# Kerberos V4

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

Audio/Video recordings of this lecture are available at:

http://www.cse.wustl.edu/~jain/cse571-07/

# Overview

- What is Kerberos?
- Kerberos V4 Concepts and Design Principles
- Replicated KDCs
- Multiple Realms
- Other details

# Overview of Kerberos

- Allows two users (or client and server)to authenticate eacn other over an insecure network
- Named after the Greek mythological character *Kerberos* (or *Cerberus*), known in Greek mythology as being the *monstrous three-headed guard dog of Hades*
- Designed originally for Project Athena at M.I.T.
- Implementation freely available from M.I.T.
- V5 is proposed as an Internet Standard (RFC 4120)
- Windows 2000/XP/Server 2003/Vista use Kerberos as their default authentication mechanism
- Apple's Mac OS X clients and servers also use Kerberos
- Apache HTTP Server, Eudora, NFS, OpenSSH, rcp (remote copy), rsh, X window system allow using Kerberos for authentication.

# Overview (Cont)

- Protects against eavesdropping and replay attacks
- Uses a trusted third party (Key Distribution Center) and symmetric key cryptography
- First 3 versions are no longer in use.
- V5 is a generalization of V4 with several problems fixed and additional features.
- It is easier to understand V5 if you know V4
- Learn V4's features and mistakes

# Sample Kerberos Exchange

Client

Authentication Server (KDC)

1. Hi! Jain@cse would like to use the network today
2. Here is a day pass for jain@cse
3. Hi! Jain@cse would like to communicate with PrintServer. Attached is his day pass.
4. Here is the ticket for jain@cse to communicate with PrintServer. It includes a session key.
5. Hi jain@cse wants to communicate with you.. Here is his ticket.
6. Perfect. Let us use the session key that was in your ticket.

Ticket Granting Server (TGS)

Service Server

# Kerberos V4 Concepts

- **Key Distribution Center (KDC):** Physically secure node with complete authentication database
- **Principal**: Authentication Server A, Ticket Granting Server G, Client (Computer) C, User (Human) U, Server S
- **Ticket Granting Server (TGS)**
- **Keys**: $K_{cg}$, $K_{cs}$, $K_{ag}$, $K_u$, $K_{gs}$
- **Ticket**: Encrypted information. All current V4 implementations use DES.
- **Ticket Granting Ticket (TGT):** Allows user to get tickets from TGS

# Concepts (Cont)

- **Authenticator**: Name and time encrypted with a session key. Sent from client to server with the ticket and from server to client.
- **Credentials**: Session key + Ticket
- **Session**: One user login/logout session
- User enters a name and password. Client converts the password to a key $K_u$.
- TGT and the session key are good for a limited time (21 hours).

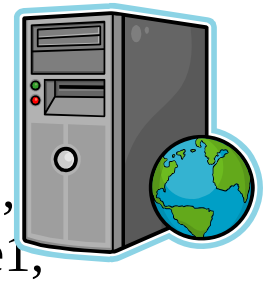
# Key Design Principles

1. The network is open ⇒ Need a proper secret key to understand the messages received
   (except message 1, which is in clear)
2. Every client and server has a pre-shared secret with the KDC.
3. KDC and Ticket Granting Server (TGS) are logically separate but share a secret key
4. Both KDC and TGS are stateless and do not need to remember the permissions granted. All the state is in the tickets. (Day pass is just a longer term ticket)
5. Longer term secrets are used less frequently. Short term secrets are created and destroyed after a limited use.

# Information Exchanged

Client

1 User, TGS



KDC

2 User, {User, Client, TGS, Kcg, SesStartTime1, SesEndTime1}Kag, {TGS, Kcg, SesStartTime1, SesEndTime2}Ku

3 Service, SvcNonce4, {User, Client, TGS, Kcg, SesSartTime1, SesEndTime1}Kag, {Client, SesTime1}Kcg

TGS

4 User, {User, Client, Service, Kcs, SvcStartTime2, SvcEndTime2}Kgs, {Service, Kcs, SvcStartTime2, SvcEndTime2, SvcNonce4}Kcg

5 {User, Client, Service, Kcs, SvcStartTime2, SvcEndTime2}Kgs, {Client, SvcTime2}Kcs

Service Server

6 {SvcTime2}Kcs.

# Kerberous Protections

- Kerberos protects against eavesdropping:
  - If someone else sends TGT, they get back a ticket, and can't decrypt the service key unless they know the client's secret key.
- Kerberos protects against replay attacks:
  - If someone sends TGT or ticket later, it is rejected.
- All clients, servers should have time synchronized within a specified limit.

