The Kerberos Authentication Protocol

Cunsheng Ding
HKUST, Hong Kong, CHINA

Agenda

- Distributed system security
- Introduction to Kerberos V4
- Kerberos Realms
- Some operating systems using Kerberos
- Two basic protocols used in Kerberos

Distributed Systems Security

Distributed Systems

- **A distributed system**: a collection of computers linked via some network.
- **Characteristic**: The components of the distributed system may be under the authority of different organizations, and may be governed by different security policies.
- Example: The Internet

Security Issues in Distributed Systems (1)

- Impersonation of user:
  - A user may gain access to a particular workstation and pretend to be another user operating from that workstation.
- Impersonation of workstation:
  - A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.

Security Issues in Distributed Systems (2)

- **Replay attacks:**
  - A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations.
- **Conclusions:**
  - In any of these cases, an unauthorized user may be able to gain access to services and data that he or she is not authorized to access.
  - Consequently, user and service authentication are necessary in distributed systems.

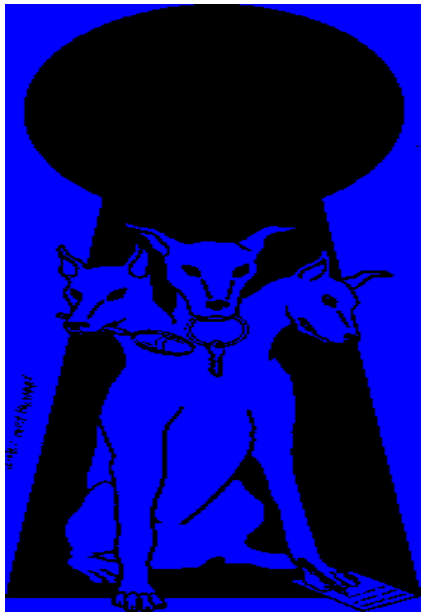
Security Services Required in Distributed Systems

- User and service authentication (covered in this lecture)
- Guarding the boundaries of internal networks
  - Firewalls (not covered in this UG course)
- Access control to distributed objects
  - Access control techniques (not covered in this course)
- Availability
  - Counter DoS techniques (not covered in this course)

Kerberos Version 4

Kerberos Version 4

- Centralized network authentication service
- Developed in the Project Athena in MIT



Environment Addressed

- An open distributed environment in which
  - Users at workstations wish to access services on servers distributed throughout the network.
  - Servers can:
    - restrict access to authorized users and
    - authenticate requests for service.
  - Workstations cannot be trusted to identify its users correctly to network services.

