Windows域认证之Kerberos协议认证

什么是Kerberos协议

Kerberos 是一种网络认证协议，其设计目标是通过密钥系统为客户机 / 服务器应用程序提供强大的认证服务。该认证过程的实现不依赖于主机操作系统的认证，无需基于主机地址的信任，不要求网络上所有主机的物理安全，并假定网络上传送的数据包可以被任意地读取、修改和插入数据。在以上情况下， Kerberos 作为一种可信任的第三方认证服务，是通过传统的密码技术（如：共享密钥）执行认证服务的

Kerberos协议的组成角色

在古希腊神话故事中，kerberos是一只具有三颗头颅的地狱恶犬，他守护在地狱之外，能够识别所有经此路过的亡灵，防止活着的入侵者闯入地狱。

kerberos协议中也存在三个角色，分别是

```
客户端（client）：发送请求的一方
服务端（Server）：接收请求的一方
密钥分发中心（Key Distribution Center，KDC），而密钥分发中心一般又分为两部分，分别是：
    AS（Authentication Server）：认证服务器，专门用来认证客户端的身份并发放客户用于访问TGS的TGT（票据授予票据）
    TGS（Ticket Granting Ticket）：票据授予服务器，用来发放整个认证过程以及客户端访问服务端时所需的服务授予票据（Ticket）
```

Kerberos认证的简单流程

举个例子：

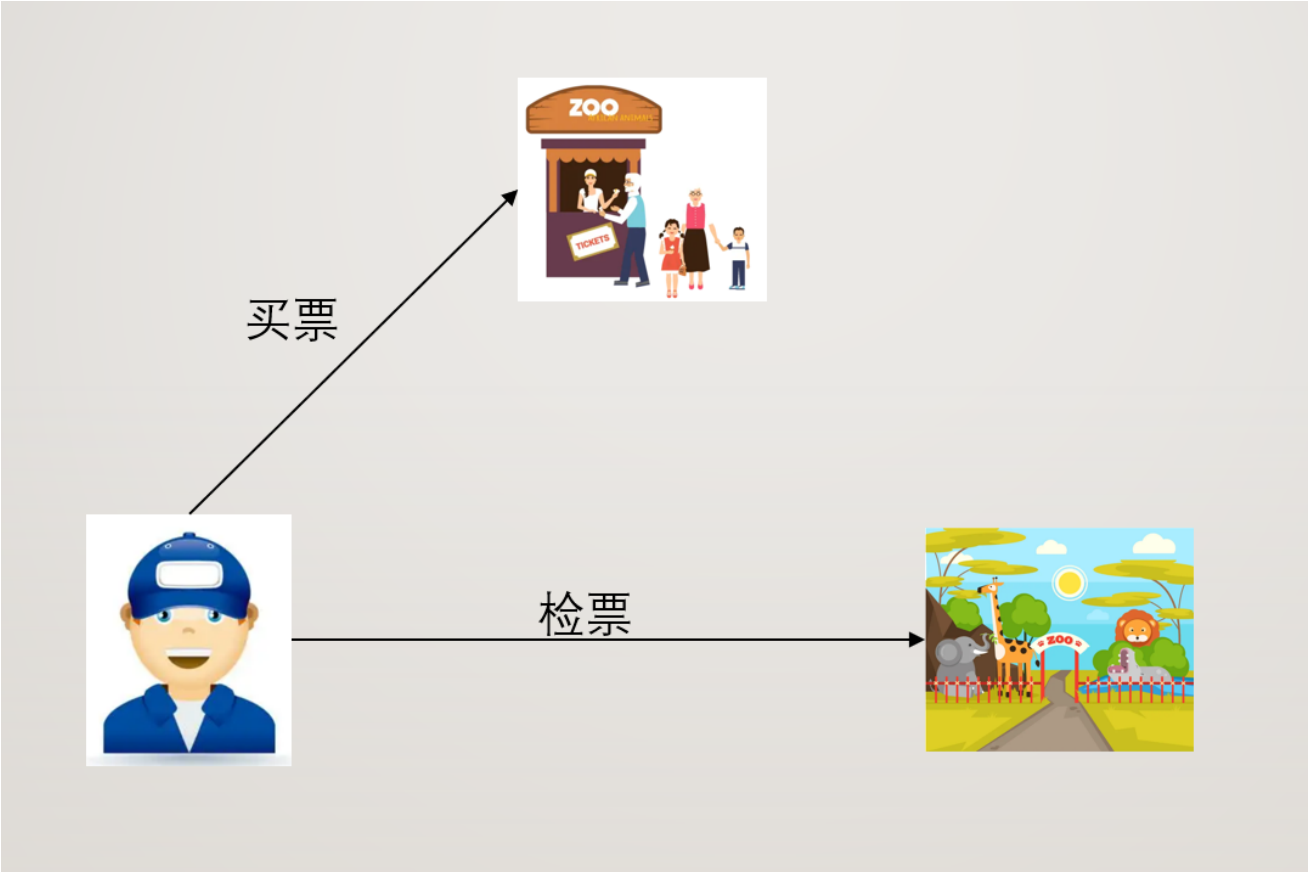
A现在想要去访问B完成一个任务。但是AB两人之间是从来没有见过面的，他们只知道对方的名字叫A，B。此时如果A直接去找B告诉B我就是A，那么B是有理由不相信A的，B同理也得不到A的认可，他们陷入了一个无 法证明我就是我的困境。