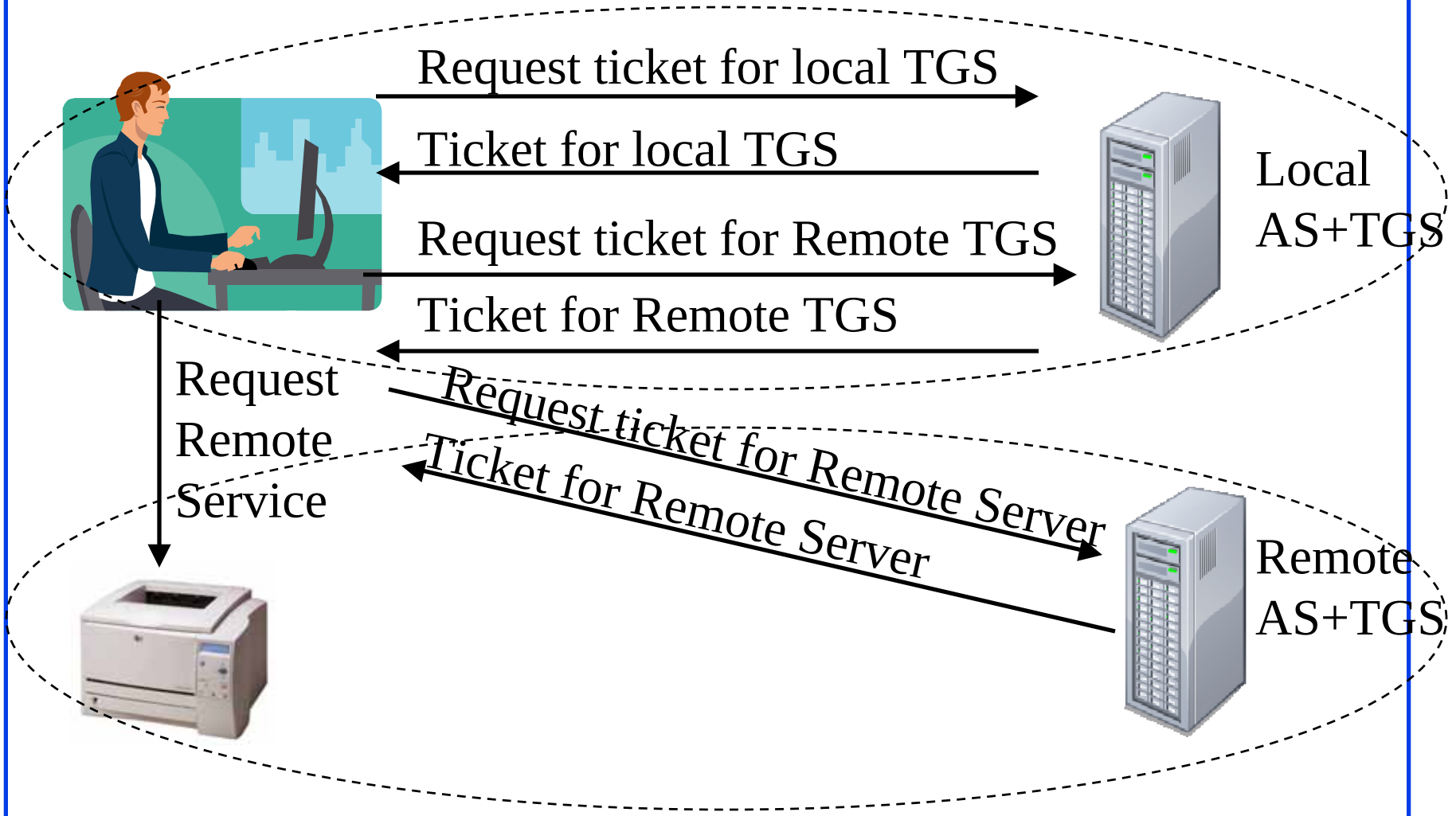
# Replicated KDCs

- KDC is a single point of failure.
- Multiple KDCs with database replication are allowed.
- One KDC keeps a master copy to which all changes are made.
- Changes propagated to other copies. All keys are already encrypted. An integrity check is added during transfers.
- Most KDC operations are read-only.

# Realms

- Realm = One organization or one trust domain
- Each realm has its own set of principles including KDC/TGT
- Each Principal's name = Name + Instance + Realm
- 40 characters each. Null terminated.
- Instance = Particular Server or Human role (administrator, game player)
- In V4, both realms should have a direct trust relationship. Chaining prohibited.

