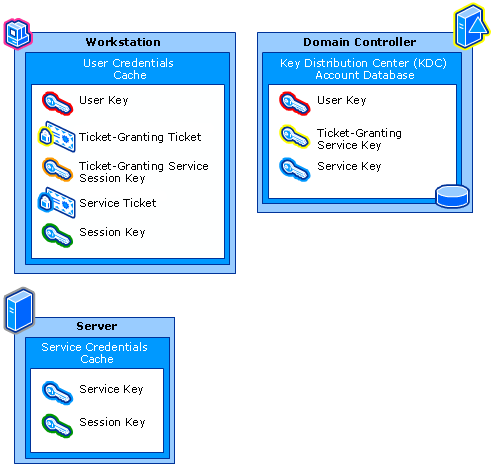
什么是Kerberos协议？

Kerberos是一种由国际标准RFC1510定义的，用来进行用户身份验证（Authentication）的协议。根据该RFC，不同的软件提供商可以开发自己特有的Kerberos程序，但是RFC可以保证一定程度上不同产生开发的Kerberos程序符合该规范，可以相互通信，例如AD内的用户可以通过Kerberos验证来访问Unix上的网站/文件资源。

接下来讨论的Kerberos均是基于微软在Windows 2003操作系统中开发的Kerberos协议。之后的操作系统都是基于该版本的Kerberos进行改进的。

Kerberos中的基本组成和术语



Kerberos验证的组要成员为客户端（Workstation），域控（Domain Controller）以及目标服务器（Server）。用户登录客户端尝试访问存储在服务器上的资源，此时服务器要求验证才可访问并支持Kerberos验证，这时就可以采用Kerberos验证。

密钥（Keys）：密钥可以理解为一串加密的字符串，当验证方发现用户提供的密钥与自己存储的密钥一致时，我们可以认为验证通过。User keys就是采用用户的密码通过加密算法所产生的加密字符串；System keys就是采用计算机的密码通过加密算法所产生的加密字符串，这里计算机可以理解为账户，也拥有自己的密码，只是该密码由操作系统自己维护；Service keys就是采用服务启动账号的密码通过加密算法所产生的加密字符串，这里服务器启动账号可以是域用户账号，也可以是Local System，就是计算机本身。此外，Inter-realm keys表示不同管理领域之间所共享的密钥，这里Realm就是指管理领域，如果采用微软的AD，就是域 (domain)，而采用第三方的目录服务，即称为Realm。Session keys表示短时存在的，为了方便短时多次验证生成的密钥，该密码可以在通过验证后一段时间内使用，使得系统不需要频繁进行验证。

上面提到的加密算法 （Encryption Algorithm） 是在操作系统中已定义的，通常包括如下算法：

DES-CBC-CRC

DES-CBC-MD5

RC4-HMAC

AES128-CTS-HMAC-SHA1-96（Windows Vista/2008以后）

AES256-CTS-HMAC-SHA1-96（Windows Vista/2008以后）

票据（Tickets）：票据可以理解为一种数据结构，是Kerberos验证时所使用的基本结构。票据中包含了颁发票据的Realm，访问服务的名称，客户端的名称，标志位，有效期，Session keys，授权信息（Authorization-Data）等（具体可以参考TechNet链接中Ticket Contents表格）。我们可以把它理解成登机牌，当您的信息（护照/身份证）已经验证通过后，验证机构会给您颁发登机牌，您就可以凭借登机牌登机（上面包含了航班，座位，起飞时间等信息），而登机就是一种资源访问。

Ticket-Granting Tickets（TGT）是一种特殊的票据，它表示用户通过了身份验证，可以采用该票据再次向票据颁发机构（Ticket-Granting Service，TGS）申请真正需要访问的资源。例如，您在申请身份证或者护照时，颁发机构需要仔细核对您的身份，颁发后您乘坐飞机/汽车/火车都可以凭借身份证和护照即可，不需要再次进行身份验证。因此这可以减少验证的次数，提供验证效率。

Service Tickets是最终享用资源的票据，即上述所述的登机牌，汽车票或者火车票。

正如之前所述，这种设计的好处就在于当客户端尝试多次访问最终资源，例如需要买多张飞机票时，只需要向TGS申请一次TGT，即身份证/护照，只要在身份证/护照有效期内，均可以直接购买。

