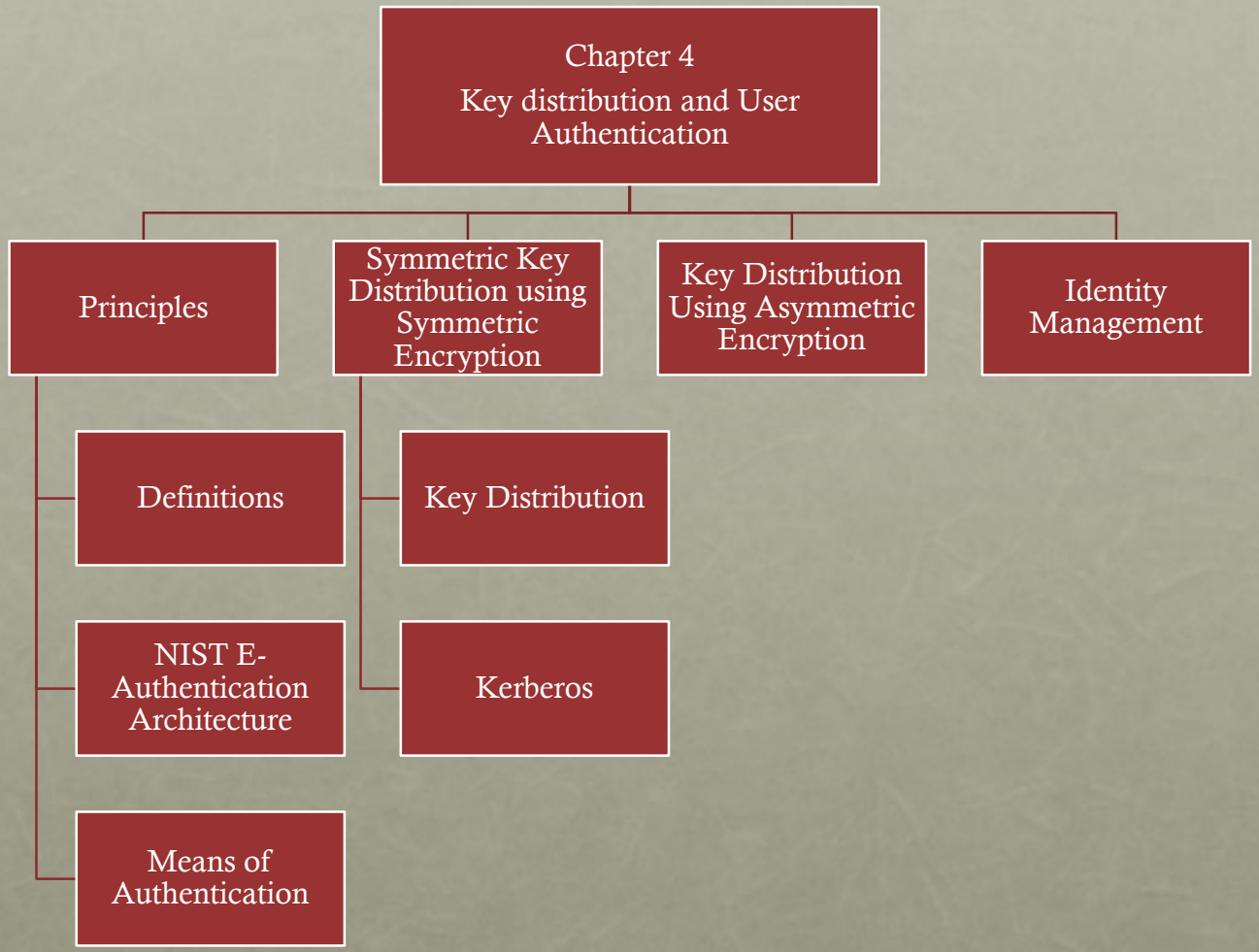
CHAPTER 4

Key Distribution and User Authentication

OVERVIEW


Chapter 4
Key distribution and User Authentication
Principles
Symmetric Key Distribution using Symmetric Encryption
Key Distribution Using Asymmetric Encryption
Identity Management
Definitions
NIST E-Authentication Architecture
Means of Authentication
Key Distribution
Kerberos

REMOTE USER AUTHENTICATION PRINCIPLES

- In most computer security contexts, user authentication is the fundamental building block and the primary line of defense
- User authentication is the basis for most types of access control and for user accountability
- RFC 4949 (Internet Security Glossary) defines user authentication as the process of verifying an identity claimed by or for a system entity
  - Identification step
    - Presenting an identifier to the security system
    - Confidence Level !!!
  - Verification step
    - Presenting or generating authentication information that corroborates the binding between the entity and the identifier

NIST MODEL FOR ELECTRONIC USER AUTHENTICATION

- NIST SP 800-63-2 (*Electronic Authentication Guideline,* August 2013 defines electronic user authentication as the process of establishing confidence in user identities that are presented electronically to an information system
- Systems can use the authenticated identity to determine if the authenticated individual is authorized to perform particular functions
- In many cases, the authentication and transaction or other authorized function take place across an open network such as the Internet
- Equally, authentication and subsequent authorization can take place locally, such as across a local area network

Applicant → Subscriber → Claimant

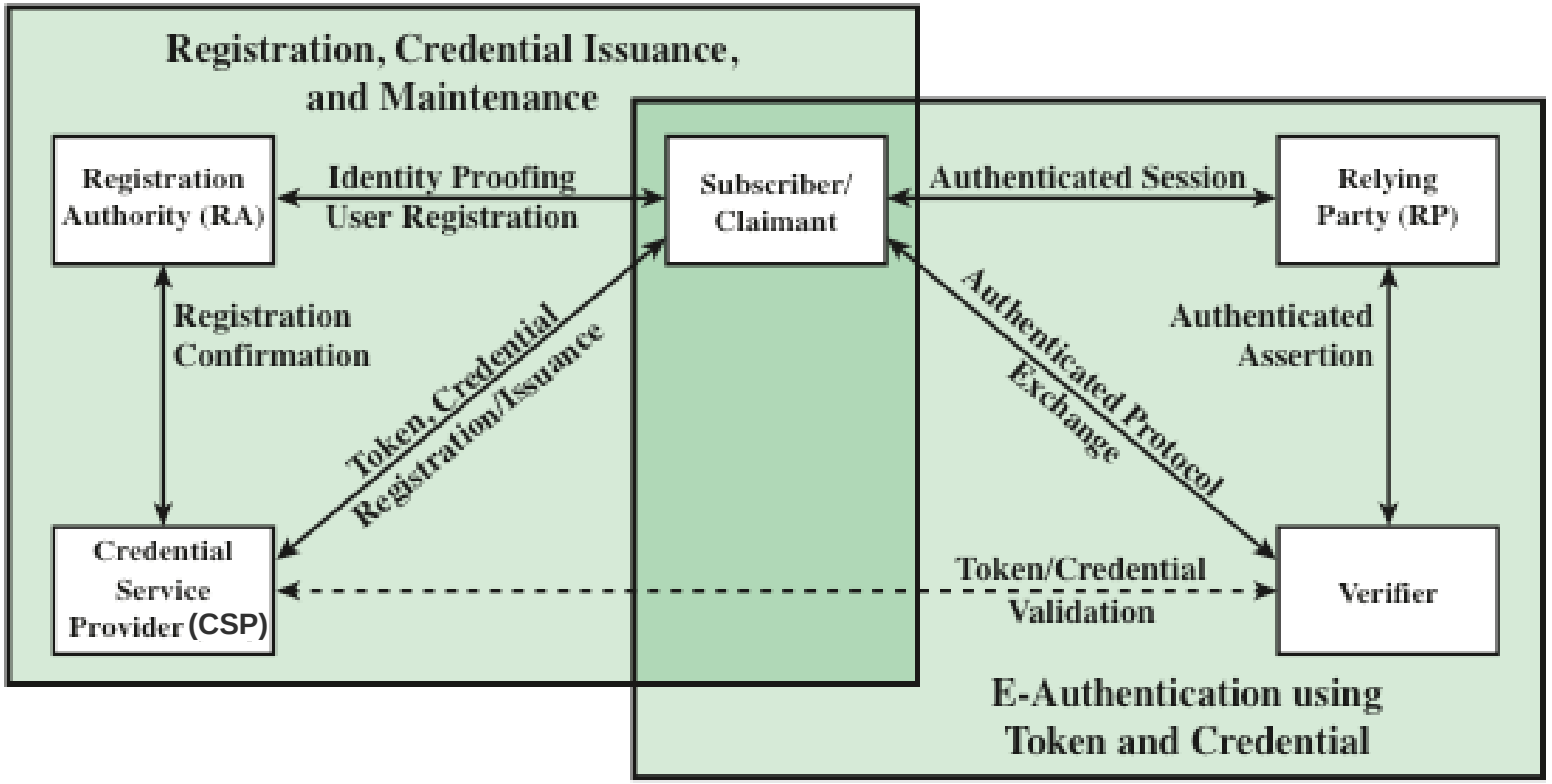
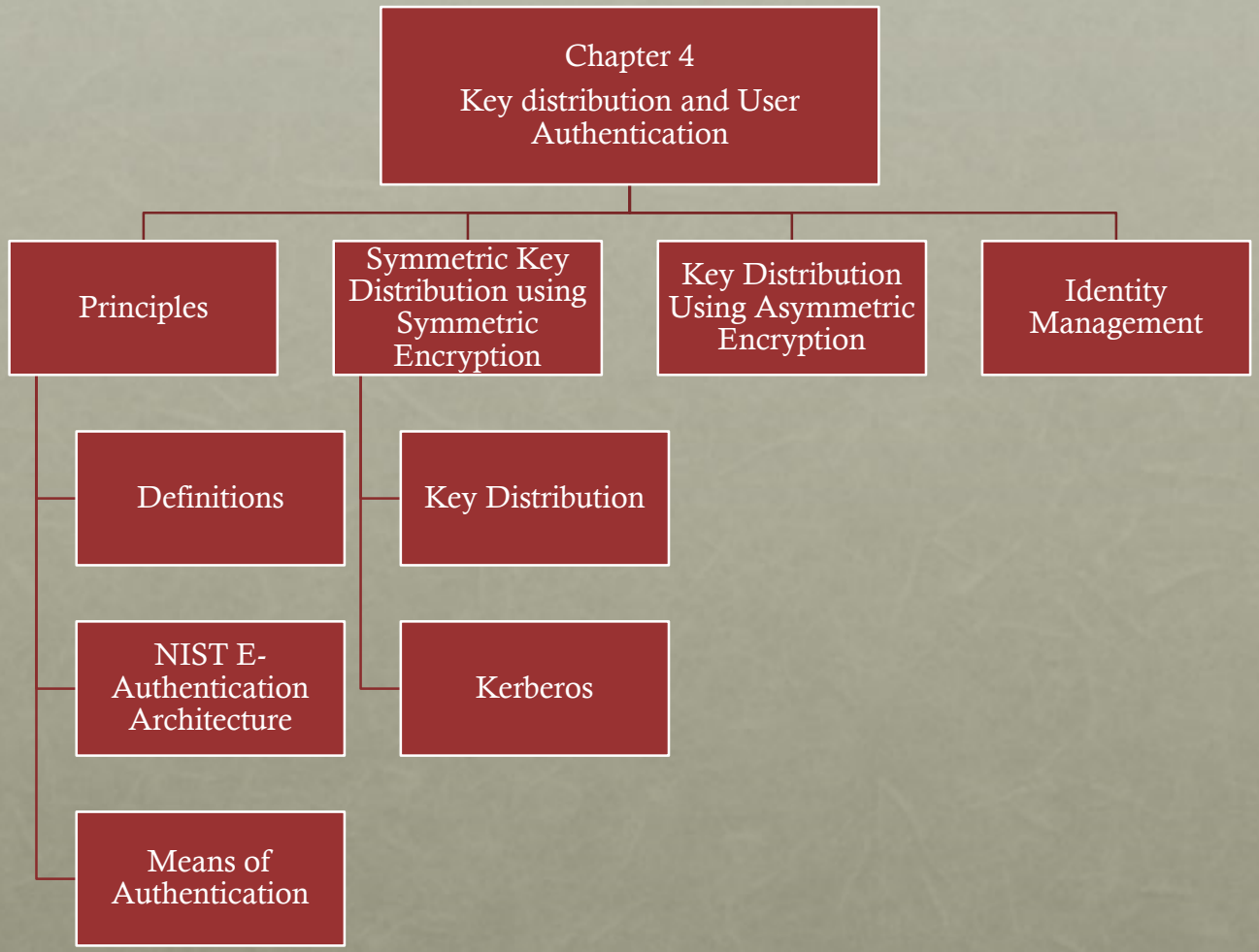


