

# Kerberos V4

Raj Jain  
Washington University in Saint Louis  
Saint Louis, MO 63130  
[Jain@cse.wustl.edu](mailto:Jain@cse.wustl.edu)

Audio/Video recordings of this lecture are available at:  
<http://www.cse.wustl.edu/~jain/cse571-09/>



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

## Overview of Kerberos

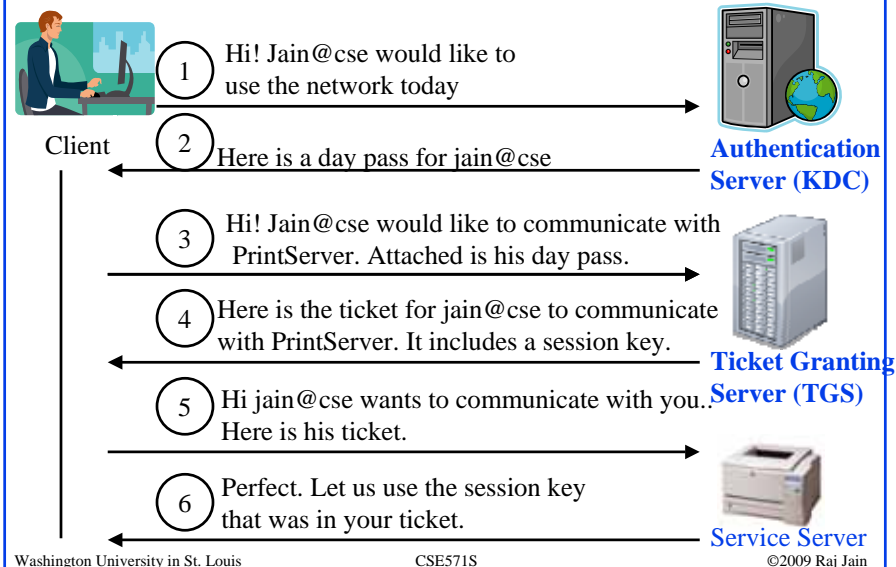


- ❑ Allows two users (or client and server) to authenticate each other over an insecure network
- ❑ Named after the Greek mythological character *Kerberos* (or *Cerberus*), known in Greek mythology as being the *monstrous three-headed guard dog of Hades*
- ❑ Designed originally for Project Athena at 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