Key Distribution Center（KDC）就是可以对用户身份进行验证（Authentication service，AS）的机构，即身份证和护照颁发机构，同时也是票据颁发机构（TGS）。在AD中，KDC就是域控，它存储了所有用户/计算机/服务账号的密码，因此可以进行验证。此外，第三方可以开发独立的验证服务器，而不提供LDAP等域控功能，因此这里统称为KDC，。在AD中，KDC的密码即为账号krbtgt的密码，该账号为系统自动创建，不能删除或者修改，它的密码由系统维护。因此TGS key就是由该密码生成的密钥。

Authenticator是用来证明客户端就是票据中所描述的客户端。这是为了防止网络中存在其它恶意客户端冒充实际客户端提供票据。客户端时间是Authenticator中最为重要的部分，默认情况下，AD中认为与域控时间不超过5分钟为可靠的时间。之后在Authentication Service Request（KRB\_AS\_REQ）中描述的结构虽然没有Authenticator，但是Pre-authentication data中也包含客户端时间。

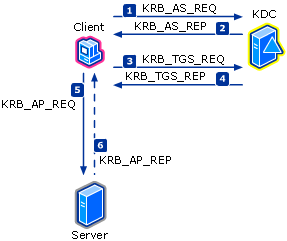
Service Principal Names（SPN）是最终服务名称在AD中的表现形式。当客户端获得TGT，即身份证/护照后，需要向TGS再次申请所要访问的资源的票据。TGS就是根据SPN来区别不同的服务提供商。在AD中存储有服务启动账号，在该账号中需要设有SPN属性，当客户端提供TGT后，TGS（即域控）就根据TGT中的SPN名称来查找AD数据库中是否存在该服务。常见的SPN格式为HTTP/WebServer1.ForestA.com。更多信息，请参考如下链接：

Service Principal Names

<http://technet.microsoft.com/en-us/library/cc961723.aspx>

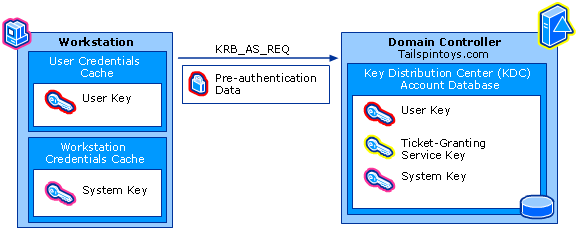
用户授权信息（Authorization data）即包含用户的权限信息，其中Privilege Attribute Certificate（PAC）包含用户的SID以及所有组相关信息，可以帮助判断用户是否有权限访问资源。

Kerberos验证流程



该标准流程的前提是用户，客户端，资源服务器均属于同一个域。例如用户采用[\\server\share](file://server/share)去访问服务器上的共享资源，或者用户尝试用浏览器打开<http://web.com>网页等。

1. Authentication Service Request（KRB\_AS\_REQ）

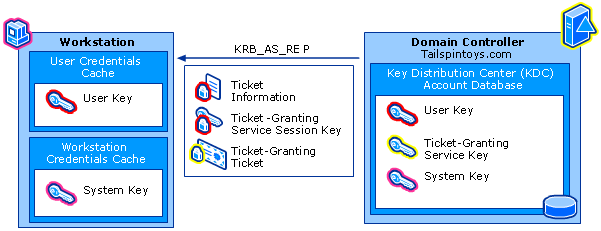


客户端发送KRB\_AS\_REQ类型的消息（网络包）给域控进行验证。该消息包括：

* 用户名（user principal name属性，UPN）
* 用户域名
* 用用户密码生成的密钥加密的Pre-authentication data（主要是客户端时间）

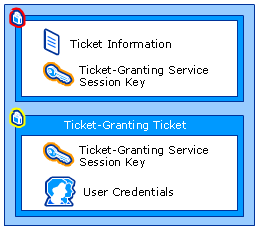
域控在收到该消息后，会

1. 根据用户名（UPN） 在数据库中查找该账户
2. 获取该账户密码并通过加密算法生成密钥存储在内存中（长期）
3. 采用上述密钥解密Pre-authentication data，对其中的客户端时间进行评判，如果符合要求，则认为用户可信任
4. Authentication Service Reply（KRB\_AS\_REP）



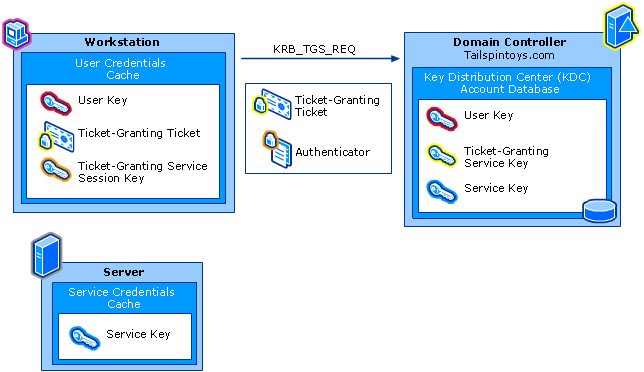
在域控通过上述验证后，会返回KRB\_AS\_REP类型的消息给客户端。该消息包括：

