CMPE 132 Information Security

Key Management and Distributions

Dr. Younghee Park

SAN JOSÉ STATE UNIVERSITY

Symmetric Key Distribution

Given parties A and B, key distribution can be achieved in a number of ways:

- **A can select a key and physically deliver it to B**
- **A third party can select the key and physically deliver it to A and B**
- **If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key**
- **If A and B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to A and B**

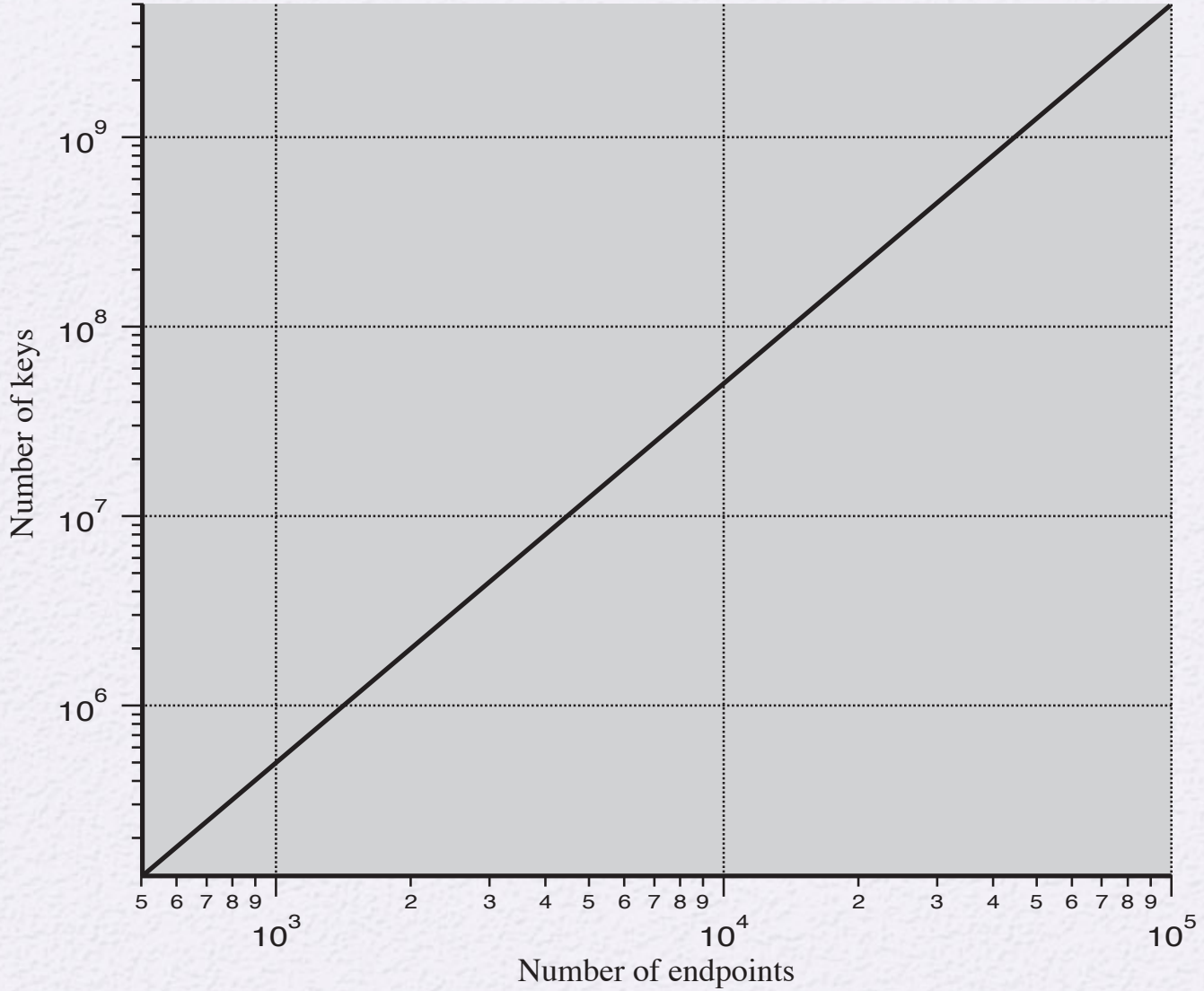


Figure 14.1 Number of Keys Required to Support Arbitrary Connections Between Endpoints

