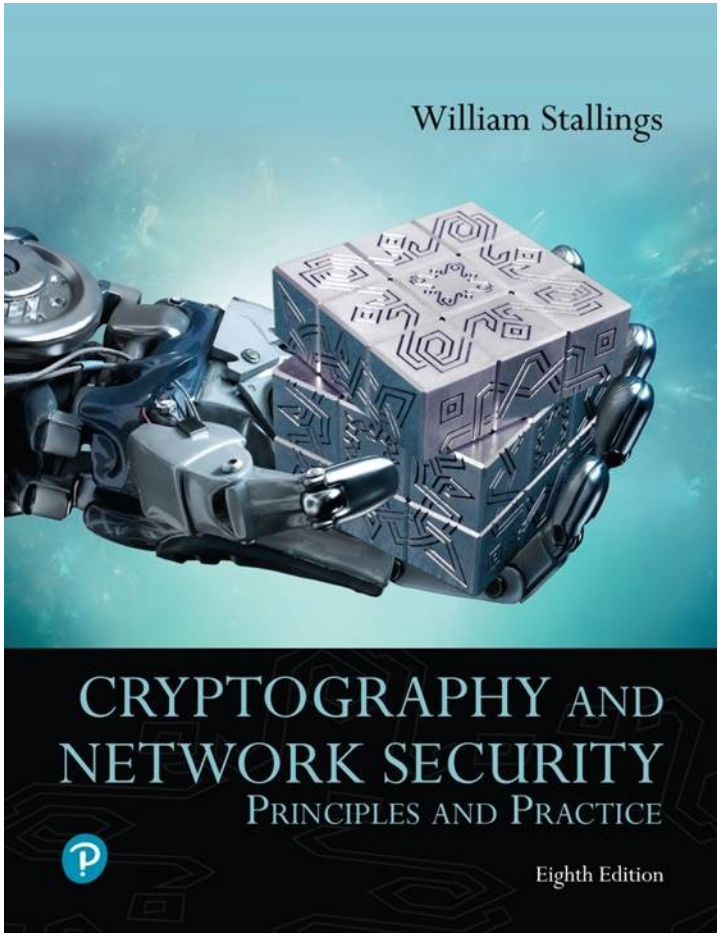
Cryptography and Network Security: Principles and Practice

Eighth Edition



Chapter 16

User Authentication

User-Authentication

- The process of determining whether some user or some application or process acting on behalf of a user is, in fact, who or what it declares itself to be
- Authentication technology provides access control for systems by checking to see if a user's credentials match the credentials in a database of authorized users or in a data authentication server
- Authentication enables organizations to keep their networks secure by permitting only authenticated users (or processes) to access its protected resources
- User authentication is distinct from message authentication
  - Message authentication is a procedure that allows communicating parties to verify that the contents of a received message have not been altered and that the source is authentic

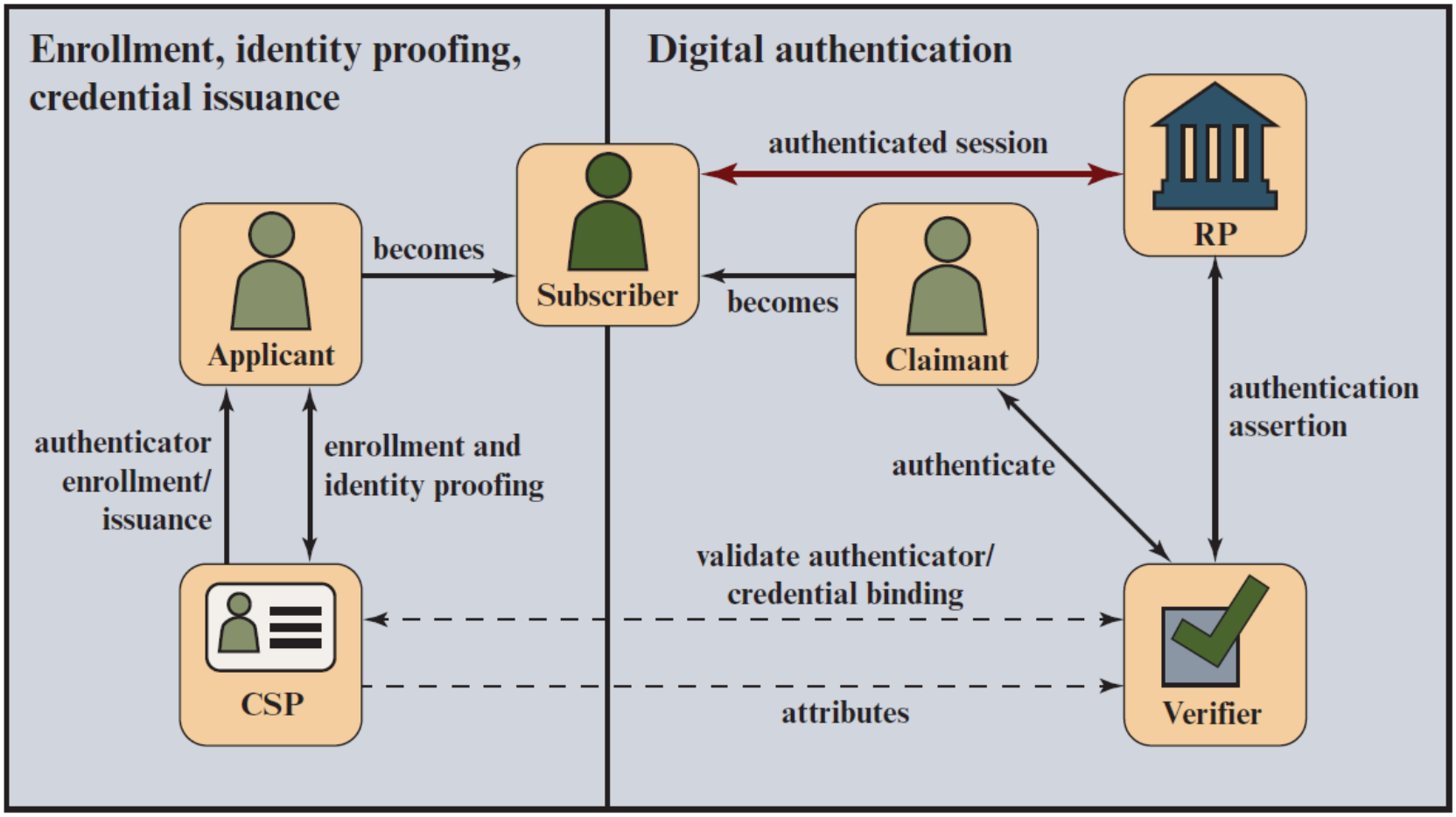
Authentication Principles (1 of 2)