Figure 4.1 The NIST SP 800-63-2 E-Authentication Architectural Model

MEANS OF AUTHENTICATION

- There are four general means of authenticating a user's identity, which can be used alone or in combination
  - Something the individual knows
    - Examples include a password, a personal identification number (PIN), or answers to a prearranged set of questions
  - Something the individual possesses
    - Examples include cryptographic keys, electronic keycards, smart cards, and physical keys
    - This type of authenticator is referred to as a *token*
  - Something the individual is (static biometrics)
    - Examples include recognition by fingerprint, retina, and face
  - Something the individual does (dynamic biometrics)
    - Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm

OVERVIEW


Chapter 4
Key distribution and User Authentication
Principles
Symmetric Key Distribution using Symmetric Encryption
Key Distribution Using Asymmetric Encryption
Identity Management
Definitions
NIST E-Authentication Architecture
Means of Authentication
Key Distribution
Kerberos

SYMMETRIC KEY DISTRIBUTION USING SYMMETRIC ENCRYPTION

- The two parties must share the same key, and that key must be protected from access by others
- Frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key
- **Key distribution technique**
  - The means of delivering a key to two parties that wish to exchange data, without allowing others to see the key
- **Key management technique**
  - Given a large number of keys, how do we preserve their safety and make them available as needed.

KEY DISTRIBUTION

- For two parties A and B, there are the following options:

1. A key can be selected by A and physically delivered to B
2. A third party can select the key and physically deliver it to A and B
3. If A and B have previously and recently used a key, one party could transmit the new key to the other, using the old key to encrypt the new key
4. If A and B each have an encrypted connection to a third party C, C could deliver a key on the encrypted links to A and B → see **KDC in next slide**

KEY DISTRIBUTION

- Two types of keys for option 4:
  1. Session key: temporary for a duration of the session
  2. Permanent key: used for distributing session keys

- key distribution center (KDC) grants two systems to establish a connection, and provides a one-time session key. Works as follows:
  1. When host A wishes to set up a connection to host B, it transmits a connection request packet KDC through encrypted link using a master (permanent) key shared only by A and the KDC.
  2. If KDC approves the request, it generates a unique one-time session key. It encrypts the session key using the key it shares with A and delivers the encrypted session key to A. Similarly, it encrypts the session key using the permanent key it shares with B and delivers the encrypted session key to B.
  3. A and B can now set up a logical connection and exchange messages and data, all encrypted using the temporary session key.

NUMBER OF KEYS

- Symmetric encryption
  - For a network of N users

$$\mathrm{K} = C_2^N = \frac{N(N-1)}{2}$$

- Public encryption
  - For a network of N users.

$$\mathrm{K} = 2N$$

- However, symmetric encryption has much less complexity for the same key size

KERBEROS

- **Motivation**: Assume a distributed environment in which users at workstations wish to access services on servers distributed throughout the network. Servers then need to restrict access to authorized users and to be able to authenticate requests for service. In this environment, a workstation cannot be trusted to identify its users correctly to network services, because:

  1. A user may gain access to a particular workstation and pretend to be another user operating from that workstation.

  2. 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.

  3. A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations.

KERBEROS

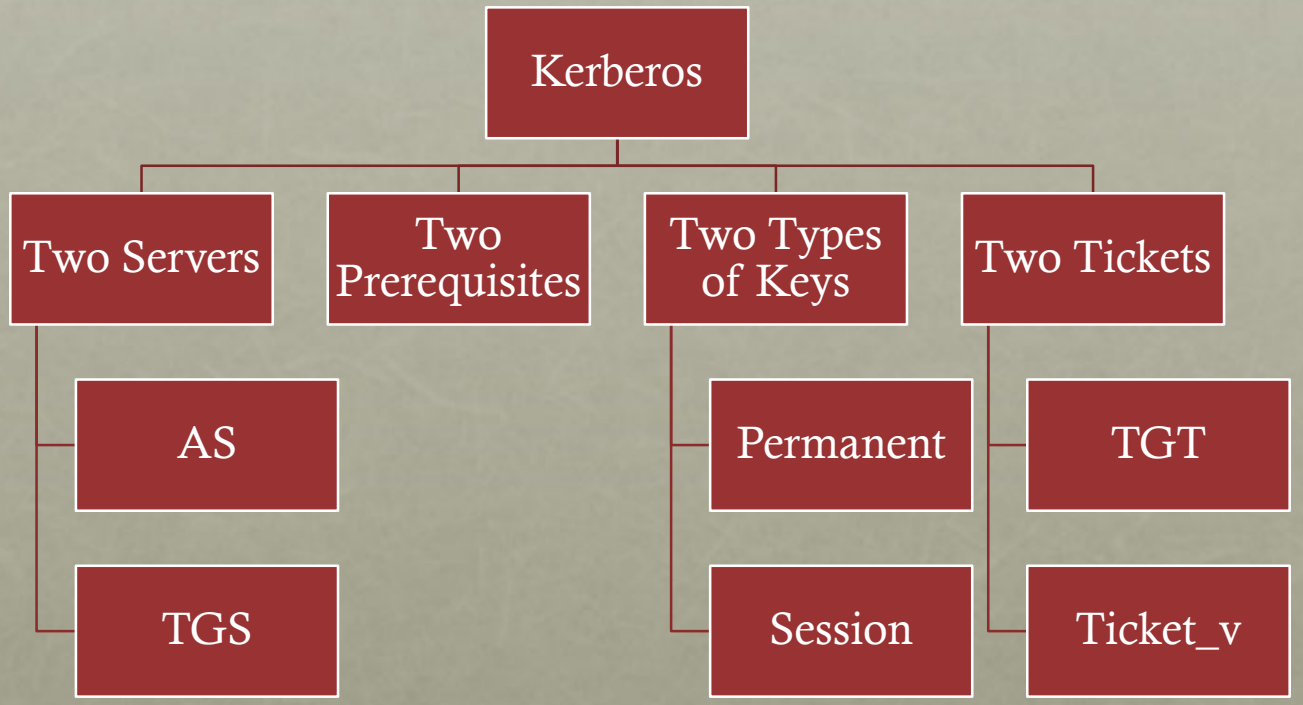
- What is Kerberos? What is it used for?
  - Key distribution and user authentication service developed at MIT
  - Provides a centralized authentication server whose function is to authenticate users to servers and servers to users
  - Relies exclusively on symmetric encryption, making no use of public-key encryption

Two Objectives

Two versions are in use

- Version 4 implementations still exist, although this version is being phased out
- Version 5 corrects some of the security deficiencies of version 4 and has been issued as a proposed Internet Standard (RFC 4120)

KERBEROS


Kerberos
Two Servers
Two Prerequisites
Two Types of Keys
Two Tickets
AS
TGS
Permanent
Session
TGT
Ticket_v

KERBEROS VERSION 4

- A basic third-party authentication scheme
- Authentication Server (AS)
  - Users initially negotiate with AS to identify self
  - AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
- Ticket Granting Server (TGS)
  - Users subsequently request access to other services from TGS on basis of users TGT
- Complex protocol using DES

KERBEROS REALM

- The Kerberos server must have the user ID and hashed passwords of all participating users in its database.

  1. ALL USERS ARE REGISTERD WITH THE KERBEROS SERVER

Two Prerequisites

- The Kerberos server must share a secret key with each server.

