

## UNIT IV

### SECURITY PRACTICE & SYSTEM SECURITY

Authentication applications - (2)

Kerberos - (2)

X.509 Authentication Services - (10)

Internet Firewalls for Trusted System - (14)

Roles of firewalls - (15)

Firewall related terminology - (17)

Types of Firewalls - (17)

Firewall designs - (16)

SET for E-commerce Transactions - (18)

Intruder - (19)

Intrusion detection system - (21)

Virus and related threats - (23)

Countermeasures - (27)

Trusted systems - (29)

Security Practice to System Security

Authentication Applications :-

- \* Key concern of security are confidentiality and timelines.
- \* To provide confidentiality one must encrypt identification field & session key.
- \* Developed to support application-level authentication and digital services.
- \* Kerberos - a private key authentication service.
- \* X.509 - a public key directory authentication service.

Kerberos:-

\* Kerberos is a centralized authentication service, whose fun is to authenticate the users to server and server to users.

Problem that kerberos address in this :-

- Workstation cannot be trusted to identify its users correctly to n/w,
- Threats,
  - > Masquerade
  - > Eaves dropping
  - > Replay



## Requirements of Kerberos:-

(3)

> Secure

> Transparent

> Reliable

> Scalable

## Kerberos version 4:-

\* This make use of DES.

\* Simple Authentication Dialogue:-

\* To overcome unauthorized users to access, it means to use an authentication

Server (AS) that stores password of all users and shows a unique secret key with each server.

(1)  $C \rightarrow AS : IP_C \parallel P_C \parallel ID_V$

(2)  $AS \rightarrow C : Ticket$

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

$Ticket = E(k_{AS}, [ID_C \parallel A_{DC} \parallel A_{DV}])$

## Steps:

\* Users logs on to a workstation & request access to server.

\* Ticket is encrypted, so it is not altered by C (or) opponent.

\* The ticket is decrypted by V and verify the user ID.

Adv:

\* AS ticket is encrypted, it prevents alteration by C. (4)

\* Inclusion of ADC in the ticket, avoids attack by an opponent.

Dis adv:

\* Each ticket can be used only once

\* Pwd is used as clear plain text PC, opponent can misuse it.

\* More Secure Authentication Dialogue:

\* This provided by use of ticket granting server (TGS) instead of authentication server (AS).

Scenario:-

once per user login session:

(1)  $C \rightarrow AS: ID_C \parallel ID_{TGS}$

1. TGS authenticates client & provide ticket

(2)  $AS \rightarrow C: E(k_c, Ticket_{TGS})$

2. ticket encrypted with session key, client will decrypt it

once per type of service:

(3)  $C \rightarrow TGS: ID_C \parallel ID_V \parallel Ticket_{TGS}$

(4)  $TGS \rightarrow C: Ticket_V$



once per service session:

(5)

(5)  $C \rightarrow V: Tpc \parallel Ticket_V$ .

Encrypted ticket shared by server & TGS.

Ticket contains following information:

$Ticket_V = E(K_V, [ID_C \parallel AD_C \parallel ID_V \parallel TS_2 \parallel Lifetime_2])$

$Ticket_{TGS} = E(K_{TGS}, [ID_C \parallel AD_C \parallel ID_{TGS} \parallel TS_1 \parallel Lifetime_1])$

Adv:-

- \* Ticket reusability.
- \* Protection of user password.
- \* Timestamps indication of issuing tickets date and time.
- \* Encryption TGS and tv prevents forgery.

\* V4 Authentication Dialogue:-

- \* Combination of simpler & more secured authentication.

V4 Message Exchanges:-

- a) Authentication server exchange to obtain ticket-granting ticket.

(1)  $C \rightarrow AS: ID_C \parallel ID_{TGS} \parallel TS_1$

(2) AS  $\rightarrow$  C:  $E(K_c, [K_c, E_{g_s}(ID_{g_s} || TS_2 || Lifetime_2 || Ticket_{g_s})])$  (6)

Ticket - Granting service exchange to obtain service-granting ticket.

