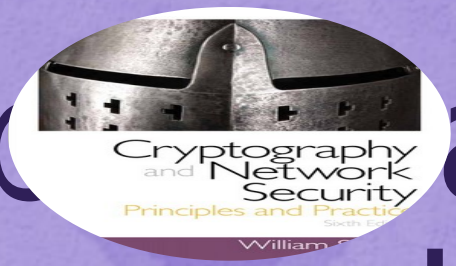
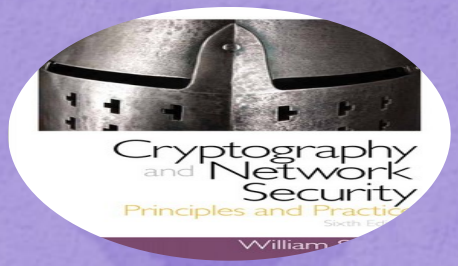
# Cryptography and Network Security




Seventh Edition
by William Stallings

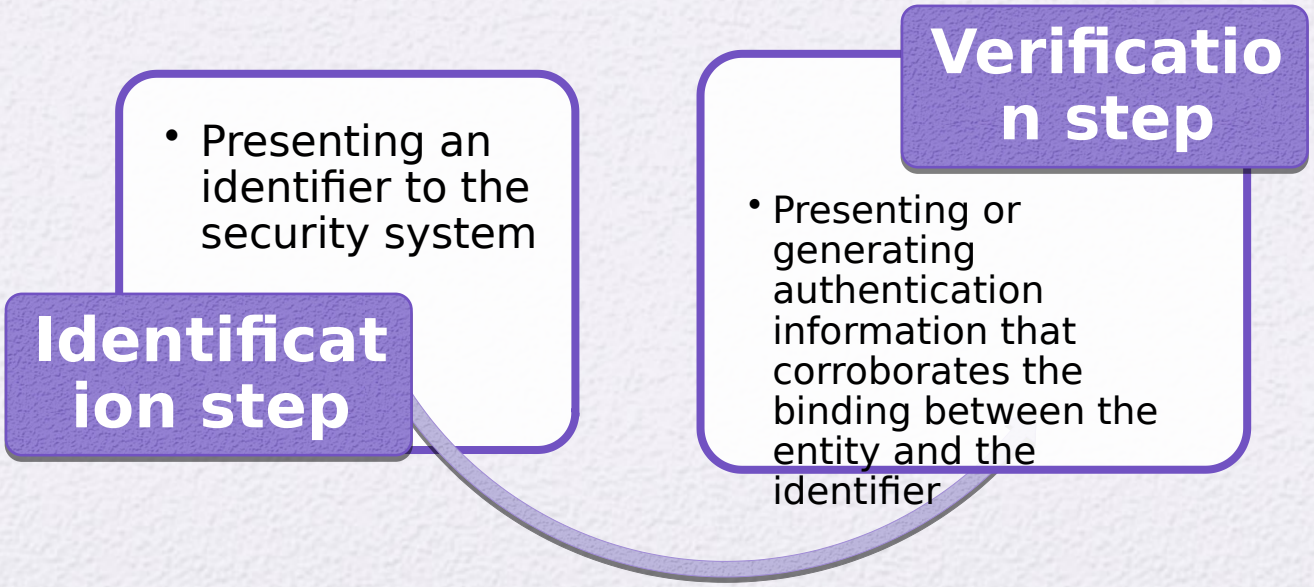



# Chapter 15

## User Authentication

# Remote User-Authentication Principles

- The process of verifying an identity claimed by or for a system entity
- An authentication process consists of two steps:



# Means of User Authentication

**Something the individual knows**

- Examples include a password, a personal identification number (PIN), or answers to a prearranged set of qu

**Something the individual possesses**

- Examples include cryptographic keys, electronic keycards, smart cards, and
  keys
  referred to as a token

