## 2.3 KERBEROS

- Kerberos is an authentication service developed by MIT and is one of the best known and most widely implemented **trusted third party** key distribution systems.
- Provides a centralized authentication server whose function is to authenticate users to servers and servers to users.
- Kerberos relies exclusively on symmetric encryption, making no use of public-key encryption.

**Kerberos Requirements**

**Secure**: A network eavesdropper should not be able to obtain the necessary information to impersonate a user.

**Reliable**: Kerberos should be highly reliable and should employ a distributed server architecture, with one system able to back up another.

**Transparent**: The user should not be aware that authentication is taking place, beyond the requirement to enter a password.

**Scalable**: The system should be capable of supporting large numbers of clients and servers. This suggests a modular, distributed architecture.

Kerberos is a basic third-party authentication scheme.

**Authentication Server (AS)**

- Knows the passwords of all users and stores these in a centralized database.
- AS shares a unique secret key with each server.
- These keys have been distributed physically or in some other secure manner
- users initially negotiate with AS to identify self
- AS provides a non-corruptible authentication credential (ticket granting ticket TGT)

**Ticket Granting server (TGS)**

- issues tickets to users who have been authenticated to AS

- users subsequently request access to other services from TGS on basis of users TGT

## Simple Authentication Dialogue

(1) $C \rightarrow AS: ID_C||P_C||ID_V$

(2) $AS \rightarrow C$: Ticket

(3) $C \rightarrow V: ID_C||Ticket$

$Ticket = E(K_V, [ID_C||AD_C||ID_V])$

Where

| | | | |
|---|---|---|---|
| C | = client | $ID_V$ | = identifier of V |
| AS | = authentication server | $P_C$ | = password of user on C |
| V | =server | $AD_C$ | = network address of C |
| $ID_C$ | = identifier of user on C | $K_V$ | = secret encryption key shared by AS and V |

Drawback of simple authentication dialogue

- The password Pc is transmitted as a simple plain text. So, there is a possibility of capturing by the attacker.

