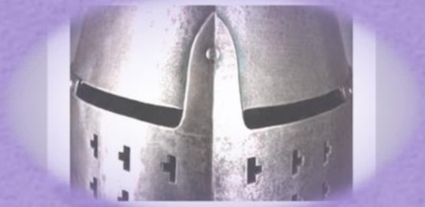


Cryptography and Network Security

Sixth Edition

by William Stallings

Chapter 14

Key Management and Distribution

“No Singhalese, whether man or woman, would venture out of the house without a bunch of keys in his hand, for without such a talisman he would fear that some devil might take advantage of his weak state to slip into his body.”

—The Golden Bough,

Sir James George Frazer

“John wrote the letters of the alphabet under the letters in its first lines and tried it against the message. Immediately he knew that once more he had broken the code. It was extraordinary the feeling of triumph he had. He felt on top of the world. For not only had he done it, had he broken the July code, but he now had the key to every future coded message, since instructions as to the source of the next one must of necessity appear in the current one at the end of each month.”

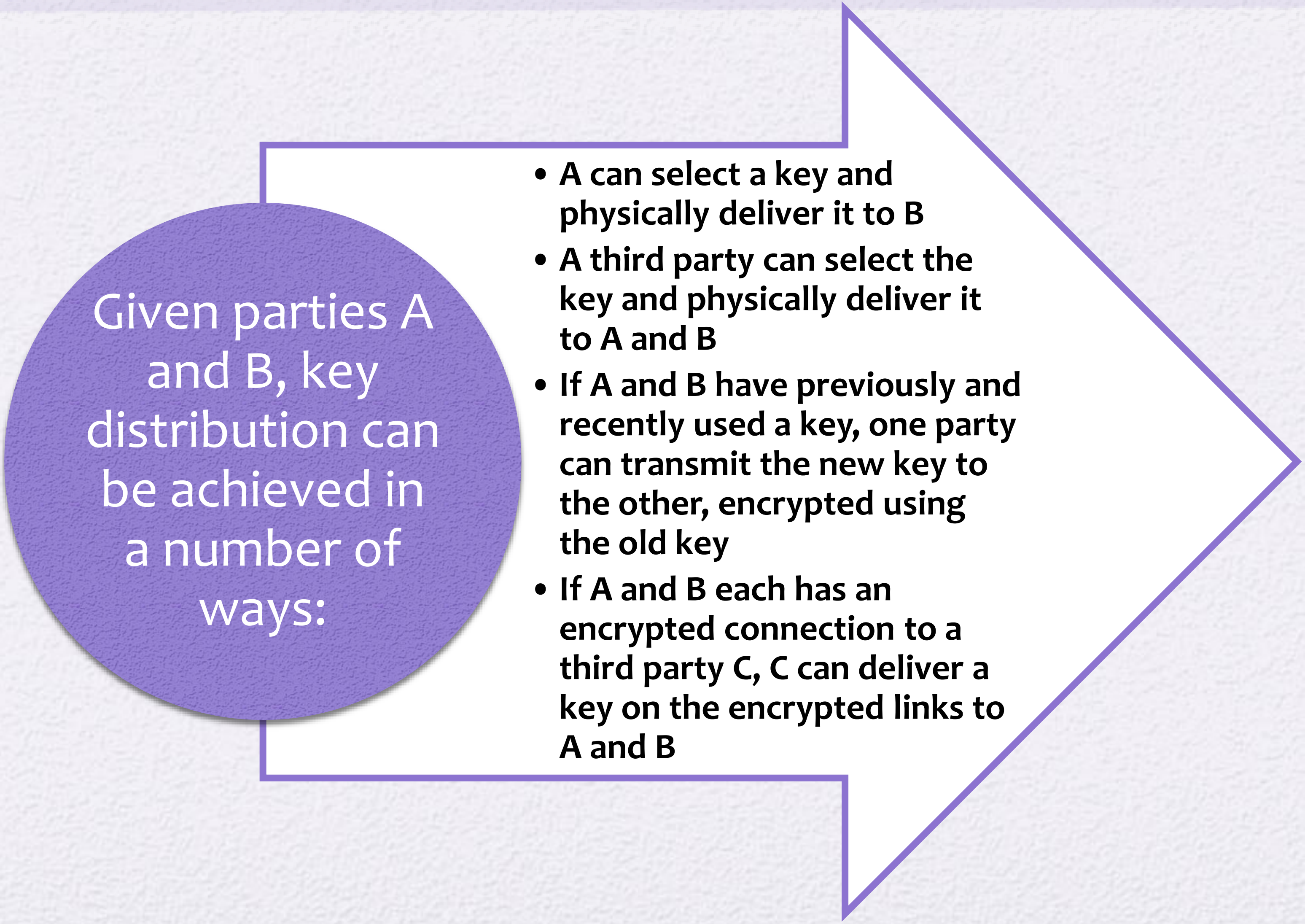
—Talking to Strange Men,

Ruth Rendall

Key Distribution Technique

- Term that refers to the means of delivering a key to two parties who wish to exchange data without allowing others to see the key
- For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others
- Frequent key changes are desirable to limit the amount of data compromised if an attacker learns the key

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

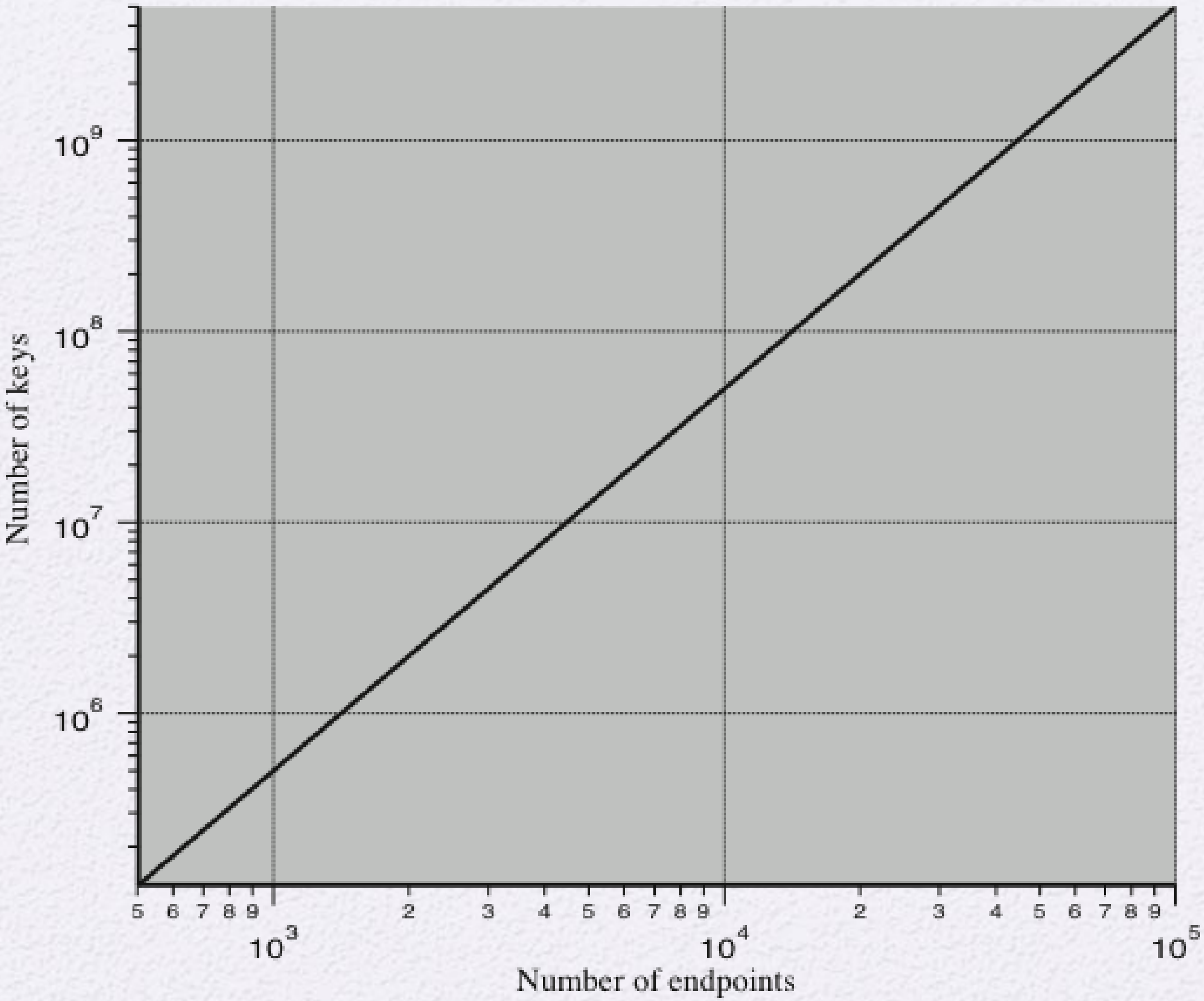
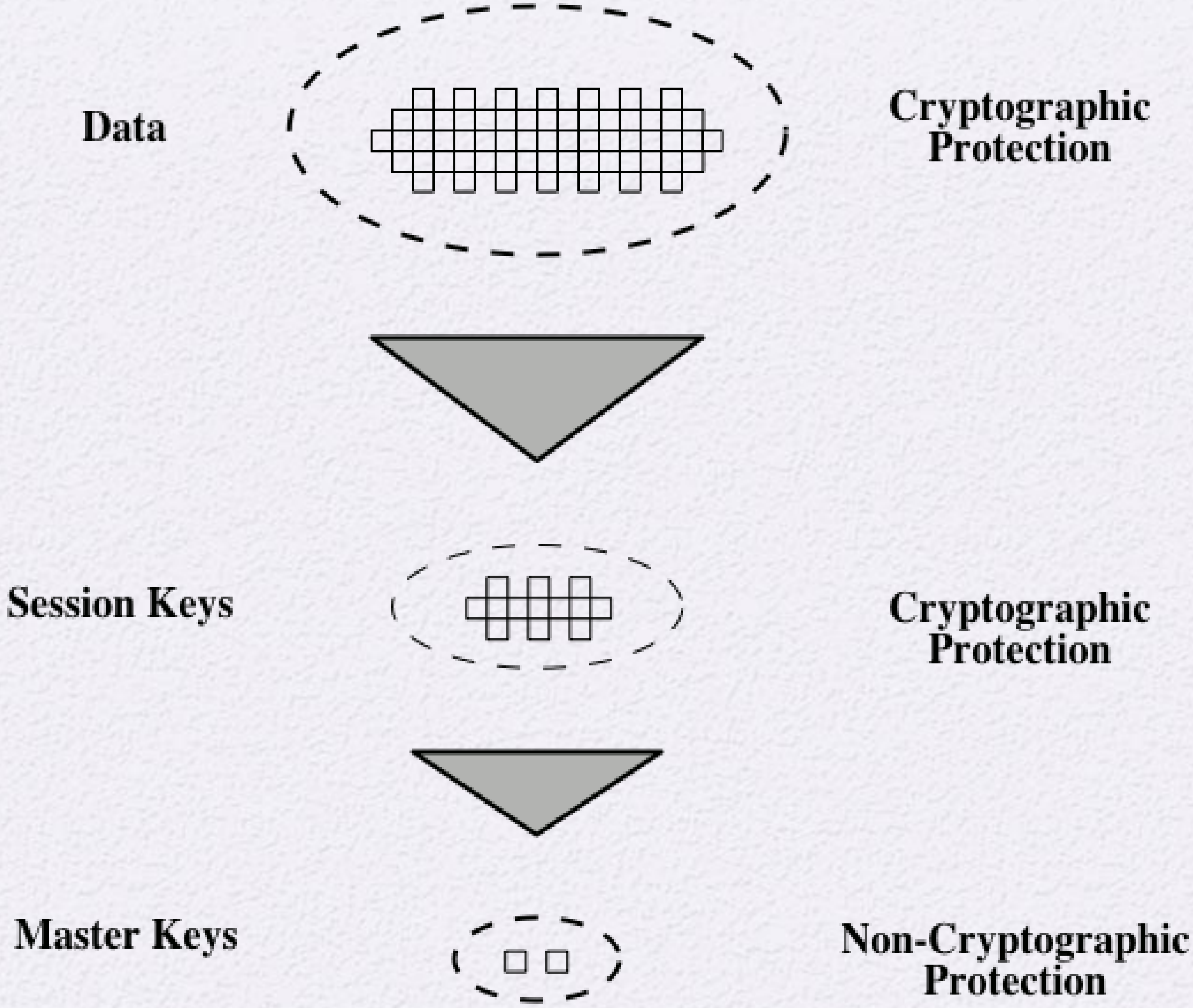


