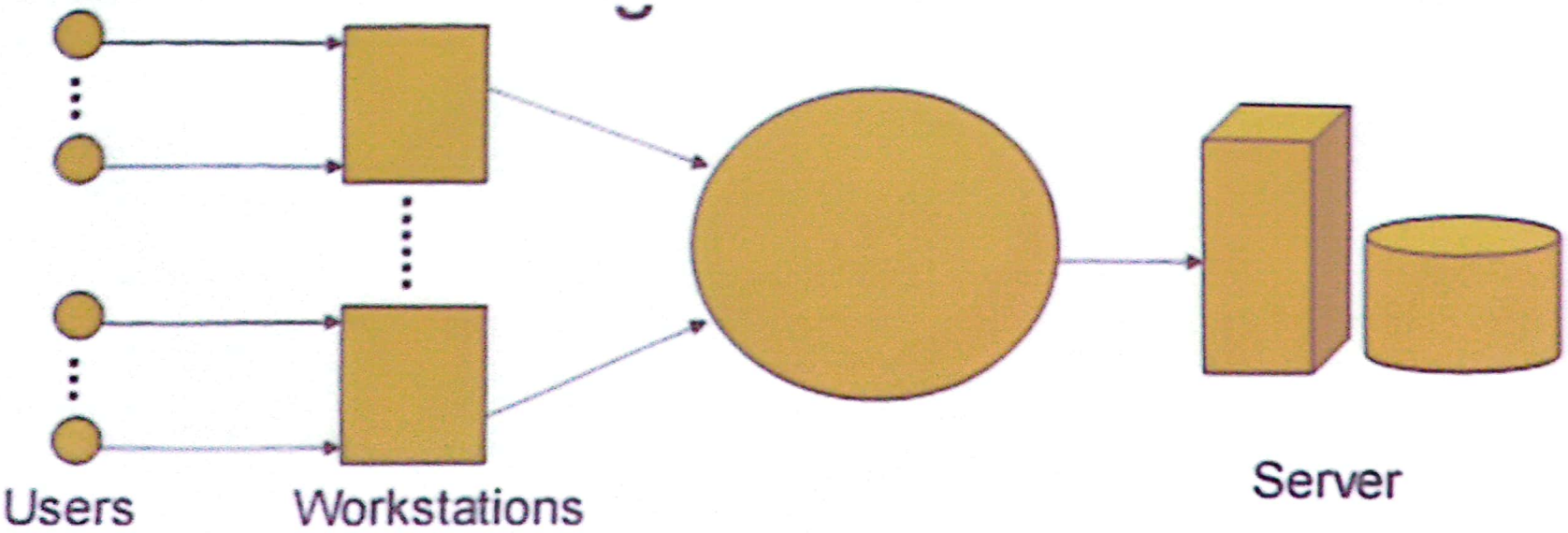
- **Kerberos历史及现状**
  - 历史回顾
  - 典型实现
- **Kerberos票据和认证符**
  - 选项与标志
  - 票据构成
  - 认证符
- **Kerberos协议**
  - 设计目标
  - 设计思路
  - 基本协议流程
  - 消息元素
  - 跨域认证
  - Kerberos Version 5
- **Kerberos消息**
  - 消息构成
  - 消息交换
  - 加密算法规范

扫描全能王 创建

设计目标

- Kerberos目标

  - 在一个开放的分布式环境中
  - 工作站的用户希望访问分布在网络各处的服务器上的服务
  - 服务器能够认证用户身份，并授权。



扫描全能王 创建

设计思路

- 两种票据
  - 票据授权票据 (Ticket granting ticket)
    - 票据授权票据由 AS 发放;
    - $Ticket_{tgs}$ 在用户登录时向 AS 申请一次，可多次重复使用
  - 服务授权票据 (Service granting ticket)
    - 是客户访问应用服务器时需要提供的票据;
    - 用 $Ticket_V$ 表示访问应用服务器 V 的票据。