## More secure authentication Dialogue

Table: Kerberos Version 4 Message Exchanges

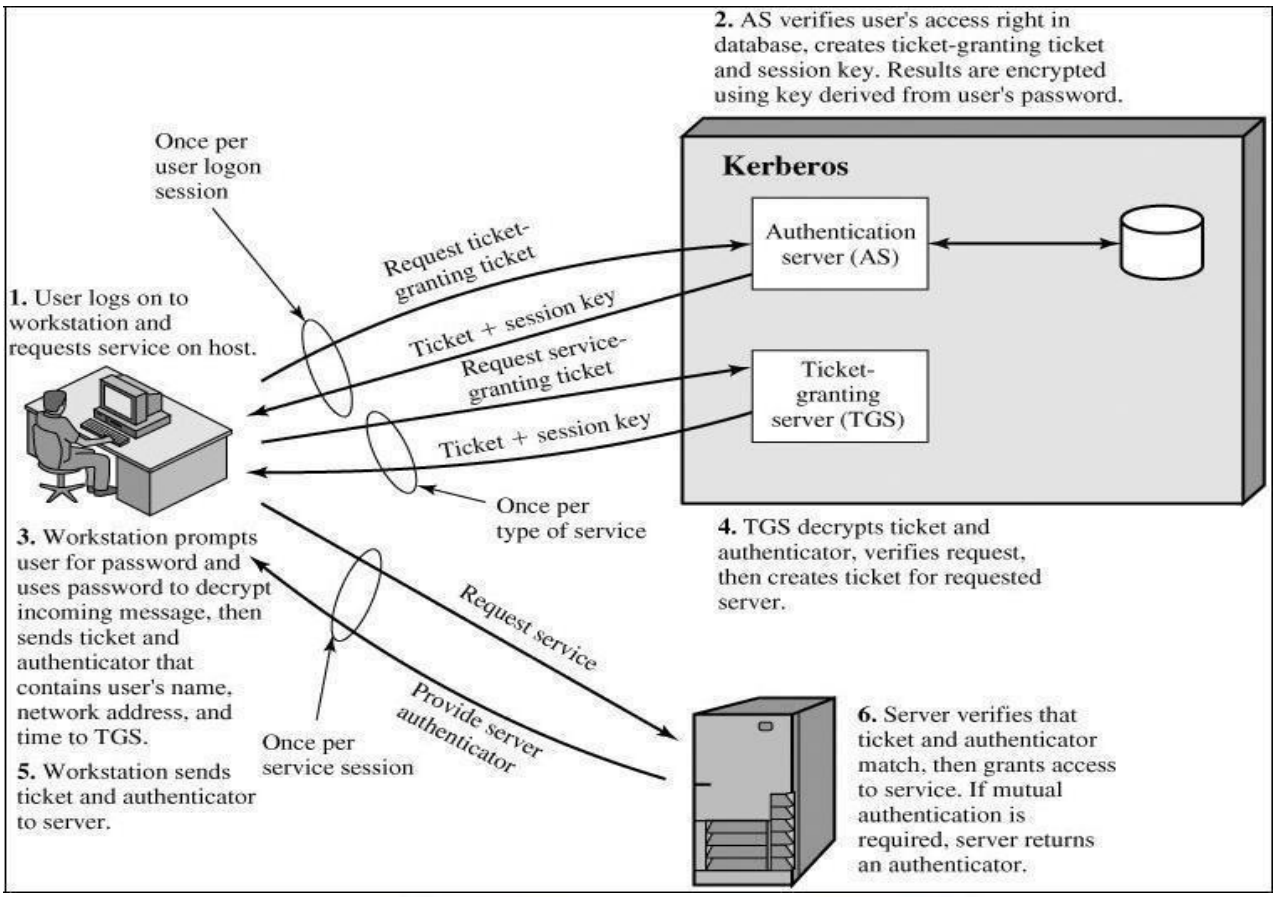
| | | |
|---|---|---|
| (1) | $C \rightarrow AS$ | $ID_c||ID_{tgs}||TS1$ |
| (2) | $AS \rightarrow C$ | $E(K_c,[K_{c,tgs}||ID_{tgs}||TS2||Lifetime2||Ticket_{tgs}])$ |
| | | $Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs}||ID_c||AD_c||ID_{tgs}||TS2||Lifetime2]$ |
| | Authentication Service Exchange to obtain ticket-granting ticket | |
| (3) | $C \rightarrow TGS$ | $ID_v||Ticket_{tgs}||Authenticator_c$ |
| (4) | $TGS \rightarrow C$ | $E(K_{c,tgs}, [K_{c,v}||ID_v||TS4||Ticket_v])$ |
| | | $Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs}||IDC||ADC||ID_{tgs}||TS2||Lifetime2])$ |
| | | $Ticket_V = E(K_V, [K_{c,v}||IDC||ADC||ID_V||TS4||Lifetime4])$ |
| | | $Authenticator_C = E(K_{C,tgs}, [IDC||ADC||TS3])$ |
| | Ticket-Granting Service Exchange to obtain service-granting ticket | |

(5) C →V $Ticket_V || Authenticator_C$

(6) V→C $E(K_{C,V}, [TS5 + 1])$ (for mutual authentication)