Figure 14.1 Number of Keys Required to Support Arbitrary Connections Between Endpoints



Data

Cryptographic Protection

Session Keys

Cryptographic Protection

Master Keys

Non-Cryptographic Protection

Figure 14.2 The Use of a Key Hierarchy

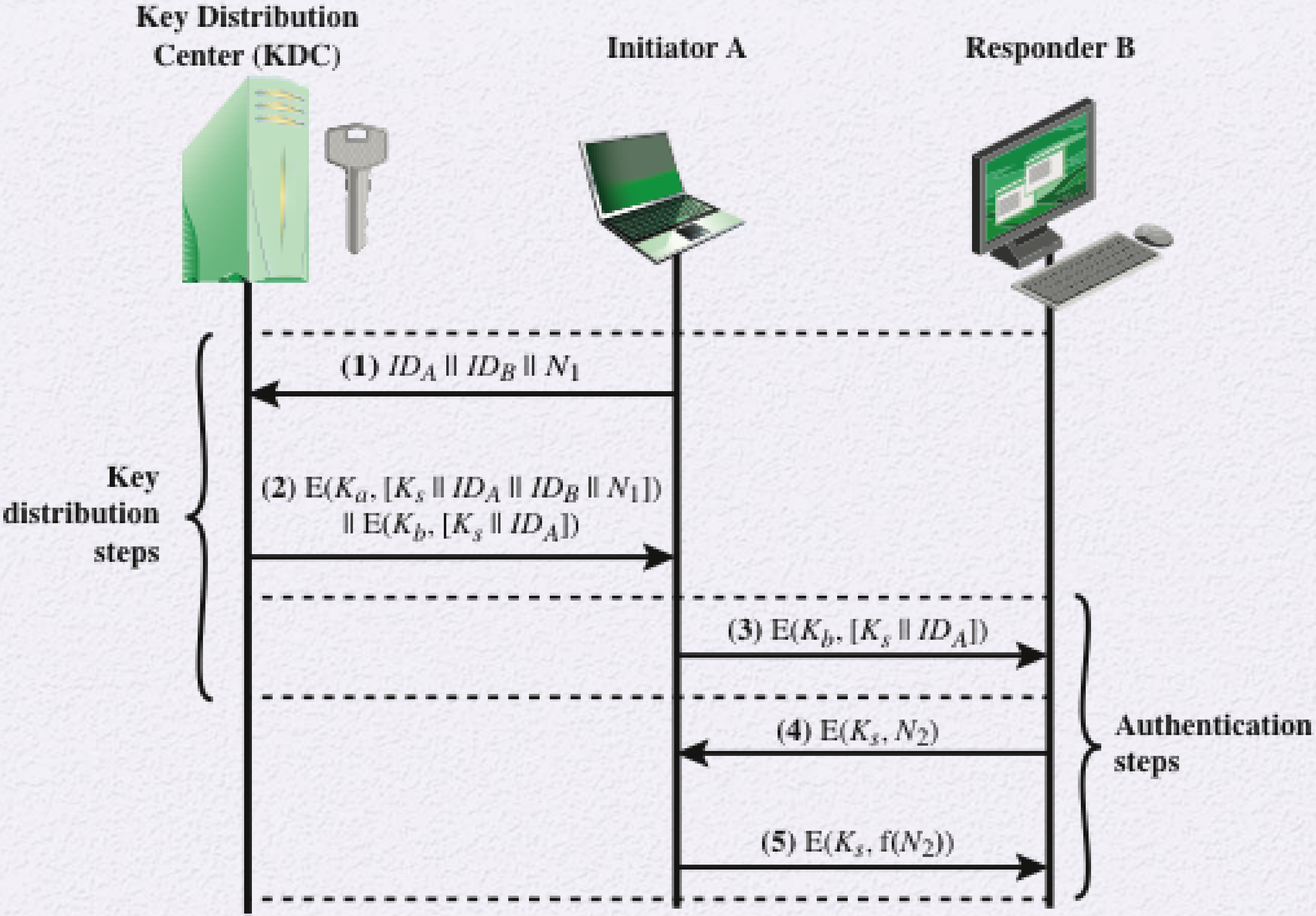
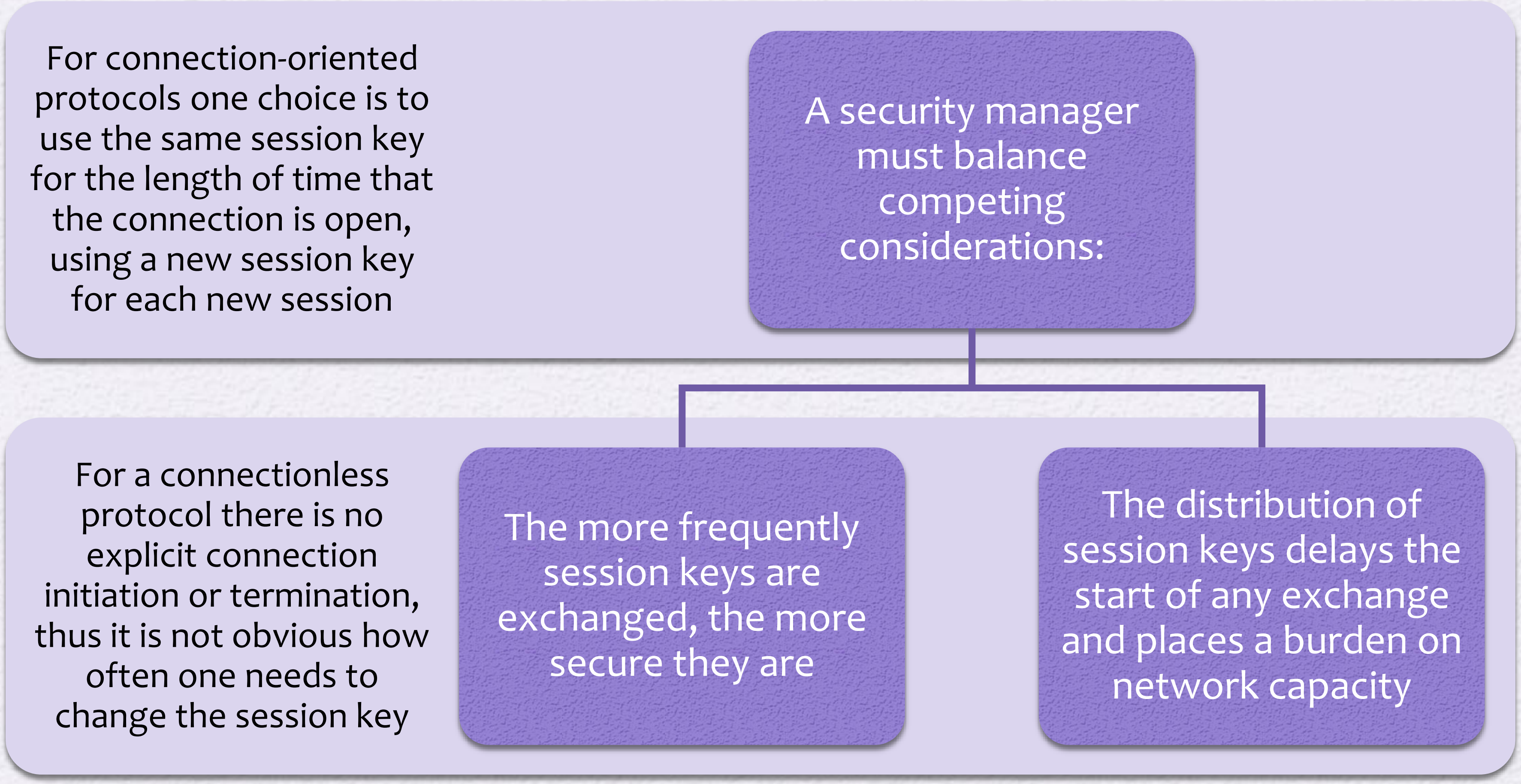


Figure 14.3 Key Distribution Scenario

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



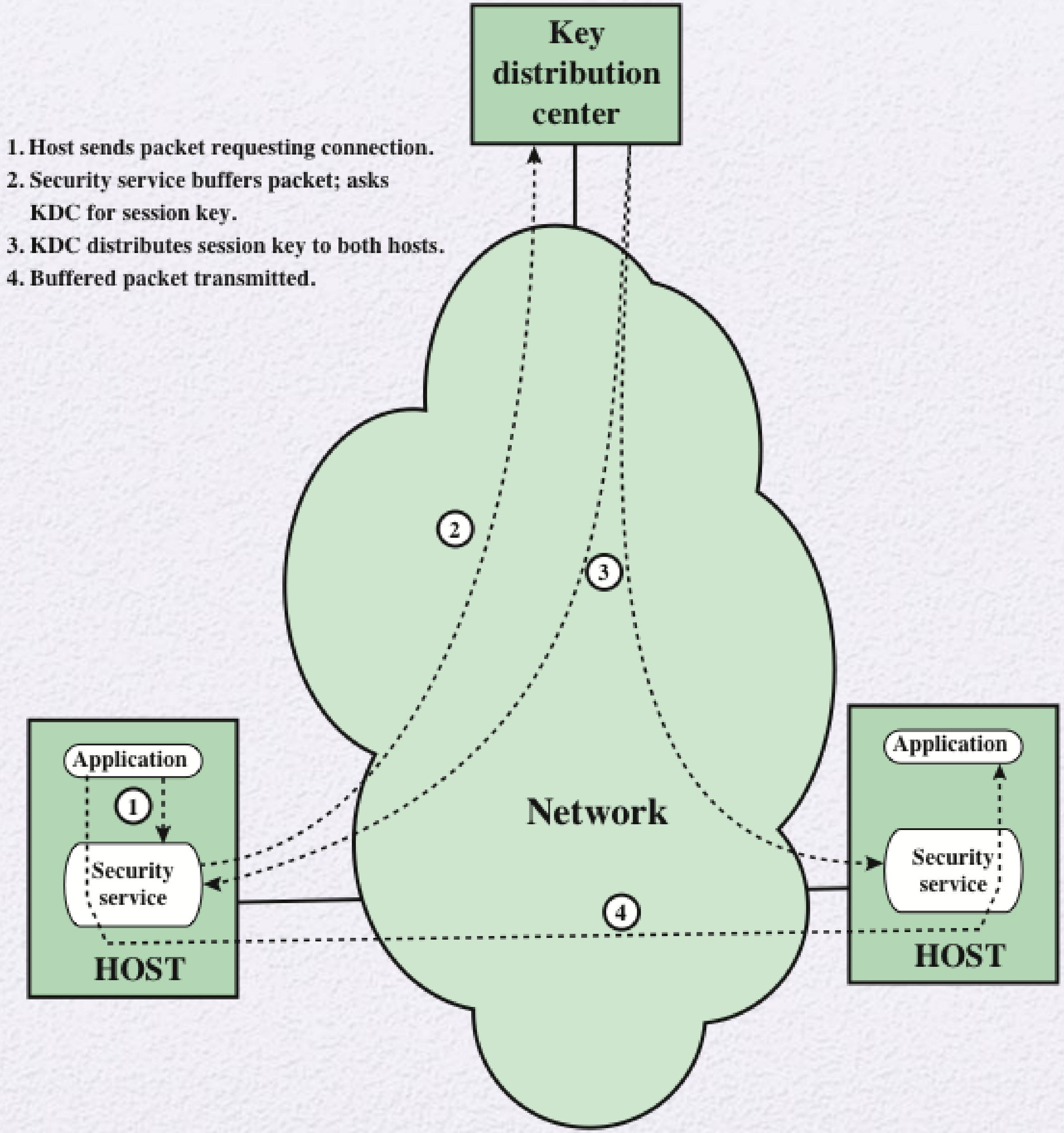
For connection-oriented protocols one choice is to use the same session key for the length of time that the connection is open, using a new session key for each new session

A security manager must balance competing considerations:

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

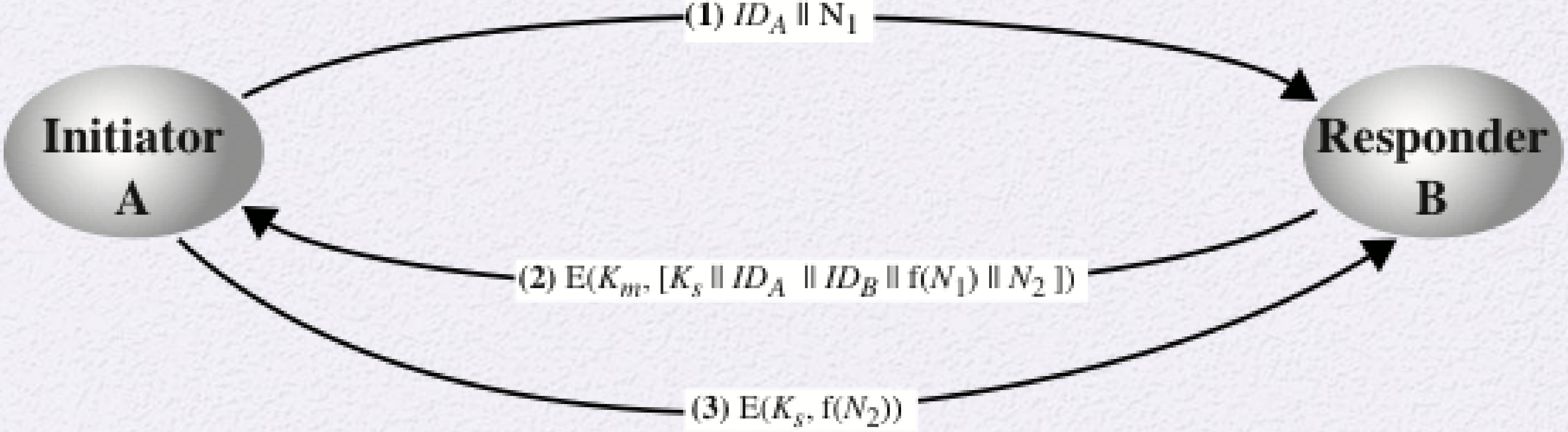
The more frequently session keys are exchanged, the more secure they are

The distribution of session keys delays the start of any exchange and places a burden on network capacity



1. Host sends packet requesting connection.
2. Security service buffers packet; asks KDC for session key.
3. KDC distributes session key to both hosts.
4. Buffered packet transmitted.

Figure 14.4 Automatic Key Distribution for Connection-Oriented Protocol



Initiator A
Responder B
(1) $ID_A \parallel N_1$
(2) $E(K_m, [K_s \parallel ID_A \parallel ID_B \parallel f(N_1) \parallel N_2])$
(3) $E(K_s, f(N_2))$

Figure 14.5 Decentralized Key Distribution

- A issues a request to B for a session key and includes a nonce, N_1.

- 2. B responds with a message that is encrypted using the shared master key. The response includes the session key selected by B, an identifier of B, the value $f(N_1)$, and another nonce, N_2.

- 3. Using the new session key, A returns $f(N_2)$ to B.

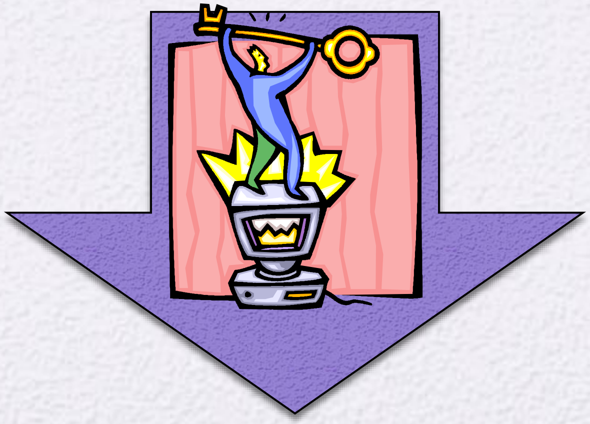
Controlling Key Usage

- The concept of a key hierarchy and the use of automated key distribution techniques greatly reduce the number of keys that must be manually managed and distributed

- It also may be desirable to impose some control on the way in which automatically distributed keys are used
  - For example, in addition to separating master keys from session keys, we may wish to define different types of session keys on the basis of use

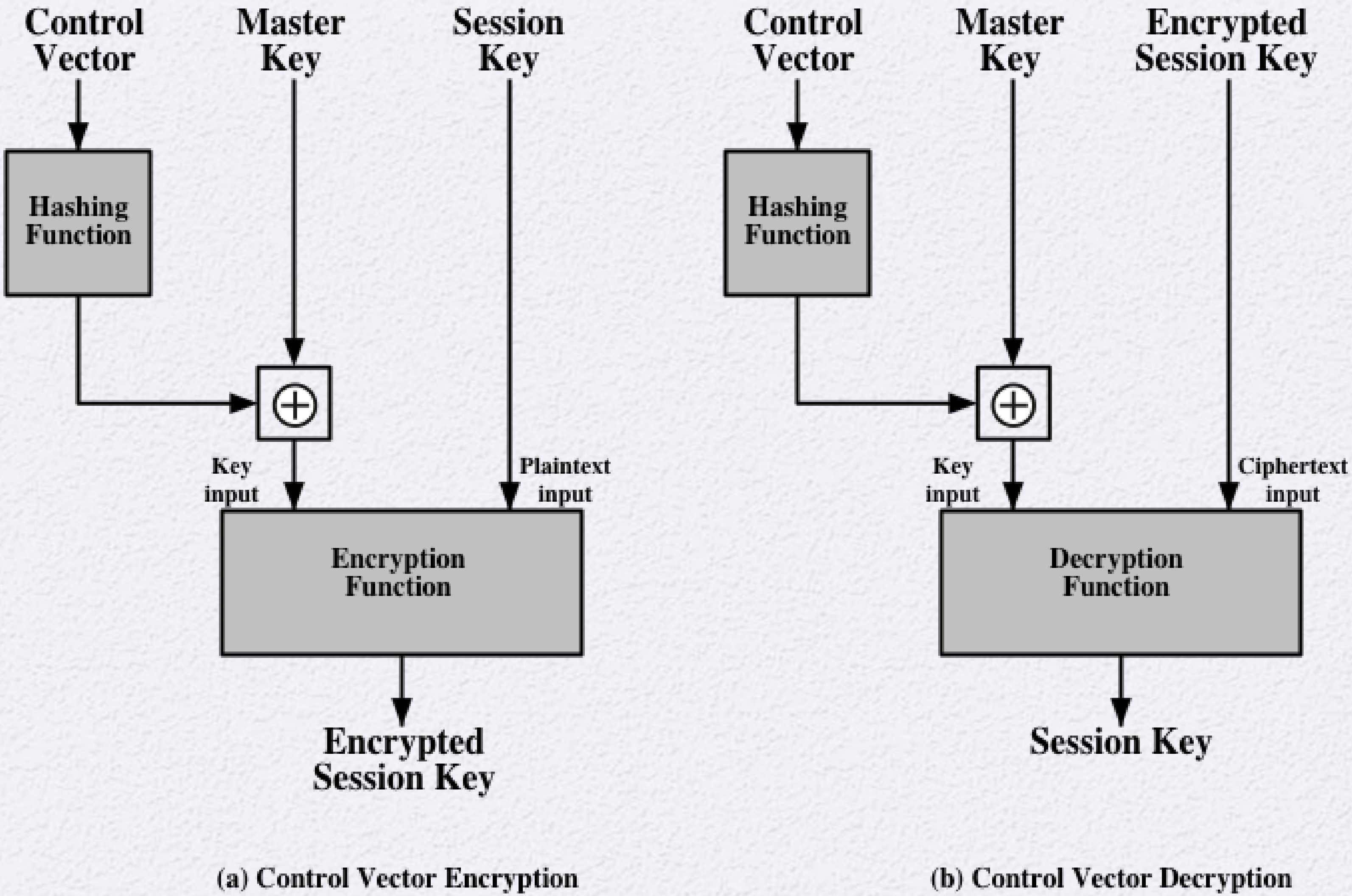
Key Controls

- Associate a tag with each key
  - For use with DES and makes use of the extra 8 bits in each 64-bit DES key
  - The eight non-key bits ordinarily reserved for parity checking form the key tag
  - Because the tag is embedded in the key, it is encrypted along with the key when that key is distributed, thus providing protection



Drawbacks:

- The tag length is limited to 8 bits, limiting its flexibility and functionality
- Because the tag is not transmitted in clear form, it can be used only at the point of decryption, limiting the ways in which key use can be controlled