于是他们就想到了一个办法，AB找到了一个他俩共同信任的人C，且这个C既认识A又认识B，所以只要C告诉B，这个A确实就是真正的A那么B就会信任这个A，同理B经过C的认可后，A也会相信B的身份。此后，A在访问B之前会先去找C，C会交给A一个凭证，代表此时的A已经得到了C的认证，这时A拿着凭证再去找B，便可以得到B的确认了。

在举个例子：

我们去动物园，动物园不认识你不让你进，你也怕进门后不是动物园，所以就很尴尬

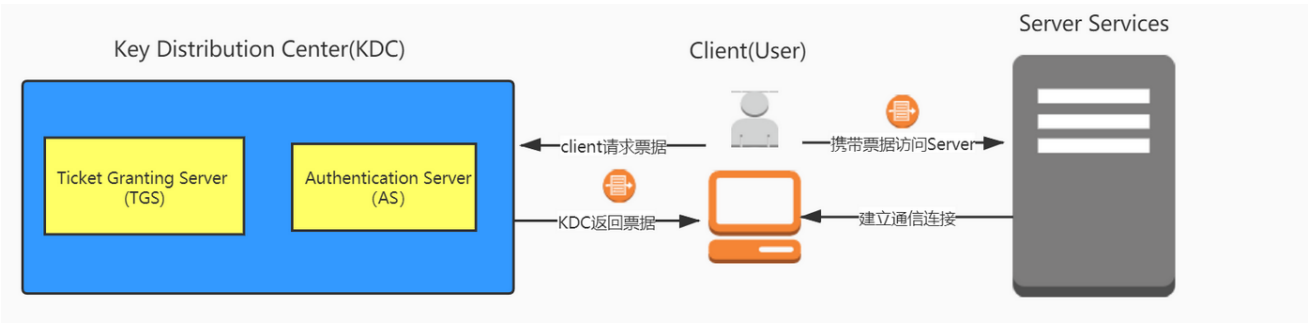
如何解决呢？我们建立一个售票窗口，只要售票处认识你和动物园，你和动物园之间就可以相互信任。



人：代表客户端
动物园：代表服务端
售票处：代表KDC

所以整个kerberos认证流程可以简化描述如下： 客户端在访问每个想要访问的网络服务时，他需要携带一个专门用于访问该服务并且能够证明自己身份的票据，当服务端收到了该票据他才能认定客户端身份正确，向客户端提供服务。所以整个认证流程可简化为两大步：