**Something the indiv (static biometrics)**

- Examples include recognition by fingerprint, retina, and face

**ng the individual namic biometrics)**

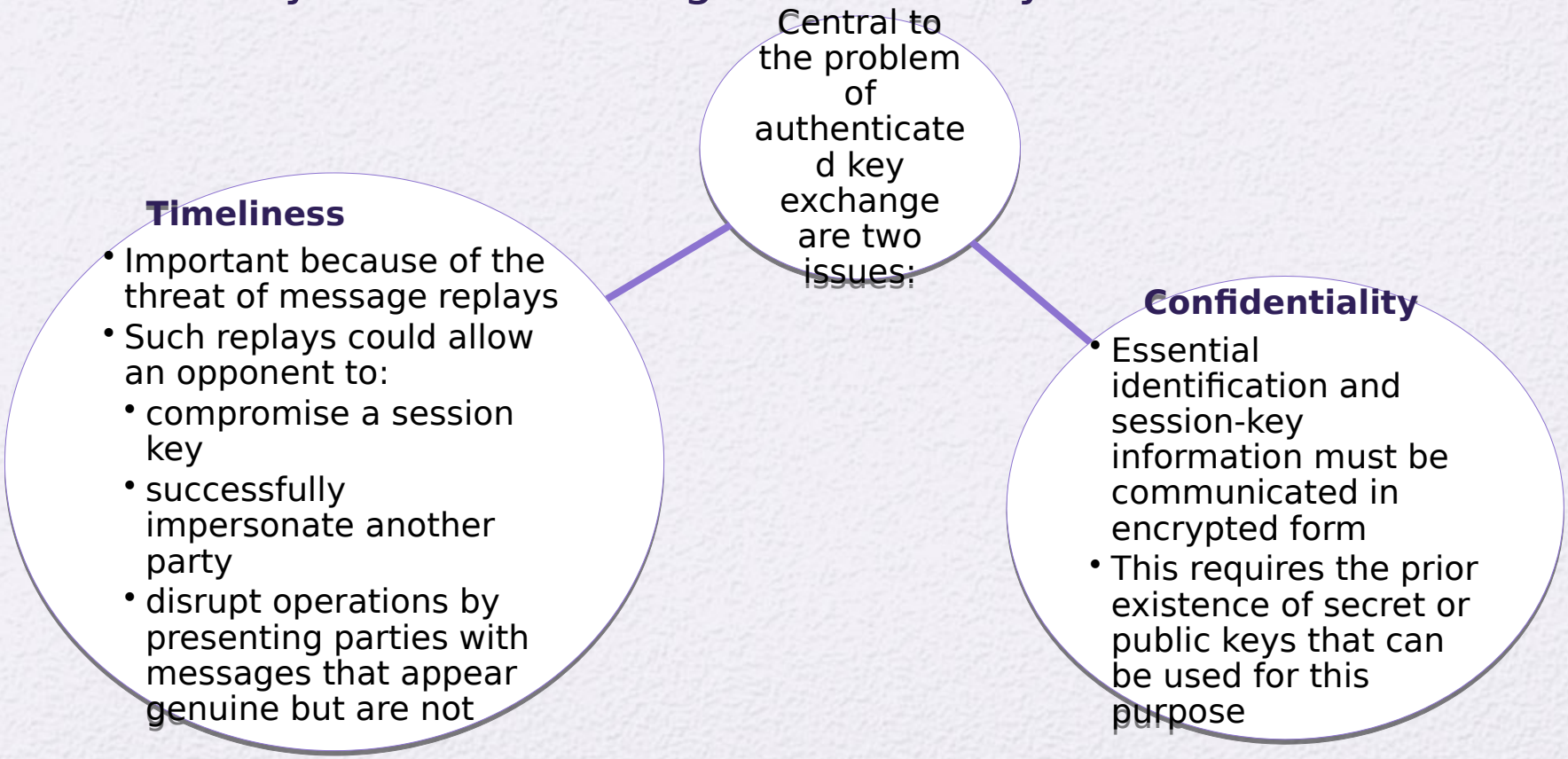
- Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm

**there are four general means of authenticating a user's identity, which can be used alone or in combination**

- F
  important methods involve cryptographic keys and something the individual knows, such as a password

# Mutual Authentication

- Protocols which enable communicating parties to satisfy themselves mutually about each other's identity and to exchange session keys



# Replay Attacks

1. The simplest replay attack is one in which the opponent simply copies a message and replays it later

2. An opponent can replay a timestamped message within the valid time window

3. An opponent can replay a timestamped message within the valid time window, but in addition, the opponent suppresses the original message; thus, the repetition cannot be detected

4. Another attack involves a backward replay without modification and is possible if symmetric encryption is used and the sender cannot easily recognize the difference between messages sent and messages received on the basis of content

# Approaches to Coping With Replay Attacks

- Attach a sequence number to each message used in an authentication exchange
  - A new message is accepted only if its sequence number is in the proper order
  - Difficulty with this approach is that it requires each party to keep track of the last sequence number for each claimant it has dealt with
  - Generally not used for authentication and key exchange because of overhead
- Timestamps
  - Requires that clocks among the various participants be synchronized
  - Party A accepts a message as fresh only if the message contains a timestamp that, in A's judgment, is close enough to A's knowledge of current time
- Challenge/response
  - Party A, expecting a fresh message from B, first sends B a nonce (challenge) and requires that the subsequent message (response) received from B contain the correct nonce value

