

KEY DISTRIBUTION (ASYMMETRIC)

Kerberos

This part is left as self-reading. Please try to go through the material and see if you can understand the material given last 3 weeks of lecture material. The protocol design has several important design principles (examinable). We will discuss these in next lecture. Details of Kerberos is not examinable

SYMMETRIC KEY DISTRIBUTION USING SYMMETRIC ENCRYPTION

- For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others
- Frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key
- Key distribution technique
 - The means of delivering a key to two parties that wish to exchange data, without allowing others to see the key

KERBEROS

- Key distribution and user authentication service developed at MIT
- Provides a centralized authentication server whose function is to authenticate users to servers and servers to users
- Relies exclusively on symmetric encryption, making no use of public-key encryption

Two versions are in use

- Version 4 implementations still exist, although this version is being phased out
- Version 5 corrects some of the security deficiencies of version 4 and has been issued as a proposed Internet Standard (RFC 4120)

KERBEROS VERSION 4

- A basic third-party authentication scheme
- Authentication Server (AS)
 - Users initially negotiate with AS to identify self
 - AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
- Ticket Granting Server (TGS)
 - Users subsequently request access to other services from TGS on basis of users TGT
- Complex protocol using DES

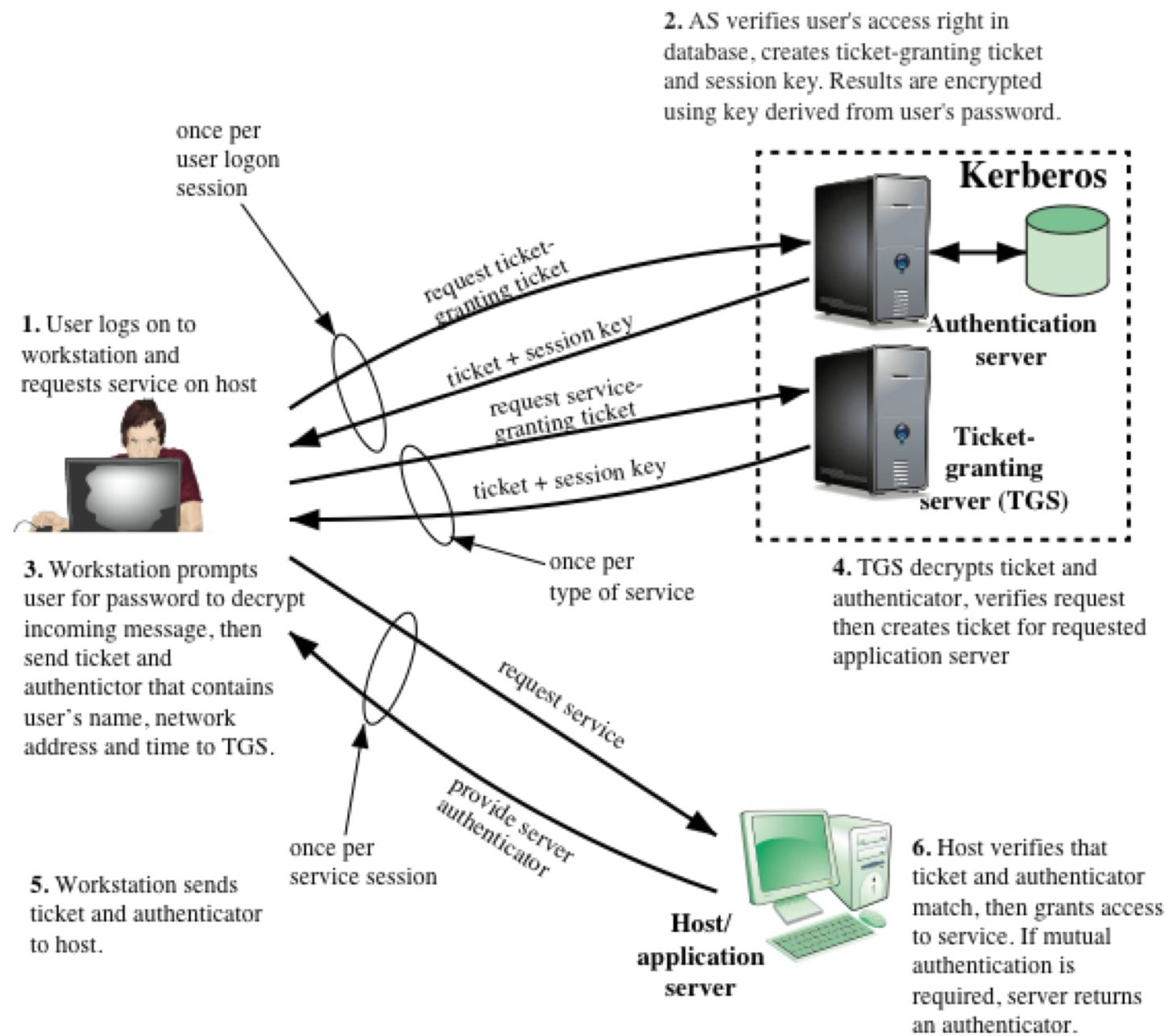


Figure 4.1 Overview of Kerberos

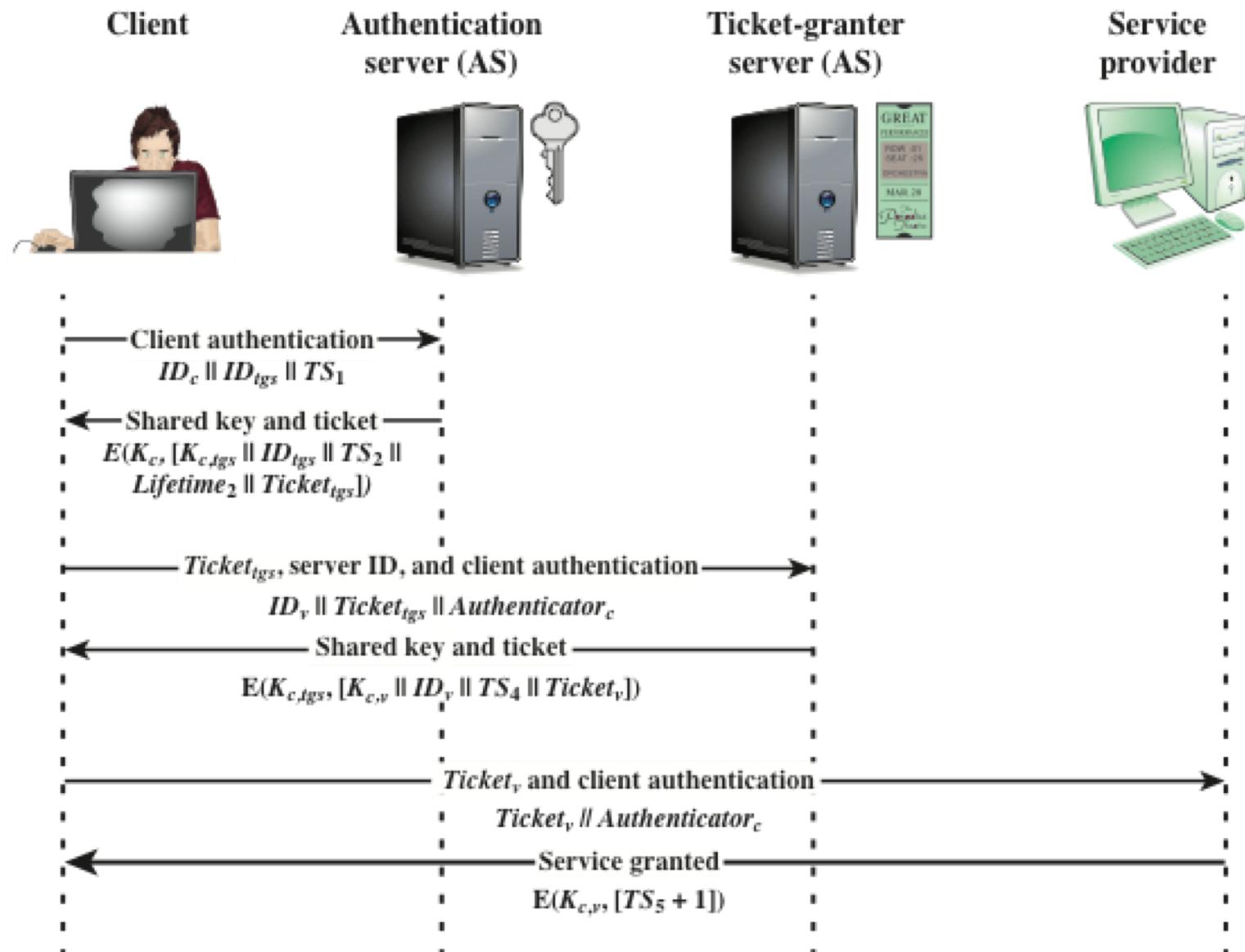


Figure 4.2 Kerberos Exchanges

TABLE 4.1
SUMMARY OF KERBEROS VERSION 4 MESSAGE EXCHANGES

(1) $C \rightarrow AS \quad ID_c \parallel ID_{tgs} \parallel TS_1$ EXAM