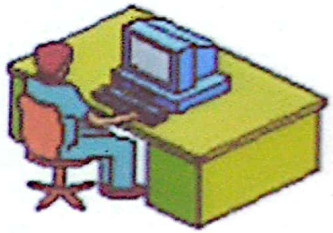
扫描全能王 创建

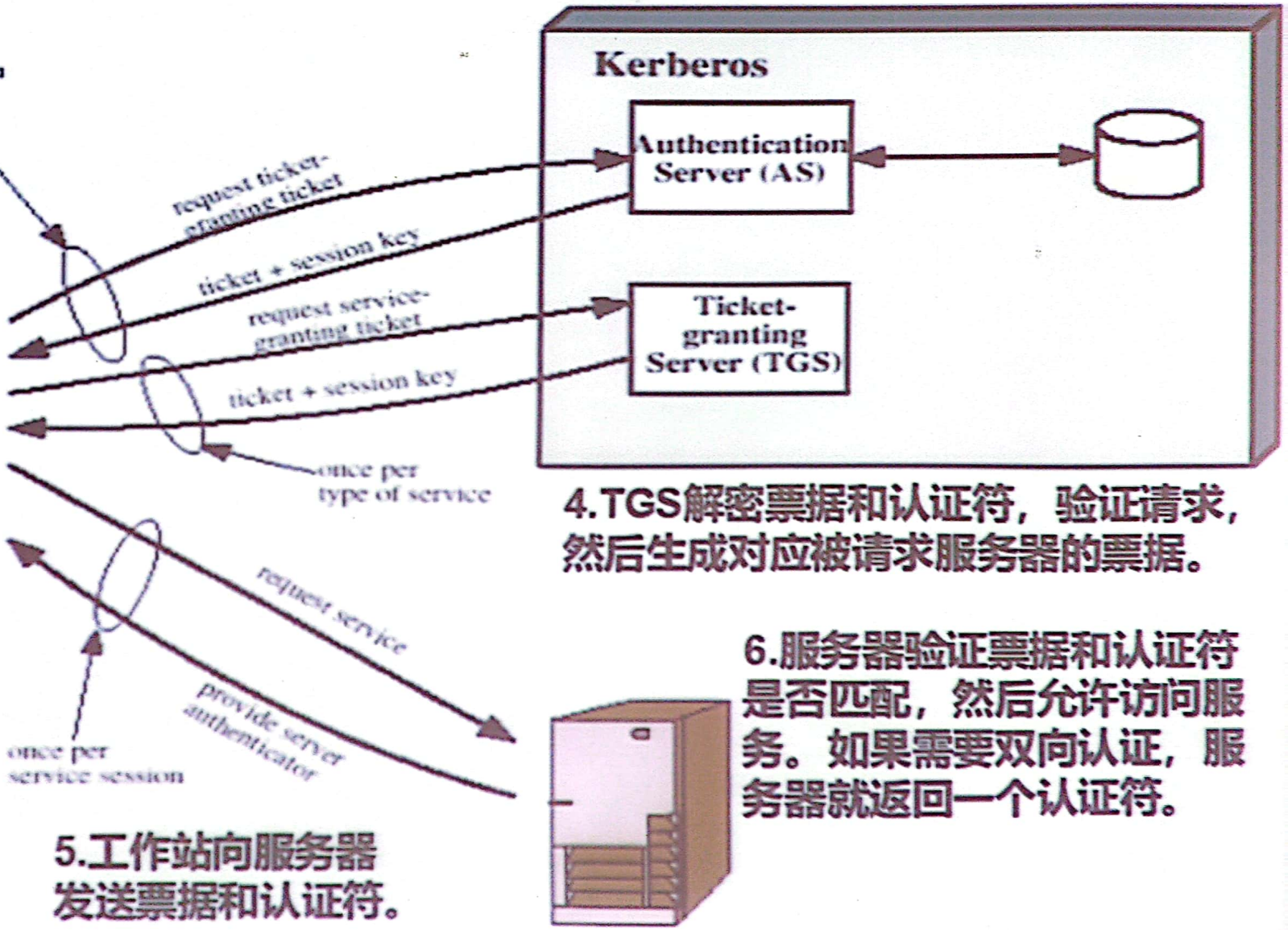
基本协议流程

Kerberos总览

2.AS在数据库中检查用户的访问权限，生成票据授权票据和会话密钥。用由用户口令派生出的密钥加密结果。

1.用户登陆工作站，并请求主机服务。





4.TGS解密票据和认证符，验证请求，然后生成对应被请求服务器的票据。

3.工作站提示用户输入口令，并用口令解密到来的消息。然后将票据和包含用户名、网络地址和时间的认证符发给TGS。