  2. ALL SERVERS ARE REGISTERD WITH THE KERBEROS SERVER

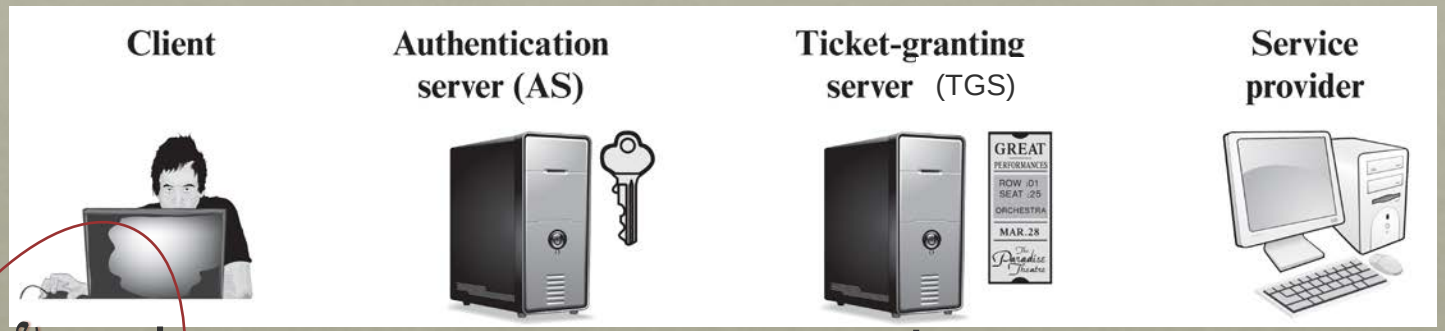
AUTHENTICATION SCENARIO

- If user C logs on to a workstation in the morning and wishes to check his/her email at server V. *Solution*: C authenticates through AS to get a ticket. But C must use a password to get a ticket for the mail server (Shouldn't send the password to server clear plaintext).
- If C wishes to check the mail several times during the day, each attempt requires reentering the password. *Solution*: Tickets reusable (i.e. the workstation can store the mail-server ticket and use it on behalf of the user for multiple accesses to the mail server)
- Still, the user would need a new ticket for every service e.g. email, printing, file server, etc.
- Hence, TGS is needed so user can get a ticket for each service using its authentication ticket.

SYMMETRIC KEYS USED IN KERBEROS

- **Permanent Symmetric Keys**

Two Types of Keys



Used as part of the authentication step → objective 1

K_c

Symmetric key between the client (C) and AS

K_{tgs}

Symmetric key between AS and TGS

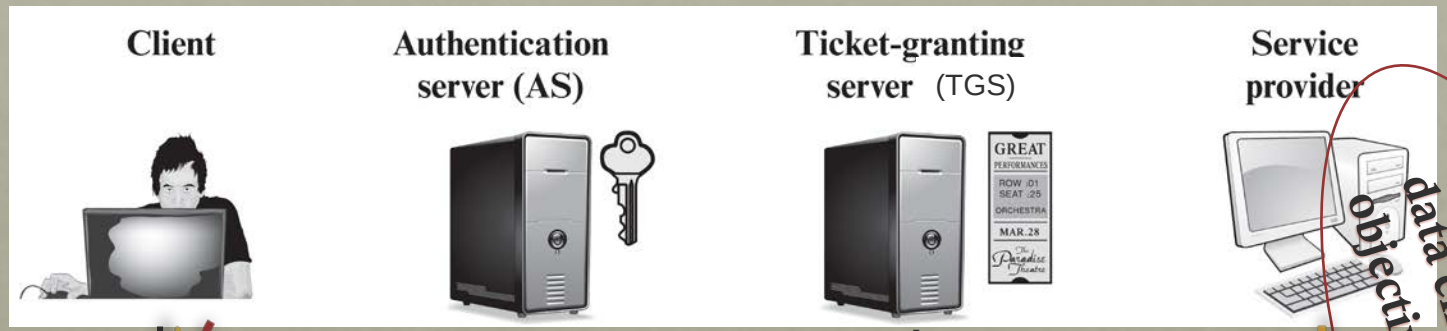
K_v

Symmetric key between TGS and Server (V)

SYMMETRIC KEYS USED IN KERBEROS

- **Session Symmetric Keys**

Two Types of Keys



$K_{c,tgs}$

Symmetric Key between C and TGS

$K_{c,v}$

Key to be used for data encryption objective 2 →

TICKETS IN KERBEROS

- Ticket-granting Ticket ($Ticket_{tgs}$)
  - Provided from AS to C
  - To allow C to start communication with TGS
  - Encrypted with K_{tgs} → only TGS can decrypt it
- Ticket ($Ticket_v$)
  - Provided from TGS to C
  - To allow C to access server V
  - Encrypted with K_v → only V can decrypt it
- **Note**: tickets are encrypted using permanent keys

Two Tickets

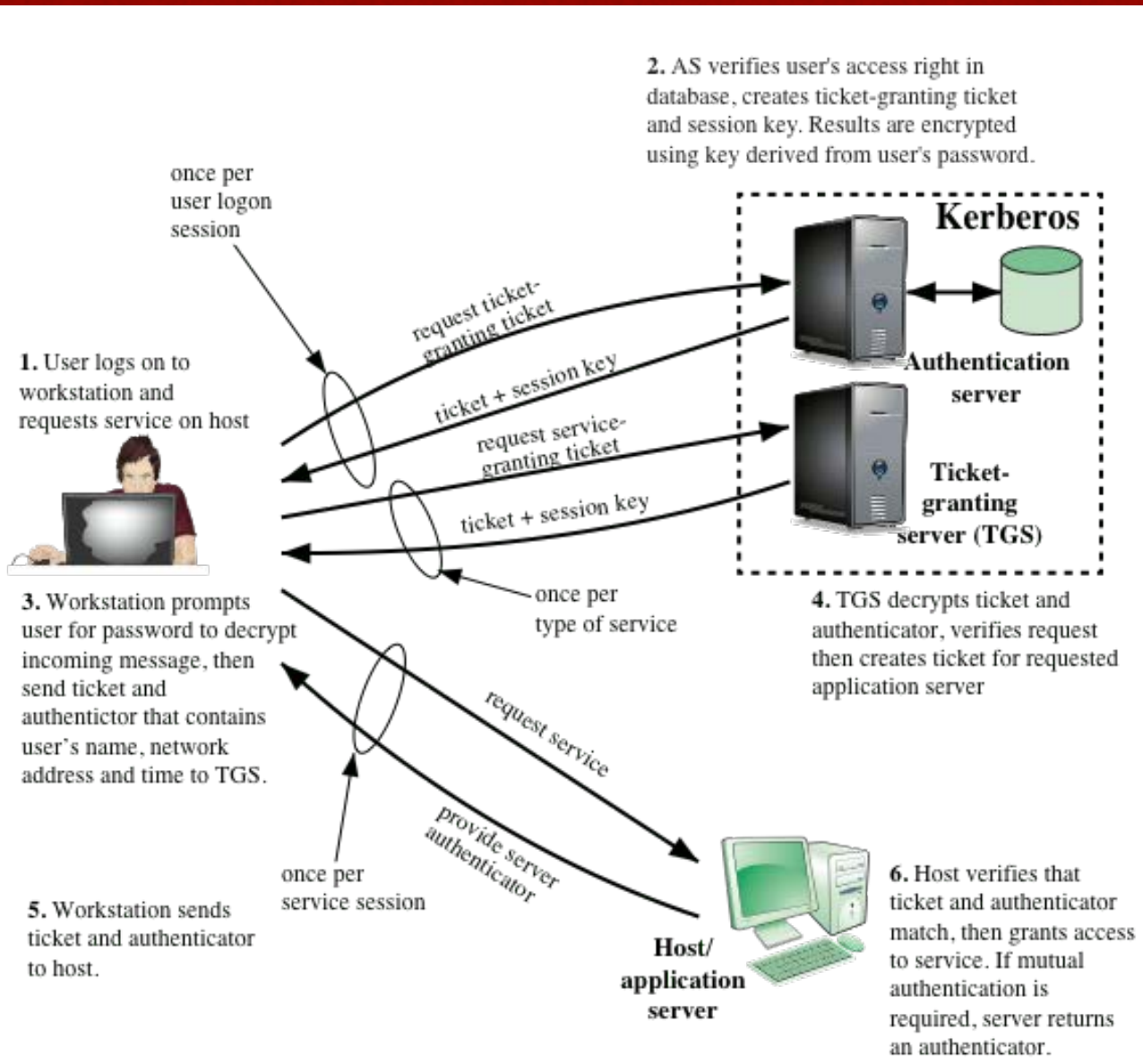
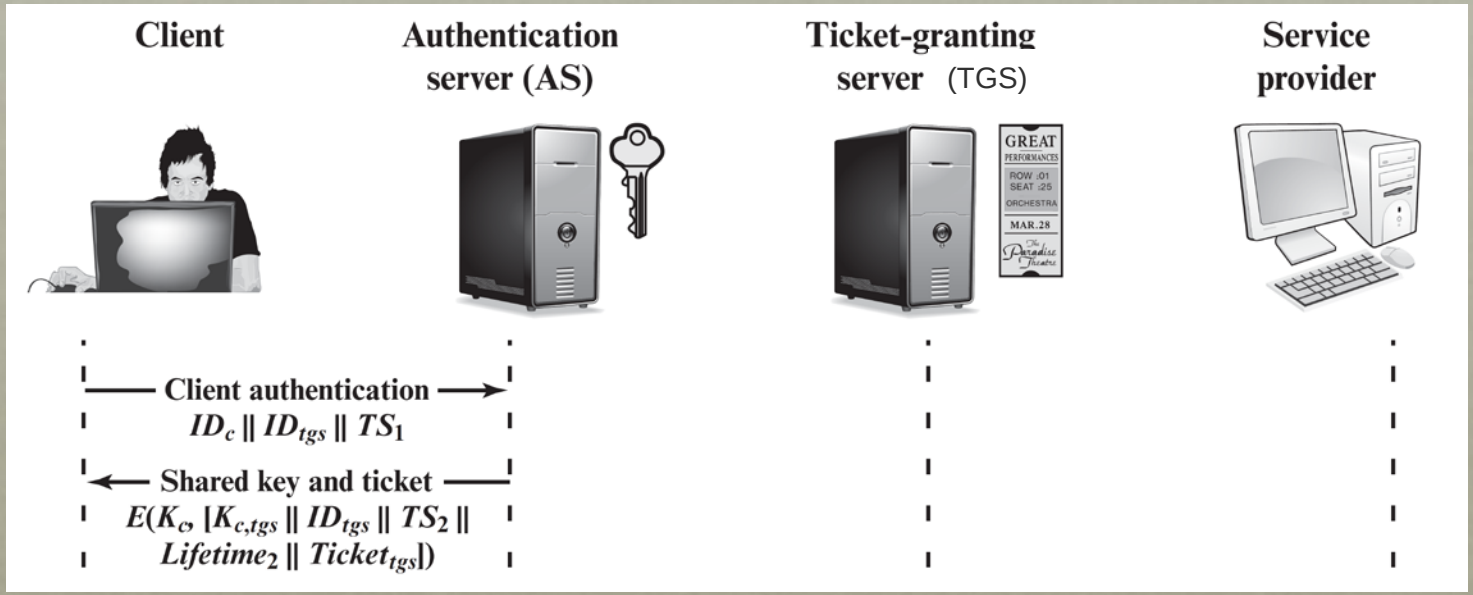