* 用用户密钥加密的票据信息和TGS Session key
* 用TGS key加密的TGT，TGT包含：
  + TGS Session key
  + 用户的授权信息（Authorization data）



当客户端收到该消息后，会采用用户登陆时已经缓存在内存中的用户密钥解密票据信息和TGS Session key。之后客户端和域控间的验证都可以采用该Session key，无需再使用用户密钥，这样最大程度上保护用户密码。Session key是短时存在的，它会在TGT过期（默认10小时）或者用户注销时失效。

1. Ticket-Granting Service Request（KRB\_TGS\_REQ）

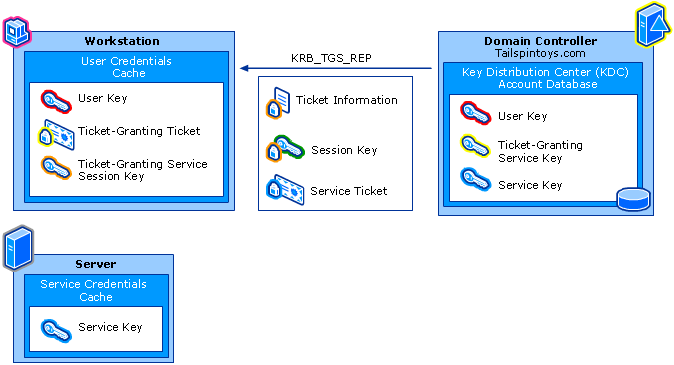


客户端会再次发送KRB\_TGS\_REQ类型消息给域控，请求最终资源的票据。该消息包含：

* 目标服务器的名称（SPN）
* 目标服务器的域名
* 原封不动的用户TGT（由于没有TGS key，用户无法解密）
* 用TGS Session key加密的验证信息（authenticator），主要是客户端的时间

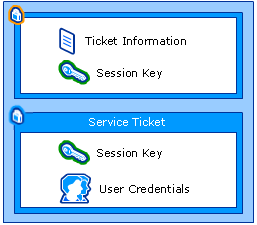
域控在收到消息后，会：

1. 用自己的TGS key解密用户TGT，获得TGS Session key（虽然在AS回复中，域控提供了TGS Session key，但是并没有保存在本地，而是使用客户端TGT中提供的）
2. 使用TGS Session key解密Authenticator，对其中的客户端时间进行评判，如果符合要求，则认为用户可信任
3. 根据SPN找到服务启动账号，把它的密码通过加密算法生成密钥存储在内存中（长期）
4. 组装服务票据（Service Ticket）并用上述密钥（Service key）加密，创建新的供客户端和服务器通讯的Session key
5. Ticket-Granting Service Reply（KRB\_TGS\_REP）



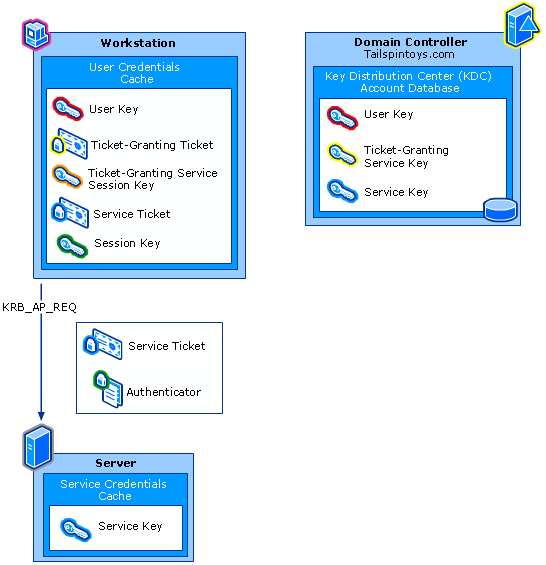
域控随后会返回KRB\_TGS\_REP类型的消息给客户端。该消息包括：

* 用TGS Session key加密的票据信息和新的供客户端和服务器通讯的Session key
* 用服务启动账号密钥（Service key）加密的服务票据（Service Ticket），它包括：
  + 新的供客户端和服务器通讯的Session key
  + 从TGT中复制来的用户授权信息（Authorization data）