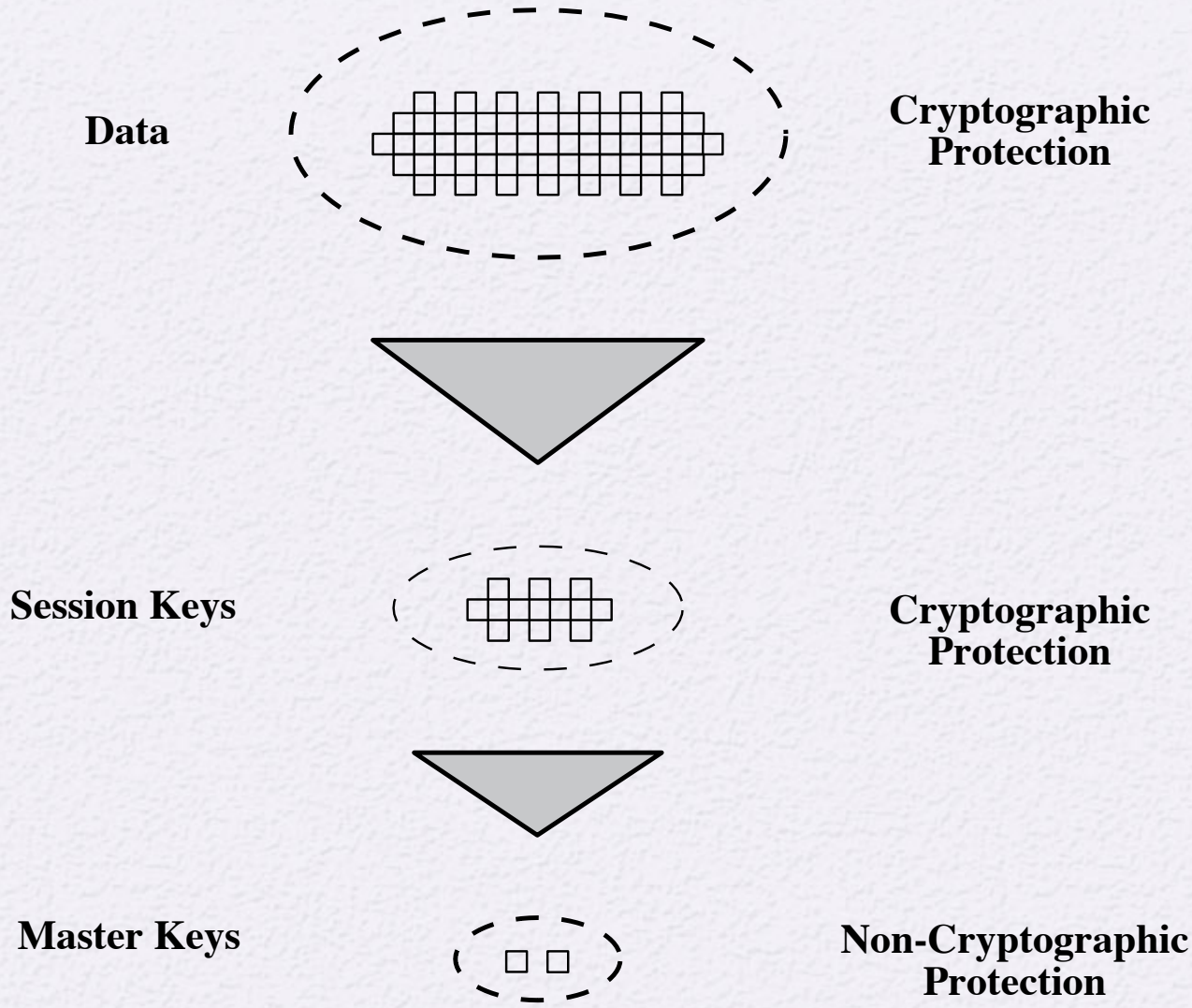


Figure 14.2 The Use of a Key Hierarchy

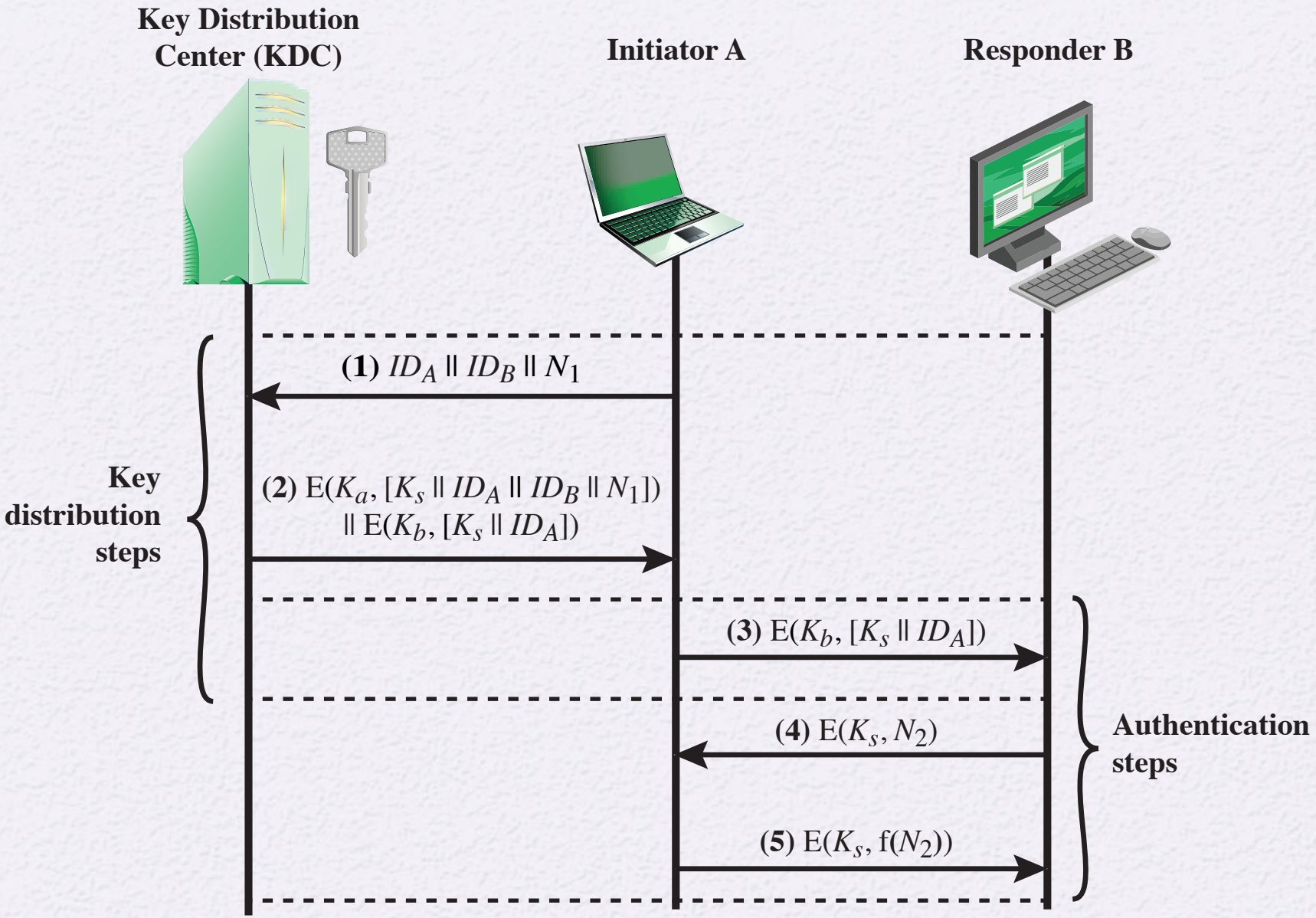


Figure 14.3 Key Distribution Scenario

Hierarchical Key Control

- For communication among entities within the same local domain, the local KDC is responsible for key distribution
  - If two entities in different domains desire a shared key, then the corresponding local KDC's can communicate through a global KDC
- The hierarchical concept can be extended to three or more layers
- Scheme minimizes the effort involved in master key distribution because most master keys are those shared by a local KDC with its local entities
  - Limits the range of a faulty or subverted KDC to its local area only

Session Key Lifetime

For connection-oriented protocols one choice is to use the same session key for the length of time that the connection is open, using a new session key for each new session

For a connectionless protocol there is no explicit connection initiation or termination, thus it is not obvious how often one needs to change the session key

A security manager must balance competing considerations:

- The more frequently session keys are exchanged, the more secure they are
- The distribution of session keys delays the start of any exchange and places a burden on network capacity

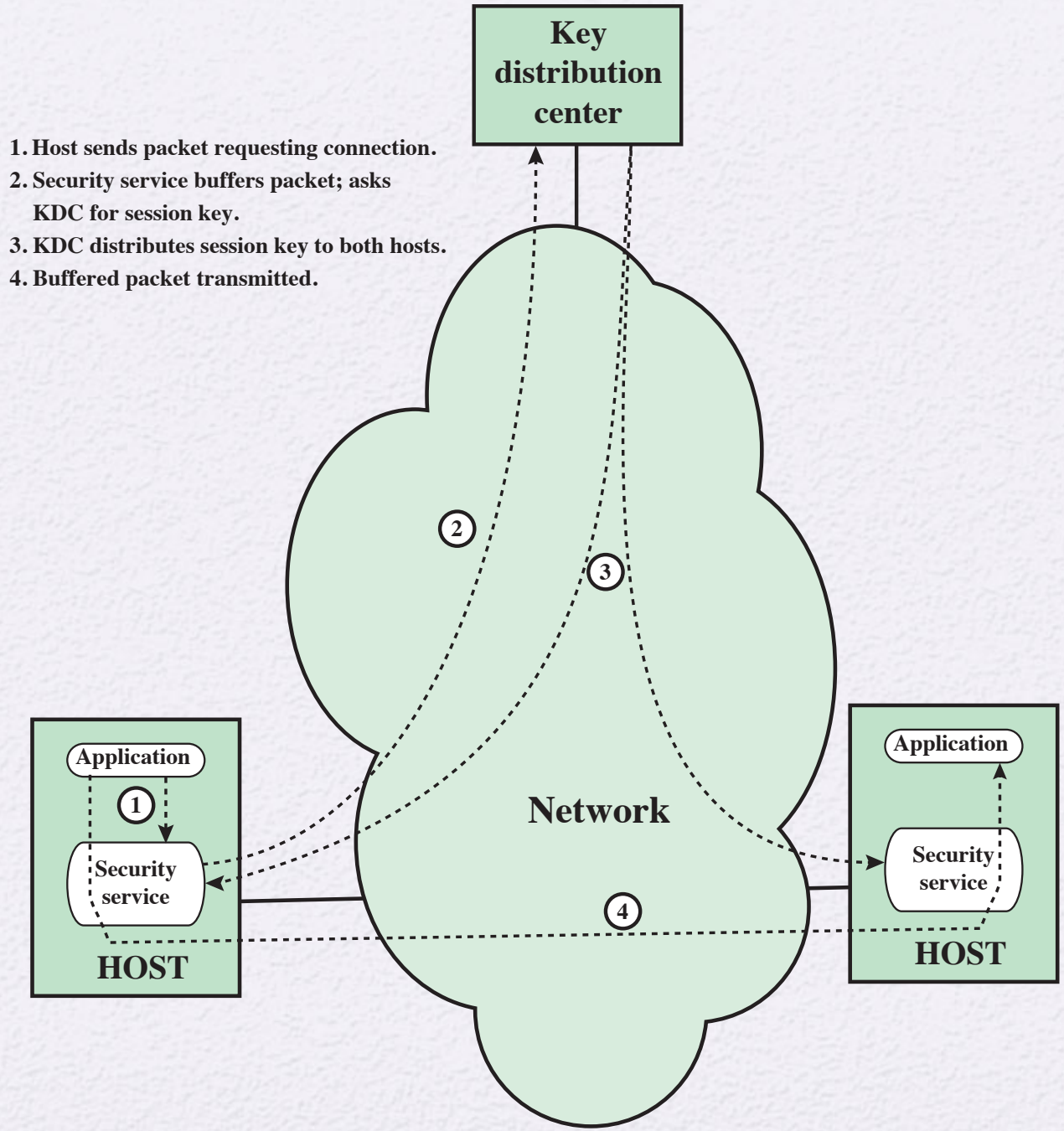
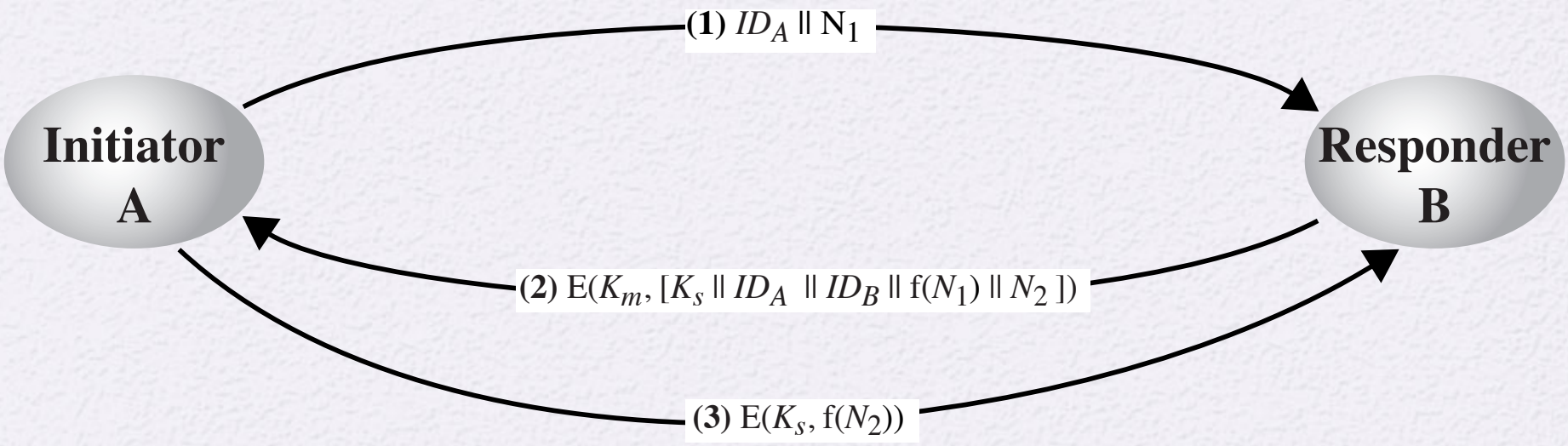