Figure 4.2 Overview of Kerberos

THREE STEPS OF KERBEROS



(1) C → AS $ID_c \| ID_{tgs} \| TS_1$

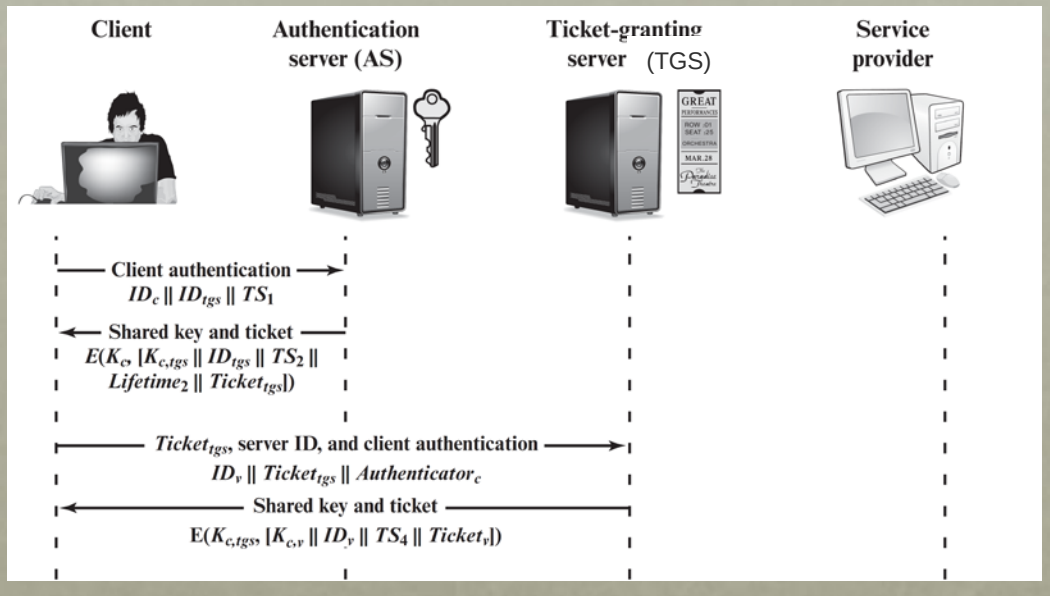
(2) AS → C $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$

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

By the end of step 1, what we have achieved so far?

THREE STEPS OF KERBEROS



(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

By the end of step 2, what do we have?

THREE STEPS OF KERBEROS

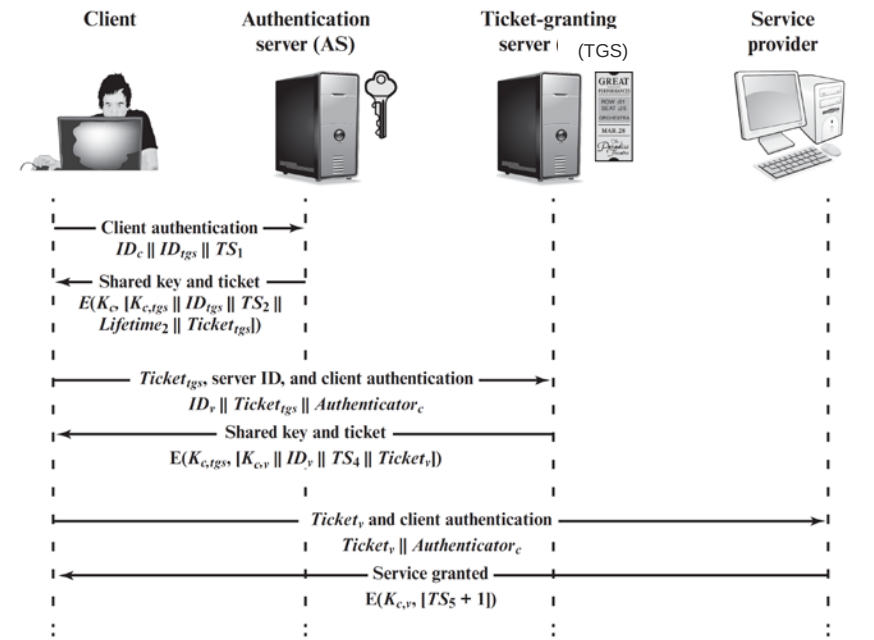


Figure 4.3 Kerberos Exchanges

(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

By the end of step 3, what do we have?

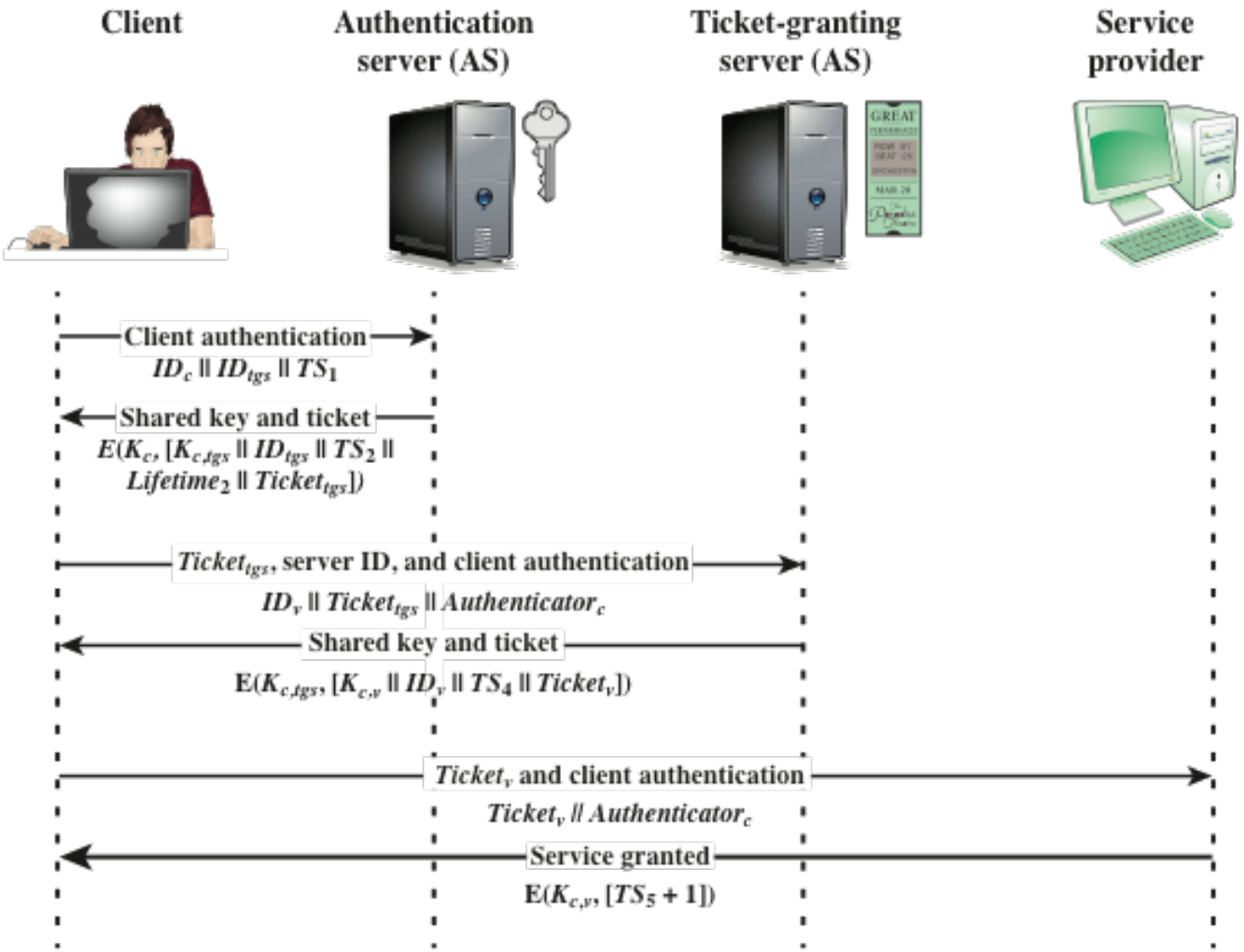


Figure 4.3 Kerberos Exchanges

TABLE 4.1
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

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.