- **Digital identity:**
  - The unique representation of a subject engaged in an online transaction
  - The representation consists of an attribute or set of attributes that uniquely describe a subject within a given context of a digital service, but does not necessarily uniquely identify the subject in all contexts
- **Identity proofing:**
  - Establishes that a subject is who they claim to be to a stated level of certitude
  - This process involves collecting, validating, and verifying information about a person

Authentication Principles (2 of 2)

- **Digital authentication:**
  - The process of determining the validity of one or more authenticators used to claim a digital identity
  - Authentication establishes that a subject attempting to access a digital service is in control of the technologies used to authenticate
  - Successful authentication provides reasonable risk-based assurances that the subject accessing the service today is the same as the subject that previously accessed the service

Figure 16.1 The NIST 800-63 Digital Identity Model



CSP = credential service provider
RP = relying party

Means of User Authentication (1 of 3)

There are three general means, or *authentication factors*, of authenticating a user's identity, which can be used alone or in combination:

- Knowledge factor (something the individual knows):
  - Requires the user to demonstrate knowledge of secret information. Routinely used in single-layer authentication processes, knowledge factors can come in the form of passwords, passphrases, personal identification numbers (PINs), or answers to secret questions

Means of User Authentication (2 of 3)

- Possession factor (something the individual possesses):
  - Physical entity possessed by the authorized user to connect to the client computer or portal. This type of authenticator used to be referred to as a token, but that term is now deprecated. The term hardware token is a preferable alternative. Possession factors fall into two categories:
    - Connected hardware tokens are items that connect to a computer logically (e.g., via wireless) or physically in order to authenticate identity. Items such as smart cards, wireless tags, and USB tokens are common connected tokens used to serve as a possession factor
    - Disconnected hardware tokens are items that do not directly connect to the client computer, instead requiring input from the individual attempting to sign in. Typically, a disconnected hardware token device will use a built-in screen to display authentication data that are then utilized by the user to sign in when prompted

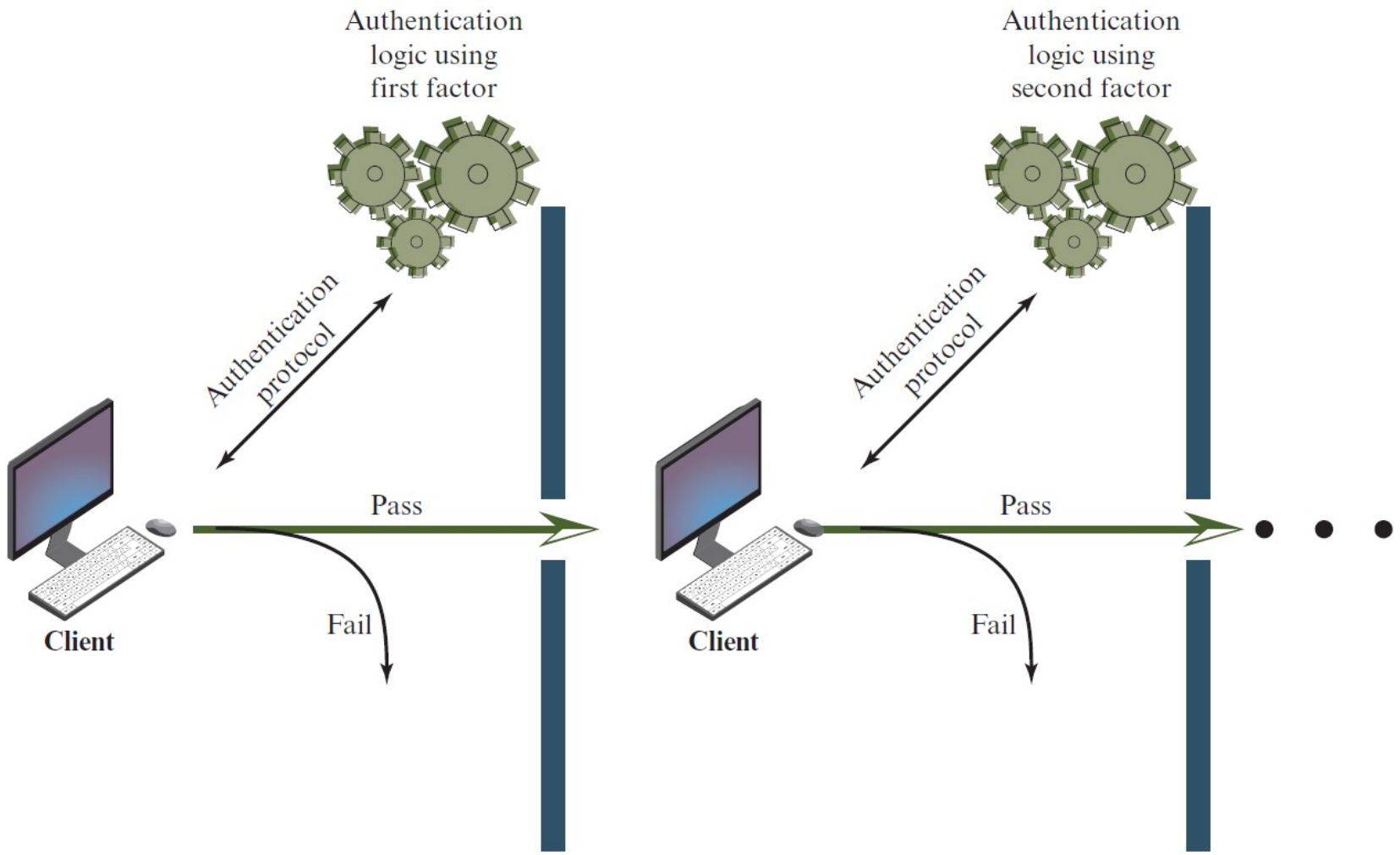
Means of User Authentication (3 of 3)