(2) $AS \rightarrow C \quad E(K_c, [K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2])$$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) $C \rightarrow TGS \quad ID_v \parallel Ticket_{tgs} \parallel Authenticator_c$

(4) $TGS \rightarrow C \quad E(K_{c,tgs}, [K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2])$$

$$Ticket_v = E(K_v, [K_{c,v} \parallel ID_C \parallel AD_C \parallel ID_v \parallel TS_4 \parallel Lifetime_4])$$

$$Authenticator_c = E(K_{c,tgs}, [ID_C \parallel AD_C \parallel TS_3])$$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) $C \rightarrow V \quad Ticket_v \parallel Authenticator_c$

(6) $V \rightarrow C \quad E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$$Ticket_v = E(K_v, [K_{c,v} \parallel ID_C \parallel AD_C \parallel ID_v \parallel TS_4 \parallel Lifetime_4])$$

$$Authenticator_c = E(K_{c,v}, [ID_C \parallel AD_C \parallel TS_5])$$

(c) Client/Server Authentication Exchange to obtain service

Table 4.2 Rationale for the Elements of the Kerberos Version 4 Protocol
 (page 1 of 3)

Message (1)	Client requests ticket-granting ticket.
ID_C	Tells AS identity of user from this client.
ID_{tgs}	Tells AS that user requests access to TGS.
TS_1	Allows AS to verify that client's clock is synchronized with that of AS.
Message (2)	AS returns ticket-granting ticket.
K_c	Encryption is based on user's password, enabling AS and client to verify password, and protecting contents of message (2).
K_{ctgs}	Copy of session key accessible to client created by AS to permit secure exchange between client and TGS without requiring them to share a permanent key.
ID_{tgs}	Confirms that this ticket is for the TGS.
TS_2	Informs client of time this ticket was issued.
$Lifetime_2$	Informs client of the lifetime of this ticket.
$Ticket_{tgs}$	Ticket to be used by client to access TGS.

