CHAPTER 24

Example Implementation

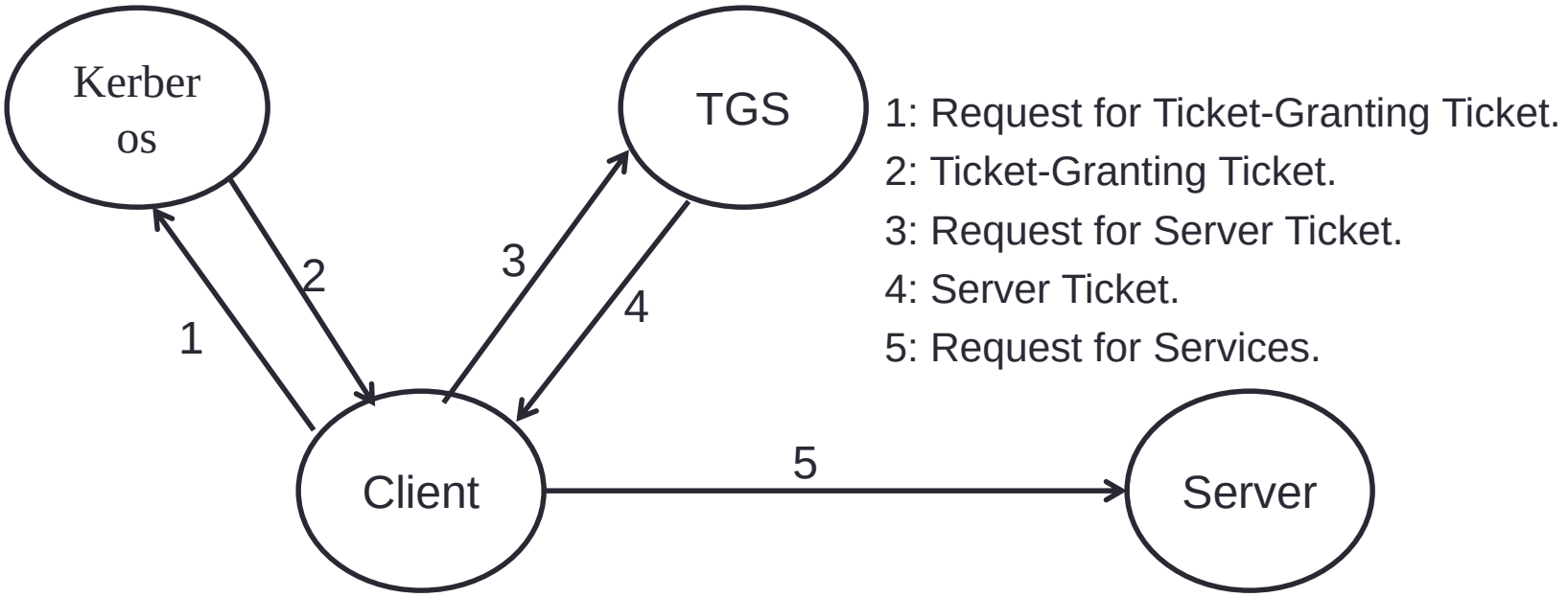
Kerberos

- Kerberos is a trusted third-party authentication protocol designed for TCP/IP network.
- A Kerberos service, sitting on the network, acts as a trusted arbitrator (বিচার/শালিশ).
- Kerberos is based on Symmetric Cryptography.
- It was originally developed at MIT for project Athena.

The Kerberos Model (2014)

- In the Kerberos model, there are entities- clients and servers- sitting on the network.
- Clients can be users, but can also be independent software programs that need to do things: download files, send messages, access databases, access printers, obtain administrative privileges, whatever.
- Kerberos keeps a database of clients and their secret keys. (For a human the secret key is an encrypted password).
- Kerberos also creates session keys which are given to a client and a server (or to two clients) and no one else.
- A session key is used to encrypt messages between two parties, after which it is destroyed.
- Kerberos v.4 provided a nonstandard mode for authentication. This mode is weak: It fails to detect certain changes to the ciphertext.
- Kerberos v.5 uses CBC mode.

Kerberos Authentication steps



How Kerberos Works (2013)

- The Kerberos protocol is straightforward (as shown on the previous slide).
- A client requests a ticket for a Ticket-Granting Service (TGS) from Kerberos.
- The ticket is sent to the client encrypted with the client's secret key.
- To use a particular server, the client requests a ticket for that server from the TGS.
- Assuming everything is in order, the TGS sends the ticket back to the client.
- The client then presents this ticket to the server along with an authenticator.

Kerberos Table of Abbreviations

c = client.

s = server.

a = client's network address.

v = beginning and ending validity time for a ticket.

t = timestamp.

K_x = x's secret key.

$K_{x, y}$ = session key for x and y.

{m} K_x= m encrypted in x's secret key.

$T_{x, y}$ = x's ticket to use y.

$A_{x, y}$ = authenticator from x to y.