- Inherence factor (something the individual is or does):
  - Refers to characteristics, called biometrics, that are unique or almost unique to the individual. These include static biometrics, such as fingerprint, retina, and face; and dynamic biometrics, such as voice, handwriting, and typing rhythm

Table 16.1 Authentication Factors

Factor	Examples	Properties
Knowledge	User ID Password PIN	Can be shared Many passwords easy to guess Can be forgotten
Possession	Smart Card Electronic Badge Electronic Key	Can be shared Can be duplicated (cloned) Can be lost or stolen
Inherence	Fingerprint Face Iris Voice print	Not possible to share False positives and false Negatives possible Forging difficult

Figure 16.2 Multifactor Authentication



Mutual Authentication (1 of 2)

- Protocols which enable communicating parties to satisfy themselves mutually about each other's identity and to exchange session keys
- Central to the problem of authenticated key exchange are two issues:
  - **Confidentiality**
    - Essential identification and session-key information must be communicated in encrypted form
    - This requires the prior existence of secret or public keys that can be used for this purpose
  - **Timeliness**
    - Important because of the threat of message replays
    - Such replays could allow an opponent to:
      - compromise a session key
      - successfully impersonate another party
      - disrupt operations by presenting parties with messages that appear genuine but are not

Mutual Authentication (2 of 2)

- Public-key encryption for session key distribution
  - Assumes each of the two parties is in possession of the current public key of the other
  - May not be practical to require this assumption
- Denning protocol using timestamps
  - Uses an authentication server (AS) to provide public-key certificates
  - Requires the synchronization of clocks
- Woo and Lam makes use of nonces
  - Care needed to ensure no protocol flaws

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 (1 of 2)

- 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

Approaches to Coping With Replay Attacks (2 of 2)

- 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

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

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*

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 (1 of 2)

- The first published report on Kerberos listed the following requirements:
- **Secure**
  - A network eavesdropper should not be able to obtain the necessary information to impersonate a user
- **Reliable**
  - Should be highly reliable and should employ a distributed server architecture with one system able to back up another

Kerberos Requirements (2 of 2)

- **Transparent**
  - Ideally, the user should not be aware that authentication is taking place beyond the requirement to enter a password
- **Scalable**
  - The system should be capable of supporting large numbers of clients and servers

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

Table 16.2 Summary of Kerberos Version 4 Message Exchanges

(1)	**C → AS**	$ID_c \| ID_{tgs} \| TS_1$
(2)	**AS → C**	$E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$
		$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$