# Inter-Realm Authentication

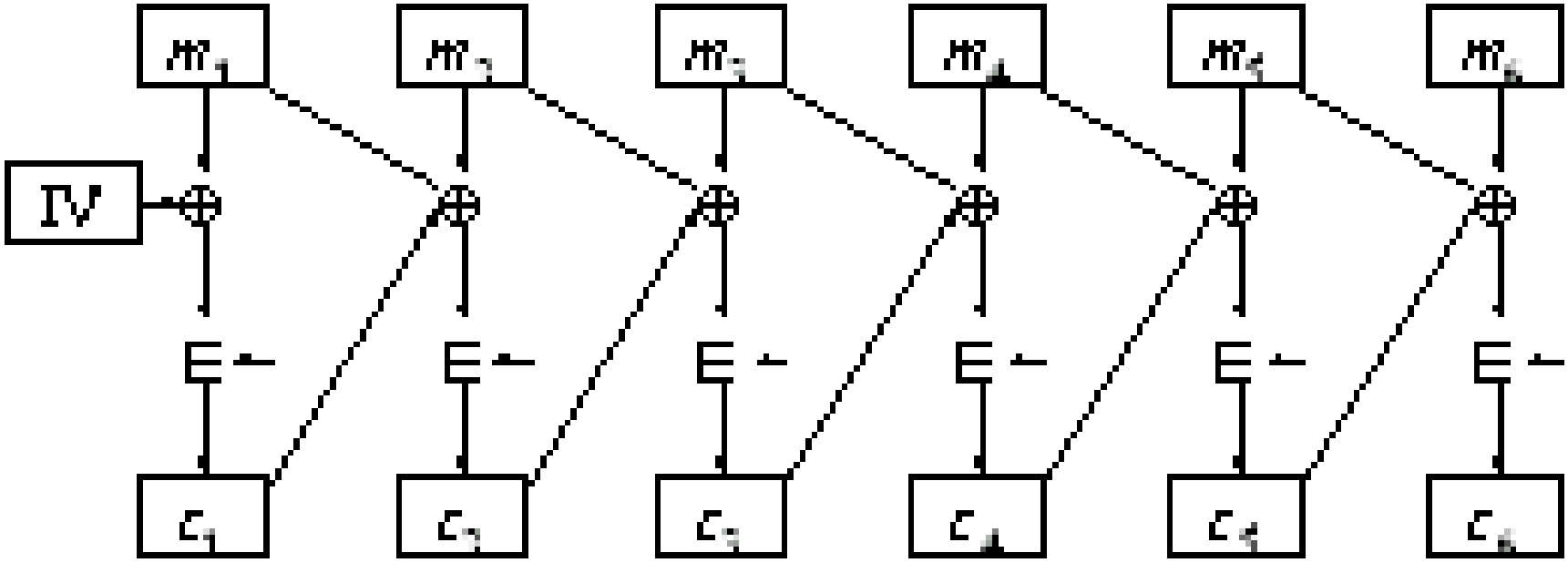


# Key Version Number

- All clients and servers remember their previous keys for a short time.
- Users have to wait after changing their password.

# Privacy and Integrity

- With CBC, only two blocks are affected by a change.
- Plaintext Cipher Block Chaining (PCBC) causes all blocks to change.
- Recognizable data is put at the end.



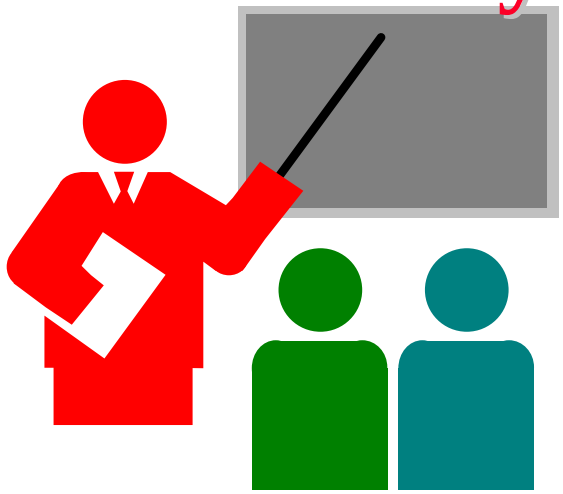
# Integrity Only

- DES too expensive.
- Kerberos uses a checksum on session key and the message
- The session key is not transmitted.
  Only message and checksum is transmitted.
- Although not broken. Not believed to be strong.
  Not used in V5.

# Network Layer Addresses in Tickets

- Ticket's contain requesters IP address.
- No one else can use the ticket without changing their IP addresses.
- Makes the delegation difficult.
- Problem for multi-homed systems
- Potential problems with Network Address Translators (NATs)
- Migration to IPv6 or other address formats

# Summary



- Kerberos is a symmetric key authentication system
- Authentication server issues Ticket Granting Tickets
- TGS issues service tickets
- Multi-realm authentication requires registration of foreign TGS with local KDC
- Requires tight time synchronization among systems

# References

- Chapter 13 of the text book.
- Wikipedia, http://en.wikipedia.org/wiki/Kerberos_%28protocol%29

# Homework 10

- Read chapter 13 of the text book. In particular, read about the format of various messages and fields.
- Submit answer to the following question:
  - In PCBC mode what is the effect of:
    - a random error in one block of cipher text $C_i$
    - interchanging ciphertext blocks $C_i$ and $C_{i+1}$