Figure 14.4 Automatic Key Distribution for Connection-Oriented Protocol



- At least n-1 master keys

Figure 14.5 Decentralized Key Distribution

Simple Secret Key Distribution

Based on Public Key

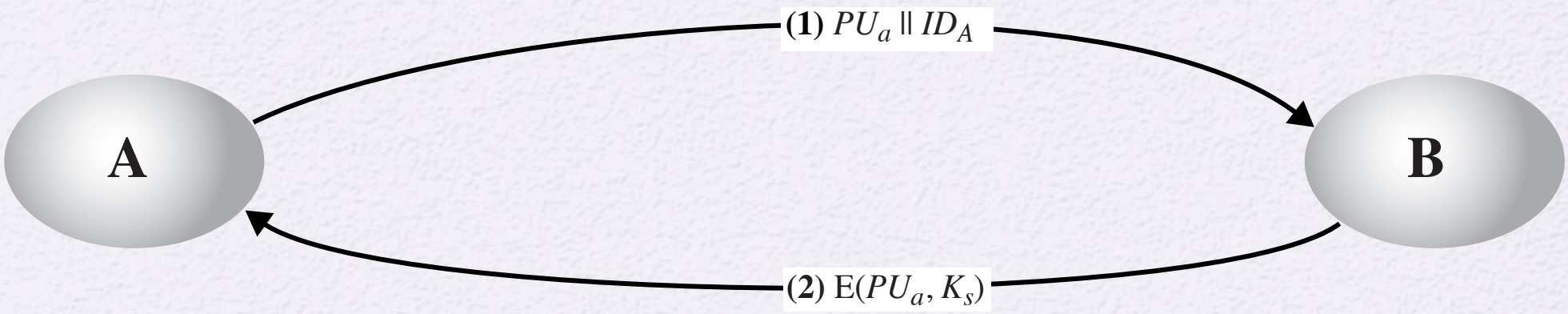


Figure 14.7 Simple Use of Public-Key Encryption to Establish a Session Key

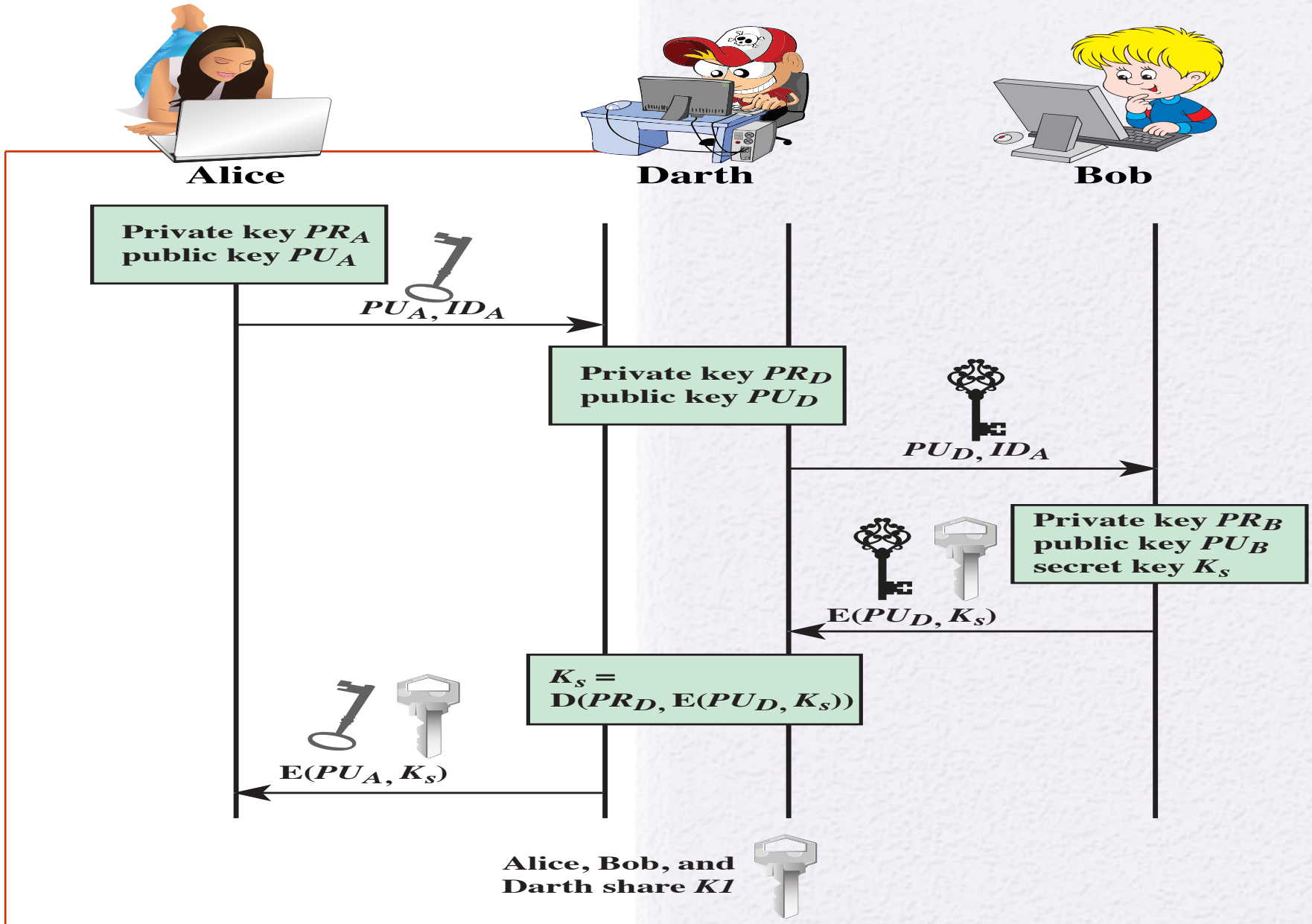


Figure 14.8 Another Man-in-the-Middle Attack

Secret Key Distribution with Confidentiality and Authentication

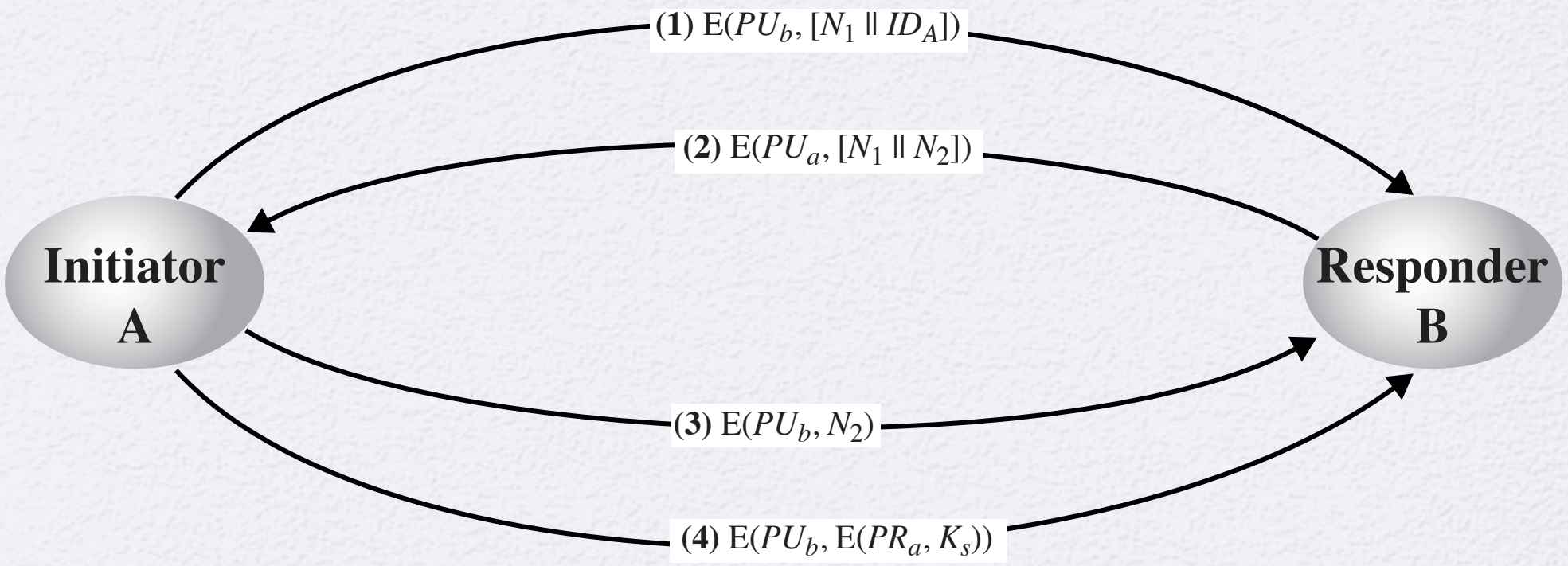


Figure 14.9 Public-Key Distribution of Secret Keys

A Hybrid Scheme

- In use on IBM mainframes
- Retains the use of a key distribution center (KDC) that shares a secret master key with each user and distributes secret session keys encrypted with the master key
- A public-key scheme is used to distribute the master keys

Rationale:

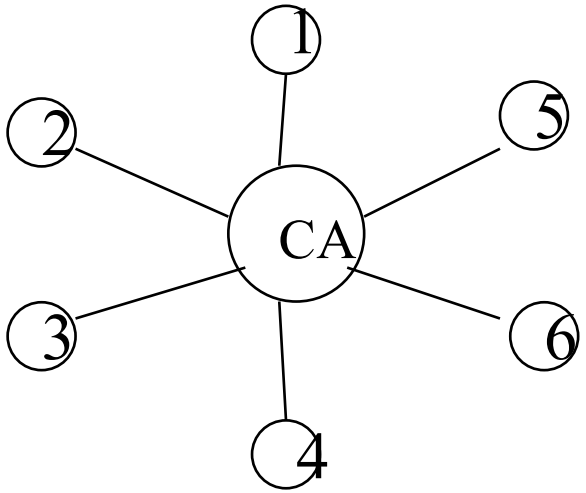
- Performance
- Backward compatibility

What Is PKI

- Informally, the infrastructure supporting the use of public key cryptography.
- A PKI consists of
  - Certificate Authority (CA)
  - Certificates
  - A repository for retrieving certificates
  - A method of revoking certificates
  - A method of evaluating a chain of certificates from known public keys to the target name