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

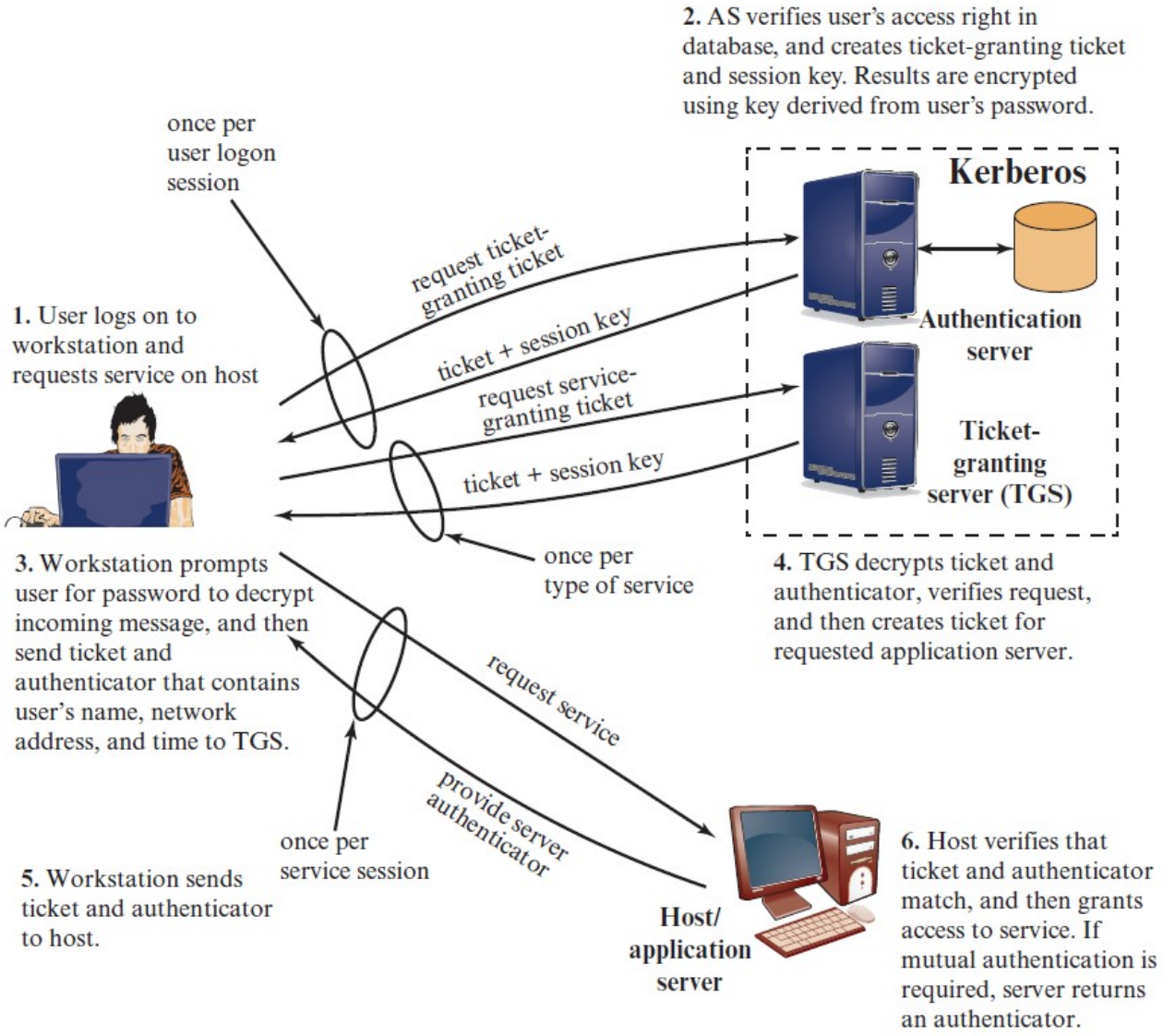
(3)	**C → TGS**	$ID_v \| Ticket_{tgs} \| Authenticator_c$
(4)	**TGS → C**	$E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$
		$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$
		$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$
		$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$

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

(5)	**C → V**	$Ticket_v \| Authenticator_c$
(6)	**V → C**	$E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)
		$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$
		$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$

(c) Client/Server Authentication Exchange to obtain service

Figure 16.3 Overview of Kerberos



Authentication Server (AS)

The authentication server (AS) is the KDC in the Kerberos protocol.

Ticket-Granting Server (TGS)

The ticket-granting server (TGS) issues a ticket for the real server (Bob).

Real Server

The real server (Bob) provides services for the user (Alice).