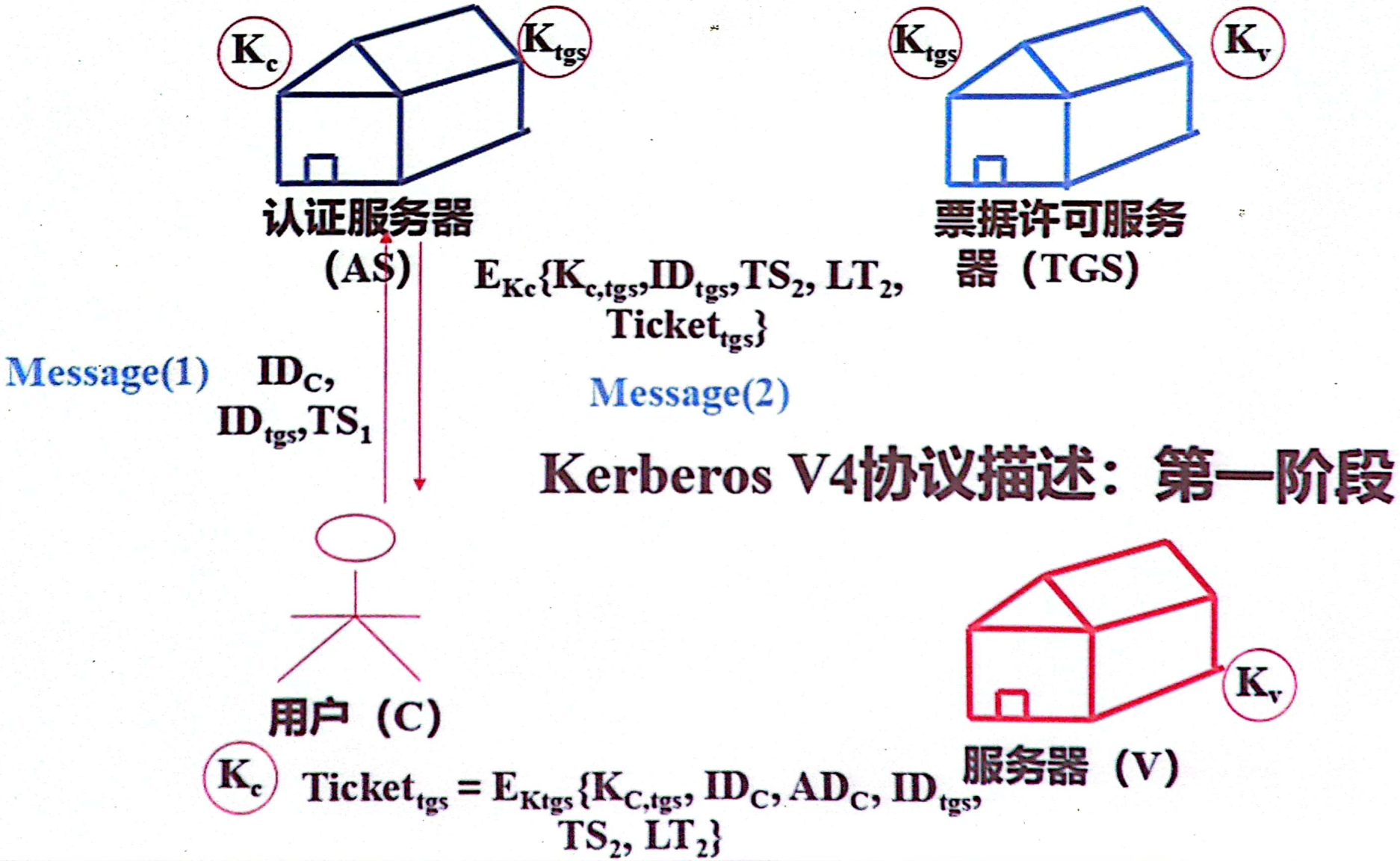
5.工作站向服务器发送票据和认证符。

6.服务器验证票据和认证符是否匹配，然后允许访问服务。如果需要双向认证，服务器就返回一个认证符。

扫描全能王 创建

基本协议流程

Kerberos整个流程可分为三个阶段



扫描全能王 创建

Kerberos V4协议描述：第二阶段

认证服务器（AS）

票据许可服务器（TGS）

Message(3)

$ID_V, Ticket_{tgs}, AU_C$

$E_{KC,tgs}\{K_{C,V}, ID_V, TS_4, Ticket_V\}$

Message(4)