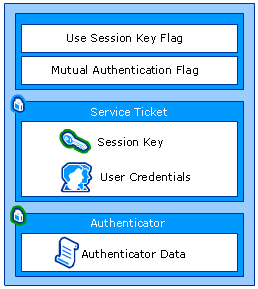
当客户端收到该消息后，会用TGS Session key解密新的和服务器通讯的Session key并把它和加密的Service Ticket存储在内存中。

1. Kerberos Application Request（KRB\_AP\_REQ）



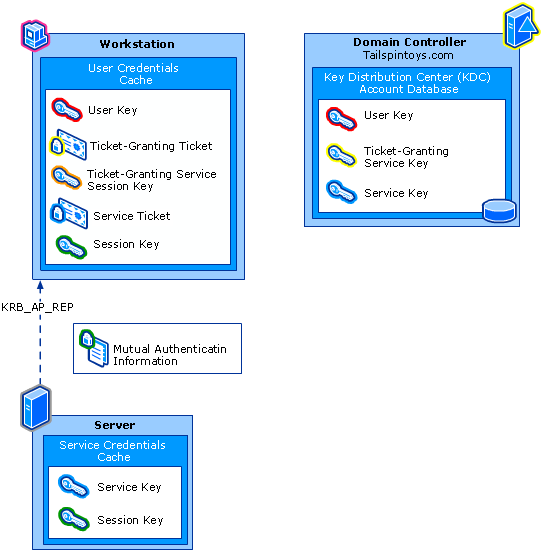
客户端最终会发送KRB\_AP\_REQ消息类型给目标服务器。该消息包括：

* 是否使用Session key标志位
* 是否需要双向验证标志位
* 原封不动的服务票据（由于没有Service key，用户无法解密）
* 用新的Session key加密的验证信息（authenticator），主要是客户端的时间



目标服务器收到该消息后，会：

1. 用自己的Service key解密服务票据，获得用户的授权信息和Session key
2. 使用Session key解密Authenticator，对其中的客户端时间进行评判，如果符合要求，则认为用户可信任
3. 查看是否需要双向验证标志位，如果需要，则使用Session key加密之前的客户端时间，返回给客户端
4. 使用用户授权信息创建用户Access token并确定用户是否有权限访问资源
5. Kerberos Application Reply（KRB\_AP\_REP）

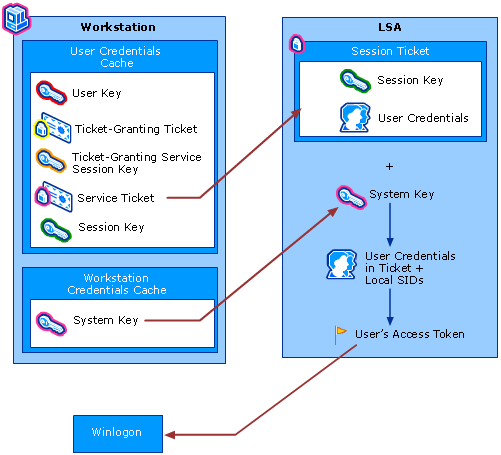


正如之前所述，该消息是根据双向验证标识位来决定返回，因此是可选的。如果标识位选中，服务器使用Session key加密之前的客户端时间返回给客户端，客户端使用Session key解密该时间后与之前的Authenticator中时间对比，如果相同，则认为服务器可信任。

扩展的Kerberos验证流程

1. 用户登录

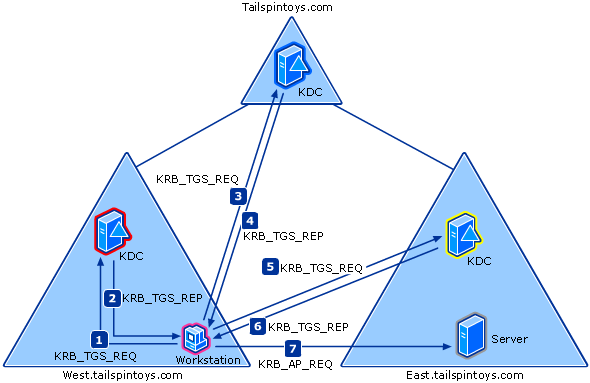
用户登录情景可以看成用户访问的资源就是计算机本身。因此域控会采用计算机的密钥加密服务票据在KRB\_TGS\_REP中返回给客户端。此时客户端的Local Security Authority（LSA）会自动采用计算机密钥解密服务器票据，在获取用户授权信息中的PAC后与本地的组信息融合生成用户的Access token。