Figure 15.7 *Kerberos servers*

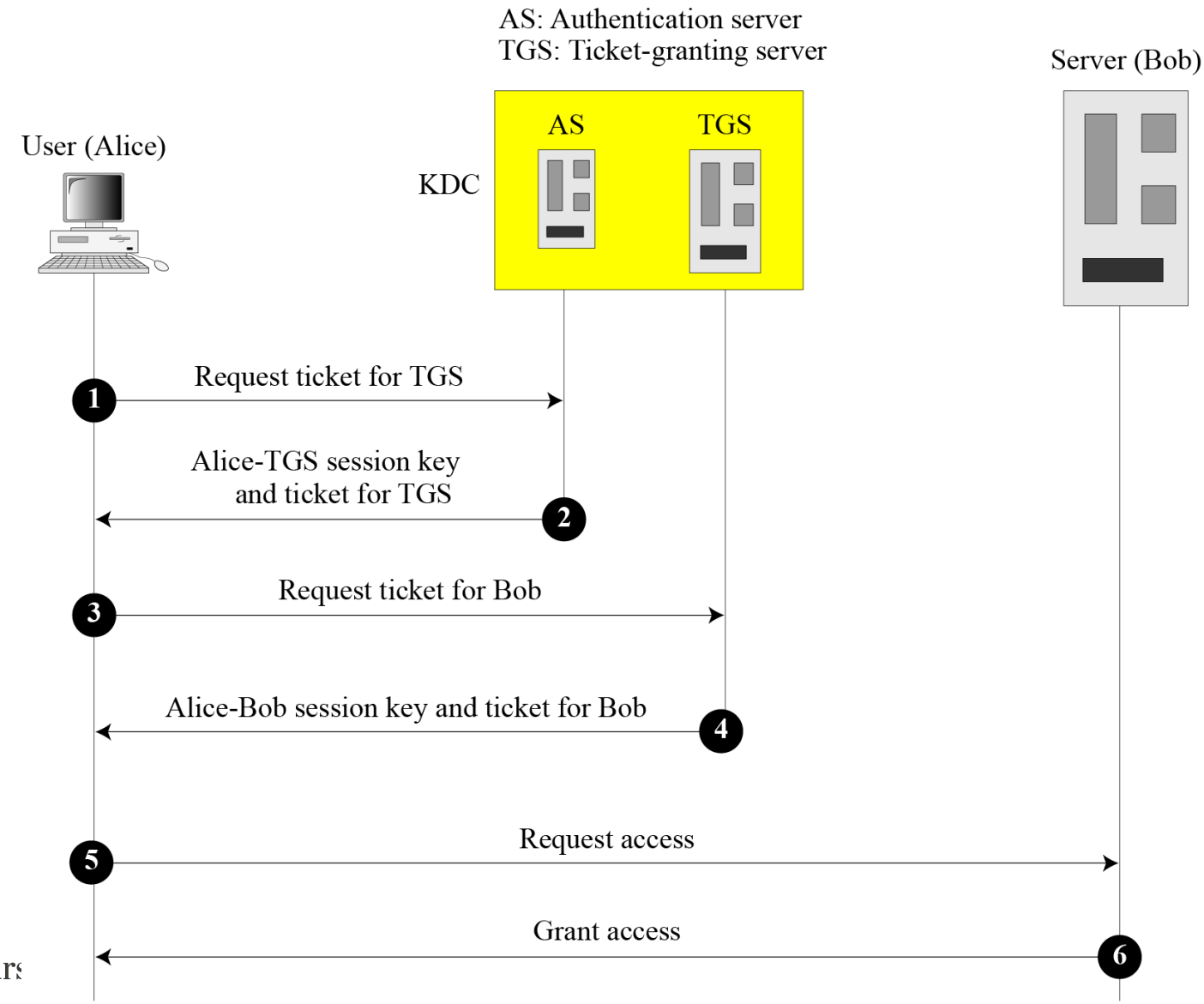


Figure 15.8 *Kerberos example*