$Ticket_V = E(K_V, [K_{C,V} || ID_C || AD_C || ID_V || TS4 || Lifetime4])$

$Authenticator_C = E(K_{C,V}, [ID_C || ADC || TS5])$

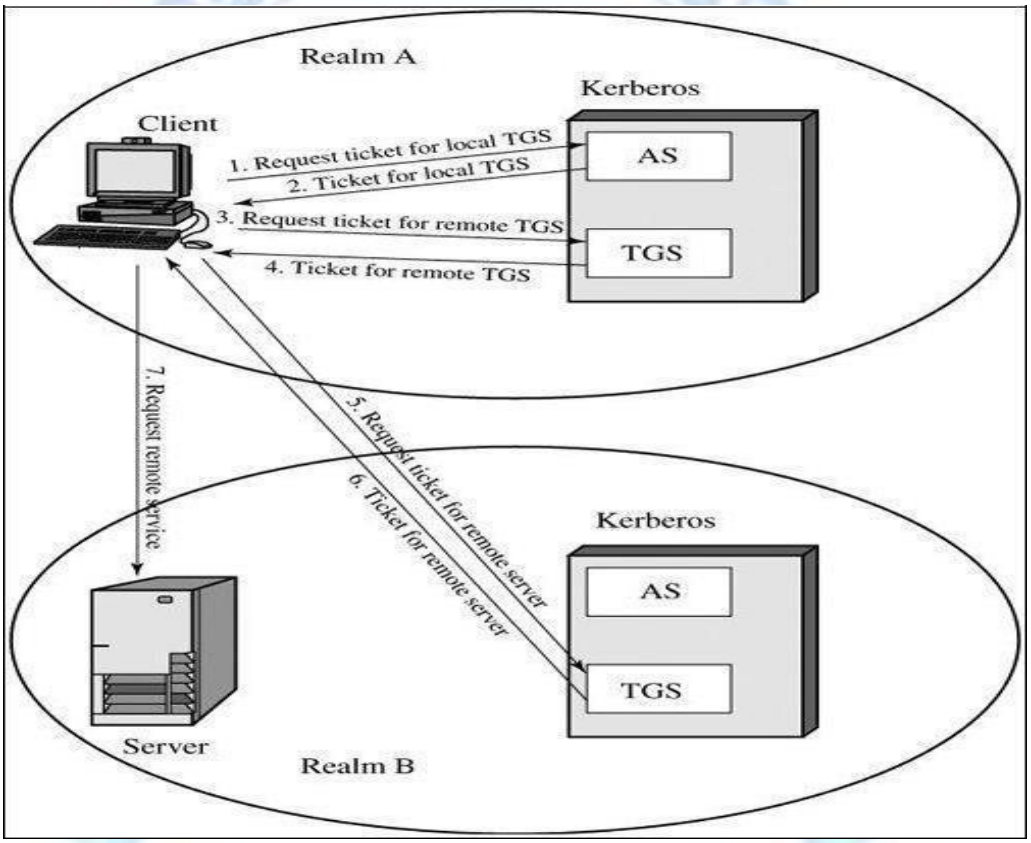



## Overview of Kerberos

- Client sends a message to the AS requesting access to the TGS.
- AS responds with a message, encrypted with a key derived from the user's password ($K_C$) that contains the ticket.
- Encrypted message also contains a copy of the session key, $K_{C,tgs}$, where the subscripts indicate that this is a session key for C and TGS.
- Session key is inside the message encrypted with $K_C$, only the user's

client can read it.

- Same session key is included in the ticket, which can be read only by the TGS.
- Thus, the session key has been securely delivered to both C and the TGS.
- Message (1) includes a timestamp, so that the AS knows that the message is timely.
- Message (2) includes several elements of the ticket in a form accessible to C. This enables C to confirm that this ticket is




for the T

## Kerberos Realms

Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers requires the following:

1. The Kerberos server must have the user ID and hashed

passwords of all participating users in its database. All users are registered with the Kerberos server.

2. The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server.
3. The Kerberos server in each interoperating realm shares a secret key with the server in the other realm. The two Kerberos servers are registered with each other.

Such an environment is referred to as a **Kerberos realm**. The concept of *realm* can be explained as follows. A Kerberos realm is a set of managed nodes that share the same Kerberos database.

**Kerberos principal**, which is a service or user that is known to the Kerberos system. Each Kerberos principal is identified by its principal name. Principal names consist of three parts: a service or user name, an instance name, and a realm name

A user wishing service on a server in another realm needs a ticket for that server. The user's client follows the usual procedures to gain access to the local TGS and then requests a ticket-granting ticket for a remote TGS (TGS in another realm). The client can then apply to the remote TGS for a service-granting ticket for the desired server in the realm of the remote TGS.