1、客户端向KDC请求获取想要访问的目标服务的服务授予票据（Ticket）；
2、客户端拿着从KDC获取的服务授予票据（Ticket）访问相应的网络服务；

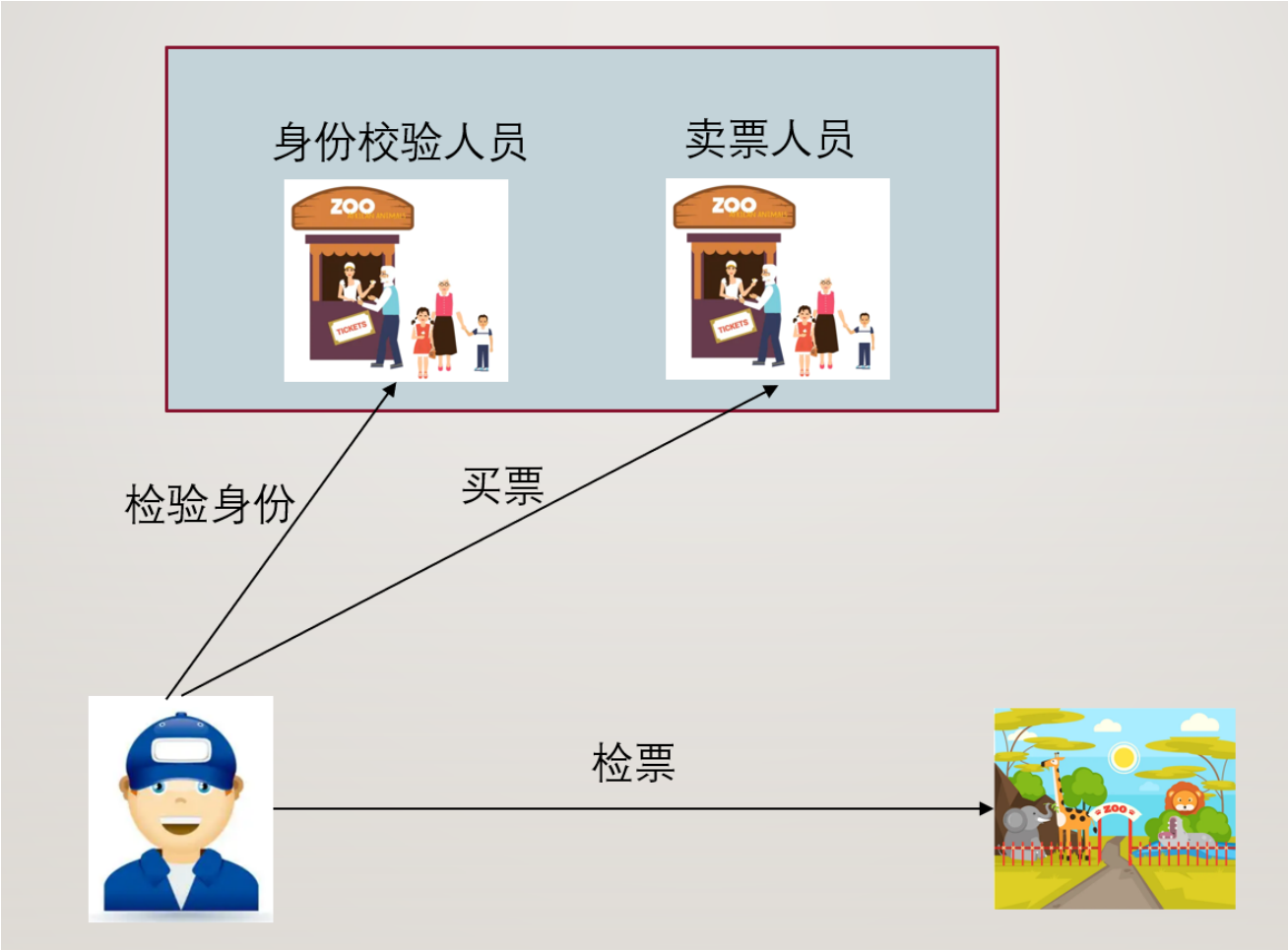


Kerberos认证完成流程

在上述的流程中，其实还有一个问题，那就是

1. KDC怎么知道你（客户端）就是真正的客户端？凭什么给你发放服务授予票据（Ticket）呢？