# One-Way Authentication

**One application for which encryption is growing in popularity is electronic mail (e-mail)**

- Header of the e-mail message must be in the clear so that the message can be handled by the store-and-forward e-mail protocol, such as SMTP or X.400
- The e-mail message should be encrypted such that the mail-handling system is not in possession of the decryption key

**A second requirement is that of authentication**

- The recipient wants some assurance that the message is from the alleged sender

# Suppress-Replay Attacks

- The Denning protocol requires reliance on clocks that are synchronized throughout the network
- A risk involved is based on the fact that the distributed clocks can become unsynchronized as a result of sabotage on or faults in the clocks or the synchronization mechanism
- The problem occurs when a sender's clock is ahead of the intended recipient's clock
  - An opponent can intercept a message from the sender and replay it later when the timestamp in the message becomes current at the recipient's site
  - Such attacks are referred to as *suppress-replay attacks*

# Remote User-Authentication Using Symmetric Encryption

**A two-level hierarchy of symmetric keys can be used to provide confidentiality for communication in a distributed environment**

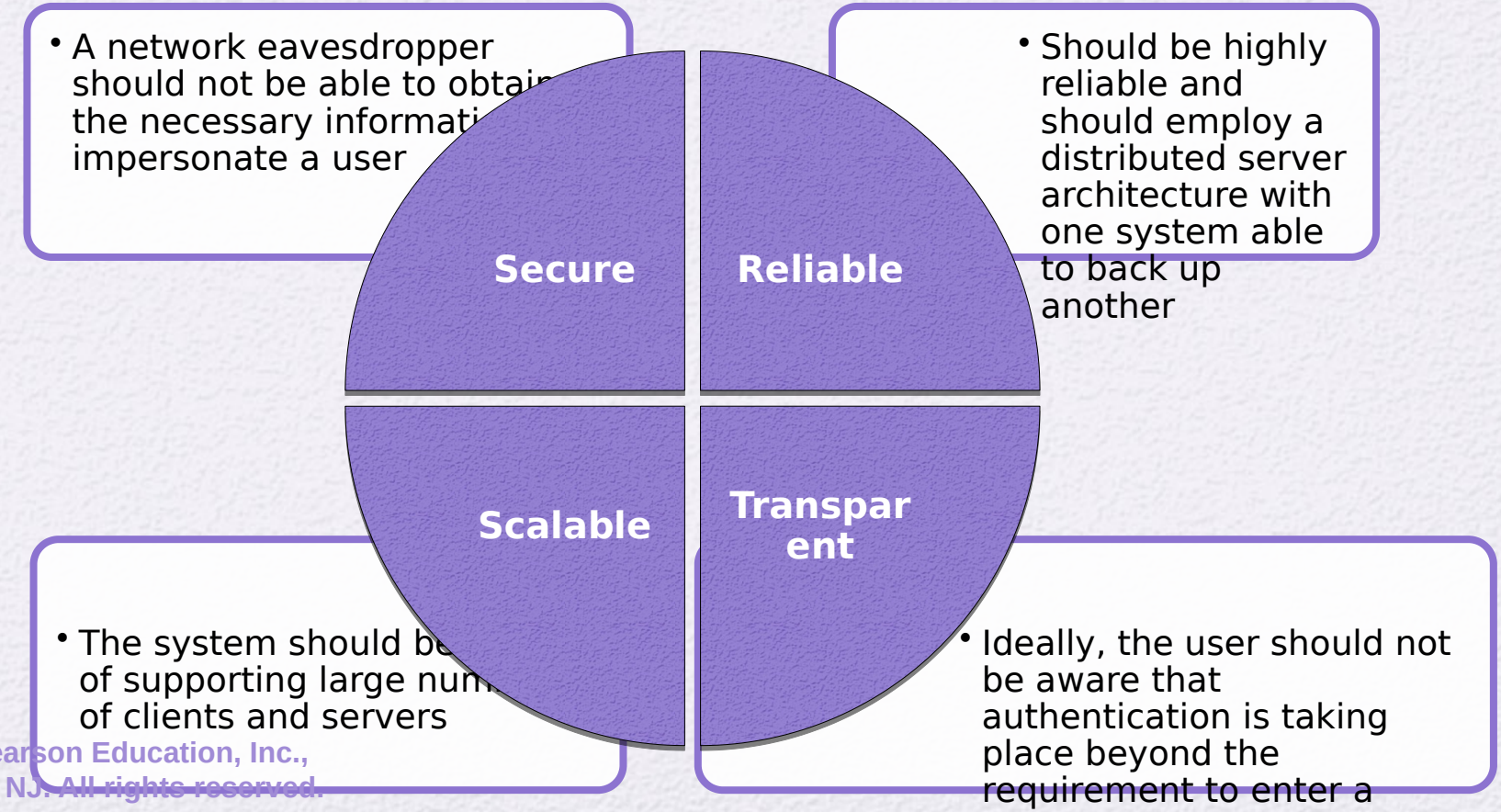
- Strategy involves the use of a trusted key distribution center (KDC)
- Each party shares a secret key, known as a master key, with the KDC
- KDC is responsible for generating keys to be used for a short time over a connection between two parties and for distributing those keys using the master keys to protect the distribution