(a) Control Vector Encryption

(b) Control Vector Decryption

Figure 14.6 Control Vector Encryption and Decryption

Simple Secret Key Distribution

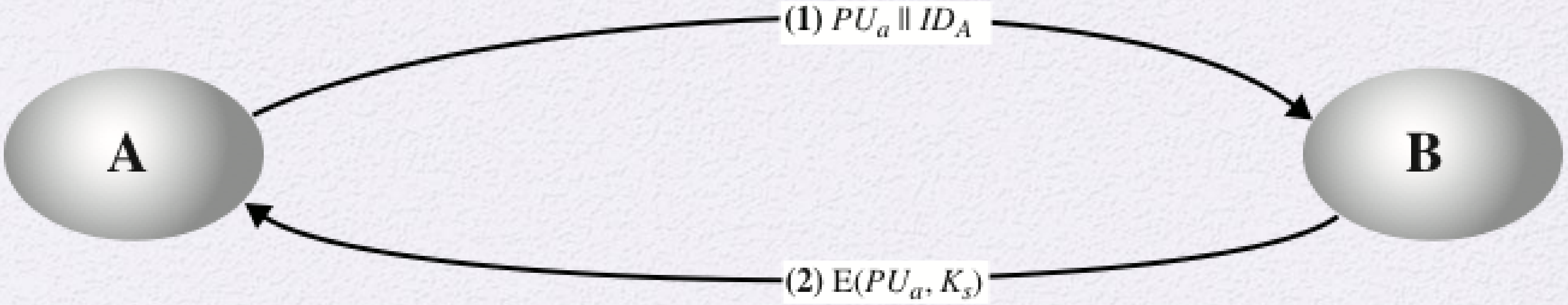


Figure 14.7 Simple Use of Public-Key Encryption to Establish a Session Key

Man-in-the-Middle Attack

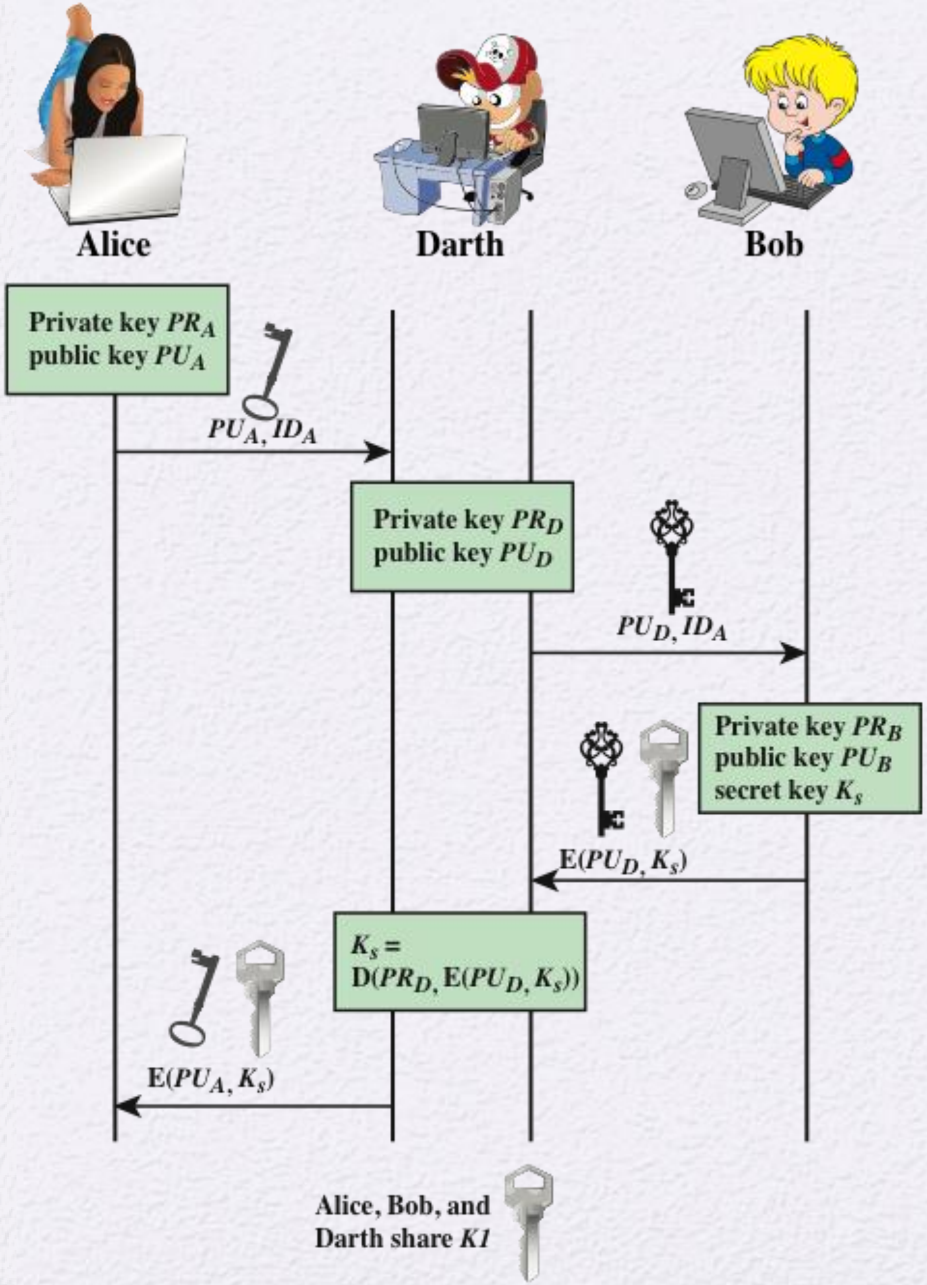


Figure 14.8 Another Man-in-the-Middle Attack

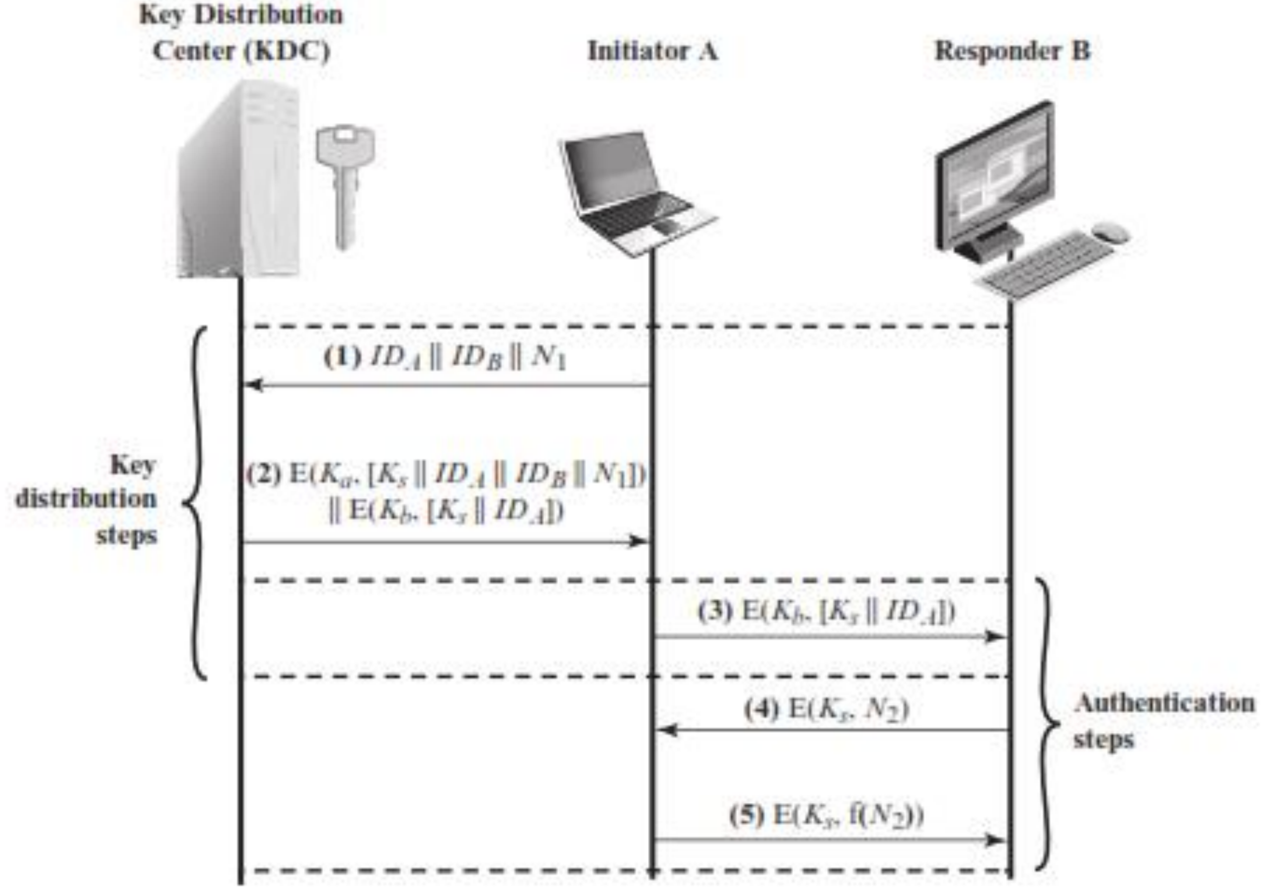
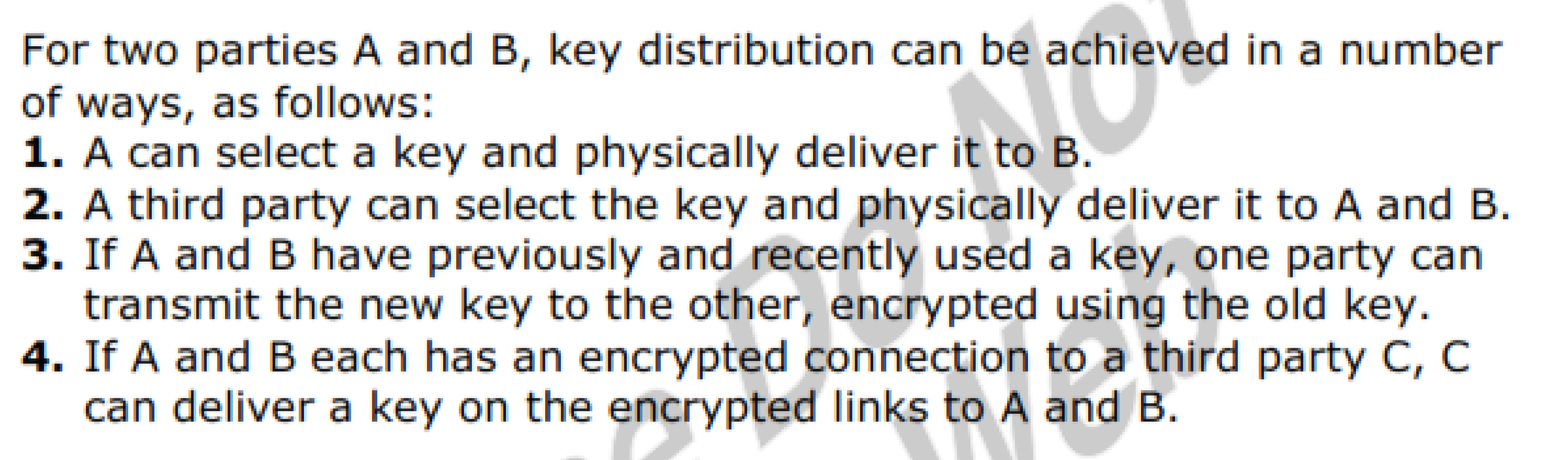

Key Distribution Center (KDC)
Initiator A
Responder B
Key distribution steps
(1) $ID_A \| ID_B \| N_1$
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s \| ID_A])$
(3) $E(K_b, [K_s \| ID_A])$
(4) $E(K_s, N_2)$
(5) $E(K_s, f(N_2))$
Authentication steps

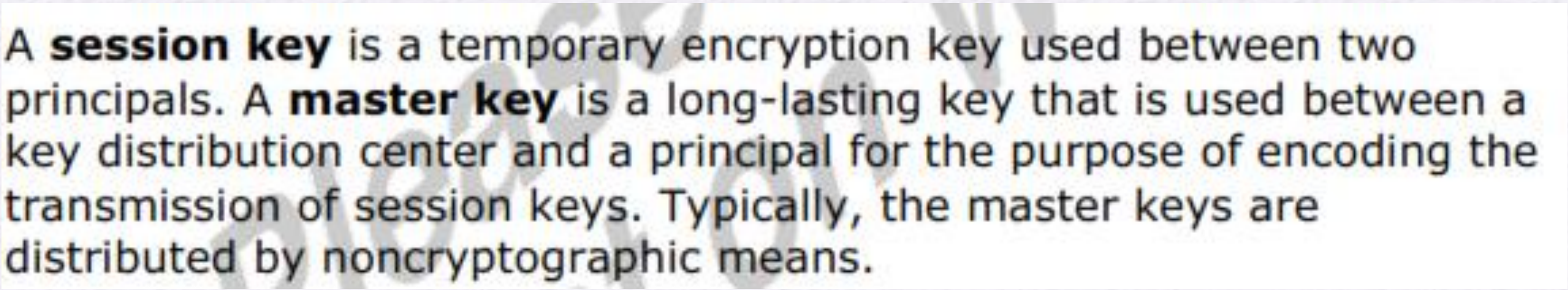
Figure 14.3 Key Distribution Scenario



For two parties A and B, key distribution can be achieved in a number of ways, as follows:

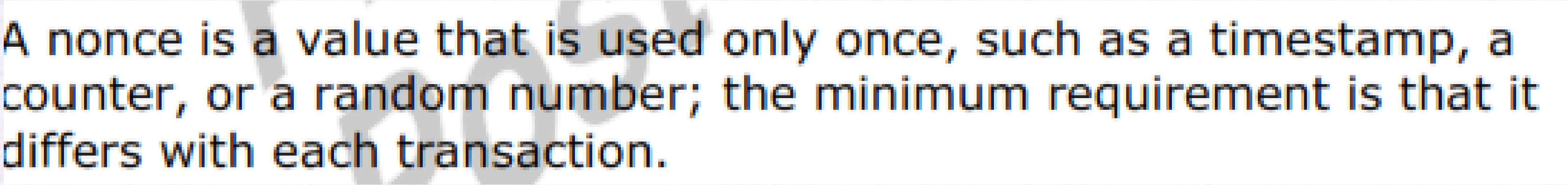
1. A can select a key and physically deliver it to B.
2. A third party can select the key and physically deliver it to A and B.
3. If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.
4. If A and B each has an encrypted connection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

- **What is the major issue in end to end key distribution? How does the key hierarchy c**oncept address that issue?



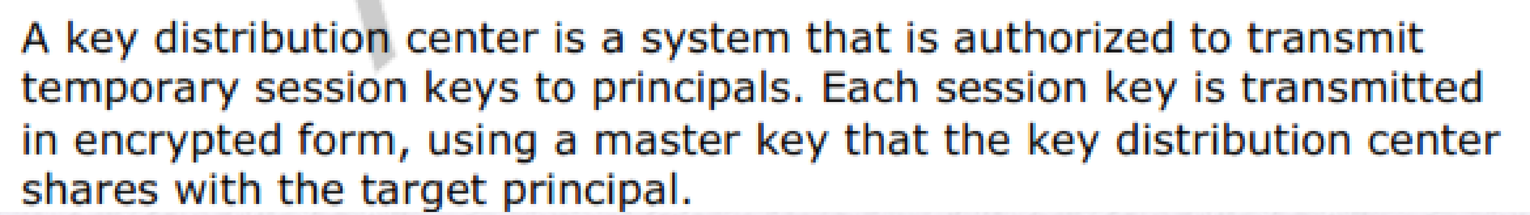
A **session key** is a temporary encryption key used between two principals. A **master key** is a long-lasting key that is used between a key distribution center and a principal for the purpose of encoding the transmission of session keys. Typically, the master keys are distributed by noncryptographic means.

