

CS355: Cryptography

Lecture 34: Kerberos.

What is Kerberos?

- Kerberos is a **network authentication protocol**
- Provides authentication for client-server applications, and data integrity and confidentiality
- Relies entirely on **symmetric cryptography**
- Developed at MIT: two versions, Version 4 and Version 5 (specified as RFC1510)
- <http://web.mit.edu/kerberos/www>
- Version 5 used in Windows 2000



Kerberos Overview

- Client wants service from a particular server
- An Authentication Server allows access
- How? Based on tickets
- **Ticket**: specifies that a particular client (authenticated by the Authentication Server) has the right to obtain service from a specified server S
- **Realm**: network under the control of an Authentication Server

Basic Authentication Protocol

$C \rightarrow AS: ID_c \parallel P_c \parallel ID_s$

$AS \rightarrow C: Ticket$

$C \rightarrow S: ID_c \parallel Ticket$

$Ticket = E_{K_s} [ID_c \parallel P_c \parallel ID_s]$

- ID represents identifiers
- P_c represents password of client
- E denotes encryption
- K_s is a key shared by Authentication Server AS and server S

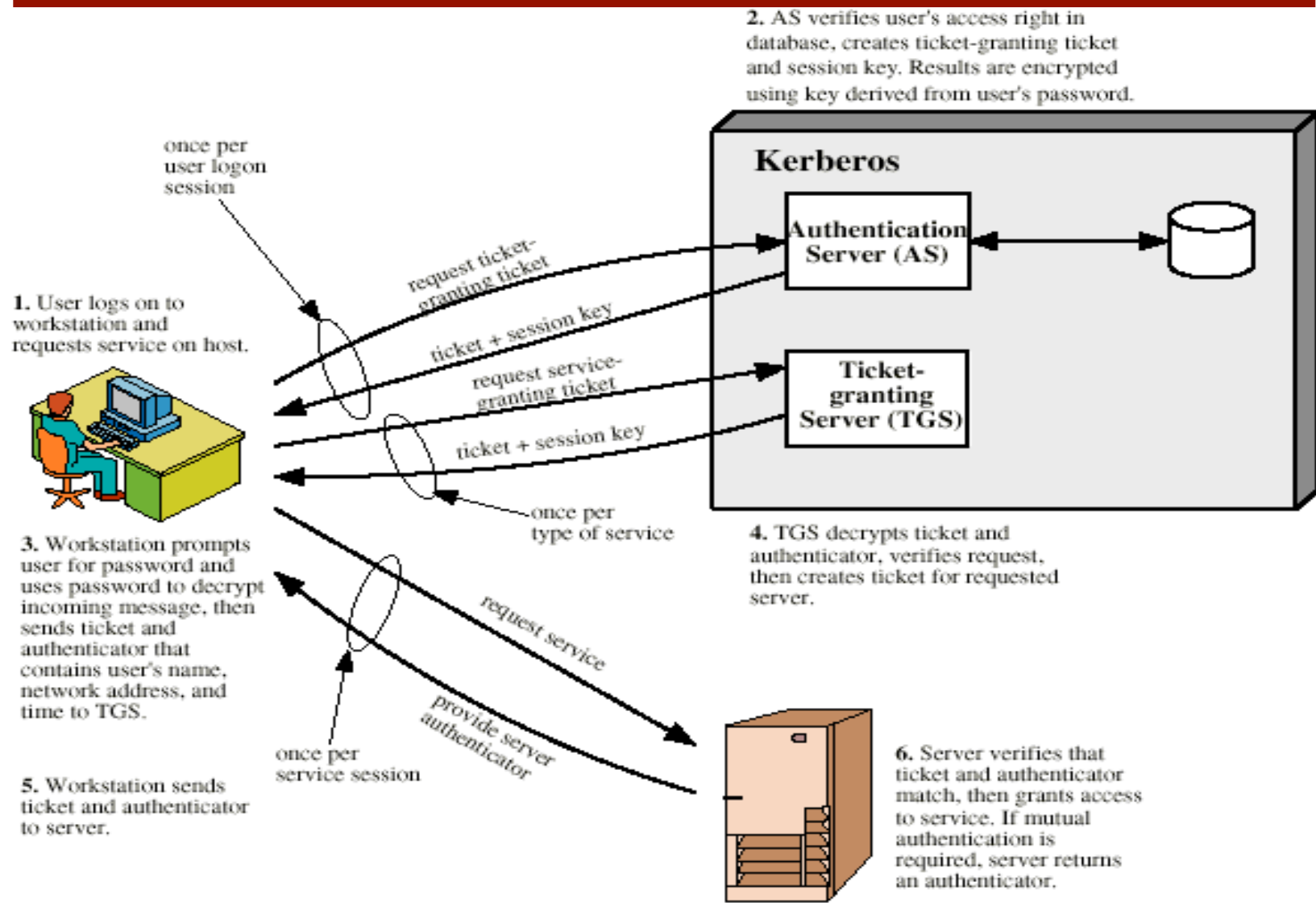
Vulnerabilities

- How long is the ticket valid?
- **Ticket valid for a short time:** then the client needs to come back and ask for another one
- **Ticket valid for a longer time:** if the ticket is stolen, somebody can reuse it before expiration
- **Replay attack:** the first message can be stored and replay later. There is no time indication associated with the ticket and the request.