# Kerberos

- Authentication service developed as part of Project Athena at MIT
- A workstation cannot be trusted to identify its users correctly to network services
  - A user may gain access to a particular workstation and pretend to be another user operating from that 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
  - A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations
- Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users
  - Relies exclusively on symmetric encryption, making no use of public-key encryption

# Kerberos Requirements

- The first published report on Kerberos listed the following requirements:



# Kerberos Version 4

- Makes use of DES to provide the authentication service
- Authentication server (AS)
  - Knows the passwords of all users and stores these in a centralized database
  - Shares a unique secret key with each server
- Ticket
  - Created once the AS accepts the user as authentic; contains the user's ID and network address and the server's ID
  - Encrypted using the secret key shared by the AS and the server
- Ticket-granting server (TGS)
  - Issues tickets to users who have been authenticated to AS
  - Each time the user requires access to a new service the client applies to the TGS using the ticket to authenticate itself
  - The TGS then grants a ticket for the particular service
  - The client saves each service-granting ticket and uses it to authenticate its user to a server each time a particular service is requested

# The Version 4 Authentication Dialogue

The lifetime associated with the ticket-granting ticket creates a problem:

- If the lifetime is very short (e.g., minutes), the user will be repeatedly asked for a password
- If the lifetime is long (e.g., hours), then an opponent has a greater opportunity for replay

A network service (the TGS or an application service) must be able to prove that the person using a ticket is the same person to whom that ticket was issued