我们以去动物园为例，售票处凭什么给你买票，你如果是一个逃犯怎么办？其实买票的过程我们可以分为两步第一才步是你拿着身份证去验证，第二步身份验证通过了才会给你票

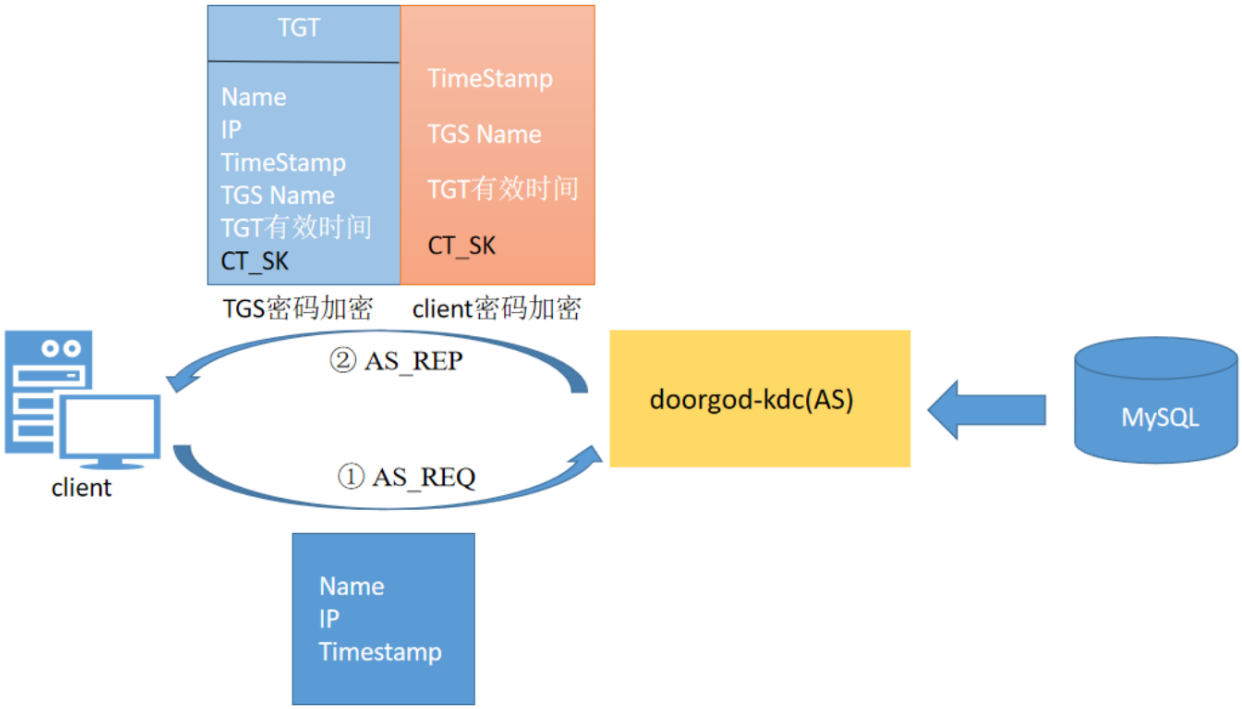


人：代表客户端
动物园：代表服务端
售票处：KDC
　　身份校验人员:AS，负责验证用户身份的合法性，和给用户一个可以买票的票（TGT）
　　卖票人员：TGS，负责客户端访问服务端时所需的服务授予票据的单位

