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
- each server has a secret key known by itself and tgs
Kerberos: Steps

1. **Client (C)** requests authentication from the **Kerberos Server (K)**.
2. The **Kerberos Server (K)** responds with a **Server Ticket (tgs)**.
3. The **User (U)** uses the **Server Ticket (tgs)** to request **authorization** from the **Ticket Granting Server (tgs)**.
4. The **Ticket Granting Server (tgs)** verifies the **Server Ticket (tgs)** and provides a **Client Ticket (Ct) for the requested service**.
5. The **Client (C)** presents the **Client Ticket (Ct)** to the **Server** for access.

**Key Components:**
- **User secret key database**
- **Server secret key database**
Protocol Overview

1. U: user id
2. $T_{u,tgs}$
3. $(T_{u,tgs}, S)$
4. $T_{C,S}$
5. $(T_{C,S}, \text{request})$
6. $T'$

Ticket Structure:

$E_{K(s)} \{C, S, K_{C,S}, \text{timestamp, lifetime}\}$
Kerberos
Phase 1

1. The user logs on to the client and the client asks for credentials for the user from Kerberos:

   \[ U \rightarrow C : \quad U \text{ (user id)} \]
   \[ C \rightarrow K: \quad (U, \text{ tgs}) \]

2. Kerberos constructs a ticket for U and tgs and a credential for the user and returns them to the client:

   \[ T_{u,\text{tgs}} = E_{K(\text{tgs})} \{ U, \text{ tgs}, K_{U,\text{tgs}}, \text{ ts}, \text{ lt} \} \]
   \[ K \rightarrow C : \quad E_{K(U)} \{ T_{U,\text{tgs}}, K_{U,\text{tgs}}, \text{ ts}, \text{ lt} \} \]

The client obtains the user's password, \( P \), and computes:

\[ K'(U) = f(P) \]

The user is authenticated to the client if and only if \( K'(U) \) decrypts the credential.
3. The client constructs an “authenticator” for user U and requests from TGS a ticket for server, S:

\[ \text{A}_U = E_{K(U, \text{tgs})} \{ \text{C, ts} \} \]
\[ \text{C} \rightarrow \text{TGS} : (\text{S, T}_{U, \text{tgs}}, \text{A}_U) \]

4. The server authenticates the request as coming from C and constructs a ticket with which C may use S:

\[ \text{T}_{C, S} = E_{K(S)} \{ \text{C, S, K}_{C, S}, \text{ts, lt} \} \]
\[ \text{TGS} \rightarrow \text{C: } E_{K(U, \text{tgs})} \{ \text{T}_{C, S}, \text{K}_{C, S}, \text{ts, lt} \} \]
Kerberos

Phase 3

5. The client builds an authenticator and send it together with the ticket for the server to $S$:

$$A_C = E_{K(C,S)} \{ C, ts \}$$

$$C \rightarrow S : (T_{C,S}, A_C)$$

6. The server (optionally) authenticates itself to the client by replying:

$$S \rightarrow C : E_{K(C,S)} \{ ts + 1 \}$$