Improved Authentication Protocol

- Use two type of tickets with two different lifetimes:
 - One ticket grants to right to ask for service; performed once per login session Ticket_{tgs}
 - For each type of service, use a ticket that grants the right to use that particular service Ticket_s
 - Every time that service is needed, used the ticket Ticket_s
- Mark time when tickets are issued and also lifetime of tickets.

Overview of Kerberos



V4: Authentication Service Exchange

Goal: Obtain Ticket-Granting Ticket

$C \rightarrow AS: ID_C \parallel ID_{tgs} \parallel TS_1$

$AS \rightarrow C: E_{K_C} [K_{C,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}]$

$Ticket_{tgs} = E_{K_{tgs}} [K_{C,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$

ID_{tgs} denotes the identifier of the Ticket Granting Server (TGS)

TS_1 and TS_2 are timestamps

K_C is the key shared by the AS and client C

$K_{C,tgs}$ is the key shared by the TGS and client C

K_{tgs} key known by AS and the TGS

$Ticket_{tgs}$...is the ticket

Lifetime is the validity of the ticket

AD is address identifier

V4: Ticket-Granting Service Exchange

Goal: Obtain Service-Granting Ticket

C → TGS: $ID_S \parallel Ticket_{tgs} \parallel Authenticator_C$

TGS → C: $E_{K_{C,tgs}} [K_{C,S} \parallel ID_S \parallel TS_4 \parallel Ticket_S]$

$Ticket_{tgs} = E_{K_{tgs}} [K_{C,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$

$Ticket_S = E_{K_S} [K_{C,S} \parallel ID_C \parallel AD_C \parallel ID_S \parallel TS_4 \parallel Lifetime_4]$

$Authenticator_C = E_{K_{C,tgs}} [ID_C \parallel AD_C \parallel TS_3]$

K_S is the key shared by the TGS and server S

V4: Client-Server Authentication Exchange

Goal: Obtain Service

$C \rightarrow S:$ $\text{Ticket}_S \parallel \text{Authenticator}_C$

$S \rightarrow C:$ $E_{K_{C,S}} [\text{TS}_5 + 1]$

$\text{Ticket}_S = E_{K_S} [K_{C,S} \parallel \text{ID}_C \parallel \text{AD}_C \parallel \text{ID}_S \parallel \text{TS}_4 \parallel \text{Lifetime}_4]$

$\text{Authenticator}_C = E_{K_{C,S}} [\text{ID}_C \parallel \text{AD}_C \parallel \text{TS}_5]$