(3) C  $\rightarrow$  TGS:  $ID_v || Ticket_{g_s} || Authenticator_C$

(4) TGS  $\rightarrow$  C:  $E(K_c, E_{g_s}(K_{c,v} || ID_v || TS_4 || Ticket_v))$

$Ticket_{g_s} = E(K_{g_s}, [K_{c,g_s}(K_{c,g_s} || ID_c || AD_c || ID_{g_s} || TS_2 || Lifetime_2)])$

$Ticket_v = E(K_v, [K_{c,v}(K_{c,v} || ID_c || AD_c || ID_v || TS_4 || Lifetime_4)])$

$Authenticator_c = E(K_c, E_{g_s}(ID_c || AD_c || TS_3))$

Client | server authenticate exchange to obtain server:

(5) C  $\rightarrow$  V:  $Ticket_v || Authenticator_c$

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

(for mutual authentication)

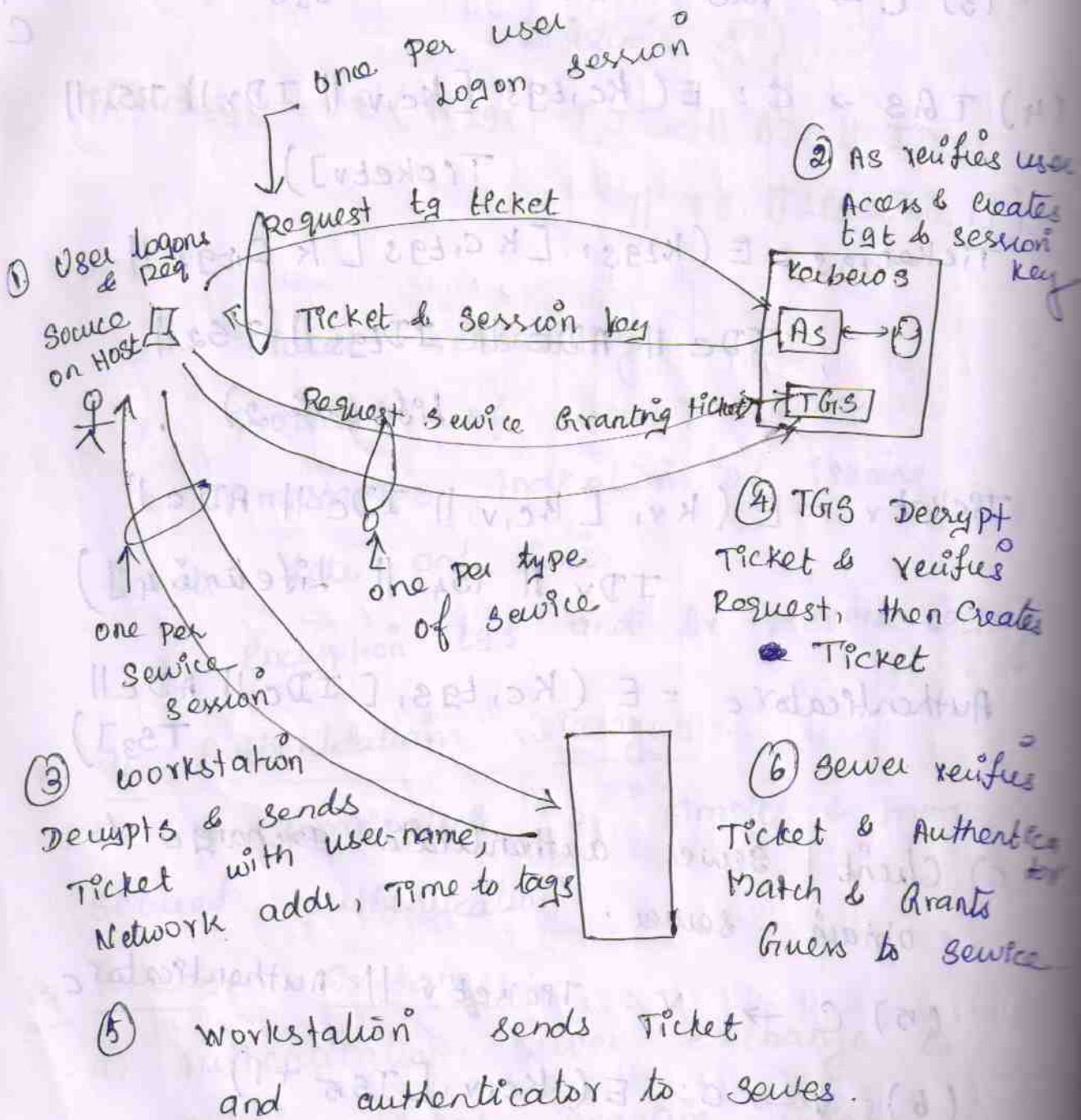


$$\text{Ticket}_v = E(k_{v,i} [k_{c,v} || ID_c || AD_c ||$$

$$ID_v || TS_4 || \text{Lifetime}_4])$$

$$\text{Authenticator}_c = E(k_{c,v} [ID_c || AD_c || TS_5])$$

### Overview of Kerberos: -



## Kerberos version 5 :-

8

- \*) Encryption system dependence
- \* Internet protocol "
- \* msg byte ordering
- \* Ticket lifetime
- \* Authentication forwarding is not in v4.
- \* Inter-realm authentication?