所以kerberos通信可以分为3步，我们逐步详解

通信第一步-客户端和AS进行通信

为了获得能够用来访问服务端服务的票据，客户端首先需要来到KDC获得服务授予票据（Ticket）。由于客户端是第一次访问KDC，此时KDC也不确定该客户端的身份，所以第一次通信的目的为KDC认证客户端身份，确认客户端是一个可靠且拥有访问KDC权限的客户端，



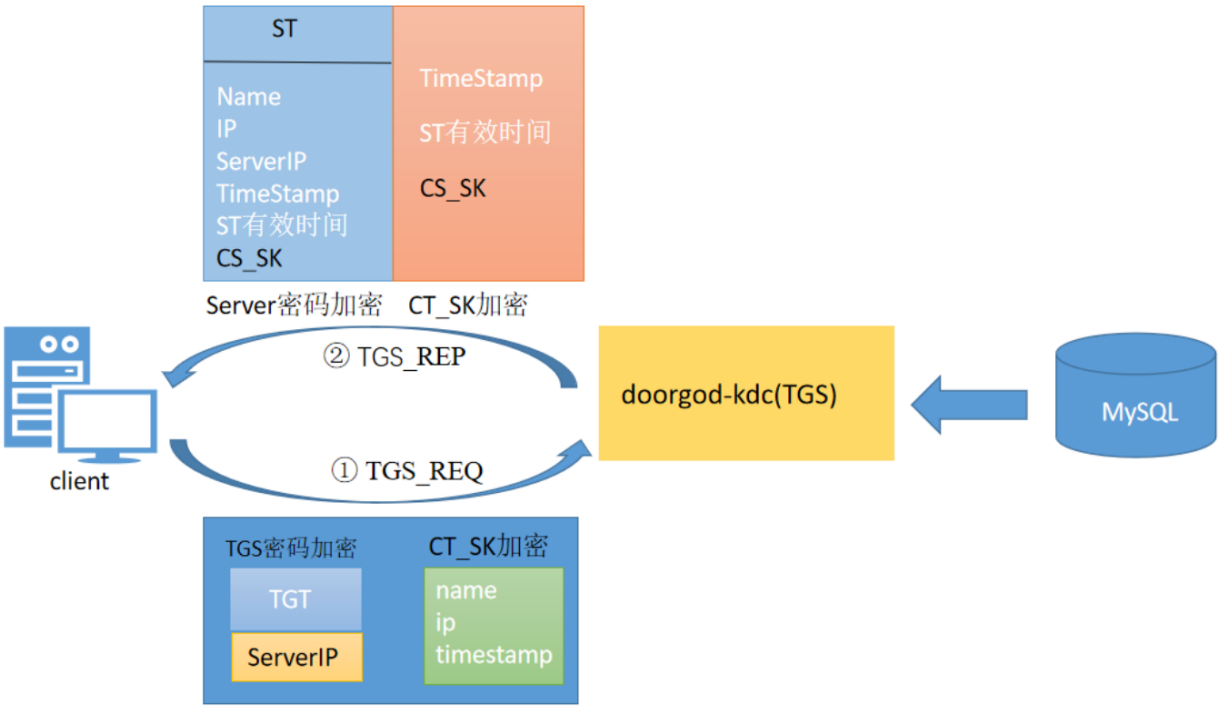
1、客户端用户向KDC以明文的方式发起请求。该次请求中携带了自己的用户名，主机IP，和当前时间戳；
2、KDC当中的AS（Authentication Server）接收请求（AS是KDC中专门用来认证客户端身份的认证服务器）后去kerberos认证数据库中根据用户名查找是否存在该用户，此时只会查找是否有相同用户名的用户，并不会判断身份的可靠性；
3、如果没有该用户名，认证失败，服务结束；如果存在该用户名，则AS认证中心便认为用户存在，此时便会返回响应给客户端，其中包含两部分内容：