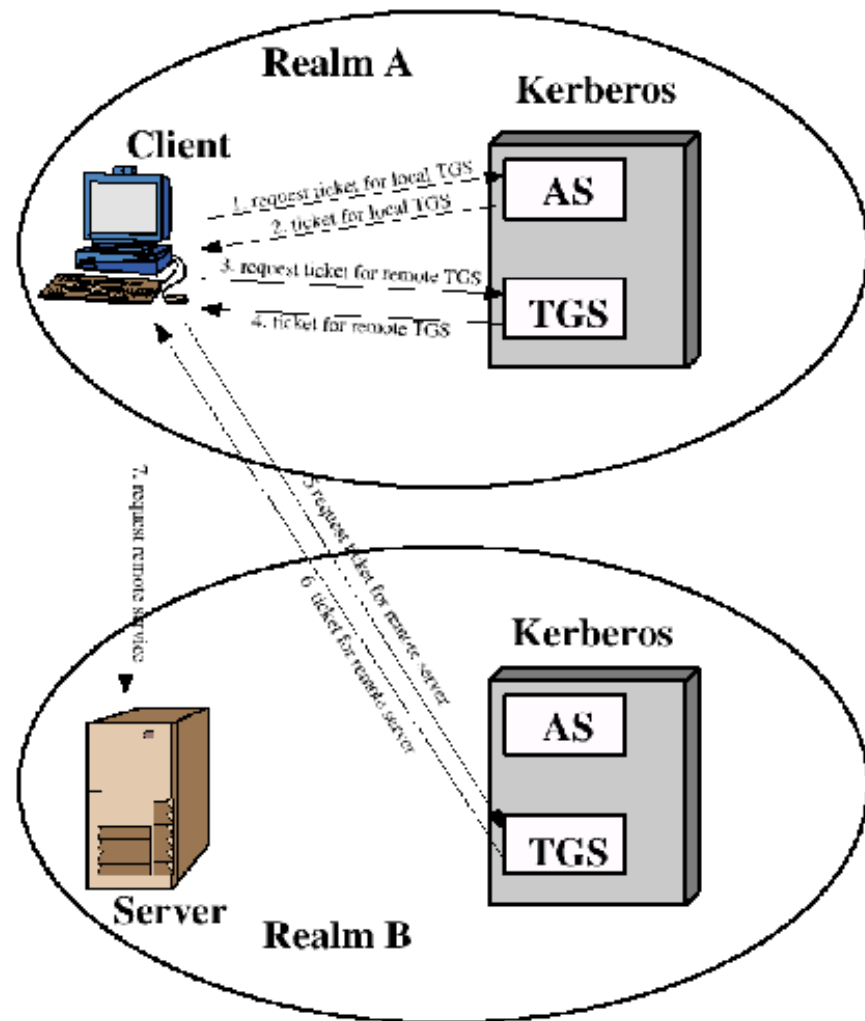
Protocol Design Motivations

- AS knows passwords for all clients
- Confidentiality, requires shared keys: include shared keys distribution in the protocols
- AS distributes keys C-TGS
- TGS distributes keys C-S
- Lifetime validity for tickets, include a time validity
- Freshness of messages to prevent replay attacks: use sequence numbers, timestamp or random numbers



Request for Service in Another Realm

- Authenticate to local AS and obtain ticket to local TGS
- Ask local TGS for ticket for remote TGS, obtain ticket for remote TGS
- Ask remote TGS for ticket for remote server S, obtain ticket for remote server S
- Ask for service from remote server S



Kerberos Version 4 vs. Version 5

- Version 5 overcomes certain deficiencies in Version 4: environment and technical (1994)
- Environment:
 - V4 was using DES as encryption and there were restrictions; not general enough
 - Depending on IP, modify such that any network type address can be used
 - Message byte ordering; move the standards that provide unambiguous byte ordering
 - Ticket lifetime: V5 allows arbitrary lifetimes
 - Inter-realm authentication: V4 requires N^2 keys; V5 is better

Kerberos Version 4 vs. Version 5

- Technical:
 - V5 eliminates one unnecessary encryption
 - V4 was using a non-standard DES encryption mode that was found vulnerable; V5 uses CBC mode.
 - Use sub-session keys
 - Include a pre-authentication protocol that makes password attacks more difficult

V5: Authentication Service Exchange

Goal: Obtain Ticket-Granting Ticket

$C \rightarrow AS$: Options || ID_C || $Realm_C$ || ID_{tgs} || Times || $Nonce_1$

$AS \rightarrow C$: $Realm_C$ || ID_C || $Ticket_{tgs}$ || $E_{K_C} [K_{C,tgs} || Times ||$
 $Nonce_1 || Realm_{tgs} || ID_{tgs}]$

$Ticket_{tgs} = E_{K_{tgs}} [Flags || $K_{C,tgs}$ || $Realm_C$ || ID_C || AD_C || Times]$

V5: Ticket-Granting Service Exchange

Goal: Obtain Service-Granting Ticket

C → TGS: Options || ID_S || Times || Nonce₂ || Ticket_{tgs} ||
Authenticator_C

TGS → C: Realm_C || ID_C || Ticket_S || E_{K_{C,tgs}} [K_{C,S} || Times ||
Nonce₂ || Realm_S || ID_S]

Ticket_{tgs} = E_{K_{tgs}} [Flags || K_{C,tgs} || Realm_C || ID_C || AD_C || Times]

Ticket_S = E_{K_S} [Flags || K_{C,S} || Realm_C || ID_C || AD_C || Times]

Authenticator_C = E_{K_{C,tgs}} [ID_C || Realm_C || TS₁]

V5: Client-Server Authentication Exchange

Goal: Obtain Service

$C \rightarrow S:$ Options || Ticket_S || Authenticator_C

$S \rightarrow C:$ $E_{K_{C,S}} [TS_2 || Subkey || Seq#]$

$Ticket_S = E_{K_S} [Flags || K_{C,S} || Realm_C || ID_C || AD_C || Times]$

$Authenticator_C = E_{K_{C,S}} [ID_C || Realm_C || TS_2 || Subkey || Seq#]$