Credentials (পরিচয়পত্র) (2013)

- Kerberos uses two types of credentials: tickets and authenticators.

- A ticket is used to pass securely to the server and it is the identitifier of the client for whom the ticket was issued.

- A Kerberos ticket takes this form:

$$T_{c,\,s} = s, \{c, a, v, K_{c,\,s}\}K_s.$$

It contains the servers name, client's name and network address, a timestamp and a session key. The information is encrypted with the server's secret key.

- Once the client gets this ticket , she can use it multiple times to access the server – until the ticket expires.

Credentials Continue…

- A Kerberos authenticator takes this form:

$$A_{c,\,s} = \{c,\, t,\, key\}K_{c,\,s}.$$

The client generates it every time she wishes to use a service on the server. It contains the client's name, a timestamp, and an optional additional session key, all encrypted with the session key shared between the client and the server.

- The authenticator serves two purposes:
  1. It contains some plaintext encrypted with the session key. This proves that it also knows the key.
  2. An eavesdropper who records both the ticket and the authenticator can't replay them two days later.

Kerberos V.5 Messages

Kerberos V.5 has Five messages:

1. Client to Kerberos: c, tgs
2. Kerberos to Client: $\{K_{c,\,tgs}\}\,K_c$, $\{T_{c,\,tgs}\}K_{tgs}$.
3. Client to TGS: $\{A_{c,\,s}\}\,K_{c,\,tgs}$,$\{T_{c,\,tgs}\}K_{tgs}$.
4. TGS to Client: $\{K_{c,\,s}\}\,K_{c,\,tgs}$,$\{T_{c,\,s}\}K_s$.
5. Client to Server: $\{A_{c,\,s}\}\,K_{c,\,s}$, $\{T_{c,\,s}\}K_s$.

Getting Initial Ticket

- The client has one piece of information that proves her identity: her password.
- The client sends a message containing her name and the name of her TGS server to the Kerberos authentication server.
- The Kerberos authentication server looks up the client in the database and upon success Kerberos generates a session key to be used between her and the TGS.
- This is called Ticket Granting Ticket(TGT).
- Kerberos encrypt that session key with that client's secret key.
- Then it creates a TGT for the client to authenticate herself to the TGS, and encrypt that in the TGS's secret key.
- The Kerberos authentication server sends both of these encrypted messages back to the client.

Getting Initial Ticket Continue…

- The client now decrypts the first message and retrieves the session key.
- The secret key is a one way hash of her password, so a legitimate user will have no trouble doing this.
- If the user were an imposter (ছদ্মবেশী), he would not know the correct password and therefore could not decrypt the response from the Kerberos authentication server.
- The client saves the TGT and session key and erases the password and the one way hash.
- The client can now prove her identity to the TGS for the lifetime of the TGS.

Getting Server Tickets (2013)

- A client has to obtain a separate ticket for each service she wants to use.
- The TGS grants tickets for individual servers.
- When a clients needs a ticket that she does not already have, she sends a request to the TGS.
- Upon receiving the request, the TGS decrypts the TGT with his secret key.
- Then he uses the session key included in the TGT to decrypt the authenticator.
- The TGS responds to a valid request by returning a valid ticket for the client to present to the server.
- The TGS also creates a new session key for the client and the server, encrypted with the session key shared by the client and the TGS.

Requesting a Service

- Now the client is ready to authenticate herself to the server.
- She creates a message very similar to the one sent to the TGS.
- The client creates an authenticator consisting of her name and network address, and a timestamp, encrypted with the session key for her and the server that the TGS generated.
- The server decrypts and checks the ticket and the authenticator and also checks the client's address and the timestamp.
- If everything checks out, the server knows that according to Kerberos, the client is who she says she is.

Kerberos V.4 (2014)

In Kerberos V.4 the five messages looked like:

1. Client to Kerberos: c, tgs
2. Kerberos to Client: $\{K_{c, tgs}, \{T_{c, tgs}\}K_{tgs}\} K_c$.
3. Client to TGS: $\{A_{c, s}\} K_{c, tgs}, \{T_{c, tgs}\}K_{tgs}, s$.
4. TGS to Client: $\{K_{c, s}, \{T_{c, s}\}K_s\} K_{c, tgs}$.
5. Client to Server: $\{A_{c, s}\} K_{c, s}, \{T_{c, s}\}K_s$.

   $T_{c, s} = \{s, c, a, v, 1, K_{c, s}\}K_s$

   $A_{c, s} = \{c, a, t\}K_{c, s}$.

Message 1, 3 and 5 are identical. The double encryption of the ticket in steps 2 and 4 has been removed in version 5.

Security of Kerberos (2014)

- Kerberos is vulnerable to password-guessing attacks.
- An intruder can collect tickets and they try to decrypt them.
- Remember that the average person does not usually choose good passwords.
- If Mallory collects enough tickets, his chances of recovering a password are good.
- Perhaps the most serious attack involves malicious software