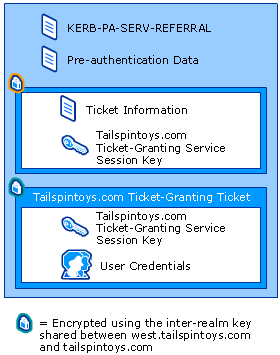
1. Cross-realm验证

当两个Realm相互信任时，它们就会共享Inter-realm key。当两个realm都是基于AD时，该信任关系就创建trust，因此密钥就是trust对象的密码生成的。此时根据信任方向，用户就可以跨域进行资源访问，双方的域控存储上述密钥并可以把对方TGS服务当作本域资源提供给用于。。这就是说，当用户尝试从用户域访问资源域的资源时，用户域域控会在KRB\_TGS\_REP中返回资源域的TGT当作服务票据提供给用户。

考虑如下示例：



1. KRB\_TGS\_REQ：客户端发送访问资源 [server@east.tailspintoys.com](mailto:server@east.tailspintoys.com) 的TGS请求，因此客户端注意到该资源可能不在本域中，因此在请求中添加了NAME\_CANONICALIZE标志位
2. KRB\_TGS\_REP：west.tailspintoys.com的域控注意到上述标志位后，会尝试查询目标服务器，当在本域中找不到后，会尝试遍历所有的信任关系以查看是否存在域east.tailspintoys.com，并创建最短路径。为该最短路径中下一个域（tailspintoys.com）的TGS服务创建TGT票据返回给客户端：



1. -7)与之前的步骤类似

当资源域在本域域控信任列表中无法找到时，域控返回错误信息，此时根据客户端的应用程序信息来决定下一步:

* 可能继续使用下一个验证方式，很大可能是NTLM
* 可能返回错误，使得应用程序弹框，让用户输入新的用户名和密码
* 可能直接失败，返回无法访问信息

Kerberos相关工具

1. Klist.exe

该工具在Windows Vista/2008以后的操作系统中自带，在之前的操作系统中需要安装Windows Resource Kit：

Windows Server 2003 Resource Kit Tools

<http://www.microsoft.com/en-in/download/details.aspx?id=17657>

该工具可以列出当前客户端的所有服务票据和TGT信息以及提供清除所有票据的选项。

具体请查看如下链接中的相关信息：

Klist.exe: Kerberos List

<http://technet.microsoft.com/en-us/library/cc738673(v=ws.10).aspx>

示例：

清除当前账户票据：klist purge

显示当前账户服务票据：klist tickets

显示当前账户TGT：klist tgt

清楚计算机账户票据：klist -li 0x3e7 purge

1. Ktpass.exe

该工具在Windows Vista/2008以后的操作系统中自带，在之前的操作系统中需要安装Windows Support Tools：

Windows Server 2003 Service Pack 2 32-bit Support Tools

<http://www.microsoft.com/en-in/download/details.aspx?id=15326>

该工具可以为非Windows的第三方服务提供Kerberos验证的支持。例如UNIX服务，通过该命令可以产生.keytab文件用来支持Kerberos验证。

具体信息请查看如下链接：

Ktpass

<http://technet.microsoft.com/en-us/library/cc753771.aspx>

Ktpass Overview

<http://technet.microsoft.com/en-us/library/cc779157(v=ws.10).aspx>

Kerberos排错

1. 在客户端或者域控系统日志中查看是否存在Kerberos或者KDC事件报错。
2. 如果可以主动重现问题，在客户端抓取Network monitor网络包。如果涉及跨域访问，可能需要同时在客户端，本域域控，对方域域控抓取网络包。但是需要注意如果有多台域控，需要先在客户端抓一次包来确定联系的域控。在抓包前先清空缓存：klist purge。
3. 查看网络包中的Kerberos报错来缩小问题范围。
4. 进一步地，还可以通过开启Kerberos ETL来进一步排查：
5. Login machine with account belonging to administrators group, then start CMD window by "run as administrator".
6. Run the following command to add a tracing task:

Logman create trace daily\_kerberos\_trace\_log -r -v mmddhhmm -p {6B510852-3583-4e2d-AFFE-A67F9F223438} 255 3 -o "c:\tmp\Kerb"

1. Run the following command to start logging:

Logman start daily\_kerberos\_trace\_log

1. Then a new file will be created under C:\TMP, named using the date/time, something like Kerb\_MMDDHHMM.etl
2. If you would like to disable and remove the tracing task, run these commands:

Logman stop daily\_kerberos\_trace\_log

Logman delete daily\_kerberos\_trace\_log

更多的排错示例，请查看M08\_S02\_Kerberos\_Authentication\_LAB.docx。