用户（C）

服务器（V）

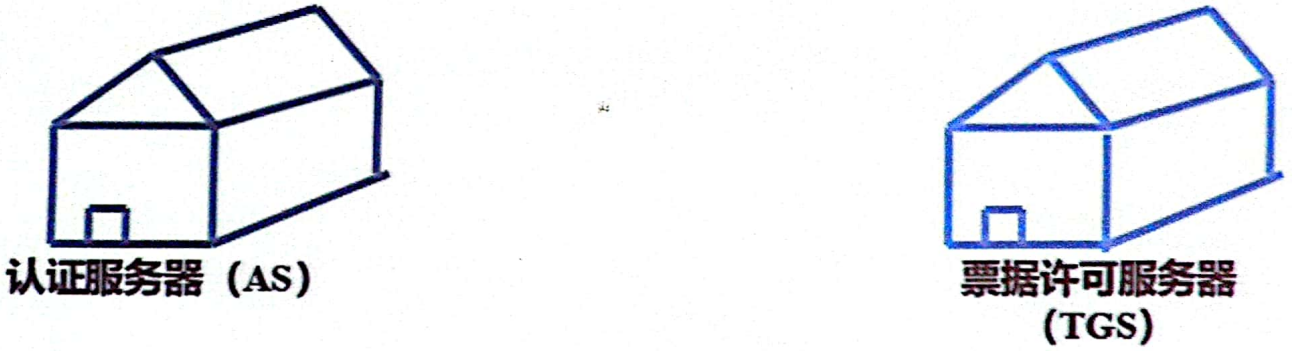
K_c

$Ticket_{tgs} = E_{Ktgs}\{K_{C,tgs}, ID_C, AD_C, ID_{tgs}, TS_2, LT_2\}$

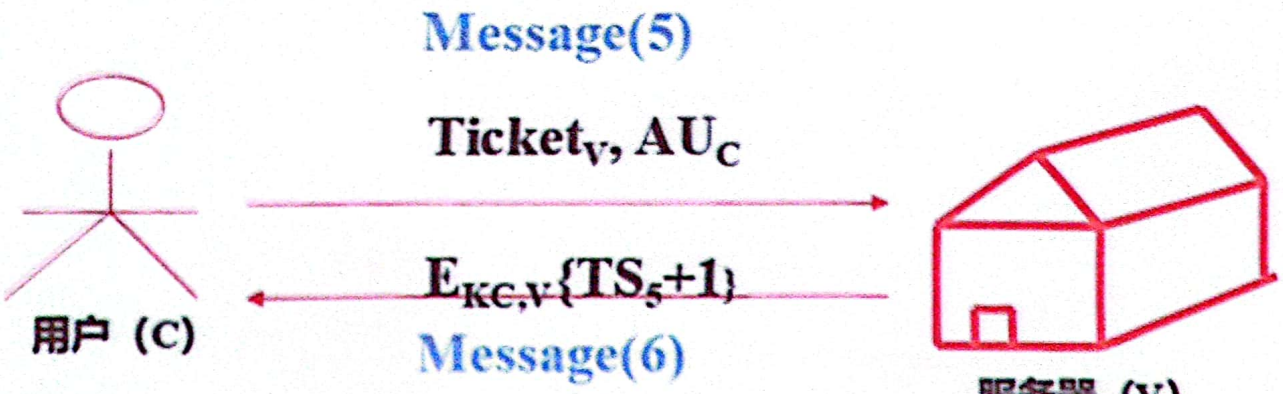
$AU_C = E_{KC,tgs}\{ID_C, AD_C, TS_3\}$

$K_{c,tgs}$

扫描全能王 创建



Kerberos V4协议描述：第三阶段



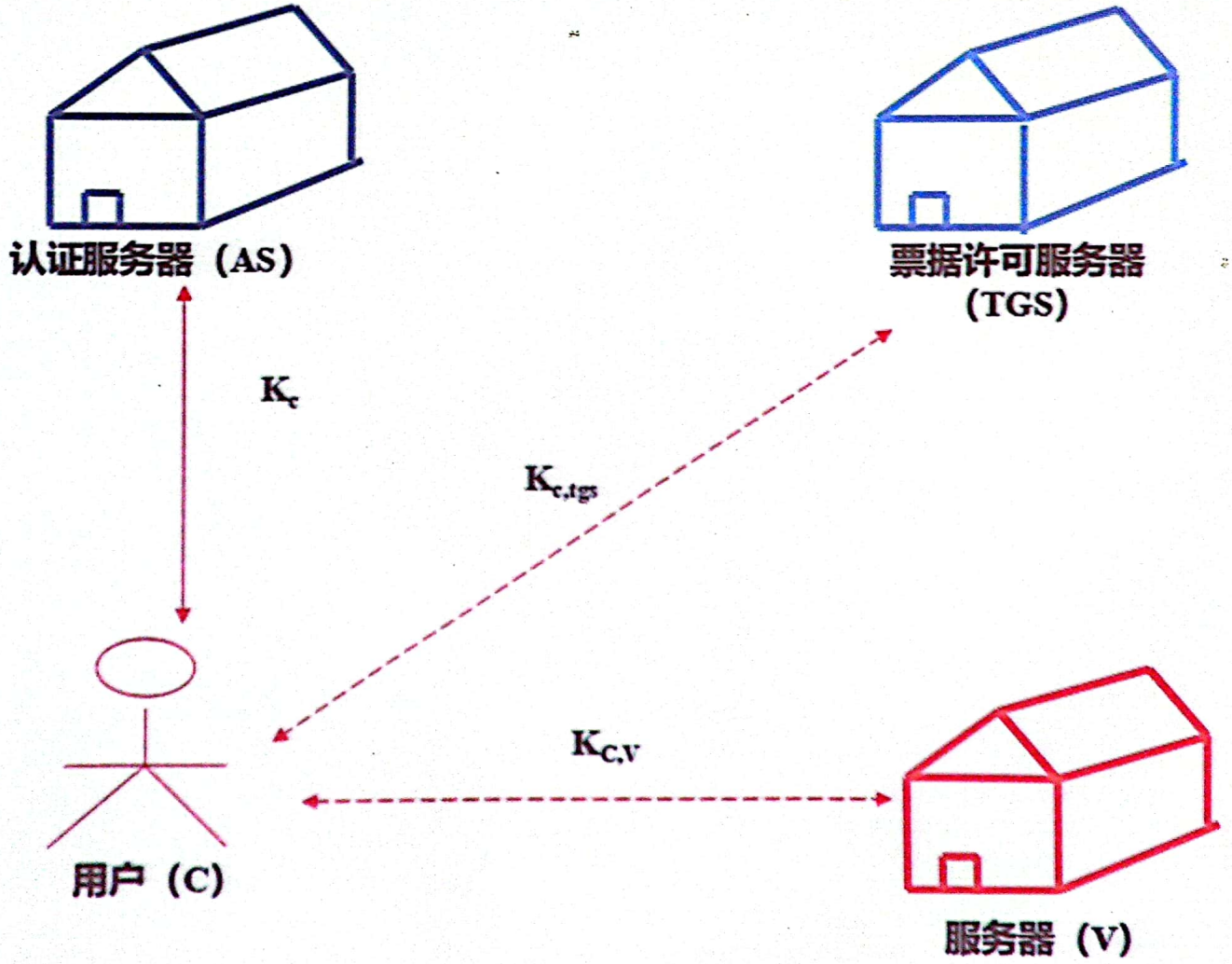
$Ticket_{tgs} = E_{Ktgs}\{K_{C,tgs}, ID_C, AD_C, ID_{tgs}, TS_2, LT_2\}$