(a) Authentication Service Exchange

Table 4.2 Rationale for the Elements of the Kerberos Version 4 Protocol
 (page 2 of 3)

Message (3)	Client requests service-granting ticket.
ID_V	Tells TGS that user requests access to server V.
$Ticket_{tgs}$	Assures TGS that this user has been authenticated by AS.
$Authenticator_c$	Generated by client to validate ticket .
Message (4)	TGS returns service-granting ticket.
$K_{c,tgs}$	Key shared only by C and TGS protects contents of message (4).
$K_{c,v}$	Copy of session key accessible to client created by TGS to permit secure exchange between client and server without requiring them to share a permanent key.
ID_V	Confirms that this ticket is for server V.
TS_4	Informs client of time this ticket was issued.
$Ticket_V$	Ticket to be used by client to access server V.
$Ticket_{tgs}$	Reusable so that user does not have to reenter password.
K_{tgs}	Ticket is encrypted with key known only to AS and TGS, to prevent Tampering.
$K_{c,tgs}$	Copy of session key accessible to TGS used to decrypt authenticator, thereby authenticating ticket.
ID_C	Indicates the rightful owner of this ticket.
AD_C	Prevents use of ticket from workstation other than one that initially requested the ticket.
ID_{tgs}	Assures server that it has decrypted ticket properly.
TS_2	Informs TGS of time this ticket was issued.
$Lifetime_2$	Prevents replay after ticket has expired.
$Authenticator_c$	Assures TGS that the ticket presenter is the same as the client for whom the ticket was issued has very short lifetime to prevent replay.
$K_{c,tgs}$	Authenticator is encrypted with key known only to client and TGS, to prevent tampering.
ID_C	Must match ID in ticket to authenticate ticket.
AD_C	Must match address in ticket to authenticate ticket.
TS_3	Informs TGS of time this authenticator was generated.

