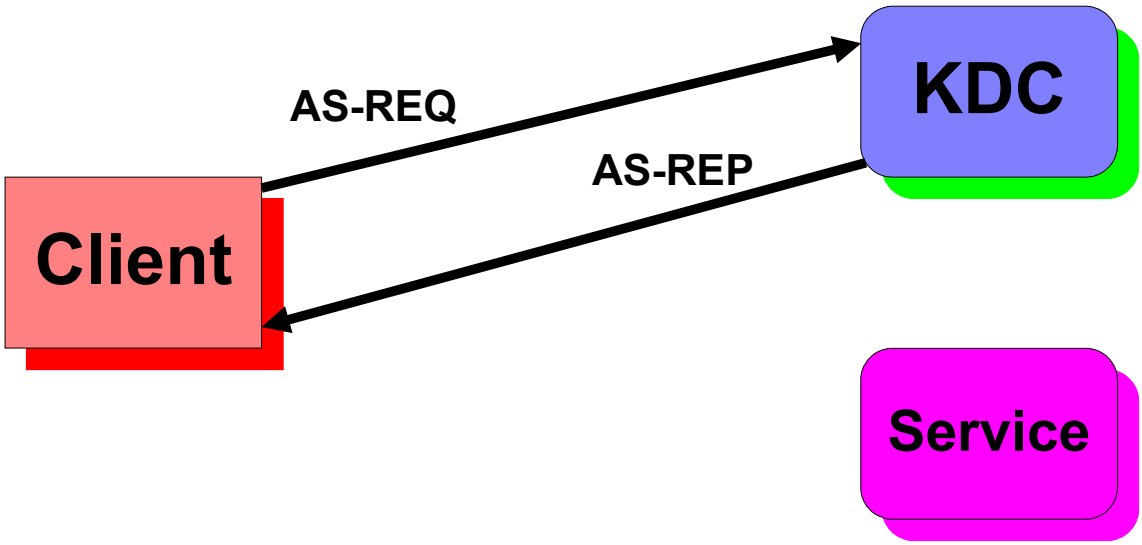
IP fonix

Kerberos Single Sign-On System

One of the greatest advantages of Kerberos is that it can be used as a *single sign-on* authentication system. This means that a client needs to authenticate itself only once to the network in order to use any number of services. In other words, in a Kerberized network, a client could use telephony, video, pay-per-view, VOD and any number of other service offerings without needing to authenticate itself to each service. Instead, the client authenticates itself once, to the network Key Distribution Center (KDC) and then requests *tickets* for other services from the KDC.

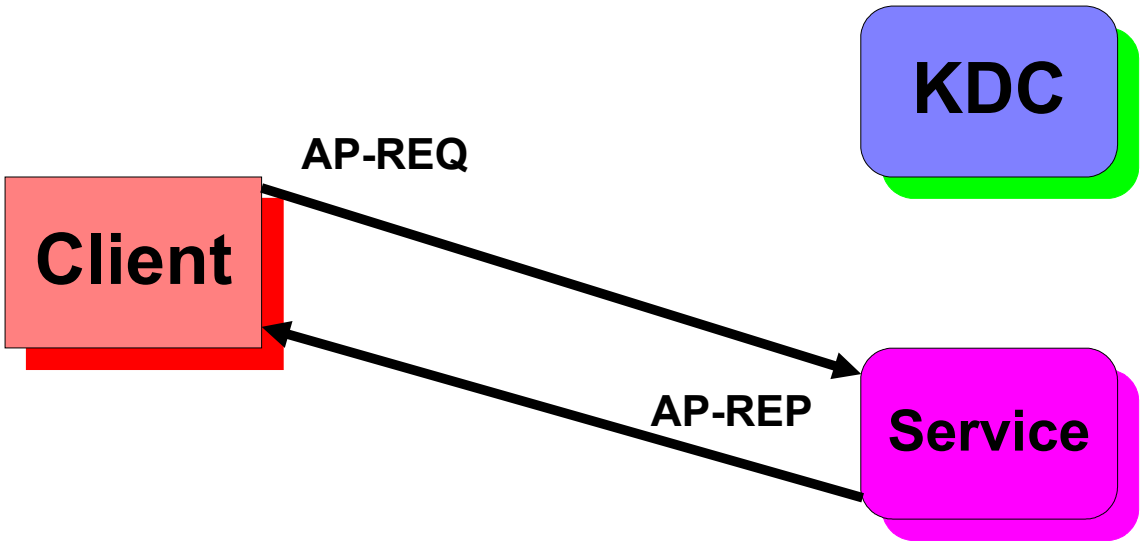
To see how this works, let's first look at the basic Kerberos protocol:



1. The client sends an AS-REQ message to the KDC. The AS-REQ contains the client's authentication credentials. In systems based on PKINIT (such as PacketCable and CableHome), these credentials include a set of digital certificates. Included in the AS-REQ is the identity of the service to which the client desires to gain access.
2. The KDC processes the AS-REQ and performs numerous authentication checks against the message itself and against the certificates contained in the message. This step is computationally expensive. The expensive of performing this step is one reason (there are several others) why authentication should occur on a

dedicated server: a server intended to provide some other real-time service (for example, call routing) should not be expected to cease all processing in order to dedicate the cycles needed to perform an incoming authentication request for a previously-unknown client.

3. The KDC returns an AS-REP message that contains a ticket for the service that was identified in the AS-REQ. The ticket is bound to the client and the service, and has a limited lifetime (typically, in PacketCable and PacketCable-like networks, the lifetime is a few days).



In order to gain access to the desired service, the following sequence then occurs:

1. The client transmits an AP-REQ message to the server. This message includes the ticket that it received in the AS-REP.
2. The service checks the validity of the ticket using lightweight cryptographic functions, and also checks other fields in the AP-REQ to ensure that the ticket is being used correctly.
3. The service returns an AP-REP message to the client. This message includes fields that may be used (for example) to define a security association between the client and the service without the need for further exchanges.

At this point, the client has access to the desired service.

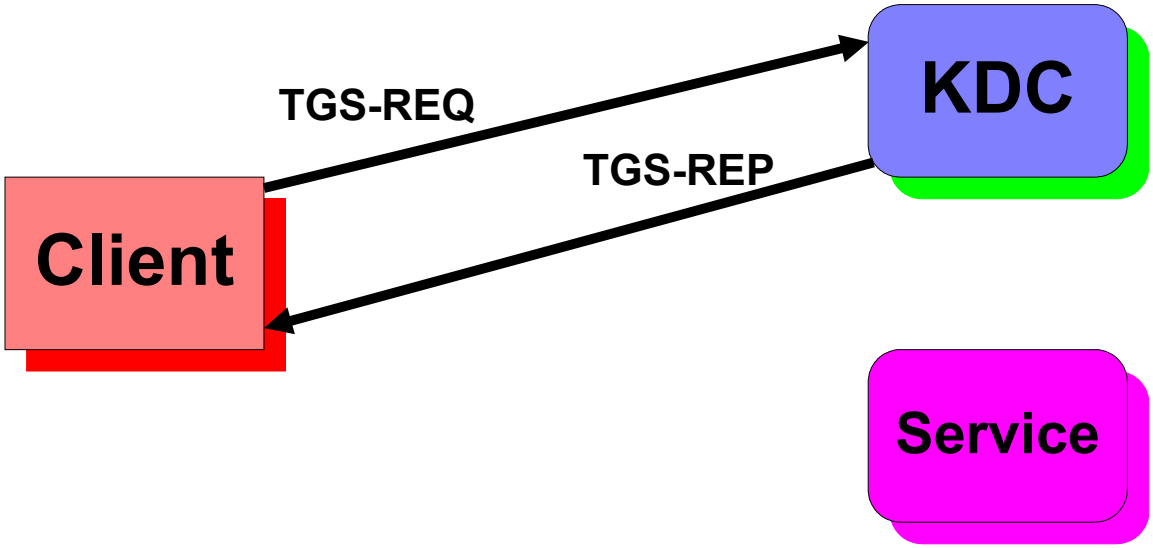
In the simplest form of Kerberos, if the client now wishes to access further services, it performs the complete sequence of exchanges again, beginning with the AS-REQ. In other words, the client authenticates itself to the KDC every time it requires a new ticket (and a new ticket is always required to access a new service). This, of course, is *not* an example of single sign-on, since authentication occurs separately for every service that the client accesses.