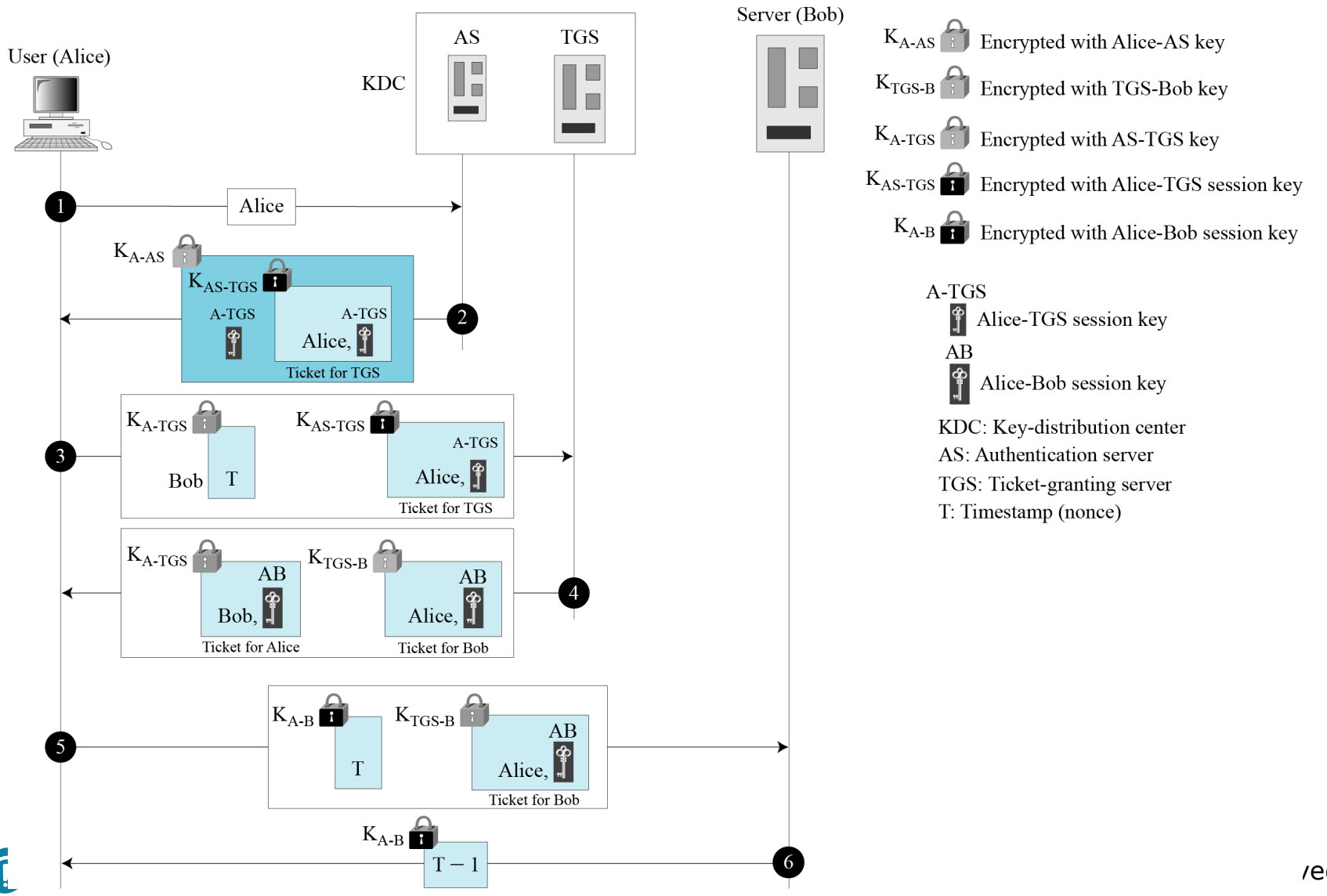
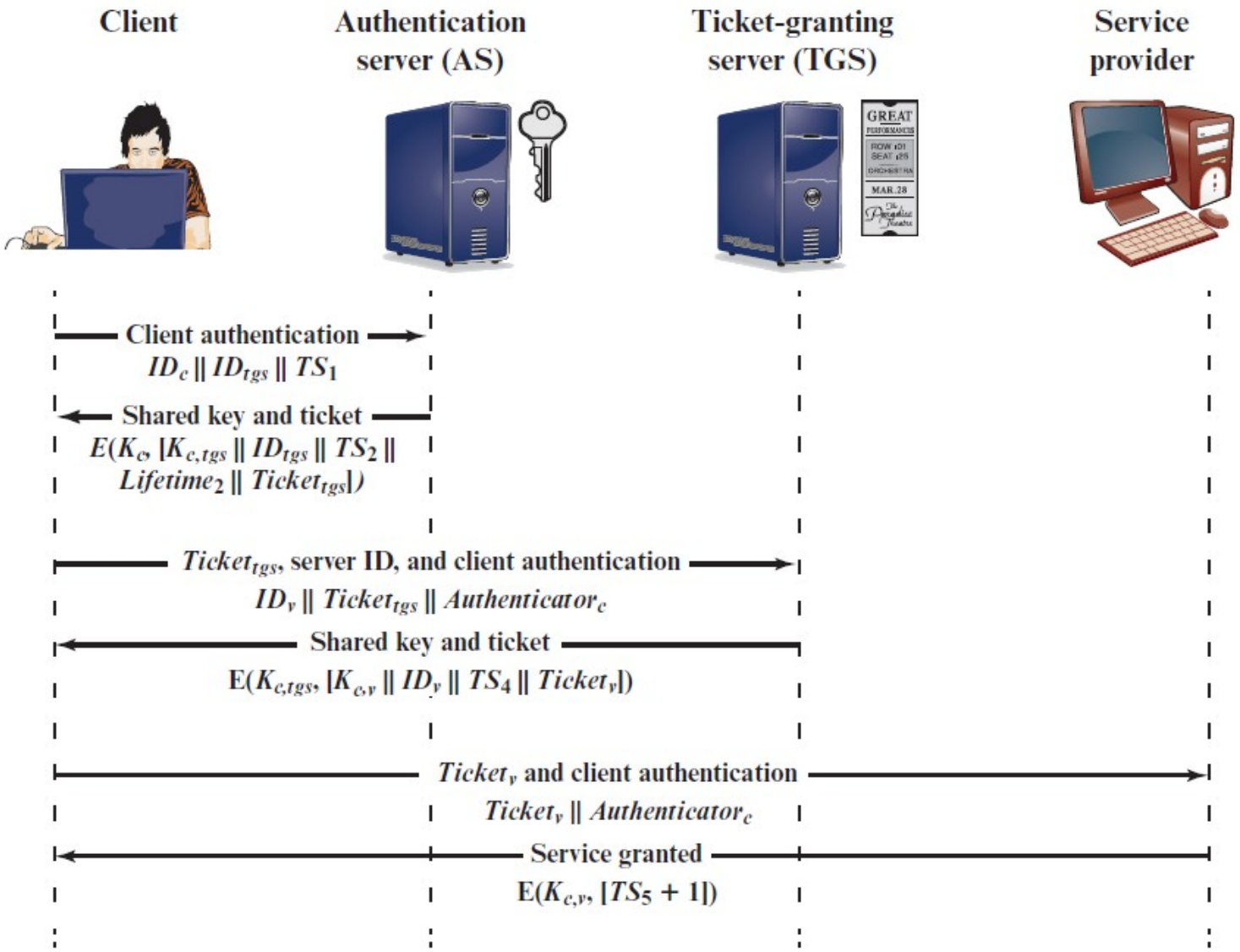