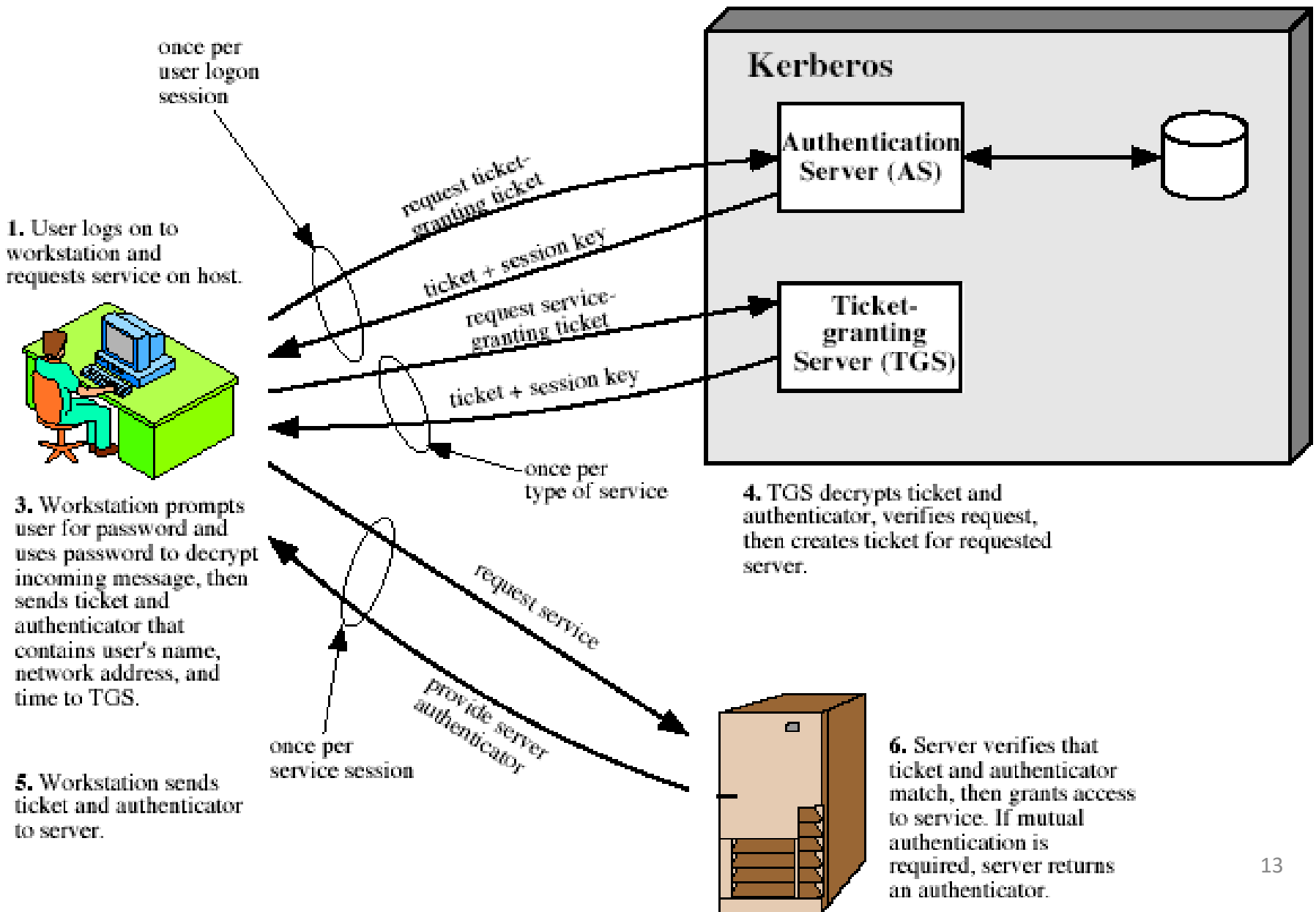
Requirements for Kerberos

- **Secure**: An opponent cannot impersonate a user and the Kerberos service should not be a weak link.
- **Reliable**: Highly reliable Kerberos service to ensure the availability of supported services of application servers.
  - I.e., reliable user and service authentication service
- **Transparent** : Users are only required to enter a password once and don't know the authentication.
- **Scalable**: System using Kerberos can support a small or large number of clients and servers.

Kerberos 4 Overview

- A basic third-party authentication scheme
- Have an Authentication Server (AS)
  - AS is responsible for identifying and authenticating a user.
  - AS is responsible for issuing an authentication credential (ticket granting ticket TGT) to a user.
- Have a Ticket Granting Server (TGS)
  - TGS is responsible for issuing service tickets to a user based on the user's TGT.

1. Each user shares a key with AS
2. TGS shares a key with AS
3. All servers are registered with TGS



Further Information

- Only one symmetric cipher, i.e., DES, is used in Version 4. In version 5, any cipher can be used.
- Each client needs to share a secret key with the AS only.
- AS and TGS share a secret key for authentication.
- Each server shares a secret key with the TGS.
- ID, timestamp, network address are used for authentication.
- The ticket-granting ticket is similar as your student card and service ticket is similar as a library card or sports facility card in a university.

Kerberos Realm

- Kerberos realm:
  - The environment that one Kerberos server (i.e., AS + TGS) can manage the authentication process.
  - A Kerberos realm is a way of logically grouping resources and identities that uses Kerberos. In Windows, realms are called domains. E.g., the HKUST domain, the CSE domain.
- The environment of one realm:
  - The Kerberos server of one realm has all users ID & hashed password of all users in the realm.
  - The Kerberos server must share a secret key with each server.
  - All servers are registered with the Kerberos server.