3.1、第一部分内容称为TGT，他叫做票据授予票据，客户端需要使用TGT去KDC中的TGS（票据授予中心）获取访问网络服务所需的Ticket（服务授予票据），TGT中包含的内容有kerberos数据库中存在的该客户端的Name，IP，当前时间戳，客户端即将访问的TGS的Name，TGT的有效时间以及一把用于客户端和TGS间进行通信的Session_key(CT_SK)。整个TGT使用TGS密钥加密，客户端是解密不了的，由于密钥从没有在网络中传输过，所以也不存在密钥被劫持破解的情况。

3.2第二部分内容是使用客户端密钥加密的一段内容，其中包括用于客户端和TGS间通信的Session_key(CT_SK),客户端即将访问的TGS的Name以及TGT的有效时间，和一个当前时间戳。该部分内容使用客户端密钥加密，所以客户端在拿到该部分内容时可以通过自己的密钥解密。如果是一个假的客户端，那么他是不会拥有真正客户端的密钥的，因为该密钥也从没在网络中进行传输过。这也同时认证了客户端的身份，如果是假客户端会由于解密失败从而终端认证流程。
至此，第一次通信完成。

通信第二步-客户端和TGS进行通信

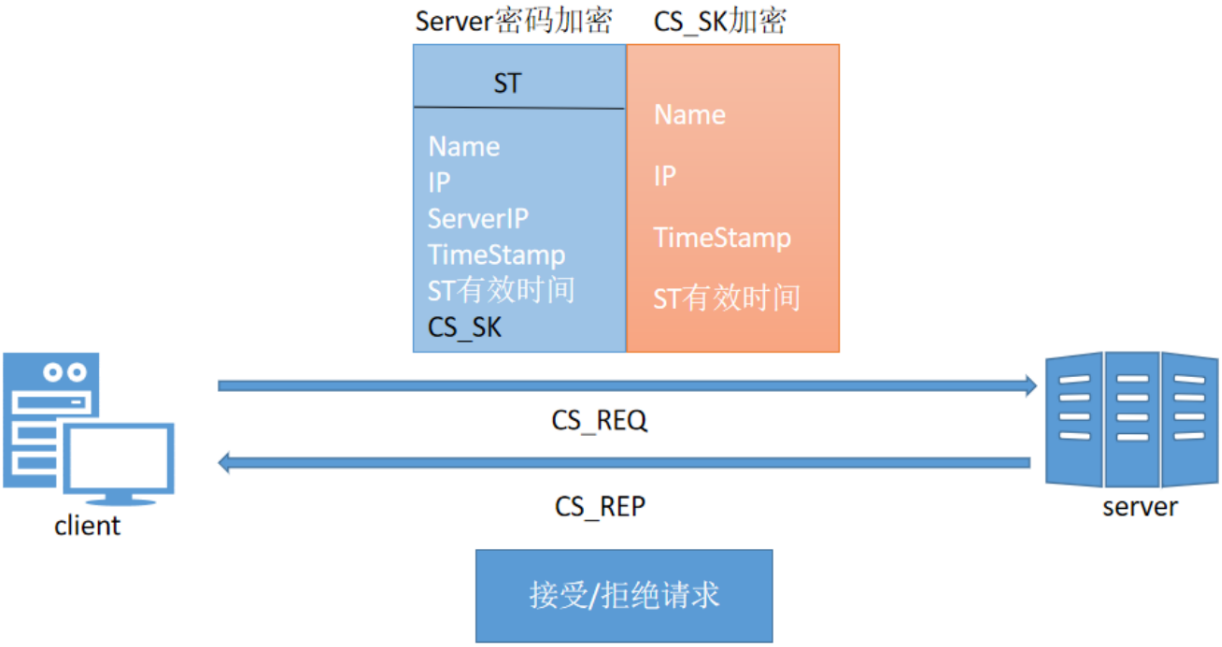
此时的客户端收到了来自KDC（其实是AS）的响应，并获取到了其中的两部分内容。此时客户端会用自己的密钥将第二部分内容进行解密，分别获得时间戳，自己将要访问的TGS的信息，和用于与TGS通信时的密钥CT_SK。首先他会根据时间戳判断该时间戳与自己发送请求时的时间之间的差值是否大于5分钟，如果大于五分钟则认为该AS是伪造的，认证至此失败。如果时间戳合理，客户端便准备向TGS发起请求