In order to support single sign-on, a new service is introduced. This is a *Ticket Granting*

Service, and resides on the KDC.

1. In order to use single sign-on, the client begins by sending an AS-REQ as before, but now it requests a ticket for the Ticket Granting Service rather than a ticket for service that it actually wishes to access.
2. The KDC authenticates the client exactly as before.
3. The KDC returns an AS-REP to the client, exactly as before. This time, since the client requested a ticket for the Ticket Granting Service, that is the ticket returned in the AS-REP.

A new exchange now occurs:



1. The client sends a TGS-REQ to the KDC. A TGS-REQ is basically a special kind of AP-REQ that requests a ticket for another service. In this case, the TGS-REQ contains a request for a ticket for the service that the client wishes to access. The TGS-REQ contains the ticket for the Ticket Granting Service that the KDC returned to the client in the AS-REP.
2. The KDC processes the TGS-REQ much like a service processes an ordinary AP-REQ. In particular, the KDC does not re-authenticate the client, relying instead on the ticket in the TGS-REQ to assure itself that the client has been previously authenticated by the KDC.
3. The KDC returns a TGS-REP to the client. This message includes a ticket for the service that was named in the TGS-REQ (in other words, the ticket for the ultimate service is returned in the TGS-REP, much like the ticket for the service was returned in the AS-REP in the earlier example).

Once the client receives the TGS-REP, it proceeds as before, sending an ordinary AP-REQ to the service.

Now if the client wishes to access another service, it no longer needs to authenticate itself to the KDC with the expensive AS-REQ/AS-REP exchange. Instead, it simply sends

another lightweight TGS-REQ, containing the ticket for the Ticket Granting Service and identifying a new service in the request. The KDC proceeds exactly as before, checking that the received ticket is valid and then issuing a new ticket for the newly-identified service, returning the new ticket to the client in the TGS-REP. The client then forwards the new ticket to the additional service in an AP-REQ. In this way, the client may access as many services as desired without the need to re-authenticate itself to either the KDC or the service.

So, as an example, assume that a client wishes to access three services: VoIP, VoD and PPV. We will call these services S1, S2 and S3 respectively. The following exchanges occur:

1. client sends an AS-REQ to the KDC, requesting a ticket for the Ticket Granting Service;
2. KDC returns an AS-REP, containing a ticket for the Ticket Granting Service;
3. client sends a TGS-REQ to the KDC, containing the ticket for the Ticket Granting Service and requesting a ticket for S1;
4. KDC returns a TGS-REP, containing a ticket for S1;
5. client sends an AP-REQ to S1, containing the ticket for S1;
6. S1 returns an AP-REP. Client now has access to S1.
7. client sends a TGS-REQ to the KDC, containing the ticket for the Ticket Granting Service and requesting a ticket for S2;
8. KDC returns a TGS-REP, containing a ticket for S2;
9. client sends an AP-REQ to S2, containing the ticket for S2;
10. S1 returns an AP-REP. Client now has access to S2.
11. client sends a TGS-REQ to the KDC, containing the ticket for the Ticket Granting Service and requesting a ticket for S3;
12. KDC returns a TGS-REP, containing a ticket for S3;
13. client sends an AP-REQ to S3, containing the ticket for S3;
14. S1 returns an AP-REP. Client now has access to S3.

The client now has access to VoIP, VoD and PPV services having authenticated itself to the network only once, in the initial AS-REQ message.