(a) Authentication Service Exchange

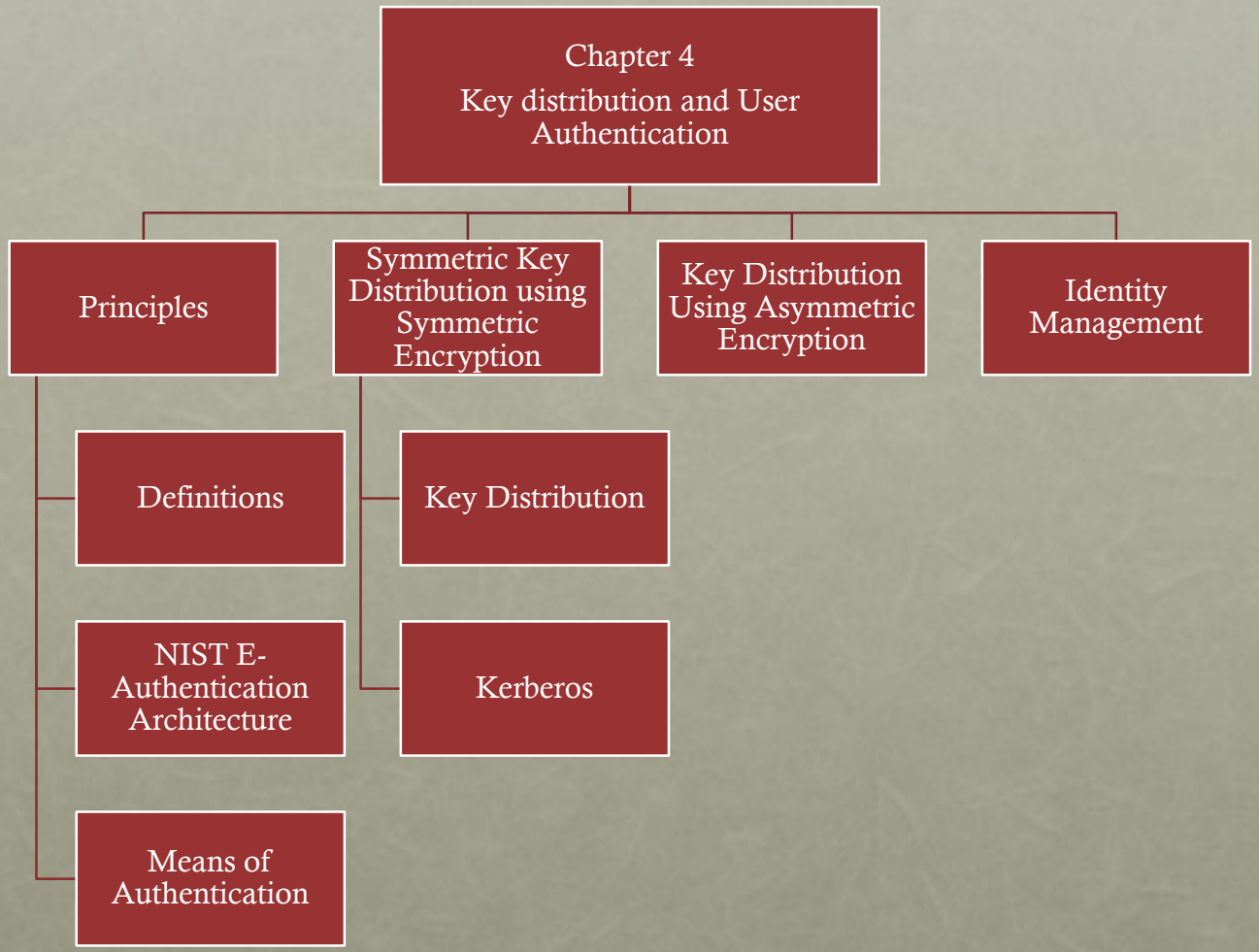
Message (3)	Client requests service-granting ticket.
ID_V	Tells TGS that user requests access to server V.
$Ticket_{tgs}$	Assures TGS that this user has been authenticated by AS.
$Authenticator_c$	Generated by client to validate ticket.
Message (4)	TGS returns service-granting ticket.
$K_{c,tgs}$	Key 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.

(b) Ticket-Granting Service Exchange

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.

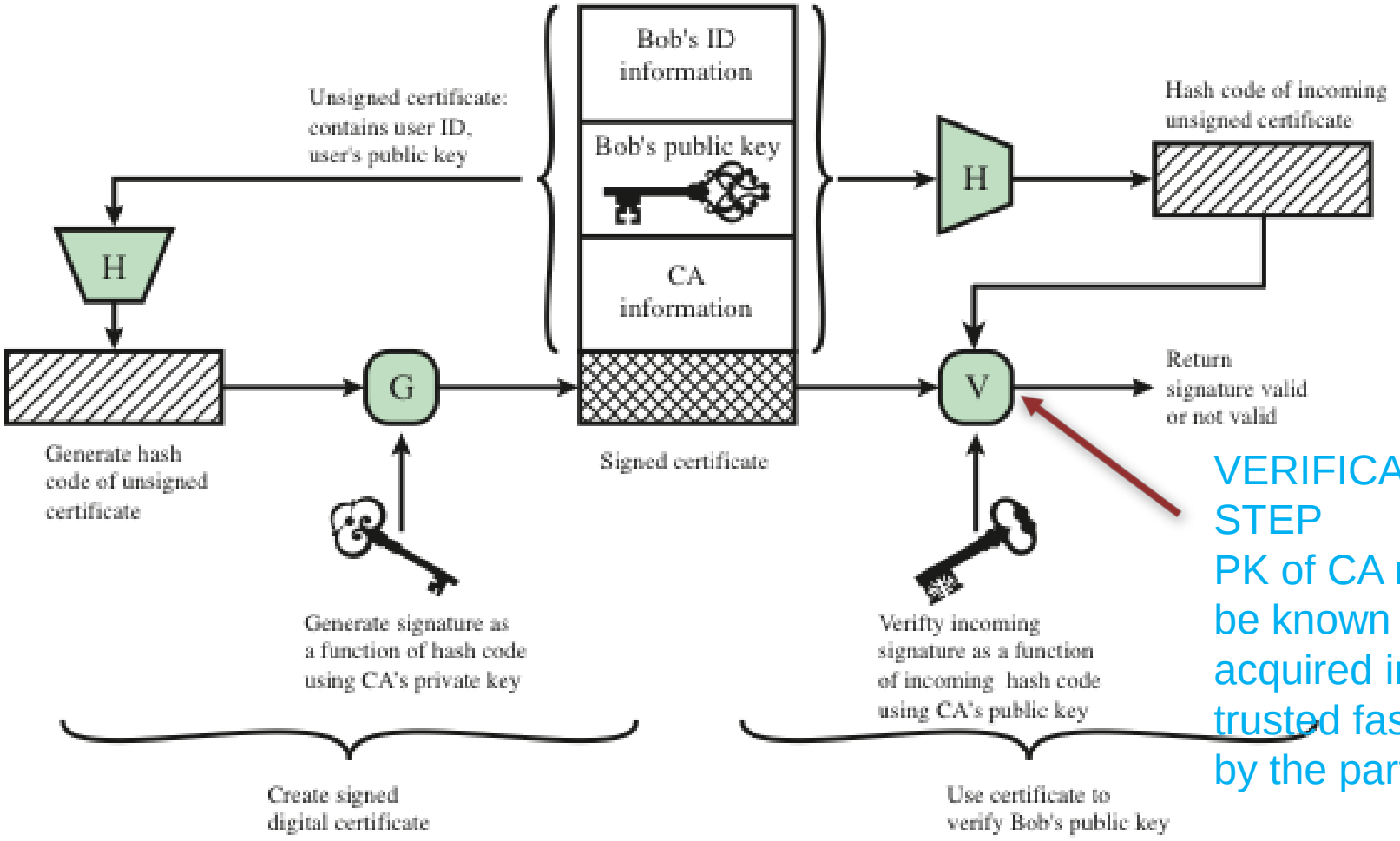
(c) Client/Server Authentication Exchange

OVERVIEW


Chapter 4
Key distribution and User Authentication
Principles
Symmetric Key Distribution using Symmetric Encryption
Key Distribution Using Asymmetric Encryption
Identity Management
Definitions
NIST E-Authentication Architecture
Means of Authentication
Key Distribution
Kerberos

KEY DISTRIBUTION USING ASYMMETRIC ENCRYPTION

- One of the major roles of public-key encryption is to address the problem of key distribution
- There are two distinct aspects to the use of public-key encryption in this regard:
  - The distribution of public keys
  - The use of public-key encryption to distribute secret keys
- Public-key certificate
  - Consists of a public key plus a user ID of the key owner, with the whole block signed by a trusted third party
  - Typically, the third party is a certificate authority (CA) that is trusted by the user community, such as a government agency or a financial institution
  - A user can present his or her public key to the authority in a secure manner and obtain a certificate
  - The user can then publish the certificate
  - Anyone needing this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature

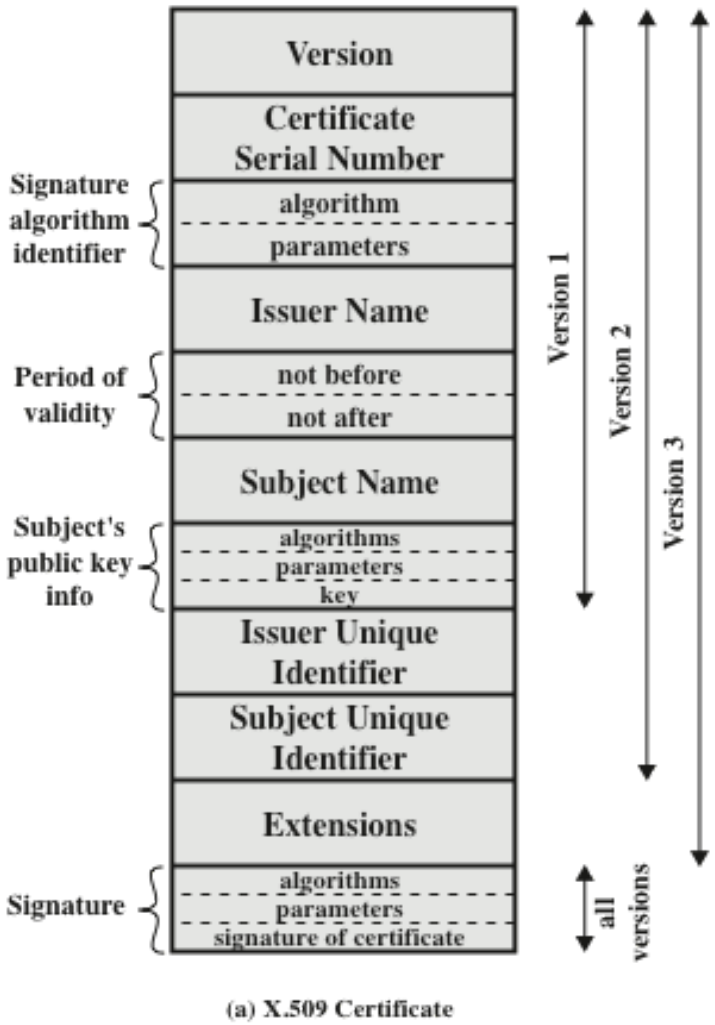


VERIFICATION STEP
PK of CA must be known or acquired in a trusted fashion by the party