客户端行为：
1、客户端使用CT_SK加密将自己的客户端信息发送给KDC，其中包括客户端名，IP，时间戳；
2、客户端将自己想要访问的Server服务以明文的方式发送给KDC；
3、客户端将使用TGS密钥加密的TGT也原封不动的也携带给KDC；
TGS行为：
1、此时KDC中的TGS（票据授予服务器）收到了来自客户端的请求。他首先根据客户端明文传输过来的Server服务IP查看当前kerberos系统中是否存在可以被用户访问的该服务。如果不存在，认证失败结束，。如果存在，继续接下来的认证。
2、TGS使用自己的密钥将TGT中的内容进行解密，此时他看到了经过AS认证过后并记录的用户信息，一把Session_KEY即CT_SK，还有时间戳信息，他会现根据时间戳判断此次通信是否真是可靠有无超出时延。
3、如果时延正常，则TGS会使用CT_SK对客户端的第一部分内容进行解密（使用CT_SK加密的客户端信息），取出其中的用户信息和TGT中的用户信息进行比对，如果全部相同则认为客户端身份正确，方可继续进行下一步。
4、此时KDC将返回响应给客户端，响应内容包括：
　　第一部分：用于客户端访问网络服务的使用Server密码加密的ST（Servre Ticket），其中包括客户端的Name，IP，需要访问的网络服务的地址Server IP，ST的有效时间，时间戳以及用于客户端和服务端之间通信CS_SK（Session Key）。
　　第二部分：使用CT_SK加密的内容，其中包括CS_SK和时间戳，还有ST的有效时间。由于在第一次通信的过程中，AS已将CT_SK通过客户端密码加密交给了客户端，且客户端解密并缓存了CT_SK，所以该部分内容在客户端接收到时是可以自己解密的。
至此，第二次通信完成。