Certification Authorities (CA)

- A CA is a trusted node that maintains the public keys for all nodes (Each node maintains its own private key)



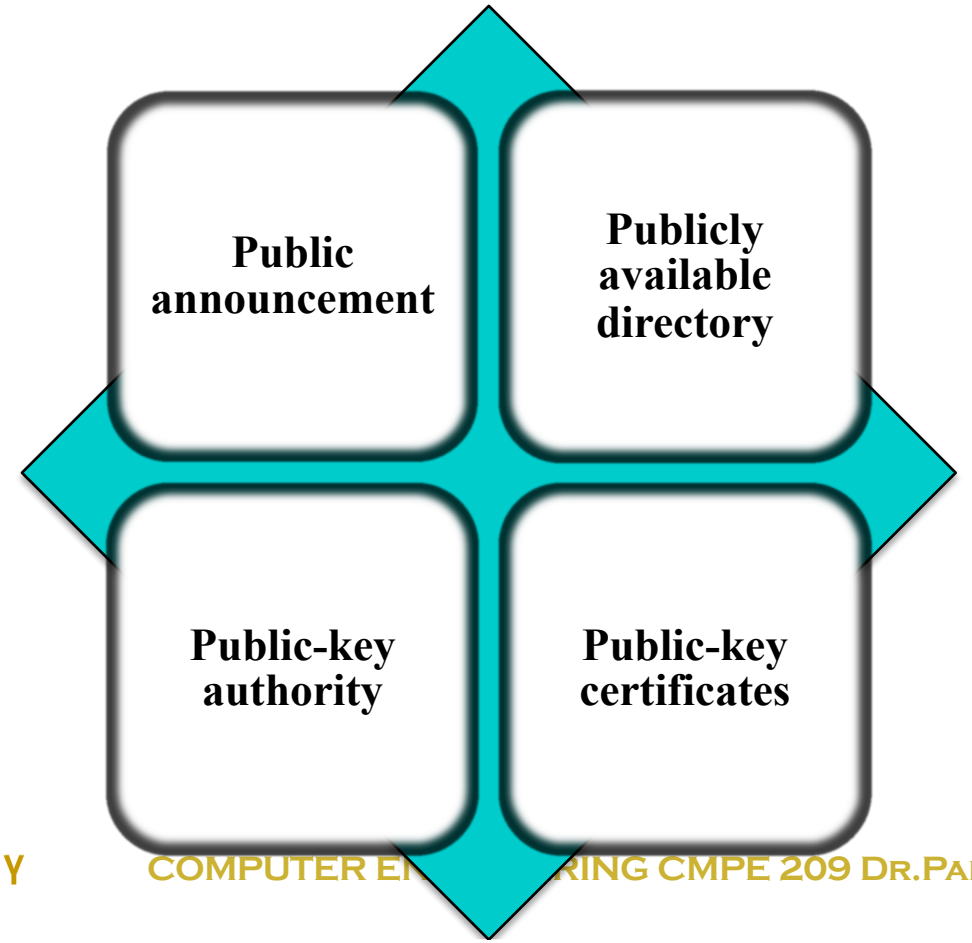
If a new node is inserted in the network, only that new node and the CA need to be configured with the public key for that node

Certificates

- Certificates can hold expiration date and time
- Alice keeps the same certificate as long as she has the same public key
- Alice can append the certificate to her messages so that others know for sure her public key

Distribution of Public Keys

- Several techniques have been proposed for the distribution of public keys. Virtually all these proposals can be grouped into the following general schemes:



Public Announcement

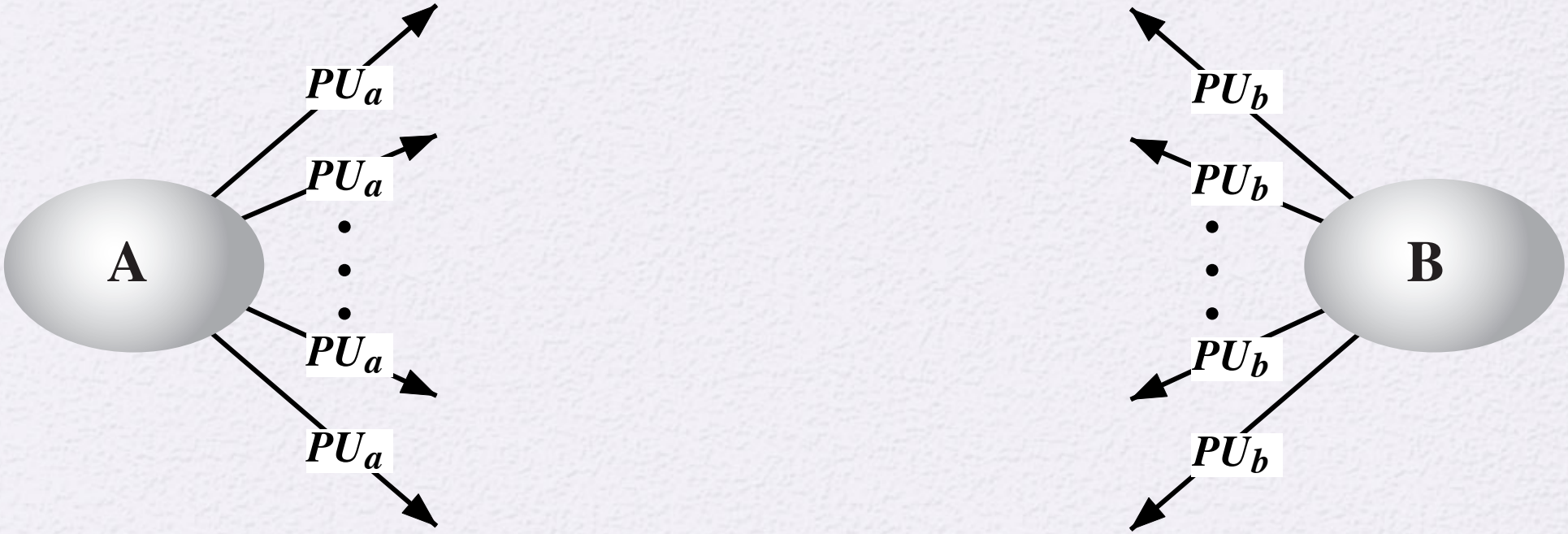


Figure 14.10 Uncontrolled Public Key Distribution

Publicly Available Directory

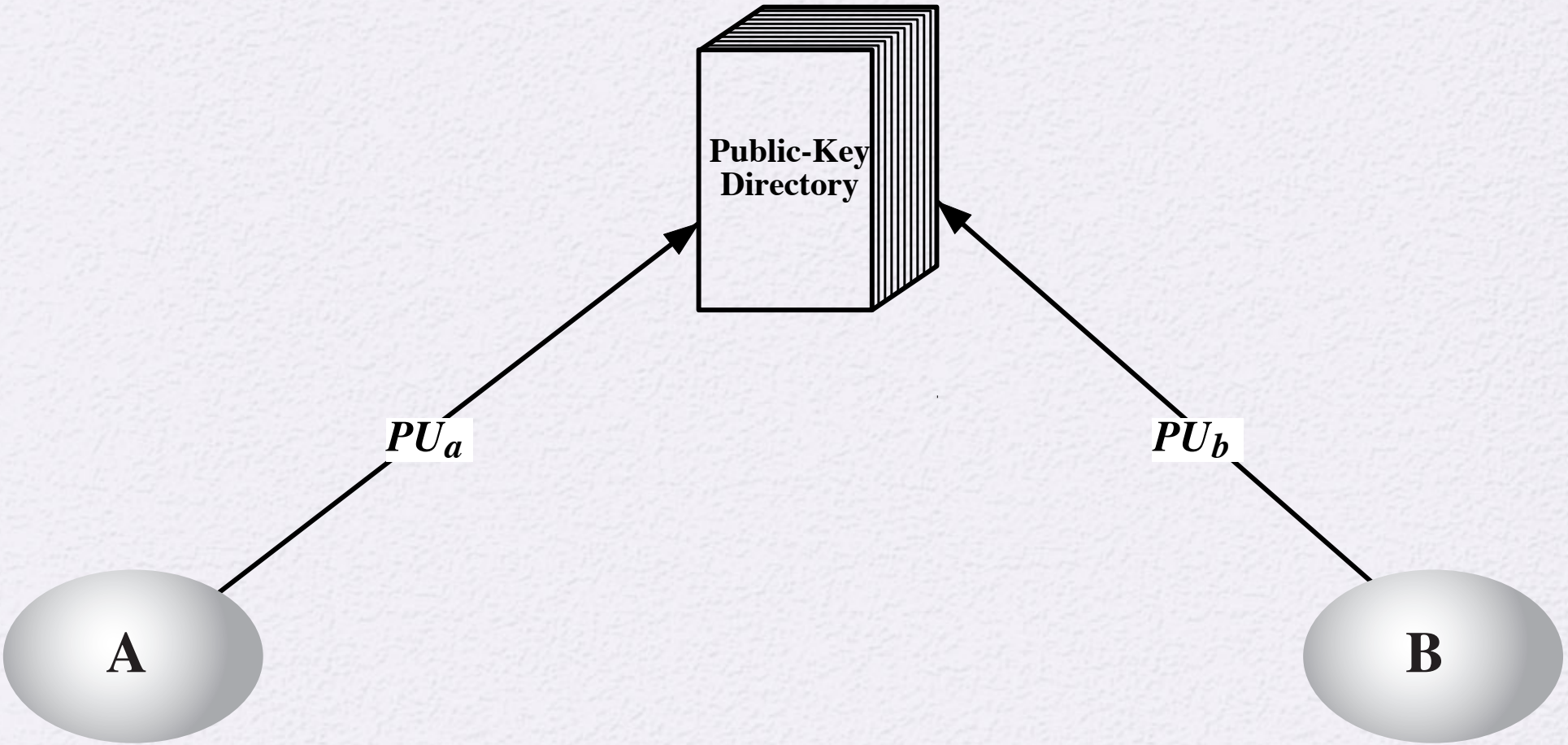


Figure 14.11 Public Key Publication

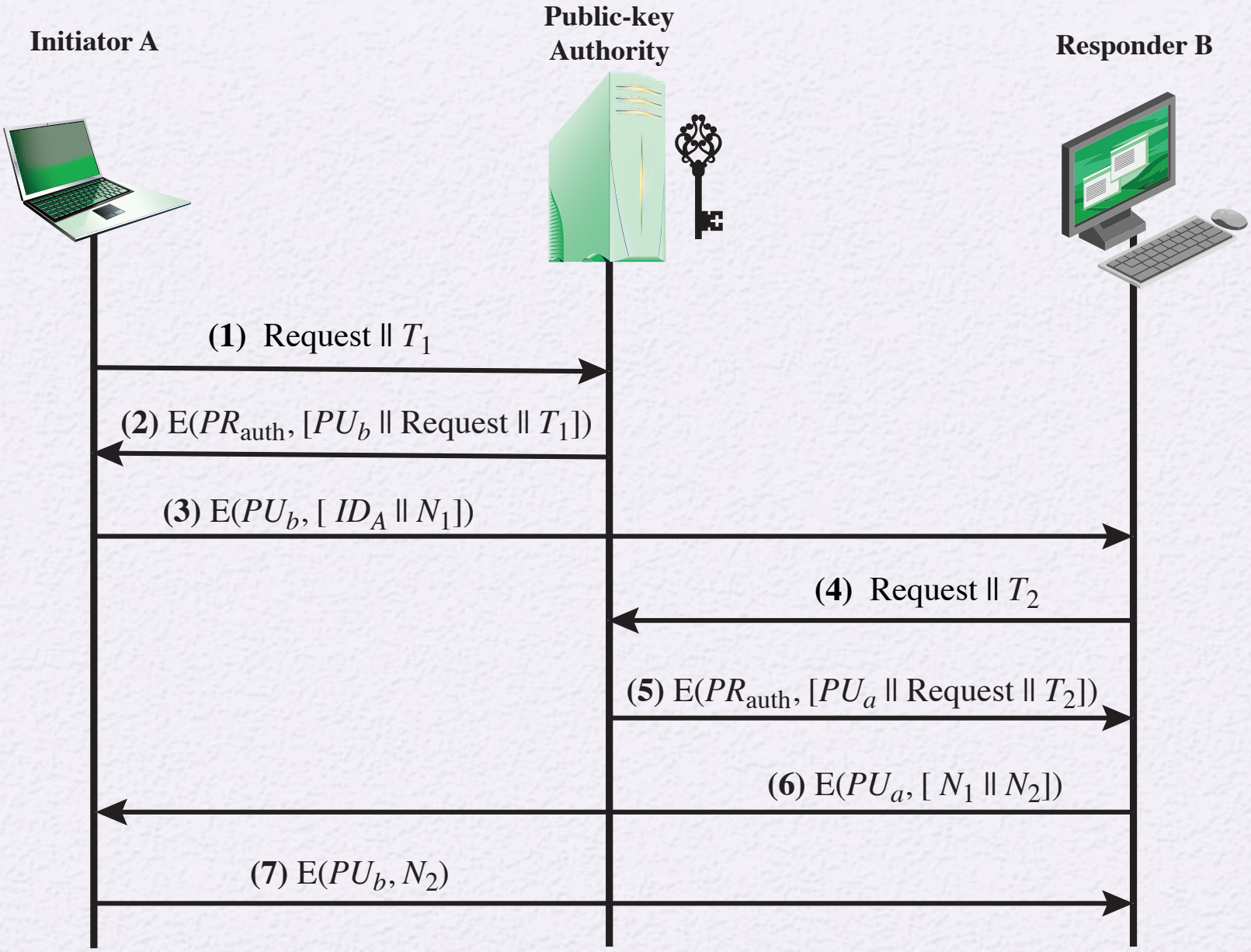
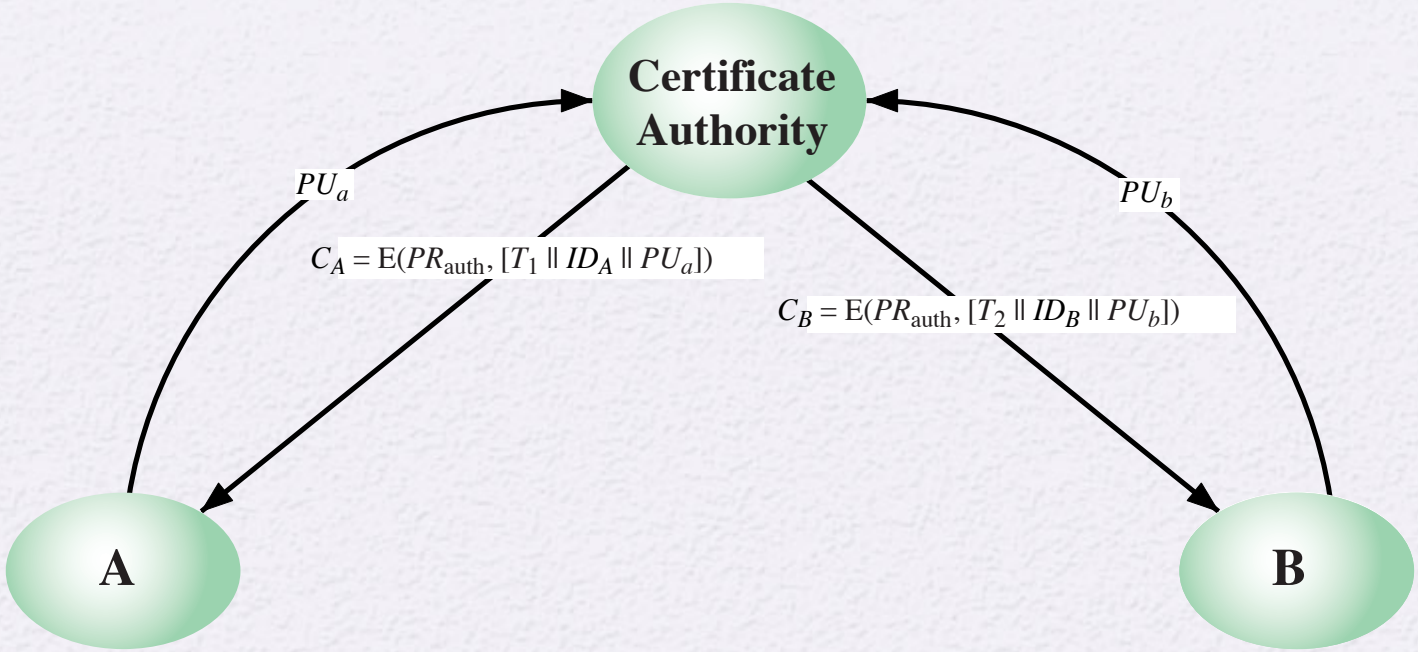
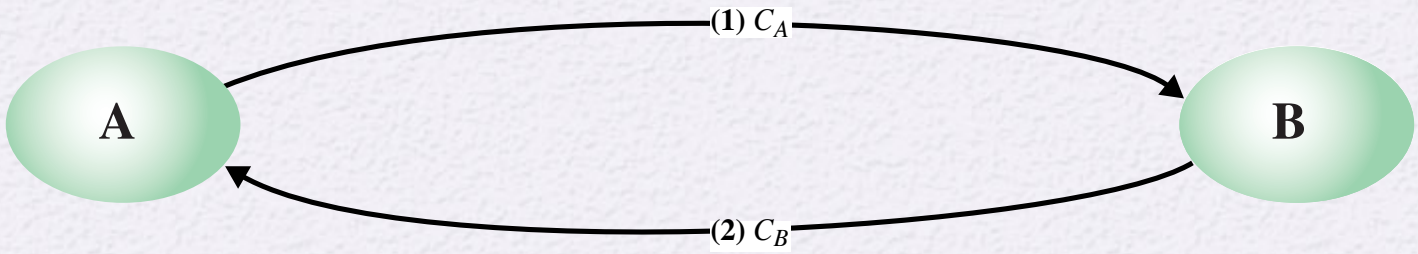


Figure 14.12 Public-Key Distribution Scenario



(a) Obtaining certificates from CA



(b) Exchanging certificates

Figure 14.13 Exchange of Public-Key Certificates

X.509 Certificates

- Part of the X.500 series of recommendations that define a directory service
  - The directory is, in effect, a server or distributed set of servers that maintains a database of information about users
- X.509 defines a framework for the provision of authentication services by the X.500 directory to its users
  - Was initially issued in 1988 with the latest revision in 2000
  - Based on the use of public-key cryptography and digital signatures
  - Does not dictate the use of a specific algorithm but recommends RSA
  - Does not dictate a specific hash algorithm
- Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority
- X.509 defines alternative authentication protocols based on the use of public-key certificates