$Ticket_V = E_{KV}\{K_{C,V}, ID_C, AD_C, ID_V, TS_4, LT_4\}$

$AU_C = E_{KC,tgs}\{ID_C, AD_C, TS_3\}$

扫描全能王 创建

Kerberos V4协议描述：共享密钥及会话密钥



扫描全能王 创建

Kerberos协议
消息元素

扫描全能王 创建

消息元素

第一阶段：认证服务交换

Message(1) Client 请求 ticket-granting ticket

ID_C ： 告诉AS本client端的用户标识；

ID_{tgs} ： 告诉AS用户请求访问TGS；

TS_1 ： 让AS验证client端的时钟是与AS的时钟同步的；

Message(2) AS返回ticket-granting ticket

E_{K_C} ：基于用户口令的加密，使得AS和client可以验证口令，并保护Message(2)。

$K_{c,tgs}$ ： session key的副本，由AS产生，client可用于在AS与client之间信息的安全交换，而不必共用一个永久的key。

ID_{tgs} ： 确认这个ticket是为TGS制作的。

TS_2 ： 告诉client该ticket签发的时间。

$Lifetime_2$: 告诉client该ticket的有效期；

$Ticket_{tgs}$: client用来访问TGS的ticket。

扫描全能王 创建

消息元素

第二阶段： TGS交换

Message(3) client 请求service-granting ticket

ID_v: 告诉TGS用户要访问服务器V；

$Ticket_{tgs}$: 向TGS证实该用户已被AS认证；

$Authenticator_c$:由client生成，用于验证ticket；

Message(4) TGS返回service-granting ticket

$E_{K_{c,tgs}}$: 仅由C和TGS共享的密钥；用以保护Message(4)；

$K_{c,tgs}$: session key的副本，由TGS生成，供client和server之间信息的安全交换，而无须共用一个永久密钥。

ID_v : 确认该ticket是为server V签发的；

TS_4 : 告诉client该ticket签发的时间；

$Ticket_V$: client用以访问服务器V的ticket；

$Ticket_{tgs}$: 可重用，从而用户不必重新输入口令；

$E_{K_{tgs}}$:ticket用只有AS和TGS才知道的密钥加密，以预防篡改；

$K_{c,tgs}$: TGS可用的session key副本，用于解密authenticator，从而认证ticket；

IDc : 指明该ticket的正确主人；

扫描全能王 创建

消息元素

第三阶段：应用服务认证交换

Message(5) client 请求服务

$Ticket_v$: 向服务器证实该用户已被AS认证；

$Authenticator_c$:由客户生成，用于验证ticket有效；

Message(6) 客户对服务器的可选认证

$E_{k_{c,v}}$: 使C确认报文来自V；

TS_5+1: 使C确信这不使报文重放；

$Ticket_V$: client用以访问服务器V的ticket；

E_{K_v} : 用只有AS和TGS才知道的密钥加密的票据，以预防篡改；

$K_{c,v}$: 用户的会话密钥副本；

IDc : 票据的合法用户；

ADc: 防止非法使用；

IDv: 使服务器确信解密正确；

扫描全能王 创建

Kerberos中的域

- 一个域指一个Kerberos 服务器所直接提供认证服务的有效范围。
- 在通常情况下，Kerberos系统使用领域来控制一个由认证服务器进行认证的区域，每个认证服务器建立和维护自己的区域。
- Kerberos协议是支持跨域认证的，即一个域中的客户端能够被另一个域的服务器认证。

扫描全能王 创建

获得另一领域中的认证服务

- 分三步骤:

(1) 获得本地TGS的访问权;

(2) 请求一张远程TGS的票据许可票据;