Figure 4.4 Public-Key Certificate Use

X.509 CERTIFICATES

- ITU-T recommendation X.509 is part of the X.500 series of recommendations that define a directory service
- Defines a framework for the provision of authentication services by the X.500 directory to its users
- The directory may serve as a repository of public-key certificates
- Defines alternative authentication protocols based on the use of public-key certificates
  - Was initially issued in 1988, currently at ver 9 released 2021
  - Based on the use of public-key cryptography and digital signatures
- The standard does not dictate the use of a specific algorithm but recommends RSA



(a) X.509 Certificate

Figure 4.5 X.509 formats

OBTAINING A USER'S CERTIFICATE

- User certificates generated by a CA have these characteristics:
  - **CA PK can verify the user PK that was certified**
  - No party can modify the certificate other than CA
- Certificates are unforgeable, they can be placed without protection.
- If all users subscribe to same CA, there is a common trust of that CA.
  - All user certificates can be placed in the directory for access by all users.
  - In addition, a user can transmit his or her certificate directly to other users.
- **If there is a large community of users**, it may not be practical for all users to subscribe to the same CA.
- Because it is the CA that signs certificates, each user must have a copy of the CA's PK to verify signatures.
- With many users, it may be more practical for there to be a number of CAs, each of which securely provides its PK to some fraction of the users.

X.509 MULTIPLE CA'S: PROBLEM

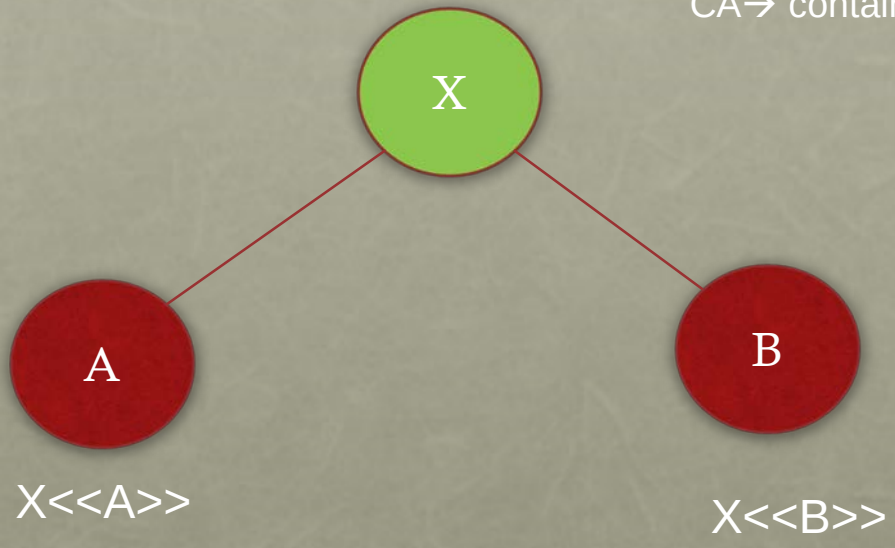
- Consider this Scenario …
  - User A obtained A's certificate from CA X1.
  - User B obtained B's certificate from CA X2.
  - If A does not know X2's public key, B's certificate is useless.
    - A can read B's certificate
    - A cannot verify the signature
- Solution: CAs X1 and X2 exchange public keys
- Then…
  - A gets X2's certificate signed by X1
  - A gets B's certificate signed by X2
  - Now, A has trusted copy of X2's public key
    - Verifies the signature
    - Obtains B's public key: Chain certificates $X_1 << X_2 >> X_2 << B >>$

MULTIPLE CA'S

- A wants to communicate with B
- B sends its certificate to A
- A must have the PK of the CA of B to verify the certificate and obtain the PK of B.

CA<<Subject>>

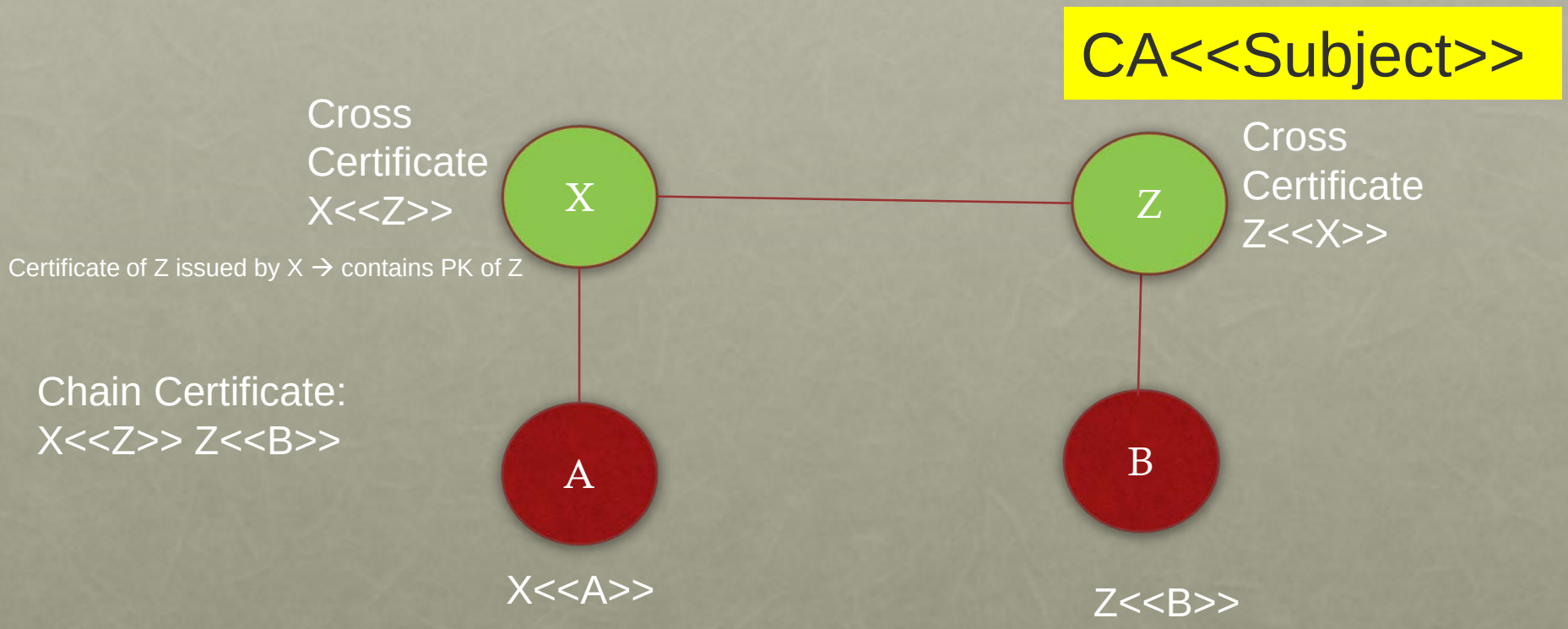
Certificate of "Subject" Issued by CA→ contains PK of "Subject"



MULTIPLE CA'S

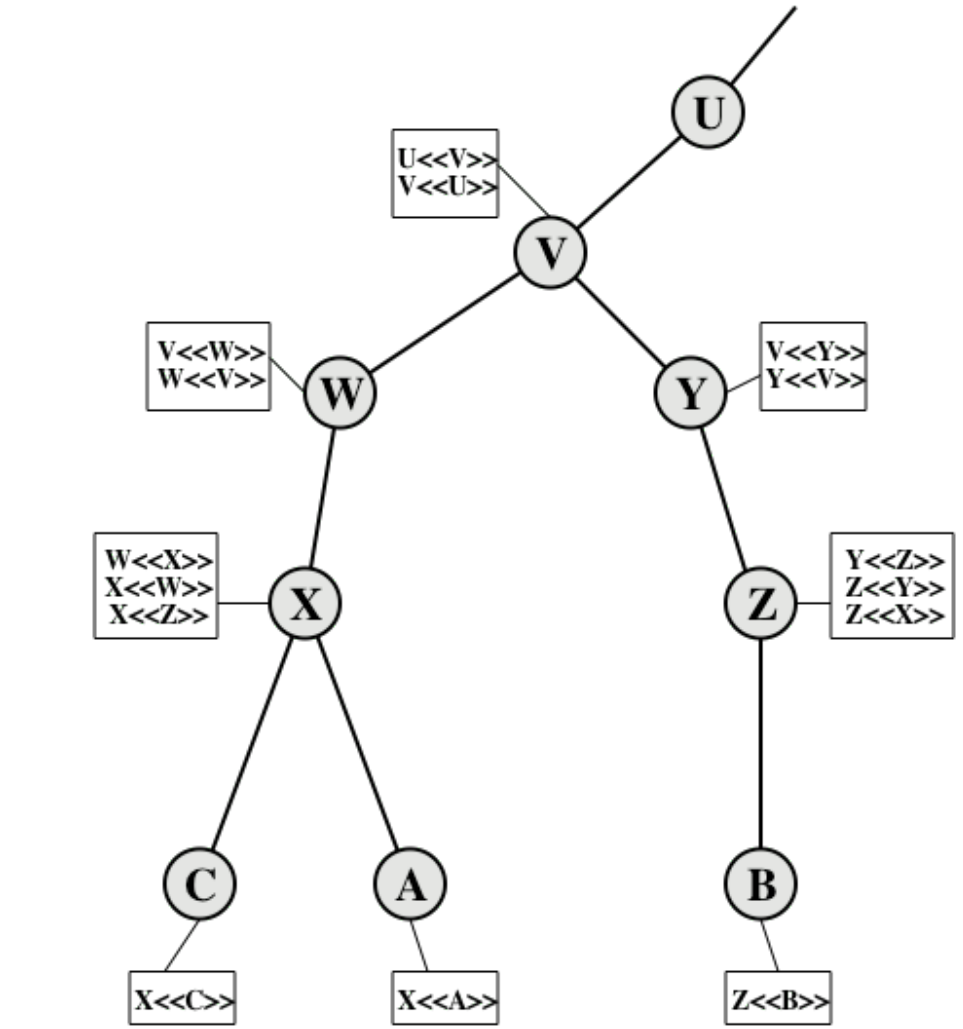
- A wants to communicate with B
- B sends its certificate to A
- A must have the PK of the CA of B to verify the certificate and obtain the PK of B.

