Kerberos: Structure

Requirements:
- Each user has a private password known only to the user.
- A user's secret key can be computed by a one-way function from the user's password.
- The Kerberos server knows the secret key of each user and the TGS.

Kerberos: Steps

Kerberos: Phase 1
1. The user logs on to the client and the client asks for credentials for the user from Kerberos.
   \[ U \rightarrow C : \ \text{U (user id)} \]
   \[ C \rightarrow K : \ (U, \text{tgs}) \]
2. Kerberos constructs a ticket for U and tgs and a credential for the user and returns them to the client.
   \[ T_{\text{u,tgs}} = E_{K(tgs)} \{ \text{U, tgs, K}_{U,tgs}, \text{ts, lt} \} \]
   \[ K \rightarrow C : \ E_{K(tgs)} \{ T_{U,tgs}, K_{U,tgs}, \text{ts, lt} \} \]
   The client obtains the user's password, P, and computes:
   \[ K(\text{U}) = f(P) \]
   The user is authenticated to the client if and only if \( K(\text{U}) \) decrypts the credential.

Kerberos: Phase 2
3. The client constructs an "attenticator" for user U and requests from TGS a ticket for server, S:
   \[ A_{\text{u}} = E_{K(U,tgs)} \{ C, \text{ts} \} \]
   \[ C \rightarrow TGS : \ (S, T_{U,tgs}, A_{\text{u}}) \]
4. The server authenticates the request as coming from C and constructs a ticket with which C may use S:
   \[ T_{C,S} = E_{K(S)} \{ C, S, K_{C,S}, \text{ts, lt} \} \]
   \[ TGS \rightarrow C : \ E_{K(U,tgs)} \{ T_{C,S}, K_{C,S}, \text{ts, lt} \} \]

Kerberos: Phase 3
5. The client builds an authenticator and sends it together with the ticket for the server to S:
   \[ A_{\text{c}} = E_{K(C,S)} \{ C, \text{ts} \} \]
   \[ C \rightarrow S : \ (T_{C,S}, A_{\text{c}}) \]
6. The server (optionally) authenticates itself to the client by replying:
   \[ S \rightarrow C : \ E_{K(C,S)} \{ \text{ts} + 1 \} \]

Protocol Overview

Ticket Structure:
\[ E_{K(tgs)} \{ C, S, K_{C,tgs}, \text{timestamp, lifetime} \} \]