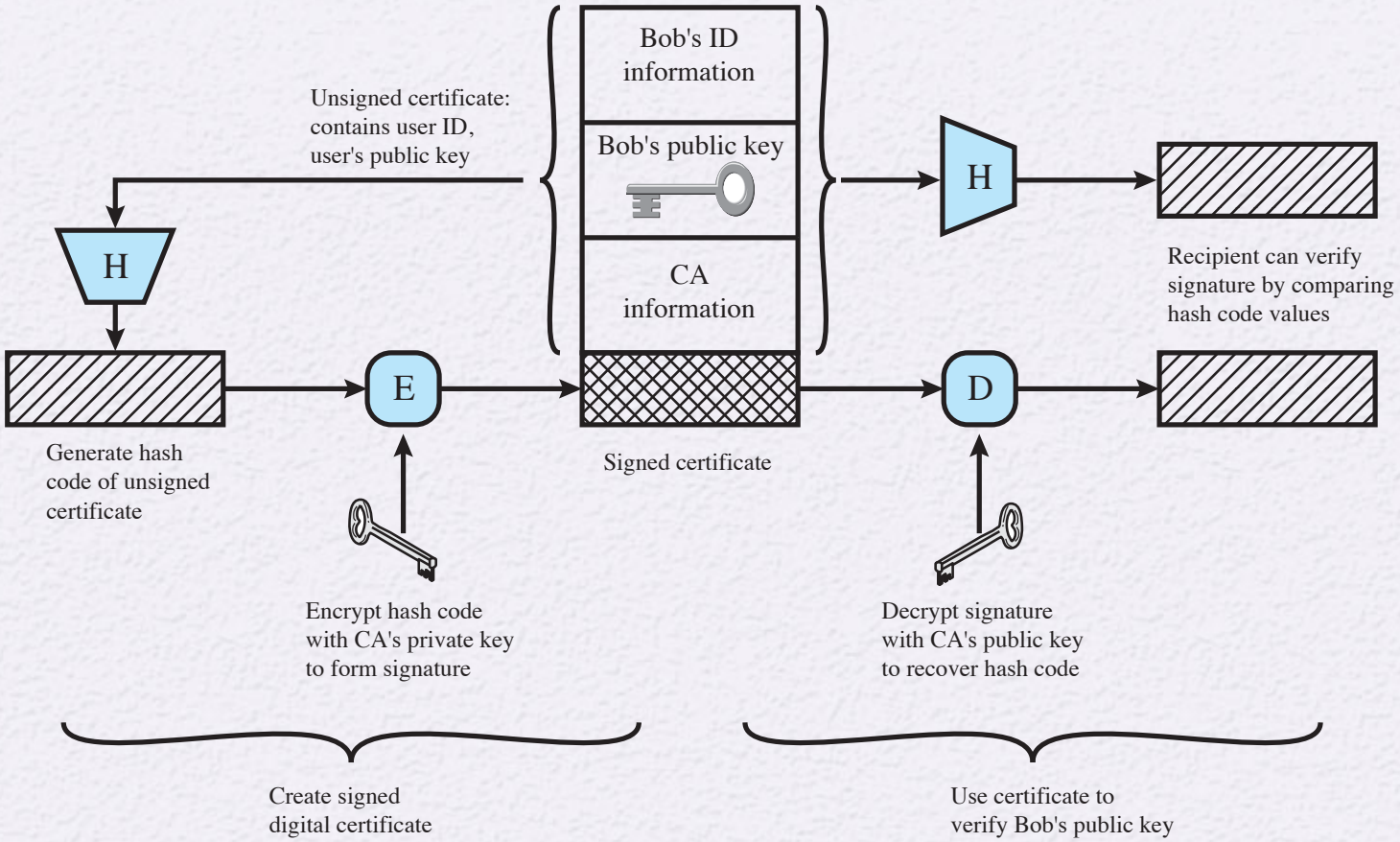
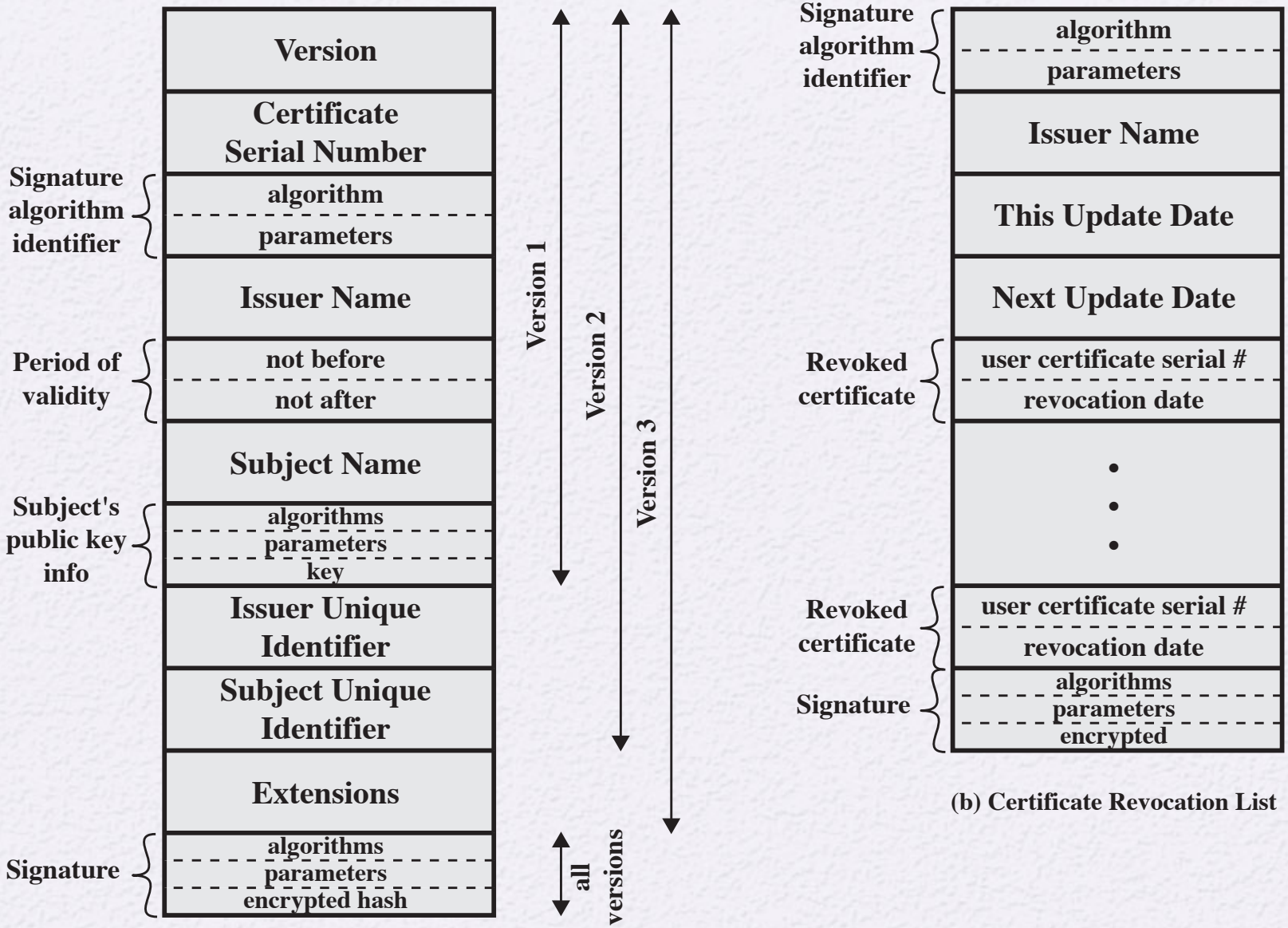


Figure 14.14 Public-Key Certificate Use

Certificates

Created by a trusted Certification Authority (CA) and have the following elements:

- Version
- Serial number
- Signature algorithm identifier
- Issuer name
- Period of validity
- Subject name
- Subject's public-key information
- Issuer unique identifier
- Subject unique identifier
- Extensions
- Signature



(a) X.509 Certificate

(b) Certificate Revocation List

Figure 14.15 X.509 Formats

Obtaining a Certificate

User certificates generated by a CA have the following characteristics:

- Any user with access to the public key of the CA can verify the user public key that was certified
- No party other than the certification authority can modify the certificate without this being detected

- Because certificates are unforgeable, they can be placed in a directory without the need for the directory to make special efforts to protect them
  - In addition, a user can transmit his or her certificate directly to other users
- Once B is in possession of A's certificate, B has confidence that messages it encrypts with A's public key will be secure from eavesdropping and that messages signed with A's private key are unforgeable

Certificate Revocation

- 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
  - The CA's certificate is assumed to be compromised
- Each CA must maintain a list consisting of all revoked but not expired certificates issued by that CA
  - These lists should be posted on the directory

Summary

- Symmetric key distribution using symmetric encryption
  - Key distribution scenario
  - Hierarchical key control
  - Session key lifetime
  - Decentralized key control
- Symmetric key distribution using asymmetric encryption
  - Simple secret key distribution
  - Secret key distribution with confidentiality and authentication
  - Hybrid scheme

- Distribution of public keys
  - Public announcement of public keys
  - Publicly available directory
  - Public-key authority
  - Public-key certificates
- X.509 Certificates
- Public-key infrastructure