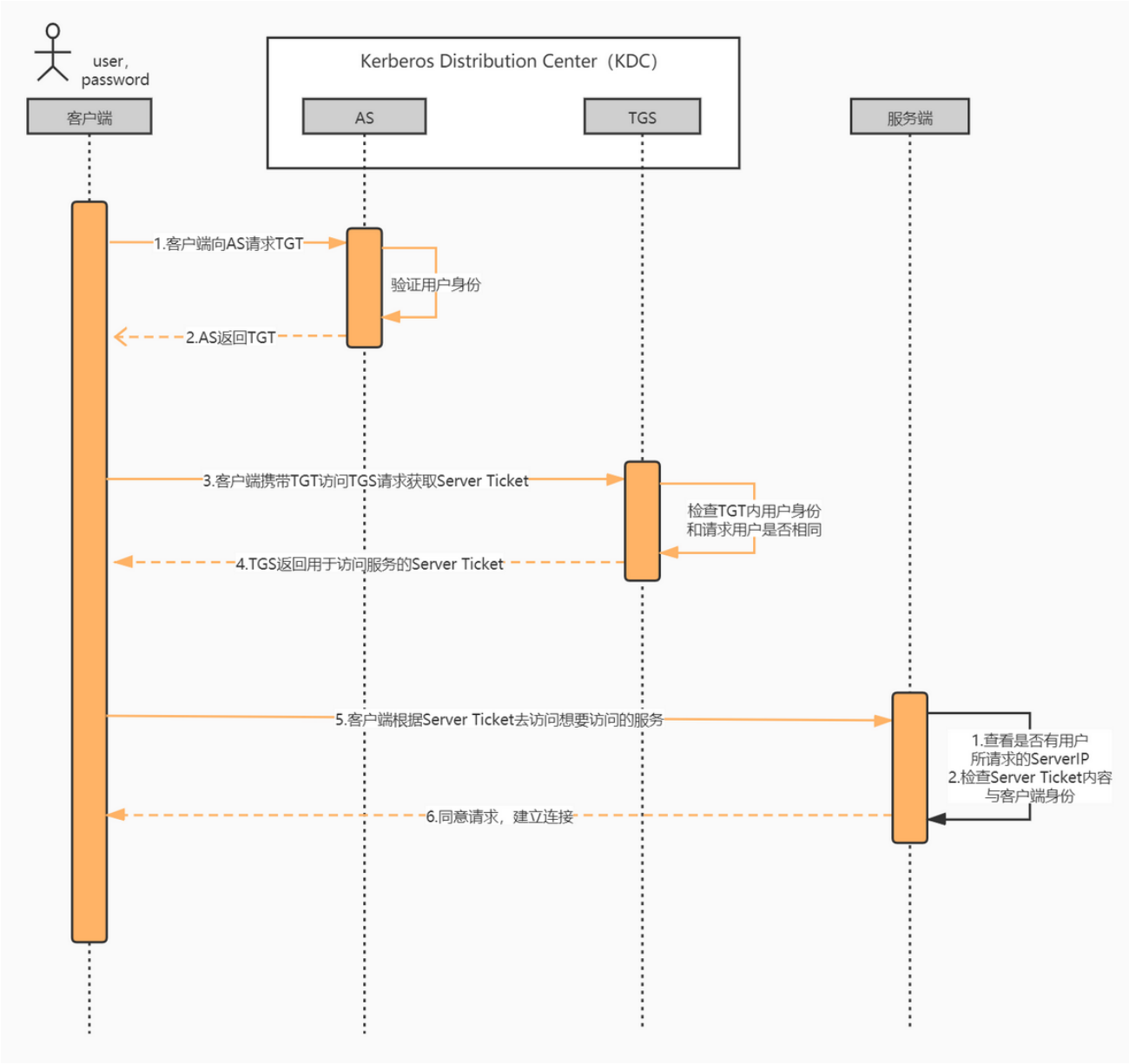
通信第三步-客户端和服务端进行通信

此时的客户端收到了来自KDC（TGS）的响应，并使用缓存在本地的CT_SK解密了第二部分内容（第一部分内容中的ST是由Server密码加密的，客户端无法解密），检查时间戳无误后取出其中的CS_SK准备向服务端发起最后的请求。



客户端：
1、客户端使用CS_SK将自己的主机信息和时间戳进行加密作为交给服务端的第一部分内容，然后将ST（服务授予票据）作为第二部分内容都发送给服务端。
服务端：
1、服务器此时收到了来自客户端的请求，他会使用自己的密钥，即Server密钥将客户端第二部分内容进行解密，核对时间戳之后将其中的CS_SK取出，使用CS_SK将客户端发来的第一部分内容进行解密，从而获得经过TGS认证过后的客户端信息，此时他将这部分信息和客户端第二部分内容带来的自己的信息进行比对，最终确认该客户端就是经过了KDC认证的具有真实身份的客户端，是他可以提供服务的客户端。此时服务端返回一段使用CT_SK加密的表示接收请求的响应给客户端，在客户端收到请求之后，使用缓存在本地的CS_ST解密之后也确定了服务端的身份（其实服务端在通信的过程中还会使用数字证书证明自己身份）。
至此，第三次通信完成。此时也代表着整个kerberos认证的完成，通信的双方都确认了对方的身份，此时便可以放心的进行整个网络通信了。

总体流程如下


user,
password
客户端
Kerberos Distribution Center（KDC）
AS
TGS
服务端
1.客户端向AS请求TGT
验证用户身份
2.AS返回TGT
3.客户端携带TGT访问TGS请求获取Server Ticket
检查TGT内用户身份
和请求用户是否相同
4.TGS返回用于访问服务的Server Ticket
5.客户端根据Server Ticket去访问想要访问的服务
1.查看是否有用户
所请求的ServerIP
2.检查Server Ticket内容
与客户端身份
6.同意请求，建立连接