CA<<Subject>>



- **Forward certificates**: Certificates of X generated by other CAs

- **Reverse certificates**: Certificates generated by X for other CAs



X<<W>> W<<V>> V<<Y>> Y<<Z>> Z<<B>>

Figure 4.6 X.509 CA Hierarchy: a Hypothetical Example

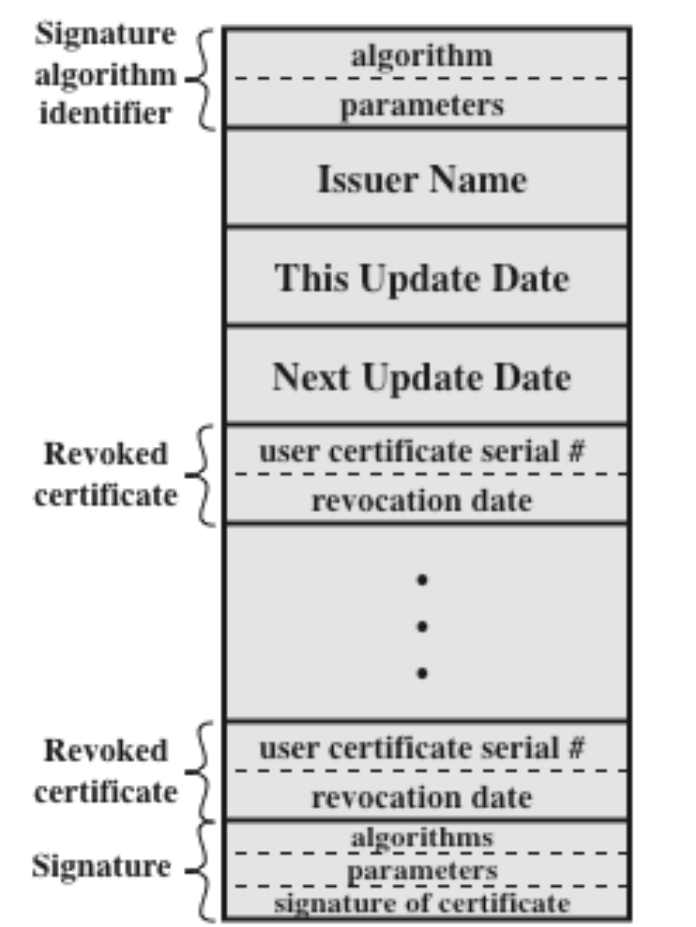
REVOCATION OF CERTIFICATES

- Each certificate includes a period of validity
- Typically a new certificate is issued just before the expiration of the old one
- It may be desirable on occasion to revoke a certificate before it expires for one of the following reasons:
  - The user's private key is assumed to be compromised
  - The user is no longer certified by this CA; reasons for this include subject's name has changed, the certificate is superseded, or the certificate was not issued in conformance with the CA's policies
  - The CA's certificate is assumed to be compromised

X.509: Certificate Revocation List (CRL)

- Each CA maintains a list of all revoked not-expired certificates.
  - issued by that CA to users
  - issued to other CAs
- Certificate Revocation List (CRL) posted to the directory is signed by the issuer and includes:
  - issuer's name
  - list creation date
  - next CRL creation date
  - List of entries, each revoked certificate entry includes:
    - serial number and
    - revocation date

X.509: CERTIFICATE REVOCATION LIST (CRL)

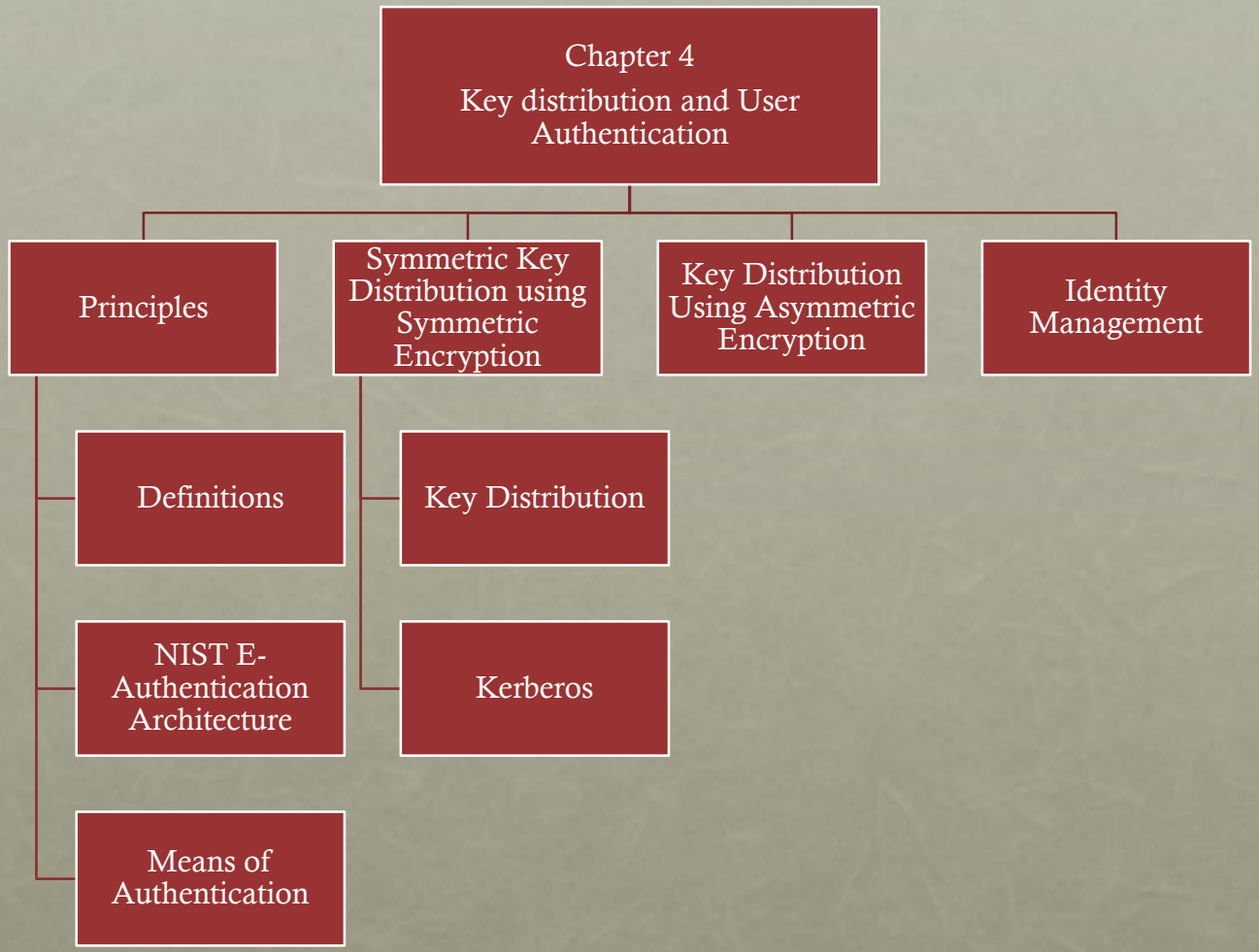


(b) Certificate Revocation List

KEY AND POLICY INFORMATION

- Certificates extensions convey additional information about the subject and issuer keys, plus indicators of certificate policy
- A certificate policy is a named set of rules that indicate the applicability of a certificate to a particular community and/or class of application with common security requirements
- For example, a policy might be applicable to the authentication of electronic data interchange (EDI) transactions for the trading of goods within a given price range.

OVERVIEW


Chapter 4
Key distribution and User Authentication
Principles
Symmetric Key Distribution using Symmetric Encryption
Key Distribution Using Asymmetric Encryption
Identity Management
Definitions
NIST E-Authentication Architecture
Means of Authentication
Key Distribution
Kerberos

IDENTITY MANAGEMENT

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

- Principal elements of an identity management system:
  - **Authentication:** SSO to access resources
  - **Authorization:** granting access based on authentication
  - **Accounting**: logging access and authorizations.
  - **Provisioning**: the process of coordinating the creation of user accounts, in the form of rules and roles, provision of physical resources associated with users.
  - **Workflow automation:** specify a sequence of events based on user role
  - **Delegated administration:** delegate admin for specific subsystems
  - **Password synchronization:** multiple passwords SSO, and RSO.
  - **Self-service password reset:** modify passwords
  - **Federation**: sharing identity across multiple systems.