Remote User-Authentication Using Kerberos

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

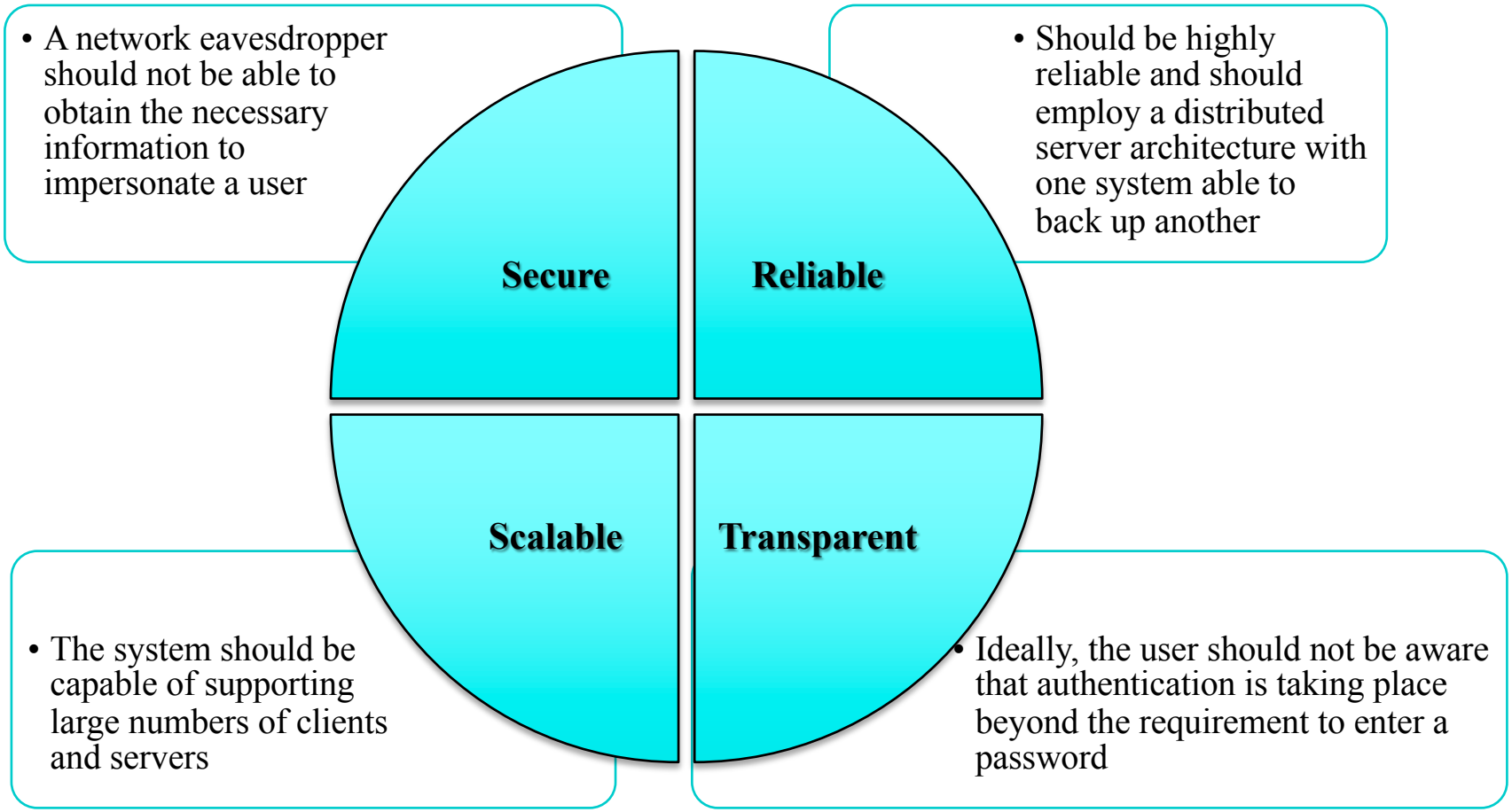
- Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users
  - Relies exclusively on symmetric encryption, making no use of public-key encryption

Authentication Service Provided by Kerberos

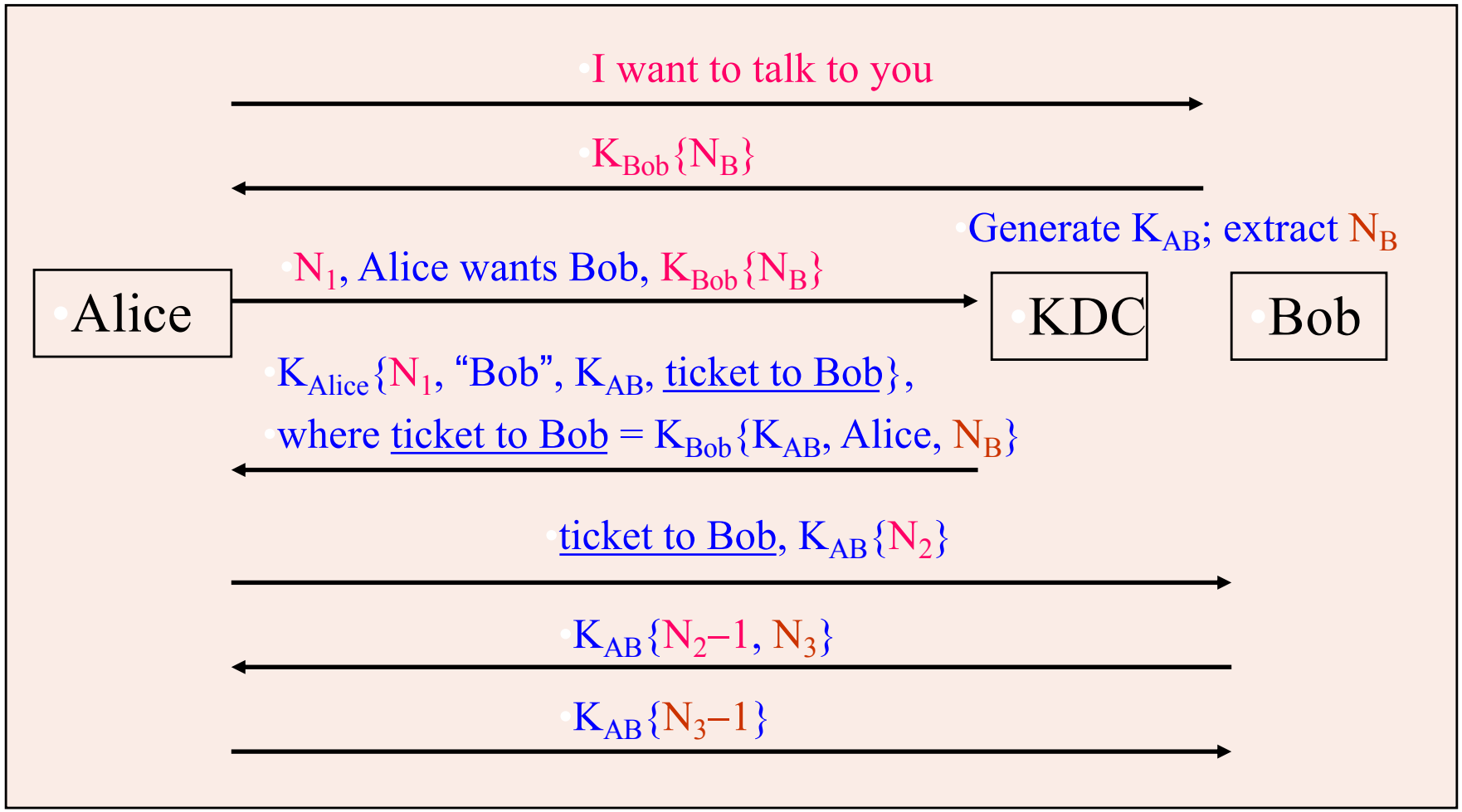
- A centralized authentication service
  - Authenticate users to services
  - Authenticate services to users
  - Servers are relieved of the burden of maintaining authentication information.
- Facts about Kerberos
  - Rely exclusively on conventional encryption.
    - Public key based Kerberos has been considered.
  - Stateless: Kerberos server does not need to maintain the state information about any entities being authenticated.

Kerberos Requirements

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



(Review) Expanded Needham-Schroeder Protocol



- The additional two messages assure Bob that the initiator has talked to KDC since Bob generates N_B

