A Private Key System

KERBEROS
**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.
- Each server has a secret key known by itself and tgs.

Diagram:
- Kerberos Server (K) connected to User (U).
- User (U) connected to Client (C).
- Client (C) connected to Ticket Granting Server (tgs).
- User secret key database.
- Server secret key database.
Kerberos: Steps

1. **Kerberos Server (K)**
   - User secret key database

2. **Client (C)**
   - User (U)
   - Server

3. **Ticket Granting Server (tgs)**
   - Server secret key database

**Authentication**
- Kerberos Server (K) authenticates the User (U) based on the user secret key.

**Authorization**
- The User (U) obtains a ticket-granting ticket from the Kerberos Server (K) to access the Ticket Granting Server (tgs).
- The Ticket Granting Server (tgs) authorizes access based on the server secret key.
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 \( \text{tgs} \) and a credential for the user and returns them to the client

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

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

   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:

\[ A_U = E_{K(U,tgs)} \{ C, ts \} \]

C --> TGS : (S, T_{U,tgs}, A_U)

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

\[ T_{C,S} = E_{K(S)} \{ C, S, K_{C,S}, ts, lt \} \]

TGS --> C: E_{K(U,tgs)} \{ T_{C,S}, K_{C,S}, 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:

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