Servers need to authenticate themselves to users

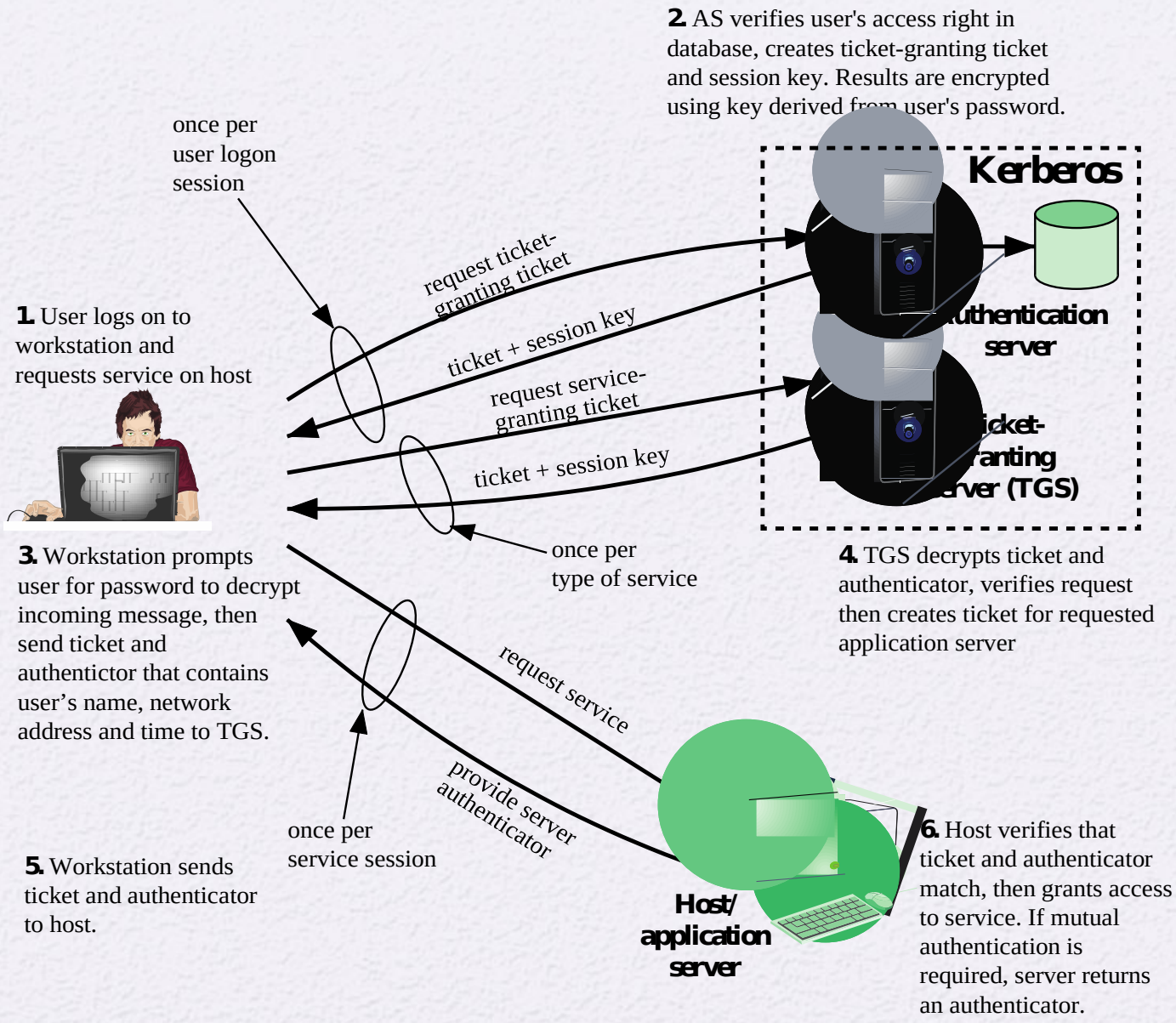



**Figure 15.2 Overview of Kerberos**

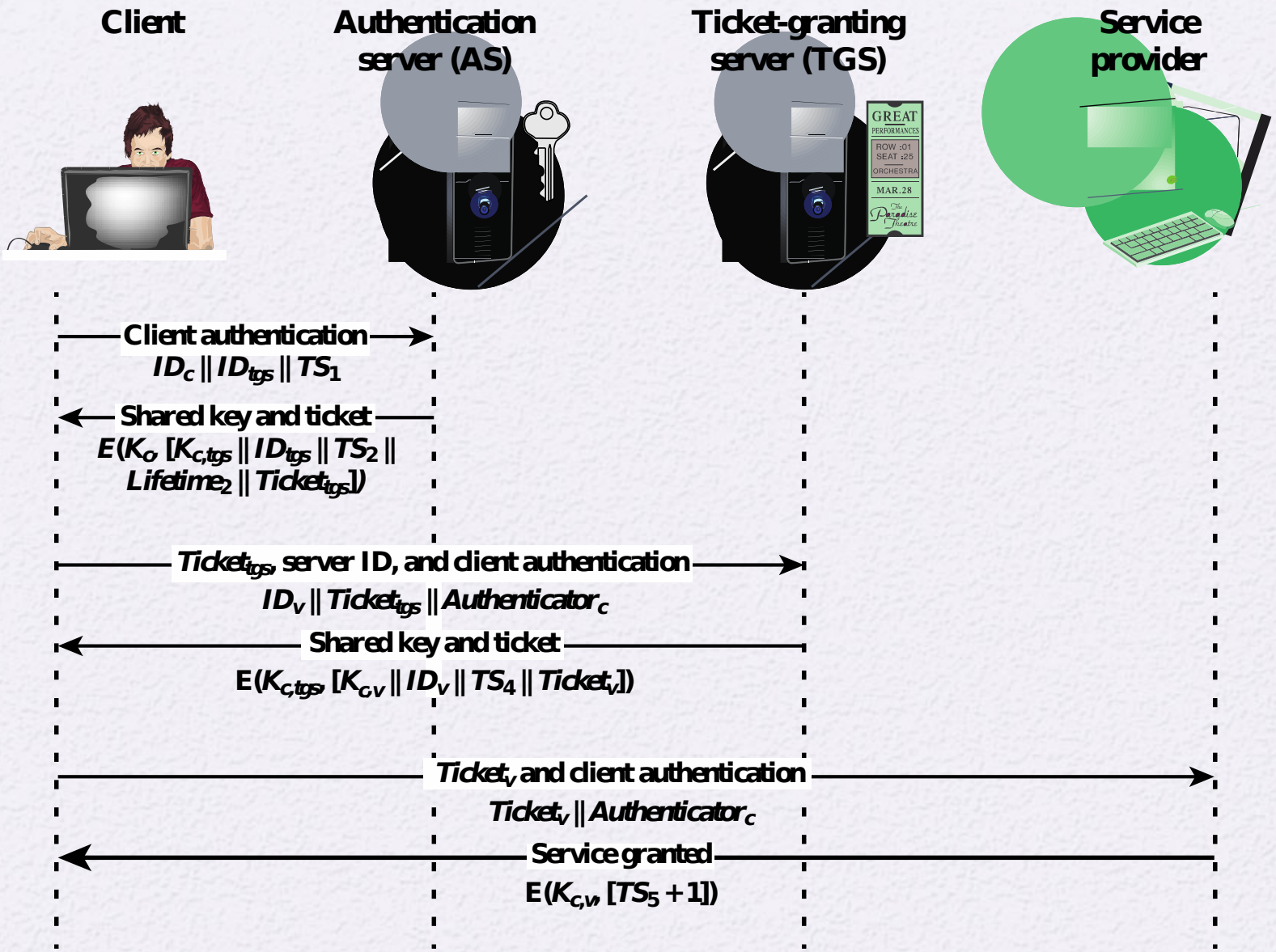



**Figure 15.3 Kerberos Exchanges**

# Table 15.2 Rationale for the Elements of the Kerberos Version 4 Protocol

(page 1 of 3)

| | |
|---|---|
| **Message (1)** | Client requests ticket-granting ticket. |
| $ID_C$ | Tells AS identity of user from this client. |
| $ID_{tgs}$ | Tells AS that user requests access to TGS. |
| $TS_1$ | Allows AS to verify that client's clock is synchronized with that of AS. |
| **Message (2)** | AS returns ticket-granting ticket. |
| $K_c$ | Encryption is based on user's password, enabling AS and client to verify password, and protecting contents of message (2). |
| $K_{c,tgs}$ | Copy of session key accessible to client created by AS to permit secure exchange between client and TGS without requiring them to share a permanent key. |
| $ID_{tgs}$ | Confirms that this ticket is for the TGS. |
| $TS_2$ | Informs client of time this ticket was issued. |
| $Lifetime_2$ | Informs client of the lifetime of this ticket. |
| $Ticket_{tgs}$ | Ticket to be used by client to access TGS. |

(This table can be found on pages 473 – 474 in the textbook)

| | |
|---|---|
| **Message (3)** | Client requests service-granting ticket. |
| $ID_V$ | Tells TGS that user requests access to server V. |
| $Ticket_{tgs}$ | Assures TGS that this user has been authenticated by AS. |
| $Authenticator_c$ | Generated by client to validate ticket . |
| | |