Kerberos in Operating Systems

- It is used in some Windows operating systems
- It is used in the following Unix-like operating systems:
  - FreeBSD, Apple's Mac OS X, Red Hat Enterprise Linux, Oracle's Solaris, IBM's AIX and Z/OS, HP's HP-UX and OpenVMS
- It is used for Kerberos authentication of **users** or **services**.

User Authentication in Kerberos

- The user/sender authentication protocol
  - A → E_k(ID_A|timestamp) → B
  - It is for sender/user authentication
  - The encrypted data is called the authenticator
  - The protocol was covered in Lecture 13

Service Authentication in Kerberos

- Service authentication is done with a service ticket
- The server TGS issues various types of service tickets to clients
- Below is the digital email service ticket
  - TGS → E_k(ID_C|AD_C|ID_V|Timestamp|Lifetime) → C
  - k is a long-term key shared by the TGS and email server V
  - ID_C is the ID of the client C; AD_C is C's network address
  - V is the email server
  - C can use it for email service multiple times as long as it is still valid.
  - Timestamp|Lifetime = Period of validity

Questions Regarding the Email Ticket

- The email service ticket
  - TGS → E_k(ID_C|AD_C|ID_V|Timestamp|Lifetime) → C
- Questions
  - If the client C forwards the ticket to the email server V, how should the email server V verify this ticket?
    - Explain the role of each entry embedded inside the ciphertext ticket
  - Who can create this ticket?
  - Why can a modification of the email service ticket by any party other than the TGS and V be detected?

Appendix: Protocol Details of Kerberos V4

Once you understand the sender authentication protocol and the email service ticket protocol discussed before, you can easily understand the protocol details of Kerberos V4 in the next slide.

You are **not** required to read the protocol details.

But if you wish to learn more, please read them offline.

Version 4 Authentication Dialogue

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

(1) **C → AS:** $ID_c \| ID_{tgs} \| TS_1$

(2) **AS → C:** $E_{K_c}[K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}]$

$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \| ID_c \| AD_c \| ID_{tgs} \| TS_2 \| Lifetime_2]$

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

(3) **C → TGS:** $ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) **TGS → C:** $E_{K_{c,tgs}}[K_{c,v} \| ID_v \| TS_4 \| Ticket_v]$

$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \| ID_c \| AD_c \| ID_{tgs} \| TS_2 \| Lifetime_2]$

$Ticket_v = E_{K_V}[K_{c,v} \| ID_c \| AD_c \| ID_v \| TS_4 \| Lifetime_4]$

$Authenticator_c = E_{K_{c,tgs}}[ID_c \| AD_c \| TS_3]$

(c) Client/Server Authentication Exchange: to obtain service

(5) **C → V:** $Ticket_v \| Authenticator_c$

(6) **V → C:** $E_{K_{c,V}}[TS_5 + 1]$ (for mutual authentication)

$Ticket_v = E_{K_V}[K_{c,v} \| ID_c \| AD_c \| ID_v \| TS_4 \| Lifetime_4]$

$Authenticator_c = E_{K_{c,V}}[ID_c \| AD_c \| TS_5]$

Index

- k_c the secret key shared between C and the AS.
- $k_{c,\,tgs}$ the session key for C and TGS, generated by the AS.
- $k_{c,v}$ the session key for C and V, generated by the TGS.
- k_{tgs} the secret key shared between the TGS and the AS.
- TS, timestamp
- ID_c, C's ID
- AD_c, C's network address.

Comments on a distributed system with Kerberos authentication

- The system is a Single Sign-On system
  - When a user signs onto any workstation, he/she signs into the whole domain.
  - It gives simple administration.
    - For instance, each user has only one user account within the HKUST domain.
  - It provides good user productivity.
    - For instance, only when a user signs into the HKUST domain, he/she inputs his/her password once, and does not need to retype the password for requesting many services later.