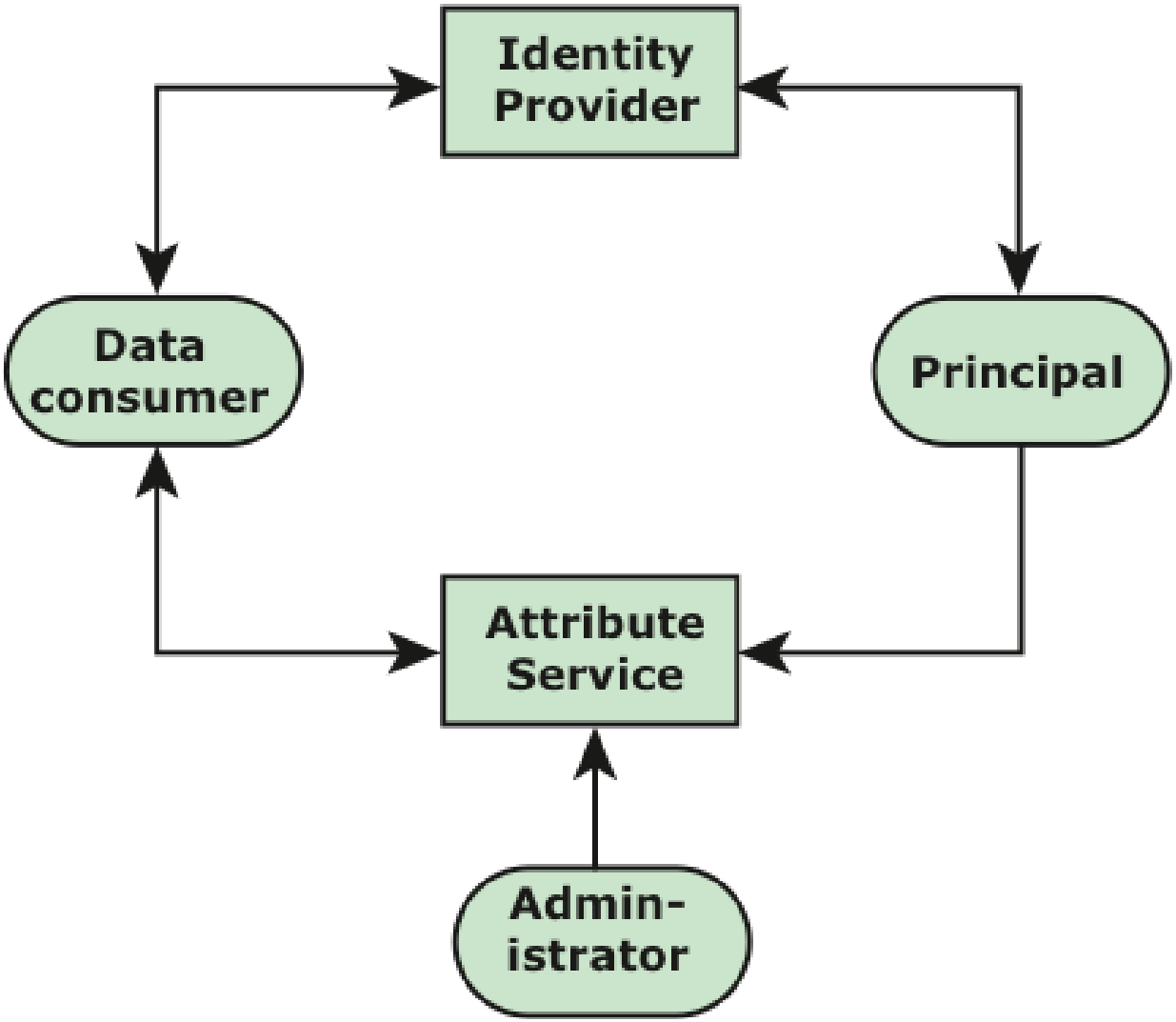
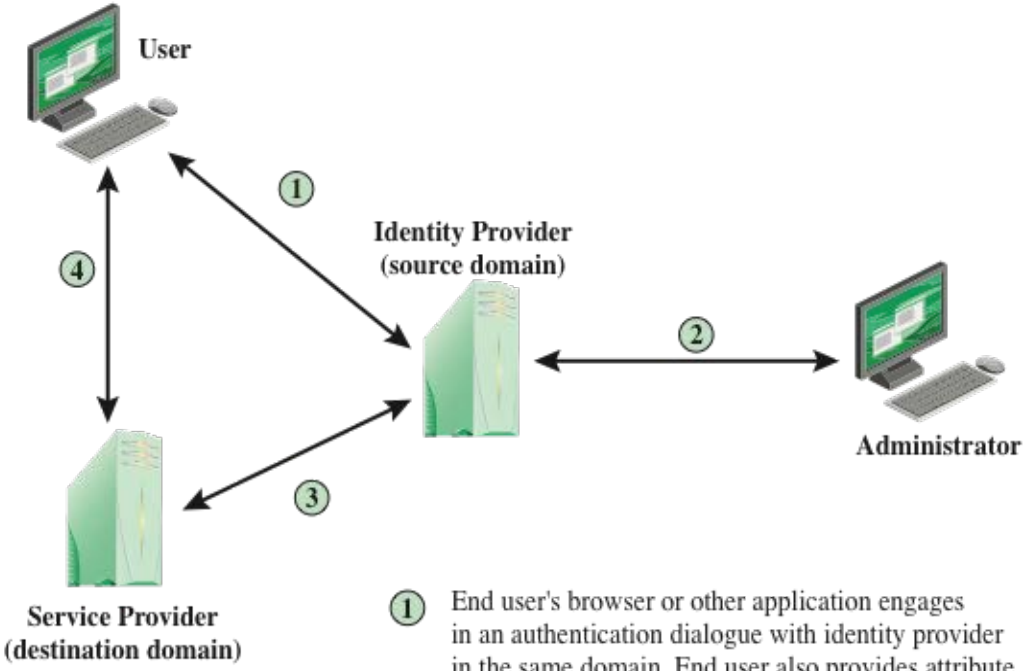


Figure 4.8 Generic Identity Management System

IDENTITY FEDERATION

- Identity federation is, in essence, an extension of identity management to multiple security domains
- Federated identity management refers to the agreements, standards, and technologies that enable the portability of identities, identity attributes, and entitlements across multiple enterprises and numerous applications
- Another key function of federated identity management is identity mapping
  - The federated identity management protocols map identities and attributes of a user in one domain to the requirements of another domain



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.

Figure 4.9 Federated Identity Operation

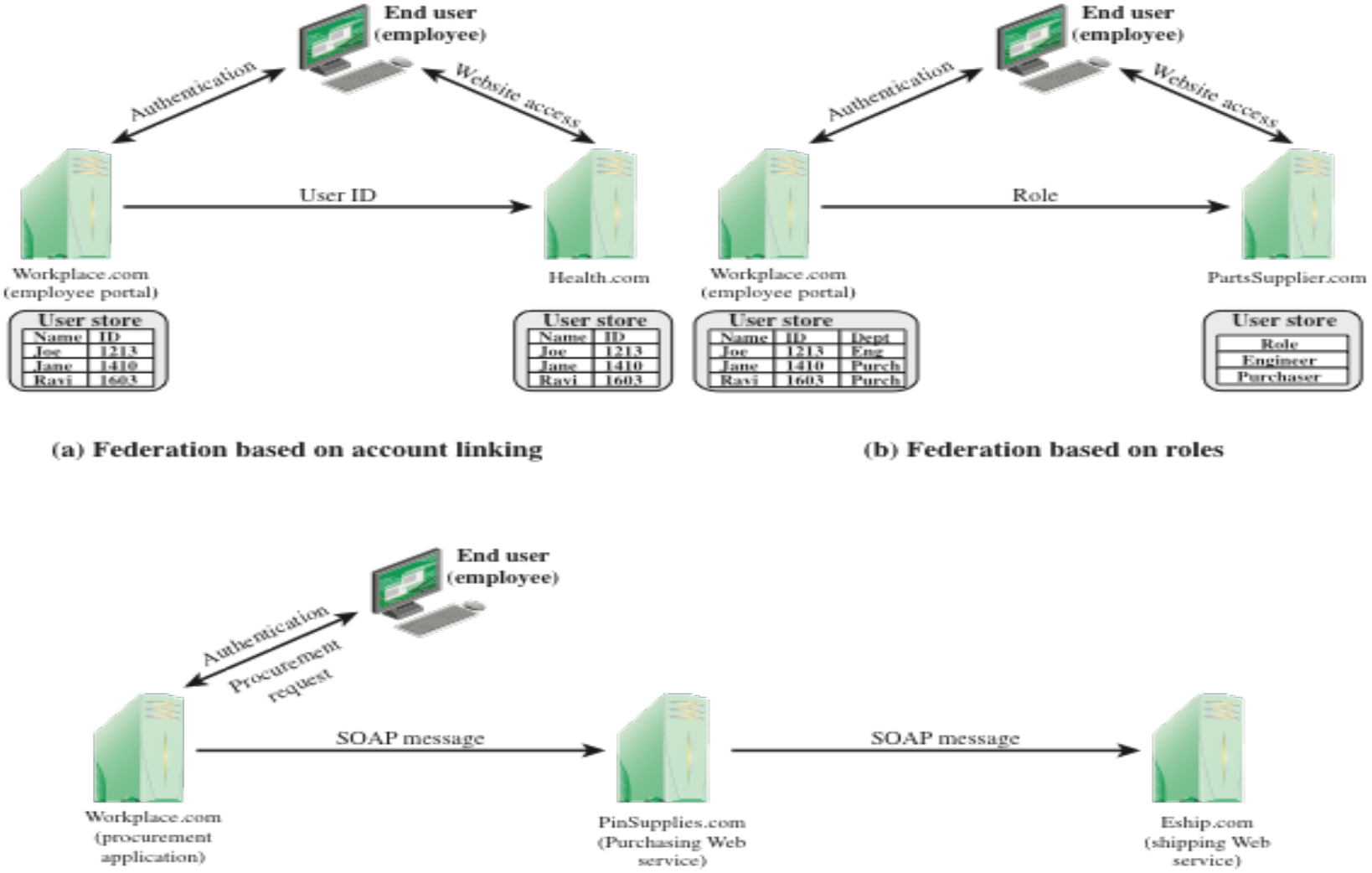


Figure 4.10 Federated Identity Scenarios

SUMMARY

- Remote user authentication principles
  - The NIST model for electronic user authentication
  - Means of authentication
- Symmetric key distribution using symmetric encryption
- Kerberos
- Key distribution using asymmetric encryption
  - Public-key certificates
  - Public-key distribution of secret keys
- X.509 certificates
  - Certificates
  - X.509 Version 3
- Federated identity management
  - Identity management
  - Identity federation