10-5

## 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

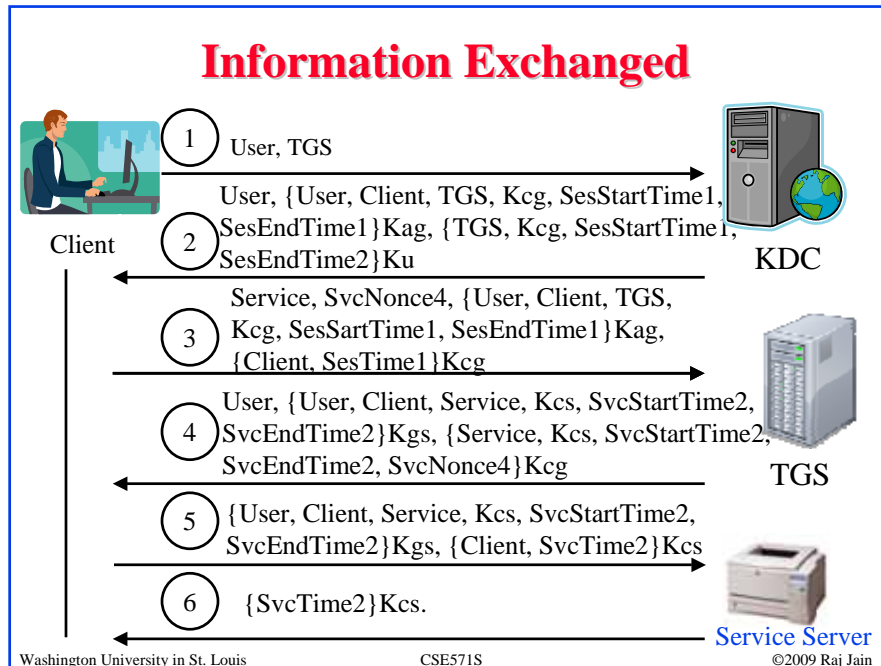
10-6

## Concepts (Cont)

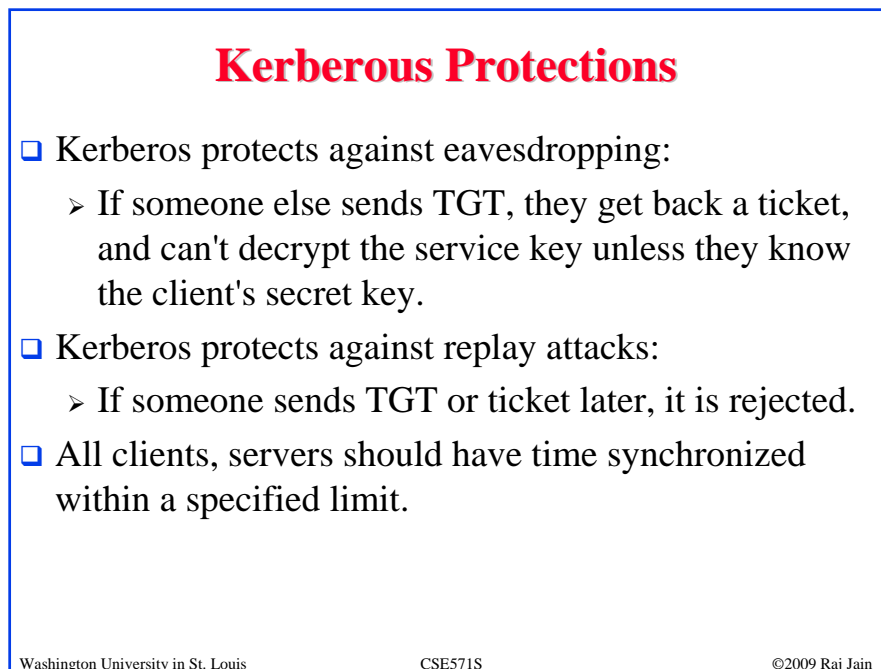
- ❑ **Authenticator:** Name and time encrypted with a session key. Sent from client to server with the ticket and from server to client.
- ❑ **Credentials:** Session key + Ticket
- ❑ **Session:** One user login/logout session
- ❑ User enters a name and password. Client converts the password to a key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

## Key Design Principles

1. The network is open  $\Rightarrow$  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.



10-9



10-10

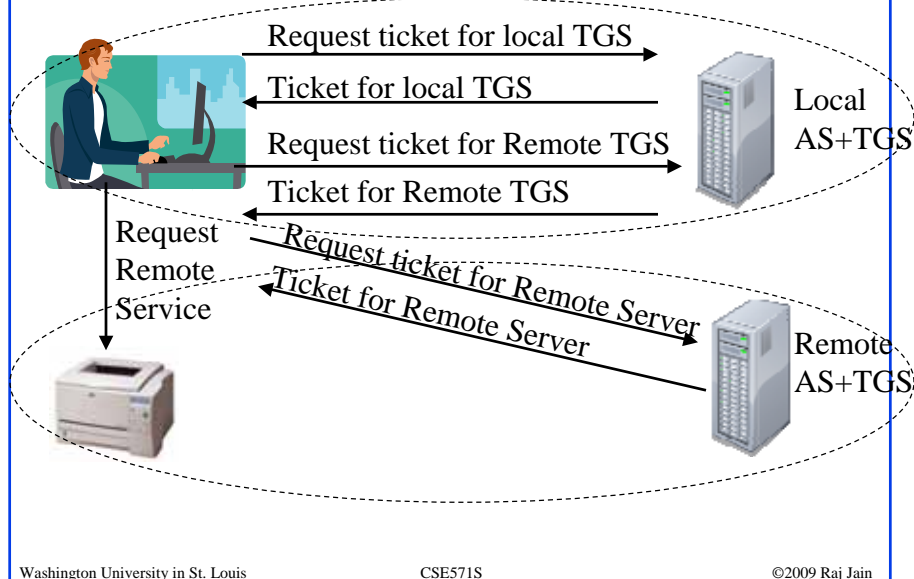
## Replicated KDCs

- ❑ 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



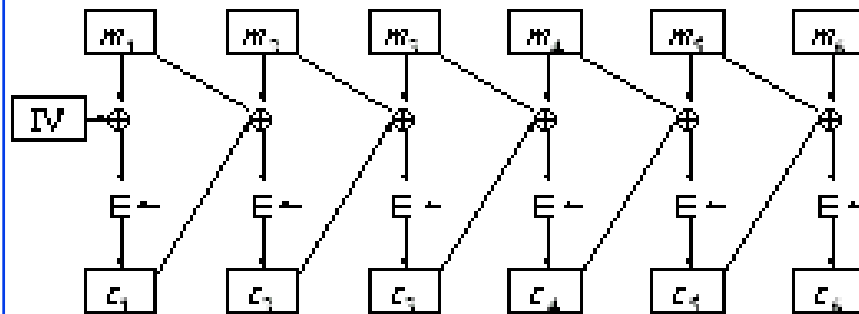
10-13

## 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.



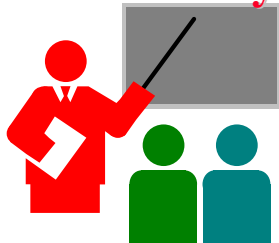
## Integrity Only

- ❑ 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](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  $C_{i+1}$

## Lab Homework 10b

Get a free digital certificates from one of the following 3 sources:

1. Verisign (60-day free trial)

<https://www.verisign.com/products-services/security-services/pki/pki-application/email-digital-id/index.html>

2. Ascertia:

<http://www.ascertia.com/onlineCA/default.aspx?linkID=40>

3. Comodo:

[http://www.comodo.com/products/certificate\\_services/email\\_certificate.html?entryURL=](http://www.comodo.com/products/certificate_services/email_certificate.html?entryURL=)

You will have received a signed email from the TA with his digital certificate. Import this certificate in your contacts list. (Use help feature on your email software for details). Now send an encrypted signed email to TA with the subject line of “CSE571S Encrypted Signed Mail Homework 10b”

## Lab Homework 10b (Cont)

The content field should contain the following information about TA’s certificate:

1. What signature algorithm was used?
2. What is the length of the public key used?
3. What is the expiration date of the certificate?
4. What is the URL to get list of revoked certificates?

Note: Use Verisign on Windows outlook and outlook express. Other combinations have not been tested.