Figure 16.4 Kerberos Exchanges



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

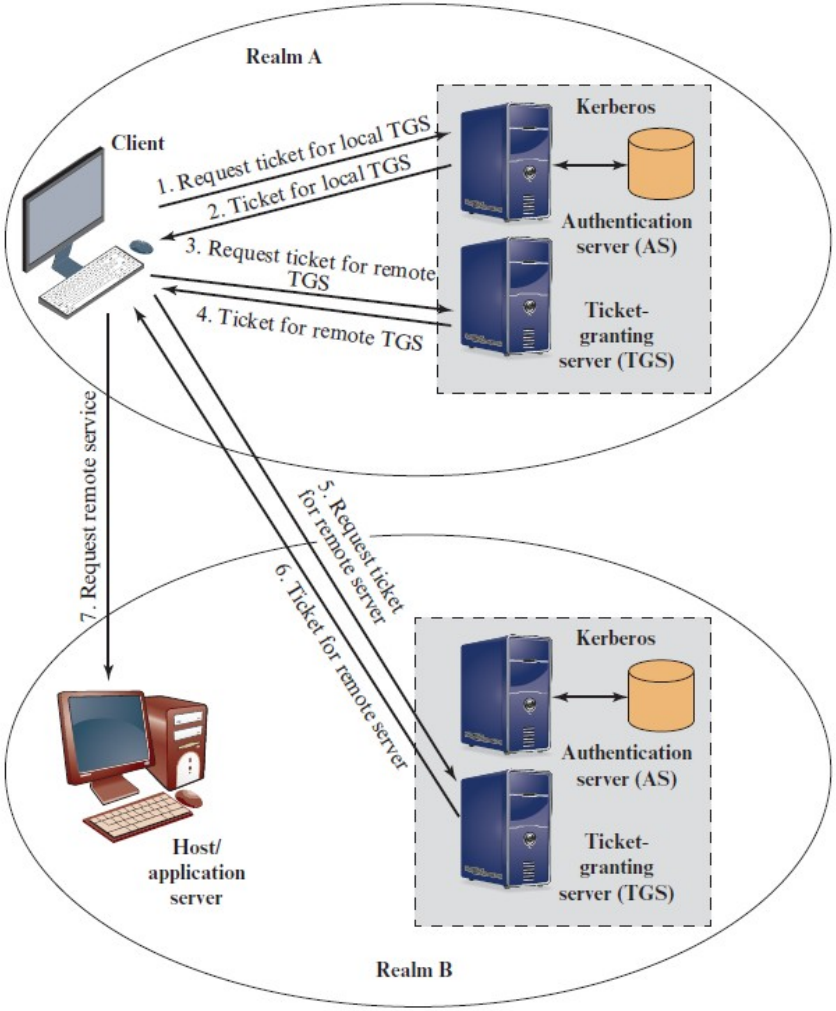
Kerberos Realm

- 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
- A service or user name
- An instance name
- A realm name
- Three parts of a principal name

Figure 16.5 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

Table 16.3 Summary of Kerberos Version 5 Message Exchanges

(1)	C → AS	$ID_c \| ID_{tgs} \| TS_1$
(2)	AS → C	$E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$
		$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$

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

(3)	C → TGS	$ID_v \| Ticket_{tgs} \| Authenticator_c$
(4)	TGS → C	$E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$
		$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$
		$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$
		$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$

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

(5)	C → V	$Ticket_v \| Authenticator_c$
(6)	V → C	$E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)
		$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$
		$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$

(c) Client/Server Authentication Exchange to obtain service

One-Way Authentication (1 of 2)

- Involves a single transfer of information from one user (A) intended for another (B)
- In its simplest form, it would establish the identity of A, the identity of B, and establish that some sort of authentication token actually was generated by A and actually was intended to be sent to B
    - An email message is an example of an application that lends itself to one-way authentication
- For confidentiality encrypt message with a one-time secret key; A also encrypts this one-time key with B's public-key
    - Only B will be able to use the corresponding private key to recover the one-time key and then use that key to decrypt the message
    - This scheme is more efficient than simply encrypting the entire message with B's public key

One-Way Authentication (2 of 2)

- If authentication is the primary concern, a digital signature may suffice
  - This method guarantees that A cannot later deny having sent the message
  - To counter fraud both the message and signature can be encrypted with the recipient's public key
- In addition to the message, A sends B the signature encrypted with A's private key and A's certificate encrypted with the private key of the authentication server
- The recipient of the message first uses the certificate to obtain the sender's public key and verify that it is authentic and then uses the public key to verify the message itself
- If confidentiality is required, then the entire message can be encrypted with B's public key
- Alternatively, the entire message can be encrypted with a one-time secret key; the secret key is also transmitted, encrypted with B's public key

Federated Identity Management

- Relatively new concept dealing with the use of a common identity management scheme across multiple enterprise and numerous applications and supporting many users
- Services provided include:
  - Point of contact
  - SSO protocol services
  - Trust services
  - Key services
  - Identity services
  - Authorization
  - Provisioning
  - Management

Identity Management

- A centralized, automated approach to provide enterprise-wide access to resources by employees and other authorized individuals
- The focus of identity management is defining an identity for each user (human or process), associating attributes with the identity, and enforcing a means by which a user can verify identity
- The central concept of an identity management system is the use of single sign-on (SSO)
- SSO enables a user to access all network resources after a single authentication