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

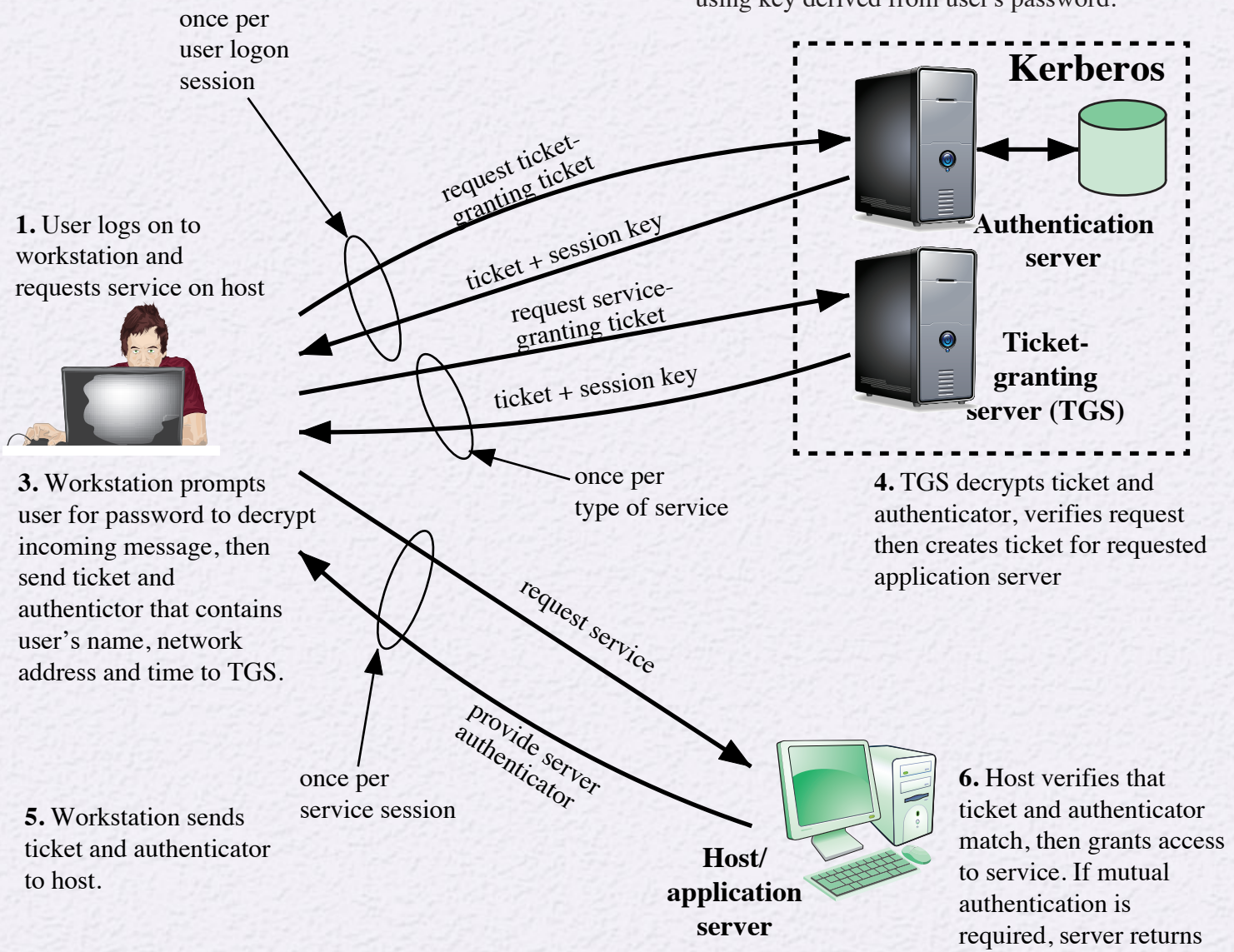


Figure 15.1 Overview of Kerberos

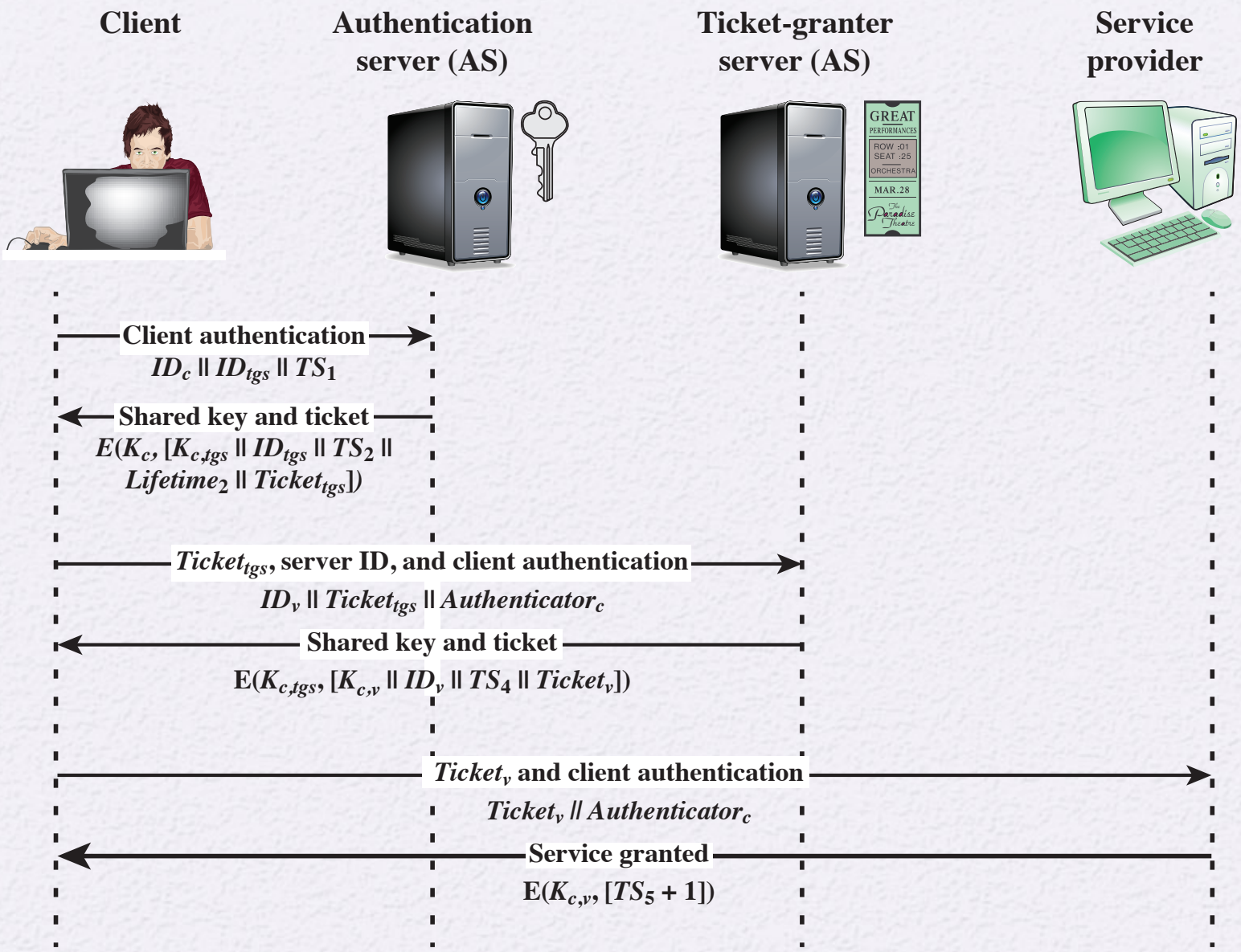


Figure 15.2 Kerberos Exchanges

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 467 – 468 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.

Table 15.1 (page 464 in textbook)

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

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
- Inter-realm authentication
  - 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.

Replicated Kerberos

- Multiple replica of Kerberos - availability and performance
- Keeping Kerberos databases consistent
  - Single master Kerberos as the point of direct update to principals' database entries
  - Updated database is downloaded from the master to all replica Kerberos
  - Periodic download or on-demand

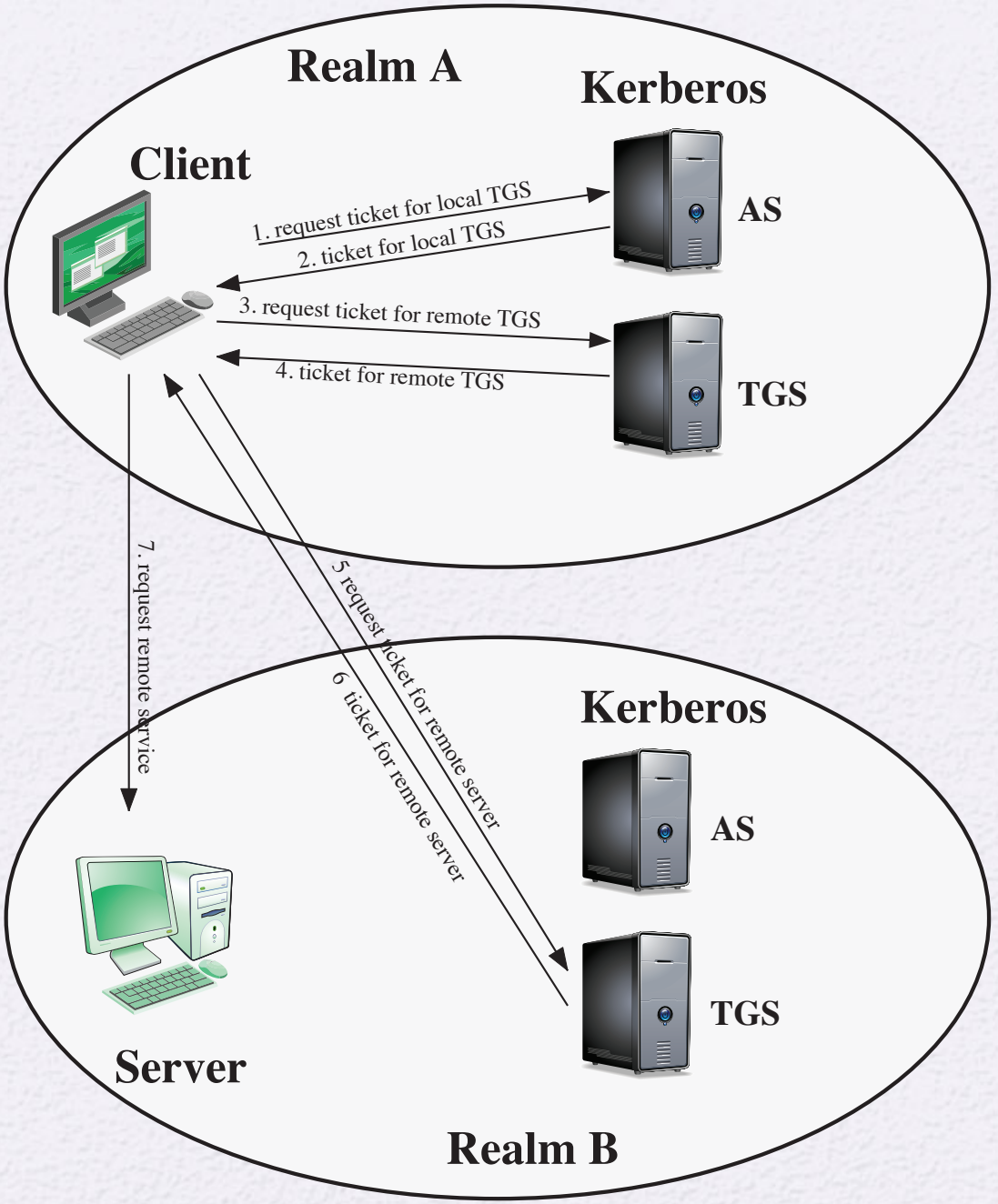


Figure 15.3 Request for Service in Another Realm