Nonce



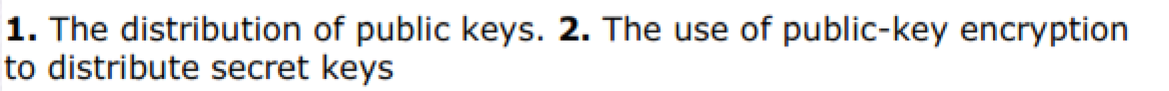
A nonce is a value that is used only once, such as a timestamp, a counter, or a random number; the minimum requirement is that it differs with each transaction.

KDC



A key distribution center is a system that is authorized to transmit temporary session keys to principals. Each session key is transmitted in encrypted form, using a master key that the key distribution center shares with the target principal.

- What are two different uses of public-key cryptography related to key distribution?



1. The distribution of public keys. **2.** The use of public-key encryption to distribute secret keys

- **List four general categories of schemes for the distribution of public keys.**
- Public announcement. Publicly available directory. Public-key authority,Public-key certificates

- Discuss the potential security issues that arise due to public key directory based system.

- **1. The authority maintains a directory with a {name, public key} entry** for each participant.

- **2. Each participant registers a public key with the** directory authority. Registration would have to be in person or by some form of secure authenticated communication.

- **3. A participant** may replace the existing key with a new one at any time, either because of the desire to replace a public key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way.

- **4. Periodically, the authority** publishes the entire directory or updates to the directory. For example, a hard-copy version much like a telephone book could be published, or updates could be listed in a widely circulated newspaper.

- **5.** Participants could also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory.

- What is a public-key certificate?
- A public-key certificate contains a public key and other information, is created by a certificate authority, and is given to the participant with the matching private key. A participant conveys its key information to another by transmitting its certificate. Other participants can verify that the certificate was created by the authority.

- **What are the requirements for the use of a public-key certificate scheme?**

- **1. Any participant can read a certificate to determine the name and** public key of the certificate's owner. **2. Any participant can verify that** the certificate originated from the certificate authority and is not counterfeit. **3. Only the certificate authority can create and update** certificates. **4. Any participant can verify the currency of the** certificate.

- **What is the purpose of the X.509 standard?**
- X.509 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. Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority.

- What is a chain of certificates?
- A chain of certificates consists of a sequence of certificates created by different certification authorities (CAs) in which each successive certificate is a certificate by one CA that certifies the public key of the next CA in the chain.

- How is an X.509 certificate revoked?
- The owner of a public-key can issue a certificate revocation list that revokes one or more certificates.

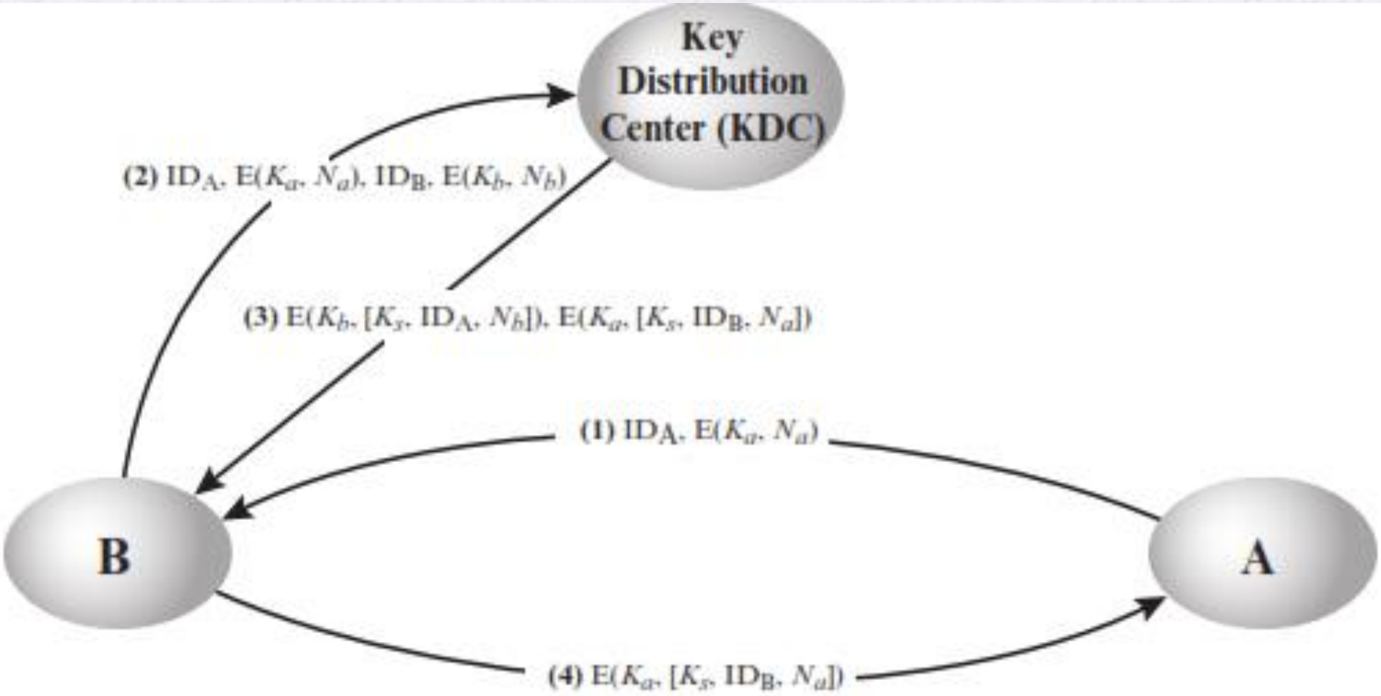

Key Distribution Center (KDC)
(2) ID_A, $E(K_a, N_a)$, ID_B, $E(K_b, N_b)$
(3) $E(K_b, [K_s, \mathrm{ID}_A, N_b])$, $E(K_a, [K_s, \mathrm{ID}_B, N_a])$
(1) ID_A, $E(K_a, N_a)$
B
A
(4) $E(K_a, [K_s, \mathrm{ID}_B, N_a])$

Figure 14.18 Figure for Problem 14.1

- A sends a connection request to B, with an event marker or nonce (Na) encrypted with the key that A shares with the KDC.
- If B is prepared to accept the connection, it sends a request to the KDC for a session key, including A's encrypted nonce plus a nonce generated by B (Nb) and encrypted with the key that B shares with

- The KDC returns two encrypted blocks to B. One block is intended for B and includes the session key, A's identifier, and B's nonce. A similar block is prepared for A and passed from the KDC to B and then to A. A and B have now securely obtained the session key and, because of the nonces, are assured that the other is authentic.

True / False

- Some sort of mechanism or protocol is needed to provide for the secure distribution of keys.
- A public-key certificate scheme alone does not provide the necessary security to authenticate the public key.
- For symmetric encryption to work the two parties to an exchange must share the same key and that key must be protected from access by others.
- X.509 defines the format for private-key certificates.

- T
- F
- T
- F

- The topics of cryptographic key management and cryptographic key distribution are complex, involving cryptographic, protocol, and management considerations.
- Frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key.
- For link encryption manual delivery is awkward.
- Each user must share a unique key with the key distribution center for purposes of key distribution.

- Typically the session key is used for the duration of a logical connection, such as a frame relay connection or transport connection, and then it is permanently stored.

- Master keys can be distributed in some noncryptographic way such as physical delivery.

- A random number would not be a good choice for a nonce.

- The distribution of session keys delays the start of any exchange and places a burden on network capacity.

- F
- T
- F
- T

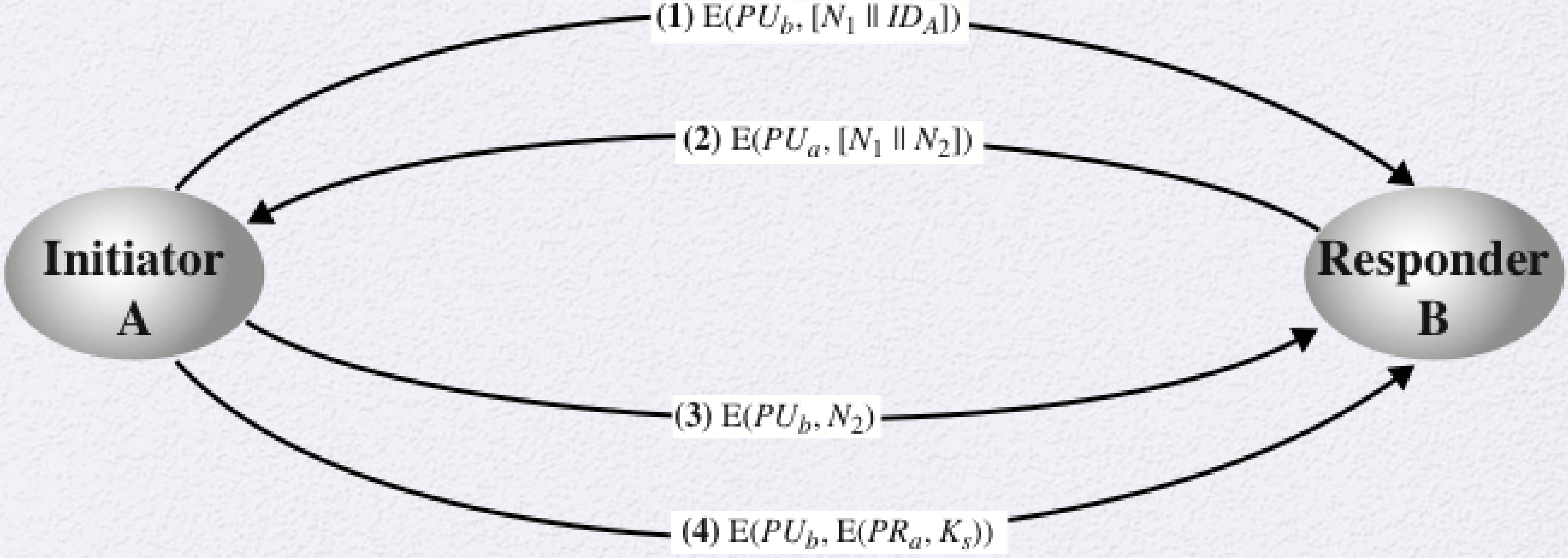
- Although public announcement of public keys is convenient, anyone can forge a public announcement.

- X.509 is an important standard because the certificate structure and authentication protocols defined in X.509 are used in a variety of contexts.

- Because certificates are forgeable they cannot be placed in a directory without the need for the directory to make special efforts to protect them.

- T
- T
- F
- T

Secret Key Distribution with Confidentiality and Authentication



(1) $E(PU_b, [N_1 \| ID_A])$
(2) $E(PU_a, [N_1 \| N_2])$
Initiator A
Responder B
(3) $E(PU_b, N_2)$
(4) $E(PU_b, E(PR_a, K_s))$

Figure 14.9 Public-Key Distribution of Secret Keys

A Hybrid Scheme

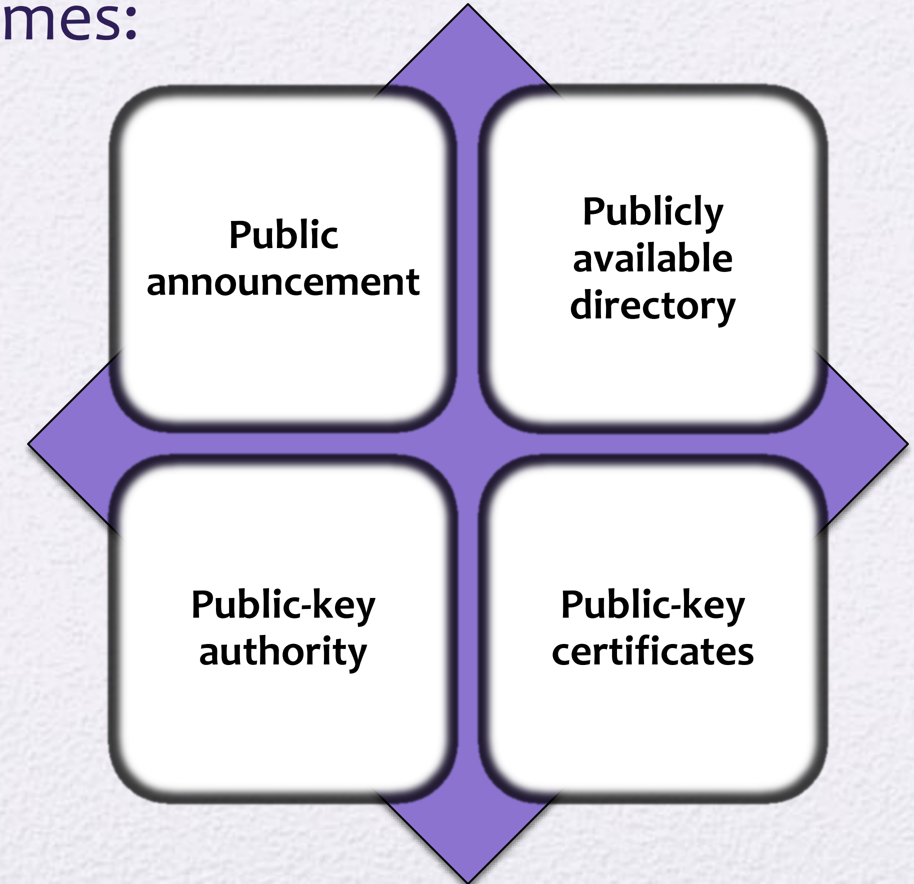
- In use on IBM mainframes
- Retains the use of a key distribution center (KDC) that shares a secret master key with each user and distributes secret session keys encrypted with the master key
- A public-key scheme is used to distribute the master keys

