


Kerberos

Ahmet Burak Can
 Hacettepe University
 abc@hacettepe.edu.tr

Information Security 1

Kerberos

- Kerberos is a **network authentication protocol**. Requirements:
 - Security
 - Reliability
 - Transparency
 - Scalability
- Cryptographic authentication for distributed systems
- Based on symmetric-key authentication with KDC
- Developed at MIT: two versions: Version 4 and Version 5 (specified as RFC1510)
 - <http://web.mit.edu/kerberos/www>

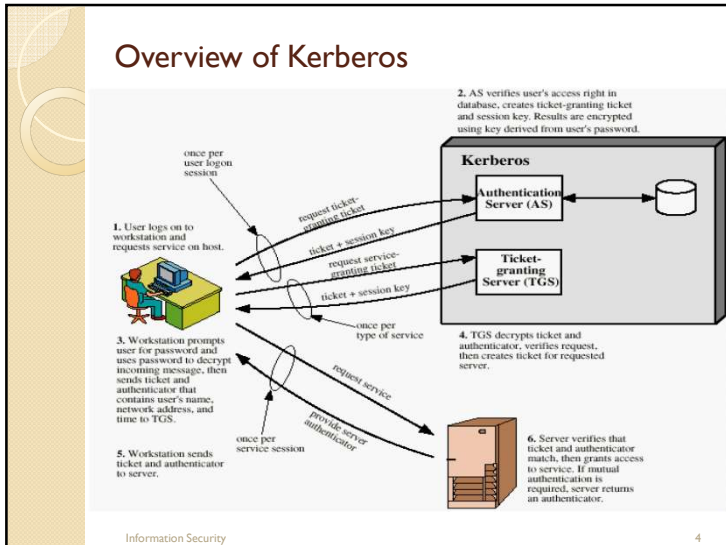


Information Security 2

Kerberos

- Advantages:
 - secure authentication
 - single sign-on
 - secure data flow
- Applications benefiting from Kerberos:
 - telnet, ftp
 - BSD rtools (rlogin, rsh, rcp)
 - NFS
 - Others (pine, eudora, etc.)

Information Security 3



Protocol Design Motivations

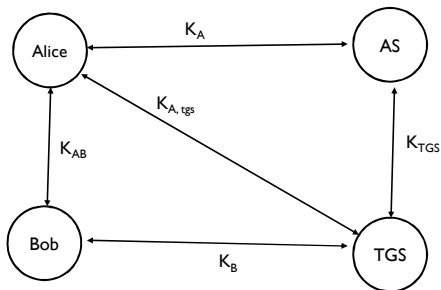
- AS knows passwords for all clients
- AS distributes keys Client-TGS
- TGS distributes keys Client-Server
- Lifetime validity for tickets, include a time validity
- Freshness of messages to prevent replay attacks: use sequence numbers, timestamp or random numbers



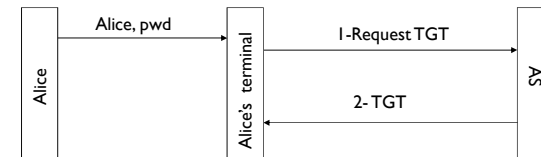
Kerberos Keys

- Each principal shares a “master key” with KDC
 - K_A : Alice’s master key. Used for initial authentication
- K_{TGS} : The key known by AS and the TGS.
- $K_{A, tgs}$: The key shared between the TGS and Alice
- **Ticket Granting Tickets (TGT)**:
 - issued to Alice by AS after login
 - encrypted with K_{TGS}
 - used to obtain session key $K_{A, tgs}$

Key Relation in Kerberos



Logging into the Network



- 1- Alice → AS: $ID_A || ID_{tgs} || TS_1$
- 2- AS → Alice: $E_{K_A} [K_{A,tgs} || ID_{tgs} || TS_2 || Lifetime_2 || Ticket_{tgs}]$

$$Ticket_{tgs} = E_{K_{tgs}} [K_{A,tgs} || ID_A || AD_A || ID_{tgs} || TS_2 || Lifetime_2]$$

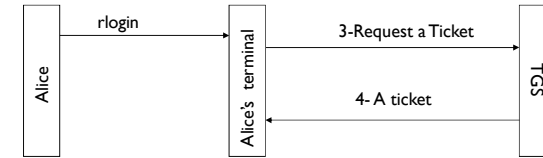
ID_{tgs} denotes the identifier of the Ticket Granting Server (TGS)
 $K_{A, tgs}$ is the key shared by the TGS and Alice
 K_{tgs} key known by AS and the TGS

Logging into the Network

The workstation,

- converts Alice's password into a DES key
- when receives the credentials from the server, decrypts them using this DES key
- if decrypts correctly, authentication is successful
- discards Alice's master key; retains the TGT.
- TGT contains all the information TGS needs about Alice's session; hence TGS can work without remembering any volatile data.

Obtaining a Ticket from TGS



3- Alice → TGS: $ID_B || Ticket_{tgs} || Authenticator_A$

4- TGS → Alice: $E_{K_{A,tgs}} [K_{AB} || ID_B || TS_4 || Ticket_B]$

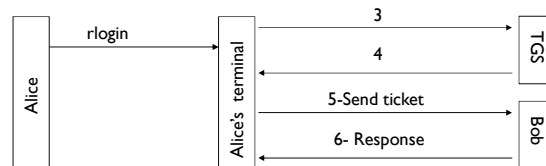
$Authenticator_A = E_{K_{A,tgs}} [ID_A || AD_A || TS_3]$

$Ticket_{tgs} = E_{K_{tgs}} [K_{A,tgs} || ID_A || AD_A || ID_{tgs} || TS_2 || Lifetime_2]$

$Ticket_B = E_{K_B} [K_{AB} || ID_A || AD_A || ID_B || TS_4 || Lifetime_4]$

K_B is the key shared by the TGS and server B

Client-Server Authentication Exchange



5- Alice → Bob: $Ticket_B || Authenticator_A$

6- Bob → Alice: $E_{K_{AB}} [TS_5 + I]$

$Ticket_B = E_{K_B} [K_{AB} || ID_A || AD_A || ID_B || TS_4 || Lifetime_4]$

$Authenticator_A = E_{K_{AB}} [ID_A || AD_A || TS_5]$