| **Message (4)** | TGS returns service-granting ticket. |
| $K_{c,tgs}$ | Key shared only by C and TGS protects contents of message (4). |
| $K_{c,v}$ | Copy of session key accessible to client created by TGS to permit secure exchange between client and server without requiring them to share a permanent key. |
| $ID_V$ | Confirms that this ticket is for server V. |
| $TS_4$ | Informs client of time this ticket was issued. |
| $Ticket_V$ | Ticket to be used by client to access server V. |
| | |
| $Ticket_{tgs}$ | Reusable so that user does not have to reenter password. |
| $K_{tgs}$ | Ticket is encrypted with key known only to AS and TGS, to prevent Tampering. |
| $K_{c,tgs}$ | Copy of session key accessible to TGS used to decrypt authenticator, thereby authenticating ticket. |
| $ID_C$ | Indicates the rightful owner of this ticket. |
| $AD_C$ | Prevents use of ticket from workstation other than one that initially requested the ticket. |
| $ID_{tgs}$ | Assures server that it has decrypted ticket properly. |
| $TS_2$ | Informs TGS of time this ticket was issued. |
| $Lifetime_2$ | Prevents replay after ticket has expired. |
| $Authenticator_c$ | Assures TGS that the ticket presenter is the same as the client for whom the ticket was issued has very short lifetime to prevent replay. |
| $K_{c,tgs}$ | Authenticator is encrypted with key known only to client and TGS, to prevent tampering. |
| $ID_C$ | Must match ID in ticket to authenticate ticket. |
| $AD_C$ | Must match address in ticket to authenticate ticket. |
| $TS_3$ | Informs TGS of time this authenticator was generated. |