Rationale:

- Performance
- Backward compatibility

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

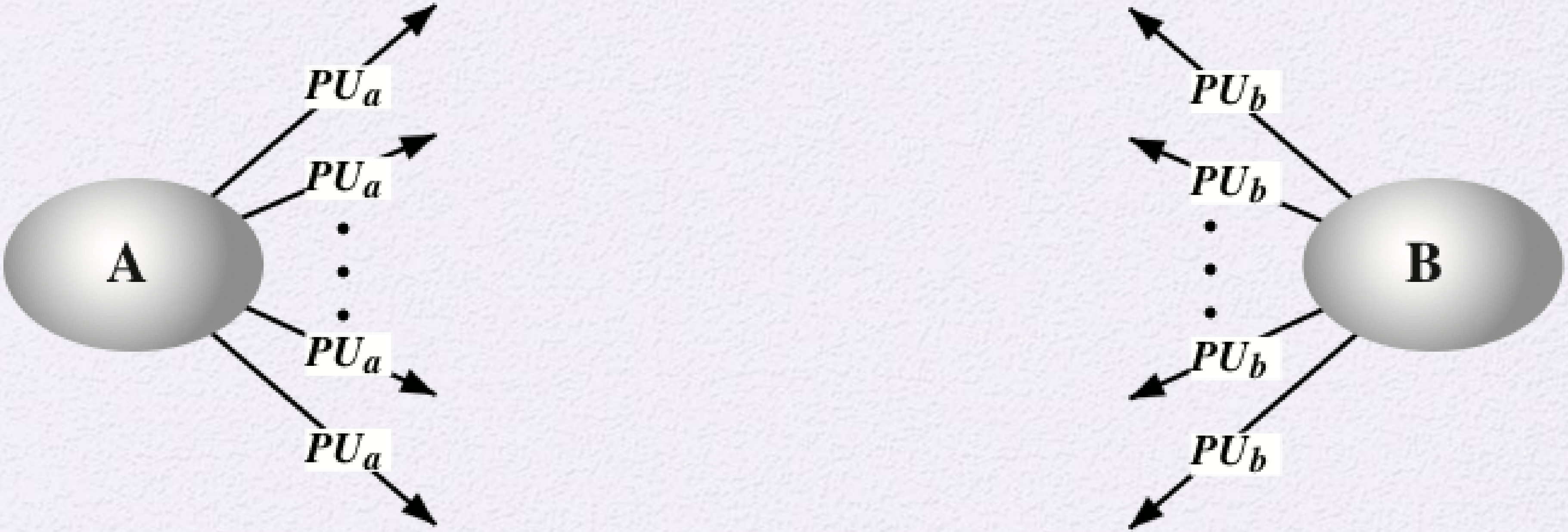


Figure 14.10 Uncontrolled Public Key Distribution

Publicly Available Directory

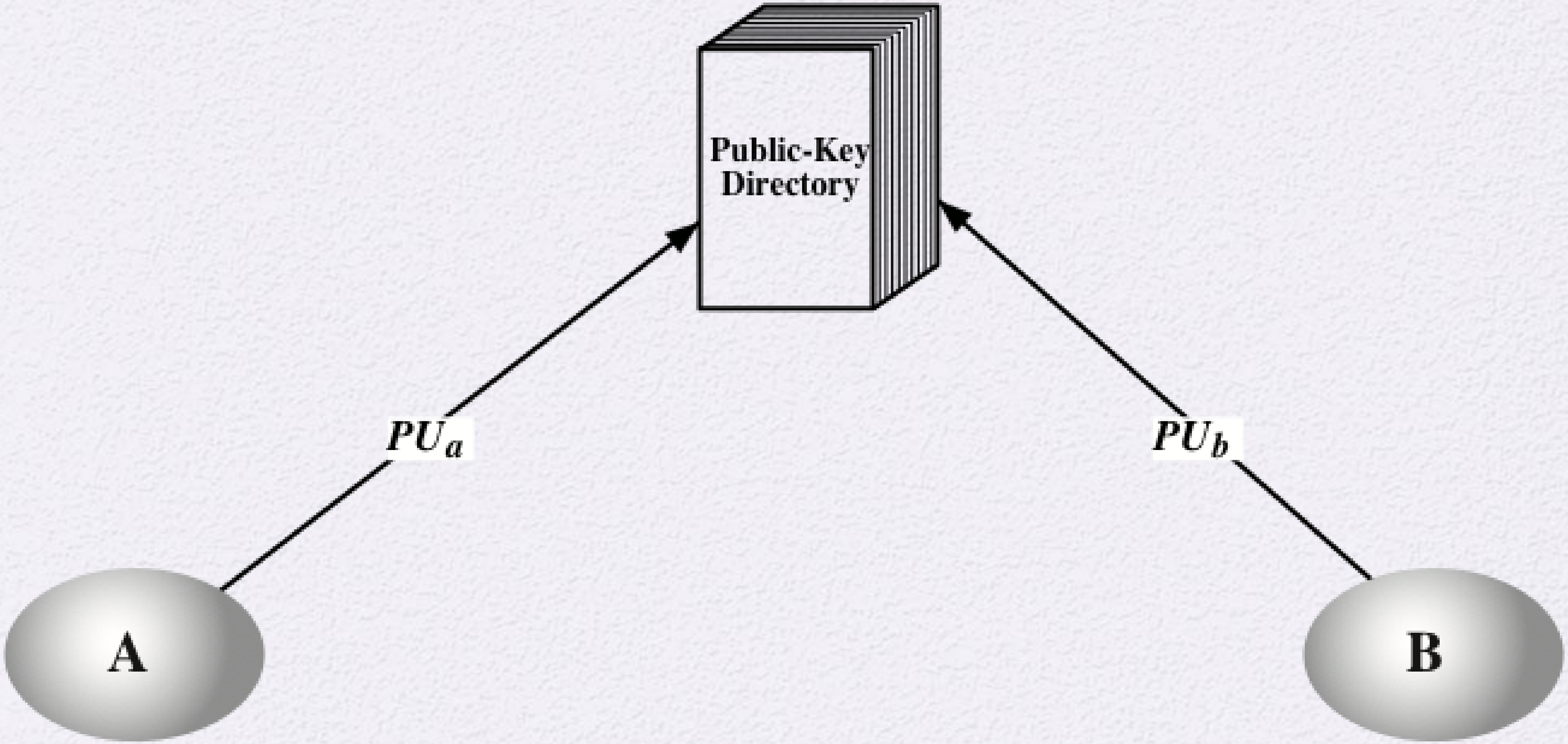


Figure 14.11 Public Key Publication

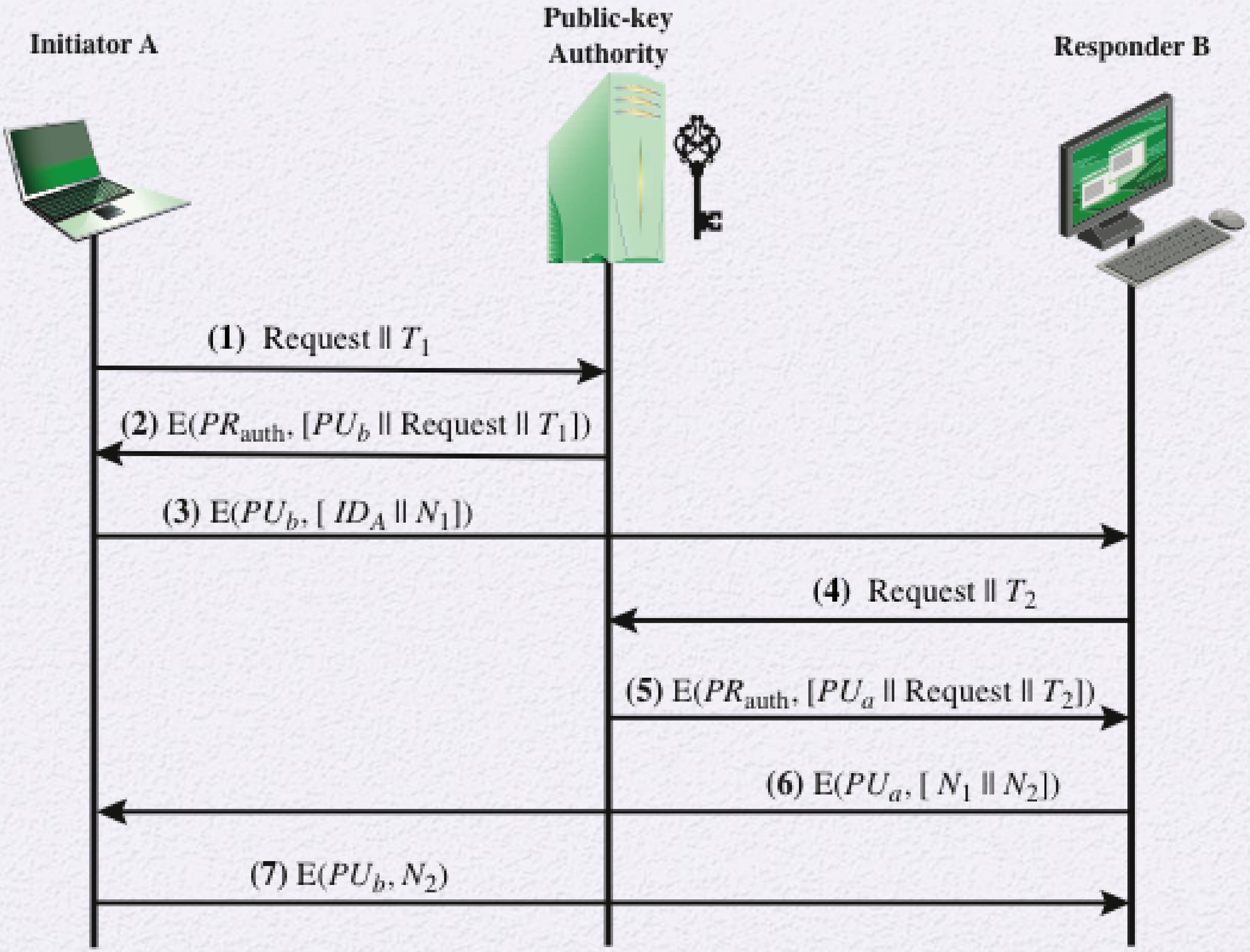
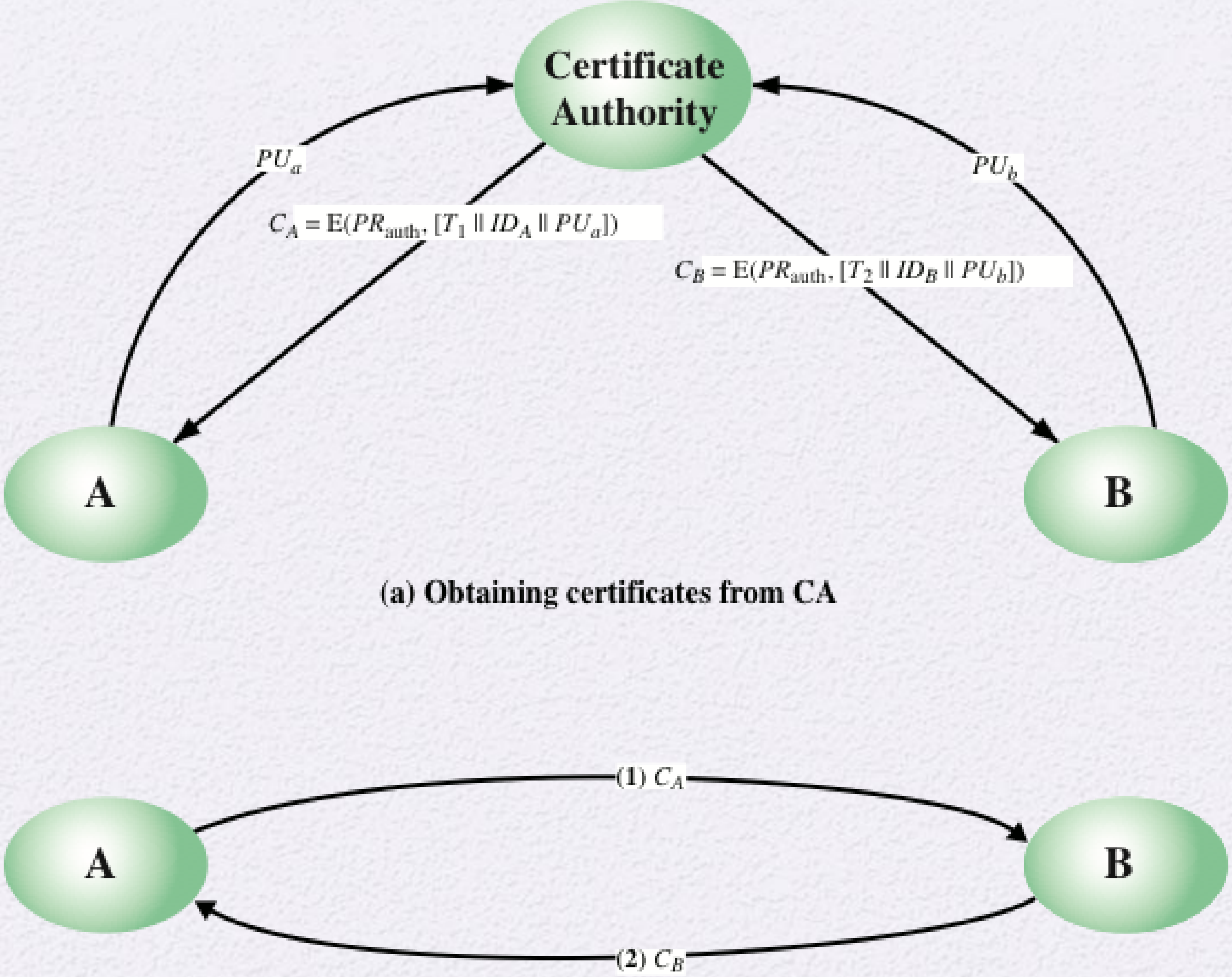