### v5 Message Exchanges :-

a) Authentication source exchange to obtain tgs :

(1)  $C \rightarrow AS : options \parallel ID_c \parallel Realm_c \parallel ID_{tgs} \parallel Times \parallel Nonce_1$

(2)  $AS \rightarrow C : Realm_c \parallel ID_c \parallel Ticket_{tgs} \parallel E(k_{c1} [k_{c1} [k_{tgs} \parallel Times \parallel Nonce_1 \parallel Realm_{tgs} \parallel ID_{tgs} ]])$

$Ticket_{tgs} = E(k_{tgs} [Flags \parallel k_{c1} \parallel ID_c \parallel AD_c \parallel Times])$

b) Ticket-granting source exchange to obtain source-granting ticket :-

(3)  $C \rightarrow TGS : options \parallel ID_v \parallel Times \parallel$

$Nonce_2 \parallel Ticket_{tgs} \parallel Authenticator_c$



(4) TGS → C : Realmc || IDc || Ticketv ||

$$E(K_{ctgs}, [K_{cv} || Times || Nonce || Realmv || IDv])$$

$$Ticket_{tgs} = E(K_{tgs}, [flags || K_{ctgs} || Realmc || IDc || ADc || Times])$$

$$Ticket_v = E(K_v, [flags || K_{cv} || Realmc || IDc || ADc || Times])$$

$$Authenticator_c = E(K_{ctgs} (IDc || Realmc || TS1))$$

c) Client | Server Authenticate exchange to obtain service :

(5) C → V : Options || Ticket v || authenticator

(6) V → C : E(K\_{cv} [TS2 || subkey || seq #])

$$Ticket_v = E(K_v, [flags || K_{cv} || Realmc || IDc || ADc || Times])$$

$$Authenticator_c = E(K_{cv} [IDc || Realmc || TS2 || subkey || seq #])$$

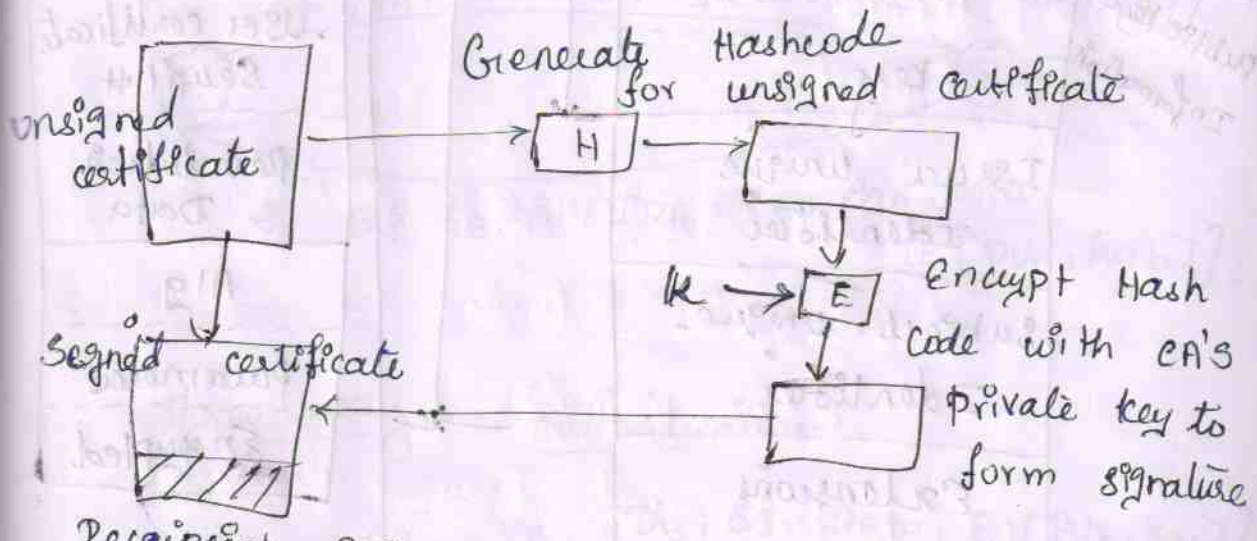
X.509 Authentication Service :-

(10)

X.509 defines format for public-key certificates. This format is widely used in variety of applications, that are,

- \* S/MIME
- \* SSL/TLS
- \* IP Security
- \* SET.

Public-key certificate :-



Recipient can verify signature using CA's public-key.

Certificate:

$$CA \langle\langle A \rangle\rangle = CA \{ V, SN, AI, CA, TA, A, AP \}$$