| | |
|---|---|
| **Message (5)** | Client requests service. |
| $Ticket_V$ | Assures server that this user has been authenticated by AS. |
| $Authenticator_C$ | Generated by client to validate ticket. |
| | |
| **Message (6)** | Optional authentication of server to client. |
| $K_{c,v}$ | Assures C that this message is from V. |
| $TS_5 + 1$ | Assures C that this is not a replay of an old reply. |
| | |
| $Ticket_V$ | Reusable so that client does not need to request a new ticket from TGS for each access to the same server. |
| $K_V$ | Ticket is encrypted with key known only to TGS and server, to prevent Tampering. |
| $K_{c,v}$ | Copy of session key accessible to client; used to decrypt authenticator, thereby authenticating ticket. |
| $ID_C$ | Indicates the rightful owner of this ticket. |
| $AD_C$ | Prevents use of ticket from workstation other than one that initially requested the ticket. |
| $ID_V$ | Assures server that it has decrypted ticket properly. |
| $TS_4$ | Informs server of time this ticket was issued. |
| $Lifetime_4$ | Prevents replay after ticket has expired. |
| | |
| $Authenticator_C$ | Assures server that the ticket presenter is the same as the client for whom the ticket was issued; has very short lifetime to prevent replay. |
| $K_{c,v}$ | Authenticator is encrypted with key known only to client and server, to prevent tampering. |
| $ID_C$ | Must match ID in ticket to authenticate ticket. |
| $AD_C$ | Must match address in ticket to authenticate ticket. |
| $TS_5$ | Informs server of time this authenticator was generated. |

# Kerberos Realms and Multiple Kerberi

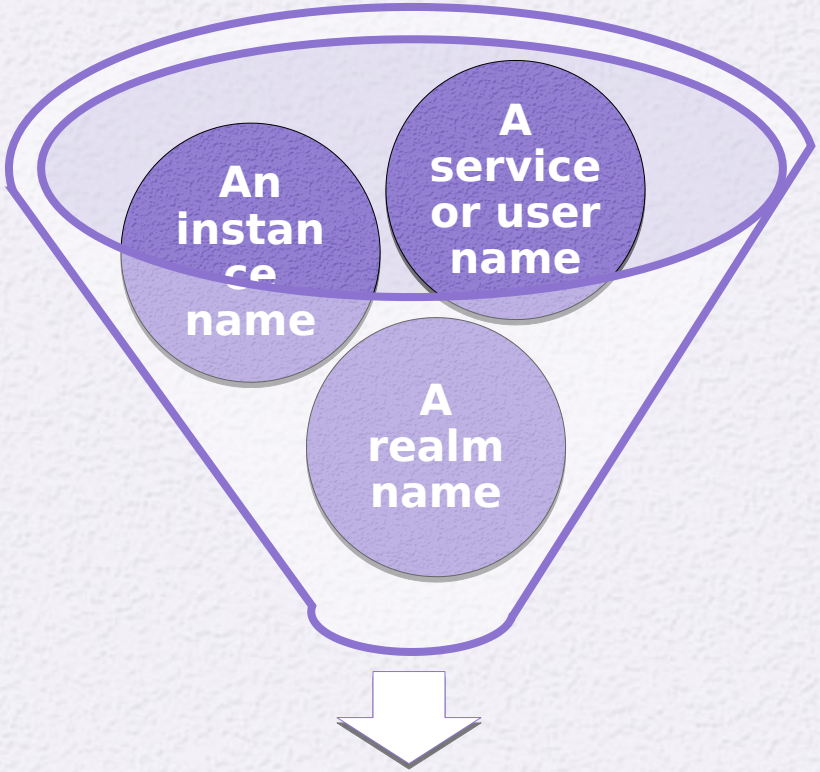
- A full-service Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers requires that:
  - The Kerberos server must have the user ID and hashed passwords of all participating users in its database; all users are registered with the Kerberos server
  - The Kerberos server must share a secret key with each server; all servers are registered with the Kerberos server
  - The Kerberos server in each interoperating realm shares a secret key with the server in the other realm; the two Kerberos servers are registered with each other

# Kerberos Realm

- A set of managed nodes that share the same Kerberos database
- The database resides on the Kerberos master computer system, which should be kept in a physically secure room
- A read-only copy of the Kerberos database might also reside on other Kerberos computer systems
- All changes to the database must be made on the master computer system
- Changing or accessing the contents of a Kerberos database requires the Kerberos master password

# Kerberos Principal

- A service or user that is known to the Kerberos system
- Identified by its principal name