Initiator A
Public-key Authority
Responder B
(1) Request ‖ T_1
(2) E(PR_{auth}, [PU_b ‖ Request ‖ T_1])
(3) E(PU_b, [ID_A ‖ N_1])
(4) Request ‖ T_2
(5) E(PR_{auth}, [PU_a ‖ Request ‖ T_2])
(6) E(PU_a, [N_1 ‖ N_2])
(7) E(PU_b, N_2)

Figure 14.12 Public-Key Distribution Scenario


Certificate Authority
PU_a
PU_b
$C_A = \mathrm{E}(PR_{\mathrm{auth}}, [T_1 \| ID_A \| PU_a])$
$C_B = \mathrm{E}(PR_{\mathrm{auth}}, [T_2 \| ID_B \| PU_b])$
A
B
(1) C_A
(2) C_B
A
B

(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

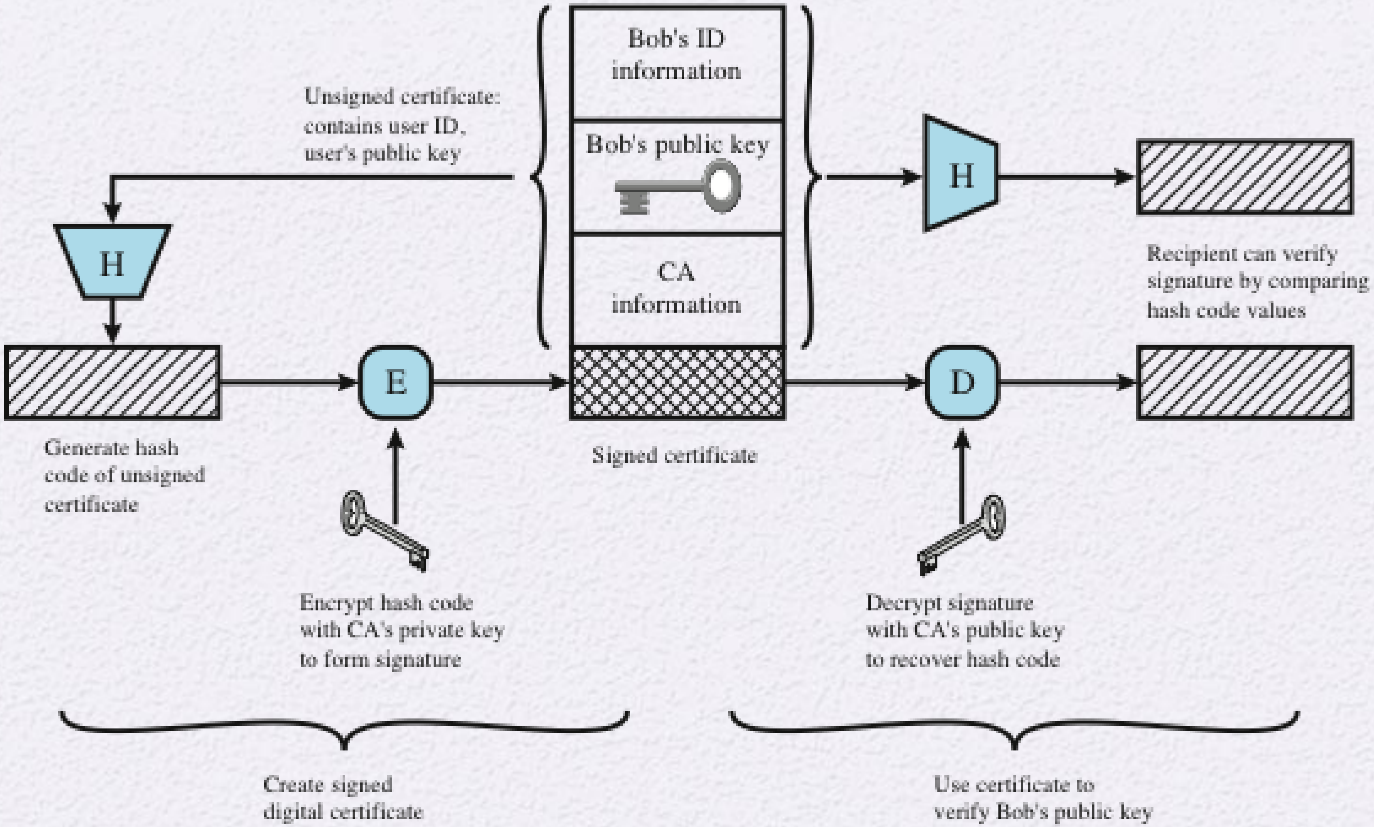
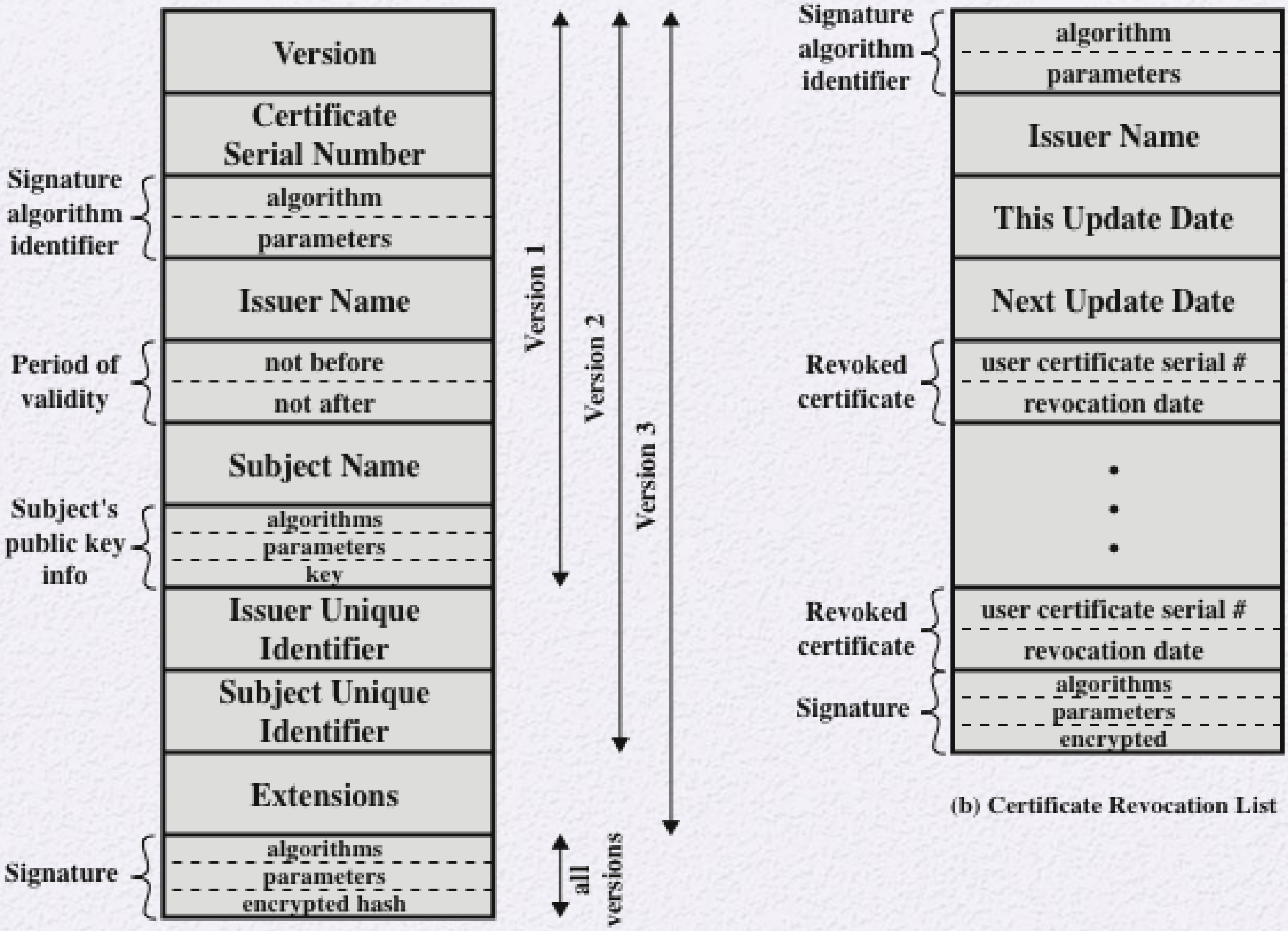

Bob's ID information
Bob's public key
CA information
Unsigned certificate: contains user ID, user's public key
H
H
Recipient can verify signature by comparing hash code values
E
D
Generate hash code of unsigned certificate
Signed certificate
Encrypt hash code with CA's private key to form signature
Decrypt signature with CA's public key to recover hash code
Create signed digital certificate
Use certificate to verify Bob's public key

