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
Kerberos: Steps

1. **Authentication Phase**
   - **Client (C)** requests a ticket from the **Kerberos Server (K)**.
   - **Kerberos Server (K)** verifies the user's identity using the **user secret key database**.

2. **Client (C) to Server**
   - **Client (C)** sends the ticket to the **Server**.

3. **Server to Ticket Granting Server (tgs)**
   - **Server** requests a ticket-granting ticket (TGT) from the **Ticket Granting Server (tgs)**.
   - **Ticket Granting Server (tgs)** verifies the server's identity using the **server secret key database**.

4. **Ticket Granting Server (tgs) to Client (C)**
   - **Ticket Granting Server (tgs)** sends the TGT to the **Client (C)**.

5. **Authorization Phase**
   - **Client (C)** uses the TGT to request access to a service.
   - **Server** verifies the ticket and grants access.

The process continues with the server handling the service requests and maintaining secure communication.
### Protocol Overview

**Kerberos Protocol Overview**

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

---

**Ticket Structure:**

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

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

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

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

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

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:

\[ A_U = E_{K(U,tgs)} \{ C, ts \} \]
\[ C \rightarrow TGS : (S, T_{U,tgs}, A_U) \]

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

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

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