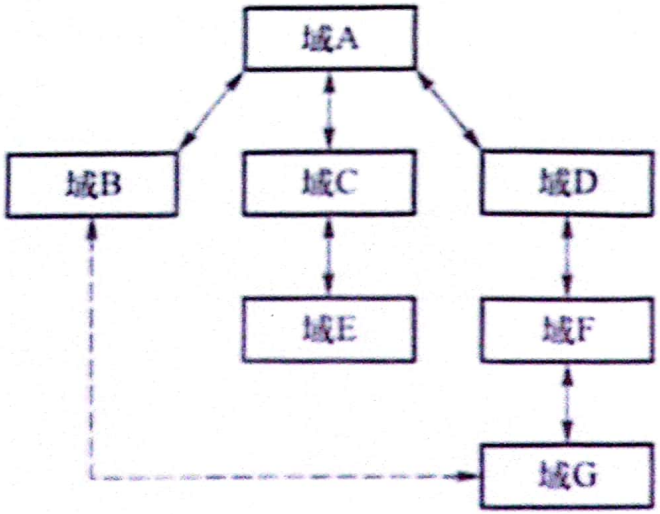
(3) 向远程TGS申请其领域内的服务许可票据

扫描全能王 创建

Kerberos跨域认证——多个域

在实际网络中会存在多个域,Kerberos允许跨越多个中间域进行认证,这些域之间构成了一个认证链,每两个直接跨域认证的域之间都共享域间密钥。

Kerberos v5将这些域组织成树状结构,如图所示。



认证路径

- 直接路径：如域B和G之间，通过A->D->F建立认证路径。
- 备份路径：建立备份路径，如B与G之间，实现直接跨域认证。

扫描全能王 创建

Kerberos Version 5

- Kerberos v5是上个世纪90年代中期开发的
- Kerberos v4与Kerberos v5的区别
  - 加密系统依赖性：v4使用DES，v5用加密类型标记密文，可以使用任何加密技术
  - Internet协议依赖性：v4使用IP地址，不支持其他地址类型；v5用类型和长度标记网络地址，允许使用任何类型的网络地址
  - 消息字节顺序：v4由发送方说明消息字节顺序；v5按编码规则确定消息字节顺序
  - 票据的生命期：v4生命期8位表示，2^8x5=1280分钟；v5有精确的起始时间和终止时间，允许任意长度生命期
  - 域间认证：v4中N个域的互操作需要N^2个Kerberos-to-Kerberos关系；v5需要的连接少

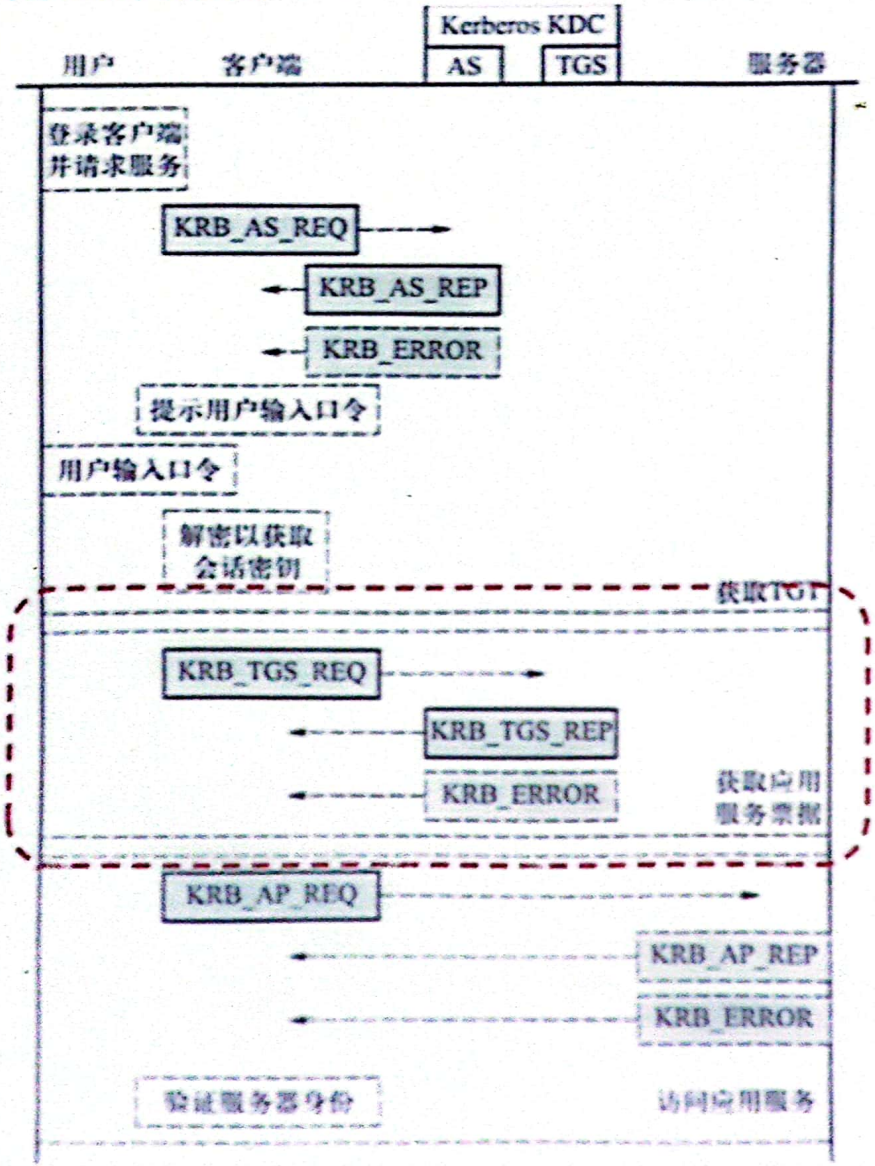
扫描全能王 创建

Kerberos消息

消息构成

扫描全能王 创建

消息构成



第二阶段： TGS交换

KRB_TGS_REQ消息格式:

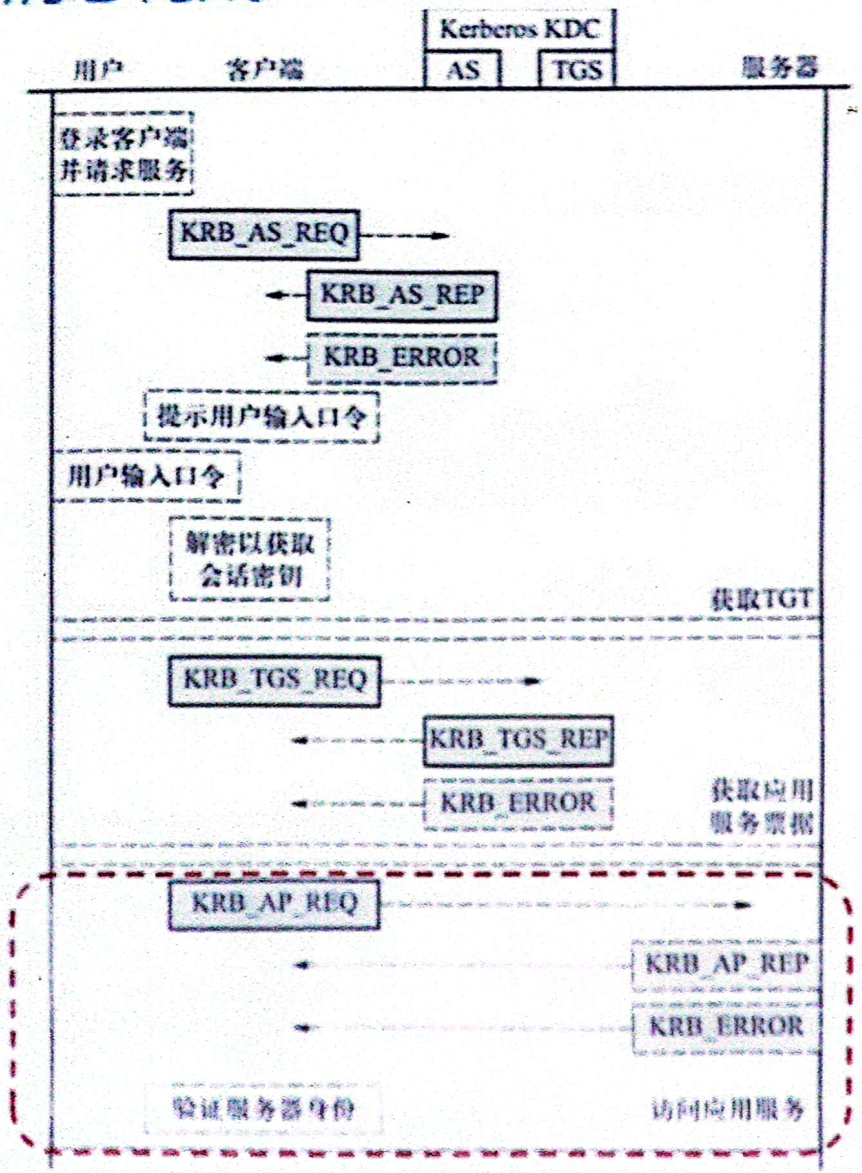
KRB_TGS_REQ
pvno(协议版本) msg-type(消息类型) padata(PA-DATA*)
req-body kdc-options(KDC选项) cname(客户端实体名) realm(服务器域) sname(服务器名) from(起始时间*) till(终止时间) rtime(更新终止时间*) nonce(随机数) etype(加密类型) addresses(主机地址*) enc-authorization-data(加密的认证数据*) additional-tickets(附加票据*)

KRB_TGS_REP消息的构成与KRB_AS_REP相同

扫描全能王 创建

消息构成

消息构成



第三阶段： 应用服务认证交换

KRB_ERROR消息格式：

KRB_ERROR
pvno(协议版本)
msg-type(消息类型)
ctime(客户端时间戳*)
cusec(客户端时间戳*)
stime(服务器时间戳)
susec(服务器时间戳)
error-code(错误码)
crealm(客户端域*)
cname(客户端实体名*)
realm(服务器域)
sname(服务器名)
e-text(描述*)
e-data(其他信息*)

扫描全能王 创建

消息交换——认证服务交换

1、客户端发送 KRB_AS_REQ 请求消息

- 请求消息中包含：
  - 是否进行预认证。
  - 所请求的票据是否可更新、可代理或可转发。
  - 是否允许推迟使用票据（推迟票据选项）。
  - 请求票据的生存期、起止时间等。

扫描全能王 创建

消息交换——认证服务交换

4、AS 构造 KRB_AS_REP 消息

- AS 使用服务器密钥加密票据中需要加密的部分。
- 将请求中的 caddr 字段复制到响应中，并处理预认证数据到 padata 字段。
- 使用客户端密钥和加密类型对响应进行加密。
- 最终生成 KRB_AS_REP 消息并返回给客户端。

扫描全能王 创建

消息交换——认证服务交换

- 认证服务交换用于两种场合：
  - 客户端初始认证向 AS（认证服务器）申请信任凭证。
  - 应用服务器通过 TGS（票据授权服务器）获取用户的服务票据

扫描全能王 创建

消息交换——信任交换

- 功能：通过 KRB_CRED 消息进行票据传递。
- 消息类型：KRB_CRED 消息，传递票据和相关的其他数据。
- 验证流程：
  - 服务器验证 KRB_CRED 消息中的票据和数据。
  - 验证票据的有效性和完整性，确保没有篡改。

扫描全能王 创建

消息交换——信任交换

- 功能：通过 KRB_CRED 消息进行票据传递。
- 消息类型：KRB_CRED 消息，传递票据和相关的其他数据。
- 验证流程：
  - 服务器验证 KRB_CRED 消息中的票据和数据。
  - 验证票据的有效性和完整性，确保没有篡改。