(b) Ticket-Granting Service Exchange

Table 4.2 Rationale for the Elements of the Kerberos Version 4 Protocol
 (page 3 of 3)

Message (5)	Client requests service.
$Ticket_V$	Assures server that this user has been authenticated by AS.
$Authenticator_c$	Generated by client to validate ticket.
Message (6)	Optional authentication of server to client.
$K_{c,v}$	Assures C that this message is from V.
$TS_5 + 1$	Assures C that this is not a replay of an old reply.
$Ticket_v$	Reusable so that client does not need to request a new ticket from TGS for each access to the same server.
K_v	Ticket is encrypted with key known only to TGS and server, to prevent Tampering.
$K_{c,v}$	Copy of session key accessible to client; used to decrypt authenticator, thereby authenticating ticket.
ID_C	Indicates the rightful owner of this ticket.
AD_C	Prevents use of ticket from workstation other than one that initially requested the ticket.
ID_V	Assures server that it has decrypted ticket properly.
TS_4	Informs server of time this ticket was issued.
$Lifetime_4$	Prevents replay after ticket has expired.
$Authenticator_c$	Assures server that the ticket presenter is the same as the client for whom the ticket was issued; has very short lifetime to prevent replay.
$K_{c,v}$	Authenticator is encrypted with key known only to client and server, to prevent tampering.
ID_C	Must match ID in ticket to authenticate ticket.
AD_c	Must match address in ticket to authenticate ticket.
TS_5	Informs server of time this authenticator was generated.

KERBEROS REALMS

- Kerberos realm
 - A set of managed nodes that share the same Kerberos database
 - The Kerberos database resides on the Kerberos master computer system, which should be kept in a physically secure room
 - A read-only copy of the Kerberos database might also reside on other Kerberos computer systems
 - All changes to the database must be made on the master computer system
 - Changing or accessing the contents of a Kerberos database requires the Kerberos master password

A Kerberos environment consists of:



A Kerberos server



A number of clients



A number of application servers

DIFFERENCES BETWEEN VERSIONS 4 AND 5

Environmental shortcomings

- Encryption system dependence
- Internet protocol dependence
- Message byte ordering
- Ticket lifetime
- Authentication forwarding
- Interrealm authentication

Technical deficiencies

- Double encryption
- PCBC encryption
- Session keys
- Password attacks

Table 4.3 Summary of Kerberos Version 5 Message Exchanges

(1) $C \rightarrow AS$ $Options \parallel IDc \parallel Realmc \parallel IDtgs \parallel Times \parallel Nonce1$

(2) $AS \rightarrow C$ $Realmc \parallel IDC \parallel Tickettgs \parallel E(Kc, [Kc,tgs \parallel Times \parallel Nonce1 \parallel Realmtgs \parallel IDtgs])$

$Tickettgs = E(Ktgs, [Flags \parallel Kc,tgs \parallel Realmc \parallel IDC \parallel ADC \parallel Times])$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) $C \rightarrow TGS$ $Options \parallel IDv \parallel Times \parallel Nonce2 \parallel Tickettgs \parallel Authenticator_c$

(4) $TGS \rightarrow C$ $Realmc \parallel IDC \parallel Ticketv \parallel E(Kc,v, [Kc,v \parallel Times \parallel Nonce2 \parallel Realmv \parallel IDv])$

$Tickettgs = E(Ktgs, [Flags \parallel Kc,tgs \parallel Realmc \parallel IDC \parallel ADC \parallel Times])$

$Ticketv = E(Kv, [Flags \parallel Kc,v \parallel Realmc \parallel IDC \parallel ADC \parallel Times])$

$Authenticator_c = E(Kc,tgs, [IDC \parallel Realmc \parallel TS1])$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) $C \rightarrow V$ $Options \parallel Ticket_v \parallel Authenticator_c$

(6) $V \rightarrow C$ $E_{Kc,v} [TS_2 \parallel Subkey \parallel Seq\#]$

$Ticketv = E(Kv, [Flags \parallel Kc,v \parallel Realmc \parallel IDC \parallel ADC \parallel Times])$

$Authenticator_c = E(Kc,v, [IDC \parallel Realmc \parallel TS2 \parallel Subkey \parallel Seq\#])$

(c) Client/Server Authentication Exchange to obtain service