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



Version

Certificate Serial Number

Signature algorithm identifier: algorithm, parameters

Issuer Name

Period of validity: not before, not after

Subject Name

Subject's public key info: algorithms, parameters, key

Issuer Unique Identifier

Subject Unique Identifier

Extensions

Signature: algorithms, parameters, encrypted hash

Version 1

Version 2

Version 3

all versions

(a) X.509 Certificate

Signature algorithm identifier: algorithm, parameters

Issuer Name

This Update Date

Next Update Date

Revoked certificate: user certificate serial #, revocation date

•
•
•

Revoked certificate: user certificate serial #, revocation date

Signature: algorithms, parameters, encrypted

(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

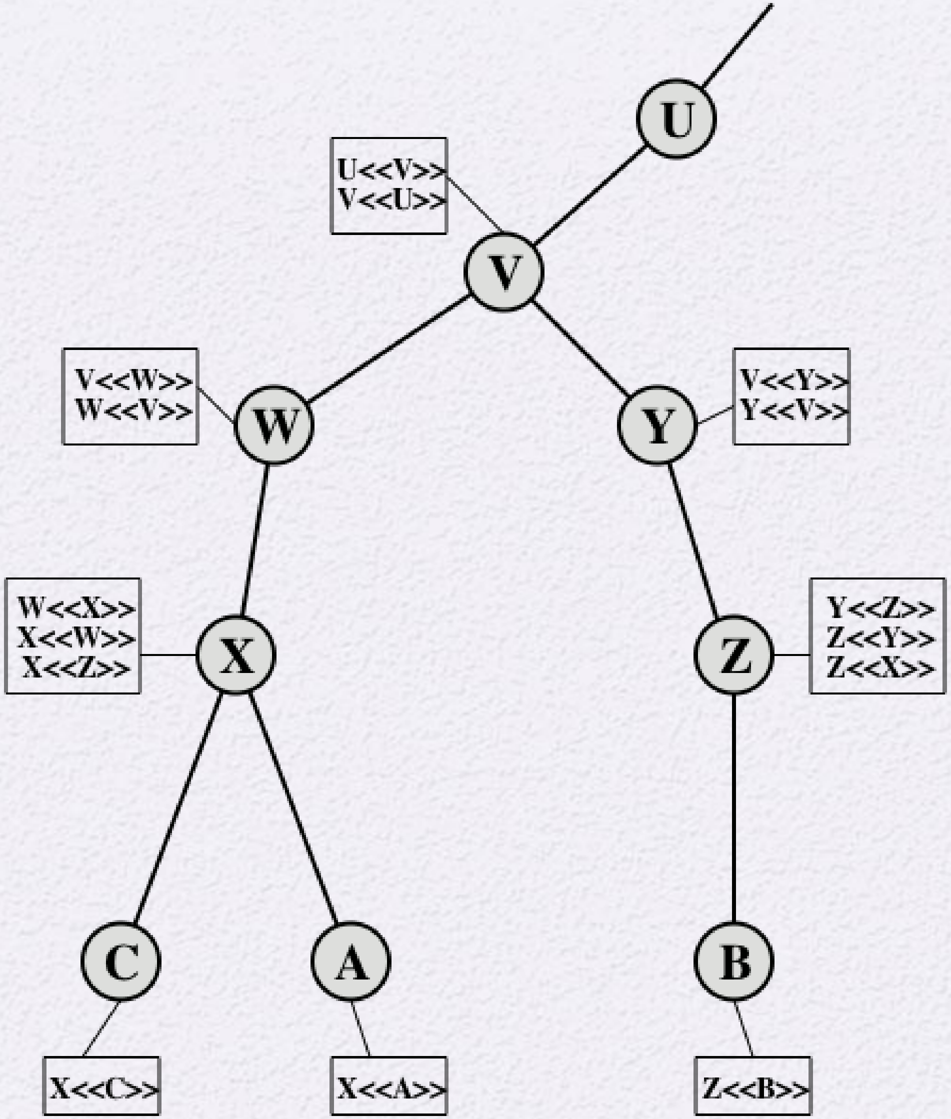


Figure 14.16 X.509 CA Hierarchy: a Hypothetical Example

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

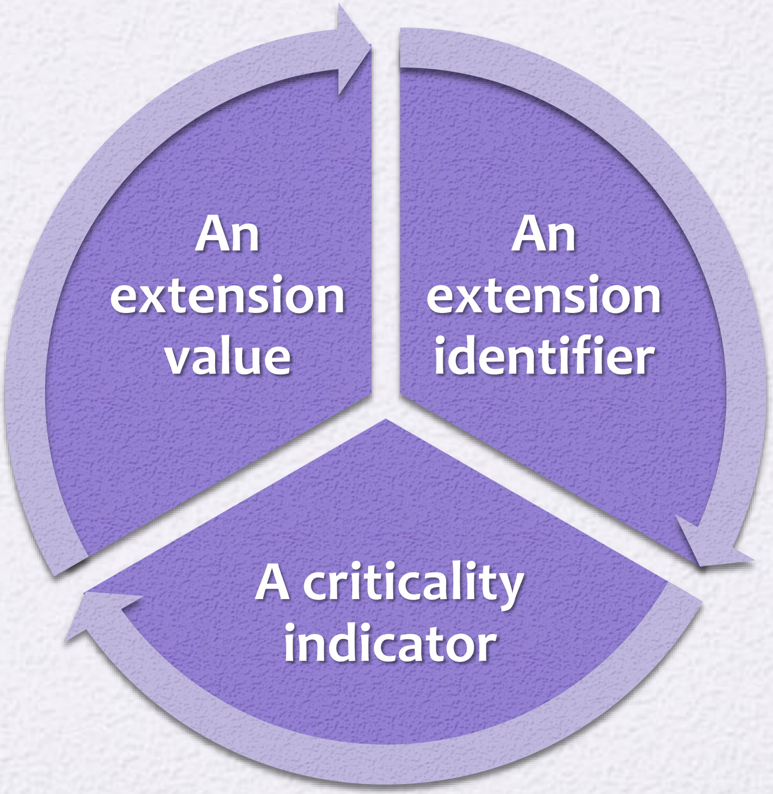
X.509 Version 3

- Version 2 format does not convey all of the information that recent design and implementation experience has shown to be needed

- Rather than continue to add fields to a fixed format, standards developers felt that a more flexible approach was needed
  - Version 3 includes a number of optional extensions

- The certificate extensions fall into three main categories:
  - Key and policy information
  - Subject and issuer attributes
  - Certification path constraints

Each extension consists of:



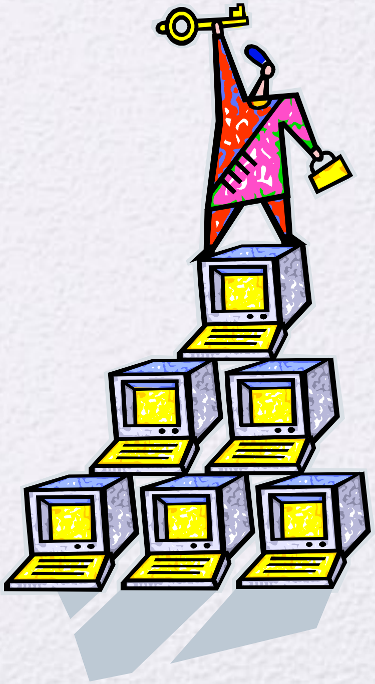
Key and Policy Information

- These 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 indicates the applicability of a certificate to a particular community and/or class of application with common security requirements

Included are:

- Authority key identifier
- Subject key identifier
- Key usage
- Private-key usage period
- Certificate policies
- Policy mappings

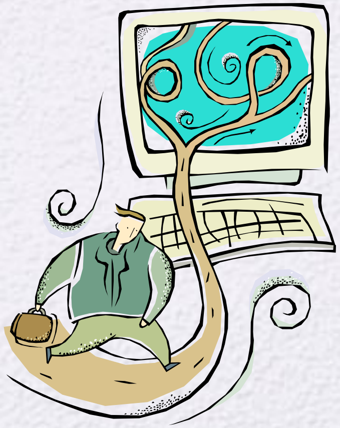


Certificate Subject and Issuer Attributes

- These extensions support alternative names, in alternative formats, for a certificate subject or certificate issuer
- Can convey additional information about the certificate subject to increase a certificate user's confidence that the certificate subject is a particular person or entity
- The extension fields in this area include:
  - Subject alternative name
  - Issuer alternative name
  - Subject directory attributes

Certification Path Constraints

- These extensions allow constraint specifications to be included in certificates issued for CAs by other CAs
- The constraints may restrict the types of certificates that can be issued by the subject CA or that may occur subsequently in a certification chain
- The extension fields in this area include:
  - Basic constraints
  - Name constraints
  - Policy constraints



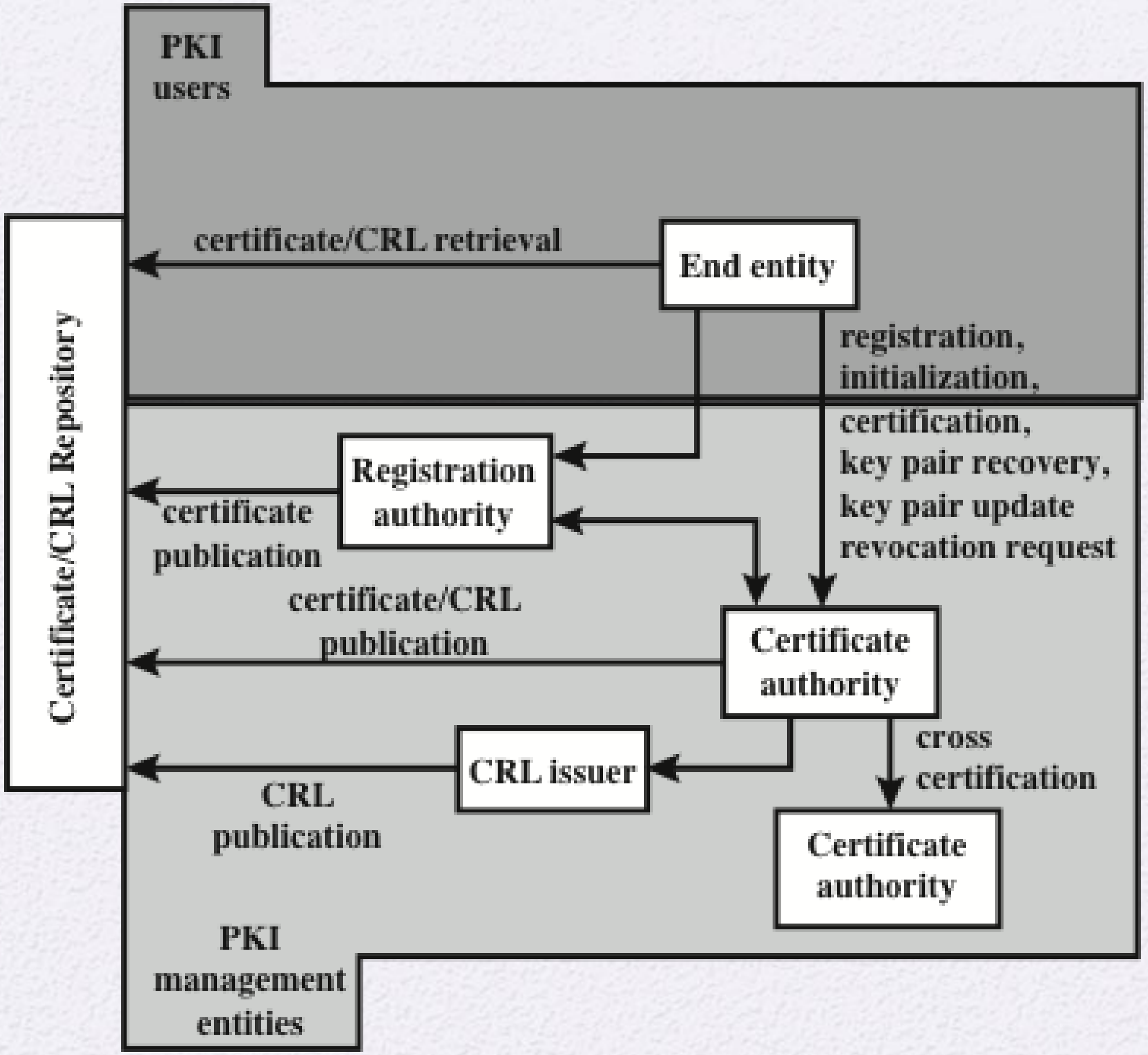


Figure 14.17 PKIX Architectural Model

PKIX Management Functions

- PKIX identifies a number of management functions that potentially need to be supported by management protocols:
  - Registration
  - Initialization
  - Certification
  - Key pair recovery
  - Key pair update
  - Revocation request
  - Cross certification

Summary

- Symmetric key distribution using symmetric encryption
  - Key distribution scenario
  - Hierarchical key control
  - Session key lifetime
  - Transparent key control scheme
  - Decentralized key control
  - Controlling key usage
- 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
  - X.509 Version 3
- Public-key infrastructure
  - PKIX management functions
  - PKIX management protocols