**Three parts of a principal name**

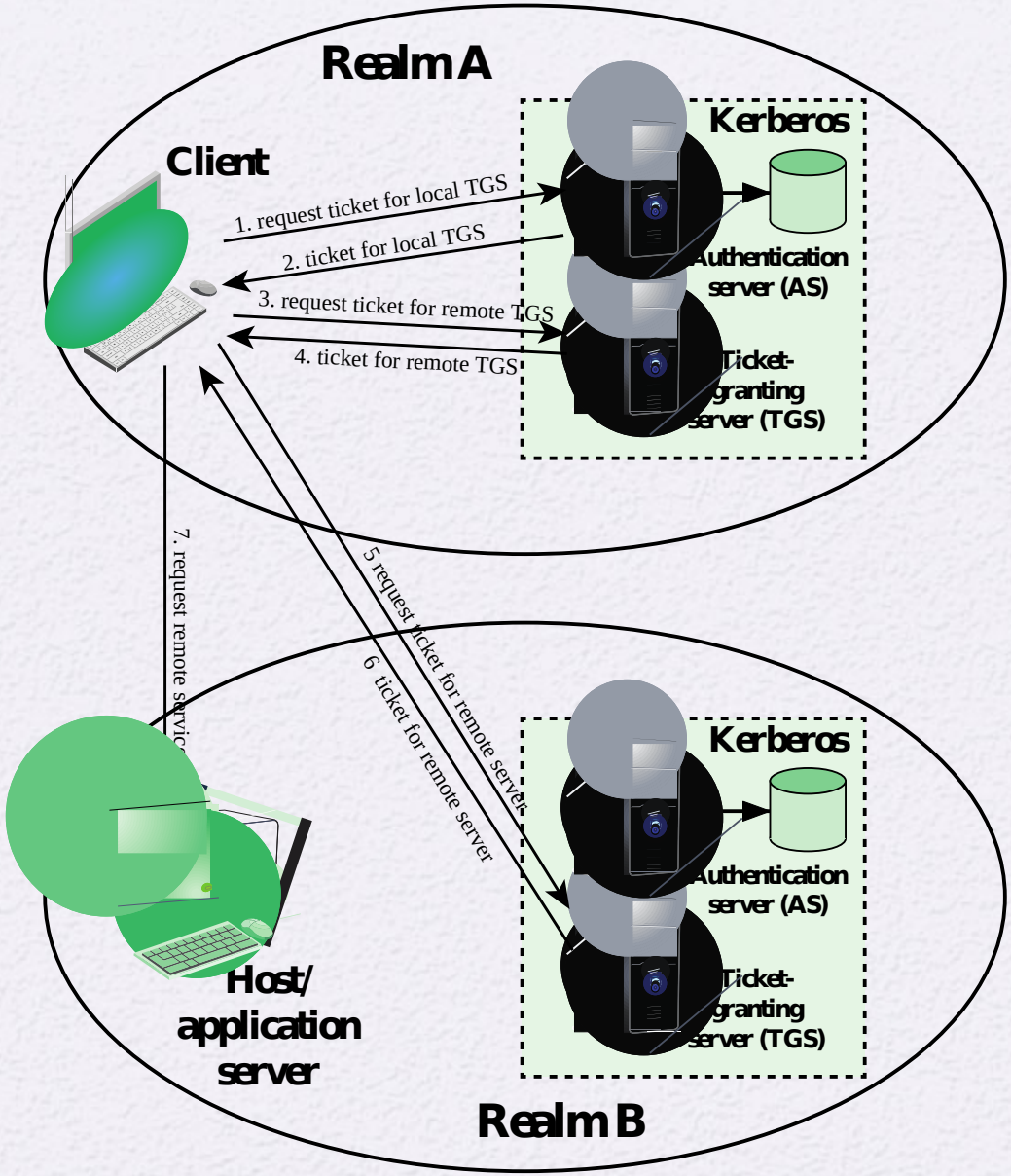



**Figure 15.4 Request for Service in Another Realm**

# Differences Between Versions 4 and 5

Version 5 is intended to address the limitations of version 4 in two areas:

## Environmental shortcomings

- Encryption system dependence
- Internet protocol dependence
- Message byte ordering
- Ticket lifetime
- Authentication forwarding
- Interrealm authentication

## Technical deficiencies

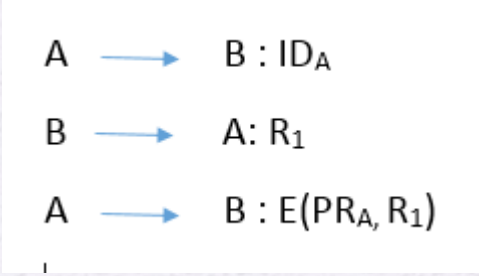
- Double encryption
- PCBC encryption
- Session keys
- Password attacks

# Questions

- What problem(s) was Kerberos designed to address?
- What are three threats associated with user authentication over a network or the Internet?
- Give an example of a Kerberos principal name.
- What is a Kerberos ticket? How many types are there? Show how they differ with an example or an analogy

# Questions

- Consider a one-way authentication technique based on asymmetric encryption as shown below:
- Explain the protocol
- What type of attack is this protocol susceptible to?



# Questions

- Consider a one-way authentication technique based on asymmetric encryption as shown below:
- Explain the protocol
- What type of attack is this protocol susceptible to?