Figure 16.6 Generic Identity Management Architecture

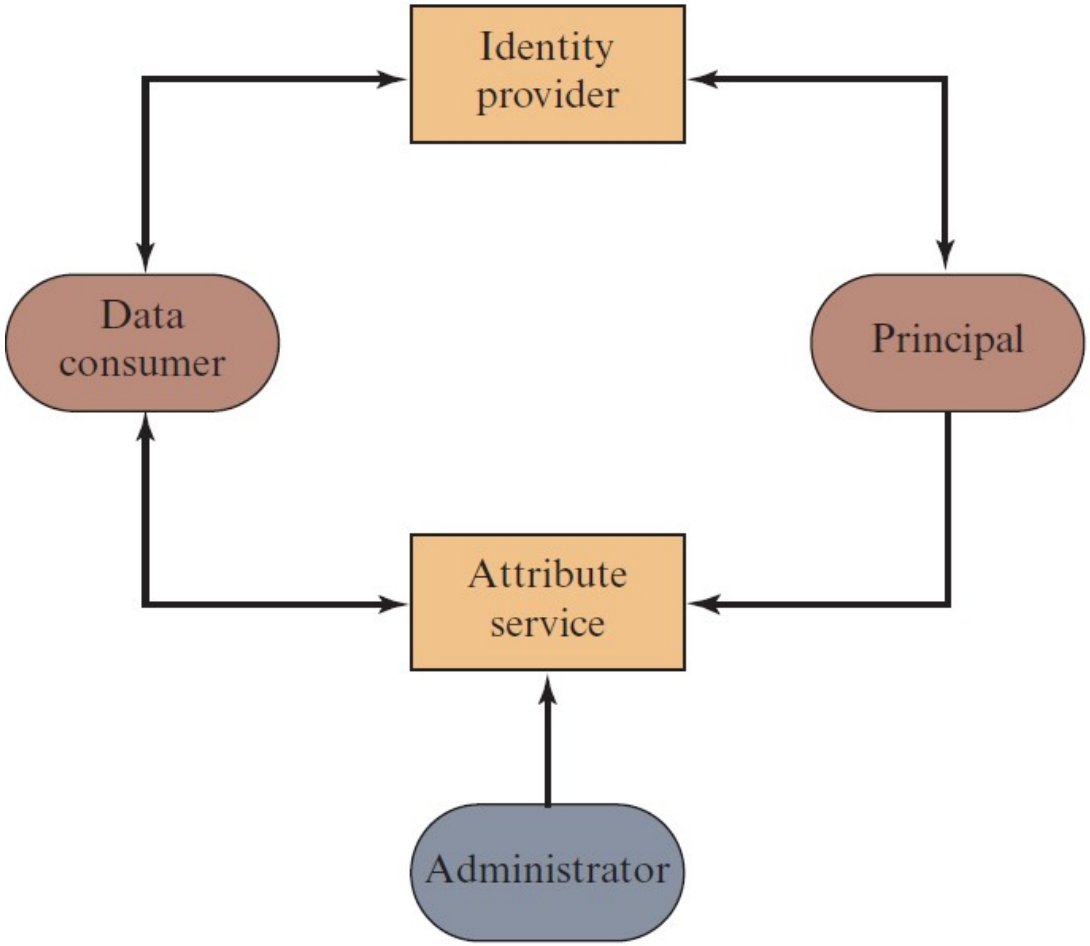
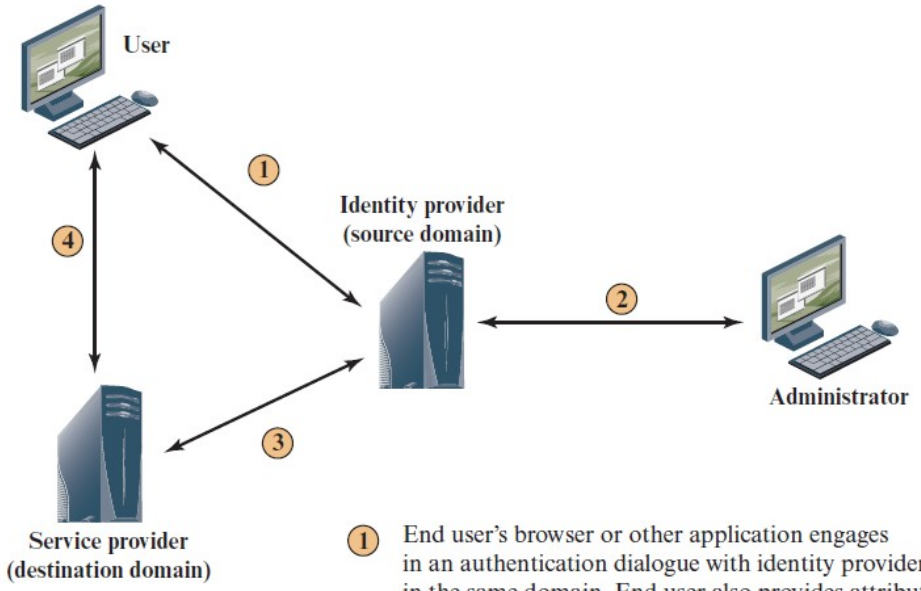


Figure 16.7 Federated Identity Operation



1. End user's browser or other application engages in an authentication dialogue with identity provider in the same domain. End user also provides attribute values associated with user's identity.
2. Some attributes associated with an identity, such as allowable roles, may be provided by an administrator in the same domain.
3. A service provider in a remote domain, which the user wishes to access, obtains identity information, authentication information, and associated attributes from the identity provider in the source domain.
4. Service provider opens session with remote user and enforces access control restrictions based on user's identity and attributes.

Summary

- Present an overview of techniques for remote user authentication using symmetric encryption
- Give a presentation on Kerberos
- Explain the differences between versions 4 and 5 of Kerberos
- Describe the use of Kerberos in multiple realms
- Present an overview of techniques for remote user authentication using asymmetric encryption
- Understand the need for a federated identity management system

Copyright