扫描全能王 创建

消息交换——应用服务认证交换

2、服务器对请求的应答

- 服务器收到 KRB_AP_REQ 消息后，验证消息中的有效性，包括：
  - 消息格式是否为 KRB_AP_REQ。
  - 请求中的密钥版本是否支持，若不支持返回 KRB_AP_ERR_BADKEYVER 错误。
  - 使用服务器的密钥解密票据，验证认证符是否一致，若篡改返回 KRB_AP_ERR_BAD_INTEGRITY 错误。
  - 其他错误处理包括 KRB_AP_ERR_SKWEW（时间偏差问题）等。

扫描全能王 创建

消息交换——TGS交换

- TGS交换用于三种场合：
  - 客户端需要访问某个应用服务器的信任状。
  - 客户端需要更新或有效化已有票据。
  - 客户端需要获得一张代理票据。

扫描全能王 创建

消息交换——TGS交换

5、客户端收到应答

- 客户端收到 KRB_AS_REP 消息后，验证消息中的客户端名和域名是否与请求中的信息匹配。
- 确认 padata 是否包含预期的加密算法和密钥信息。

扫描全能王 创建

Kerberos消息
加密算法规范

扫描全能王 创建

加密算法规范

- Kerberos 使用加密与完整性校验机制，确保消息的安全传输。每个加密和校验操作都包含了加密类型和校验类型的说明。
- 常用的校验方法为消息验证码（MAC），同时也支持循环冗余校验（CRC）用于完整性验证。

扫描全能王 创建

Kerberos票据和认证符

选项和标志

扫描全能王 创建

选项和标志

- Kerberos的票据和认证符是认证系统的核心，提供灵活性和安全性。
- 用户可以预先申请票据，服务端则以标志的形式定义票据属性。
- 票据选项和标志主要用于描述票据的属性和功能。

扫描全能王 创建

选项和标志

1、初始认证

- 初始认证相关选项包括 INITIAL, PRE-AUTHENT, HW-AUTHENT, 和 OPT-HARDWARE-AUTH。
- INITIAL 表示票据由 AS 颁发，而不是基于 TGT 颁发。
- PRE-AUTHENT 和 HW-AUTHENT 提供额外的认证信息，确保客户端的身份可信。

扫描全能王 创建

选项和标志

2、可更新票据

- 票据更新可以提高系统的灵活性和安全性，标志 RENEWABLE 允许票据在生存期内更新。
- 客户端可以请求 RENEW 选项来更新票据，KDC 会生成新的密钥和有效期。

3、可更新票据

- Kerberos 系统允许票据推迟生效，客户端可以选择 ALLOW-POSTDATE 来推迟票据的使用。
- MAY-POSTDATE 标志使票据可以在之后使用，但必须在 KDC 激活之前设定。

扫描全能王 创建

选项和标志

4、 无效票据

- INVALID 标志表示票据无效，应用服务器在收到包含此标志的票据时，必须拒绝该票据。
- 要激活无效票据，必须由KDC进行激活。

5、代理功能

- Kerberos 允许客户端指定代理以访问应用服务器。
- TGT 中的 PROXIABLE 标志允许 TGS 为另一个 IP 地址颁发票据，客户端必须明确指定该 IP 地址。
- PROXY 标志用于代理功能，确保代理可以执行客户端任务。
- OK-AS-DELEGATE 标志确保代理可以获得客户端的信任，并执行相关的认证操作。

扫描全能王 创建

票据构成

Kerberos票据
tkt-vno(票据版本)
realm(服务器域)
sname(服务器名)
enc-part(加密区域)
flags(标志)
key(会话密钥)
crealm(客户端域)
cname(客户端实体名)
transited(传输编码)
authtime(认证时间)
starttime(起始时间*)
endtime(终止时间)
renew-till(更新终止时间*)
caddr(主机地址*)
authorization-data(认证数据)

- Kerberos票据
  - tkt-vno：票据版本号
  - realm：Kerberos 服务的域
  - sname：服务器的名字

扫描全能王 创建

Kerberos票据和认证符

认证符

扫描全能王 创建

认证符

authenticator
authenticator-vno(认证符版本)
crealm(客户端域)
cname(客户端实体名)
chsum(校验和*)
cusec(客户端时间戳)
ctime(客户端时间戳)
subkey(子会话密钥*)
seq-number(序号*)
authorization-data(认证数据*)

- authenticator-vno：认证符的版本号，Kerberos v5 的取值为5。
- crealm：客户端所在的域，用于标识客户端身份。
- cname：客户端实体名称。
- chsum：校验和，用于验证认证符是否被篡改。
- subkey：子会话密钥，用于保护会话的多次连接中的通信安全。
- seq-number：序号，用于标识消息的顺序，确保消息不被重放。
- cusec：客户端时间戳，表示认证符发送的时间精度为微秒。
- ctime：客户端时间戳，表示认证符发送的时间。
- authorization-data：包含用于限制票据应用范围的附加信息。

扫描全能王 创建

小结

- **Kerberos历史及现状**

  历史回顾、典型实现

- **Kerberos协议**

  设计目标、设计思路、基本协议流程、消息元素、跨域认证、Kerberos Version 5

- **Kerberos票据和认证符**

  选项与标志、票据构成、认证符

- **Kerberos消息**

  消息构成、消息交换、加密算法规范

扫描全能王 创建