Chapter 15

Key Management

Chapter 15

Objectives

- To explain the need for a key-distribution center
- To show how a KDC can create a session key
- To show how two parties can use a symmetric-key agreement protocol to create a session key
- To describe Kerberos as a KDC and an authentication protocol
- To explain the need for certification authorities for public keys
- To introduce the idea of a Public-Key Infrastructure (PKI) and explain some of its duties

15-1 SYMMETRIC-KEY DISTRIBUTION

Symmetric-key cryptography is more efficient than asymmetric-key cryptography for enciphering large messages. Symmetric-key cryptography, however, needs a shared secret key between two parties. The distribution of keys is another problem.

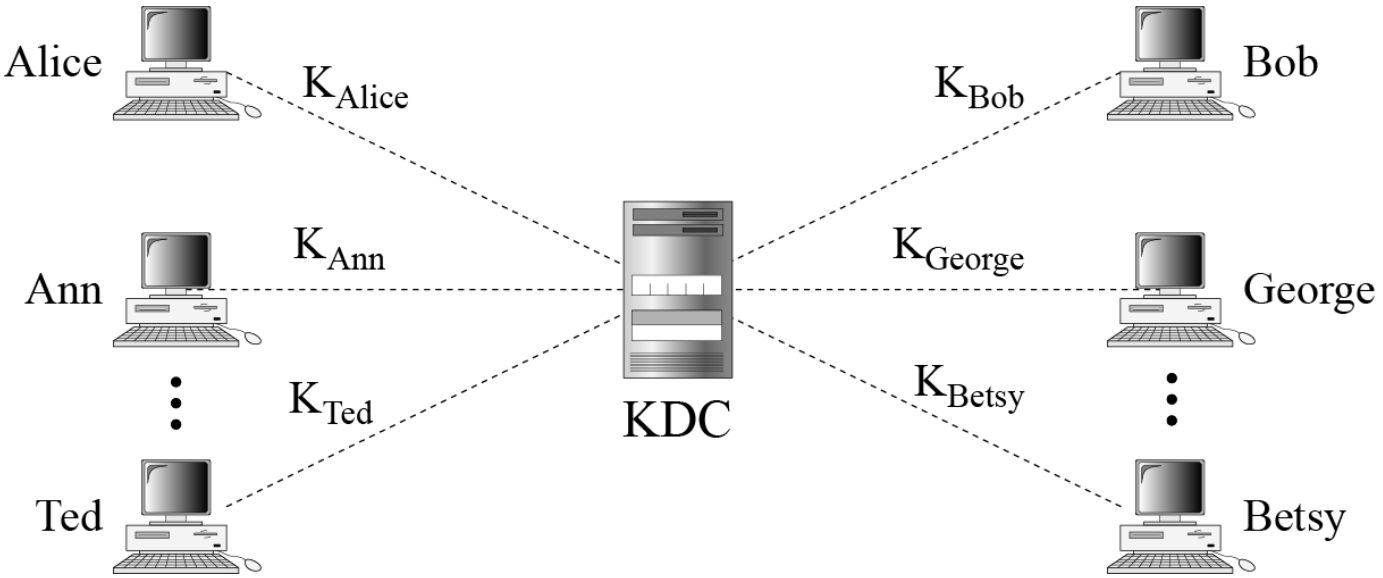
Topics discussed in this section:

15.1.1 Key-Distribution Center: KDC

15.1.2 Session Keys

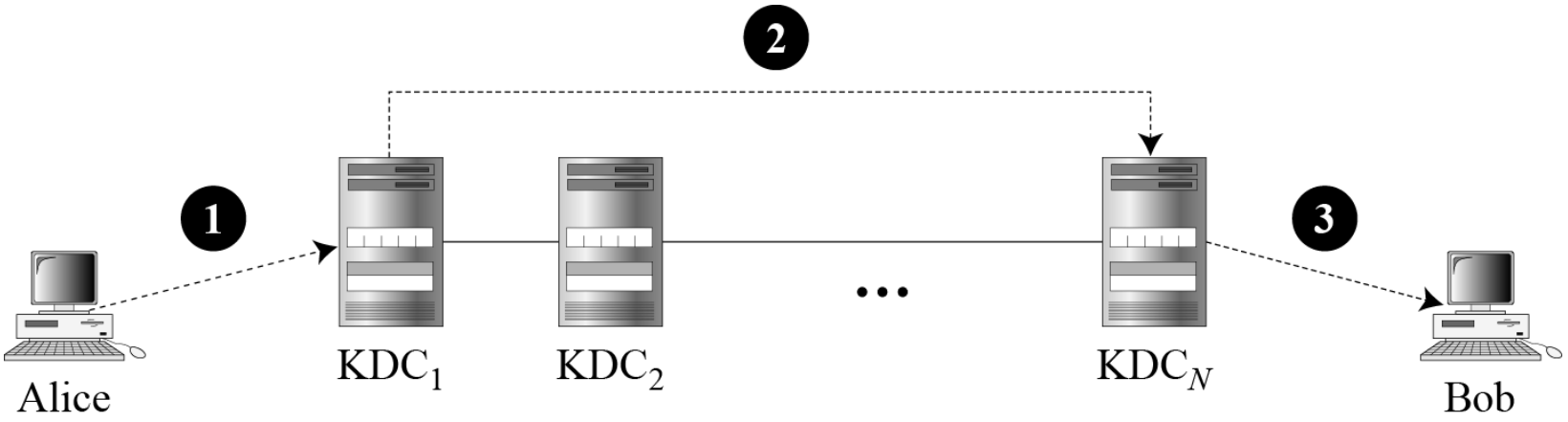
15.1.1 Key-Distribution Center: KDC

Figure 15.1 *Key-distribution center (KDC)*



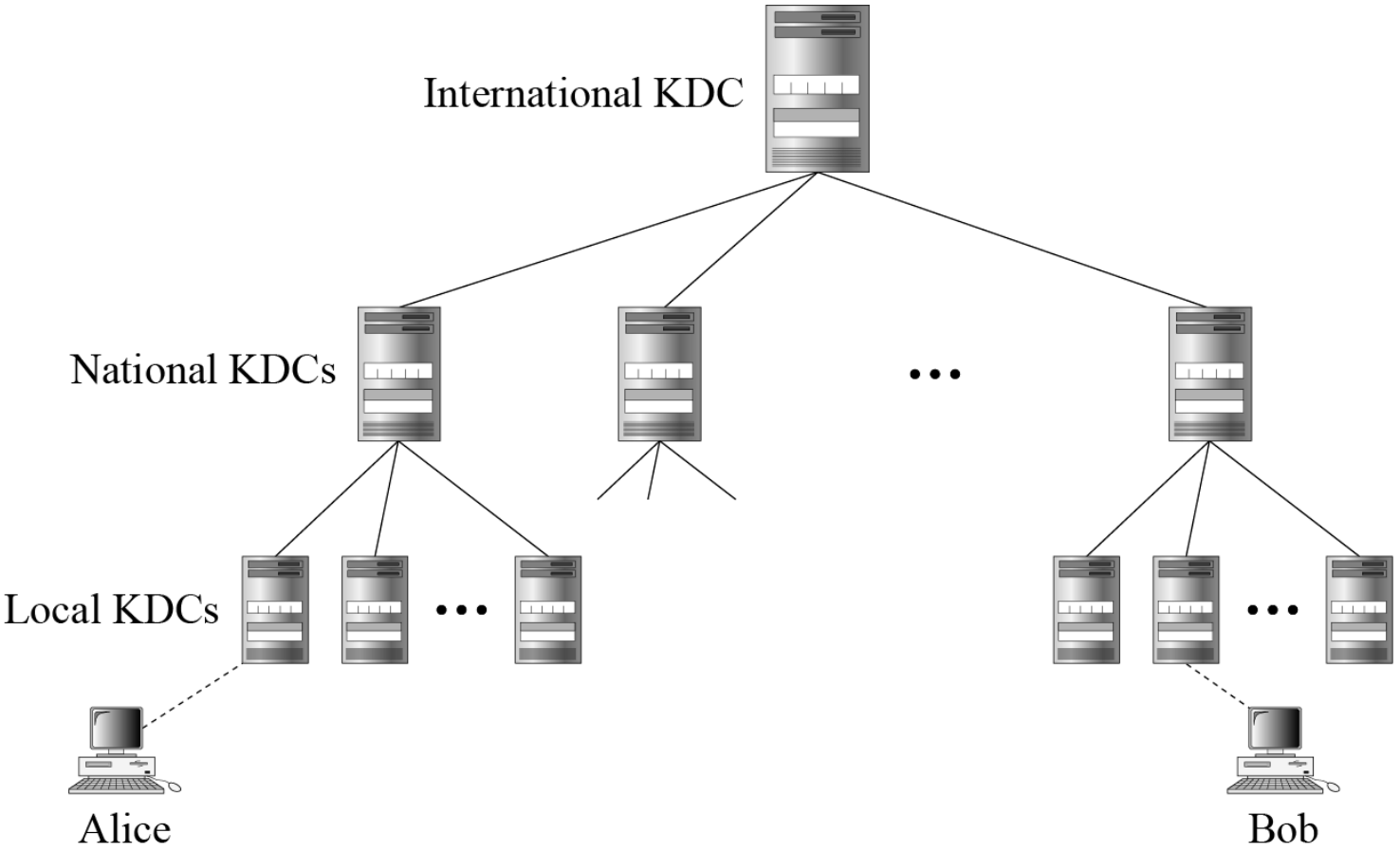
Flat Multiple KDCs.

Figure 15.2 *Flat multiple KDCs*



Hierarchical Multiple KDCs

Figure 15.3 *Hierarchical multiple KDCs*



15.1.2 Session Keys

A KDC creates a secret key for each member. This secret key can be used only between the member and the KDC, not between two members.

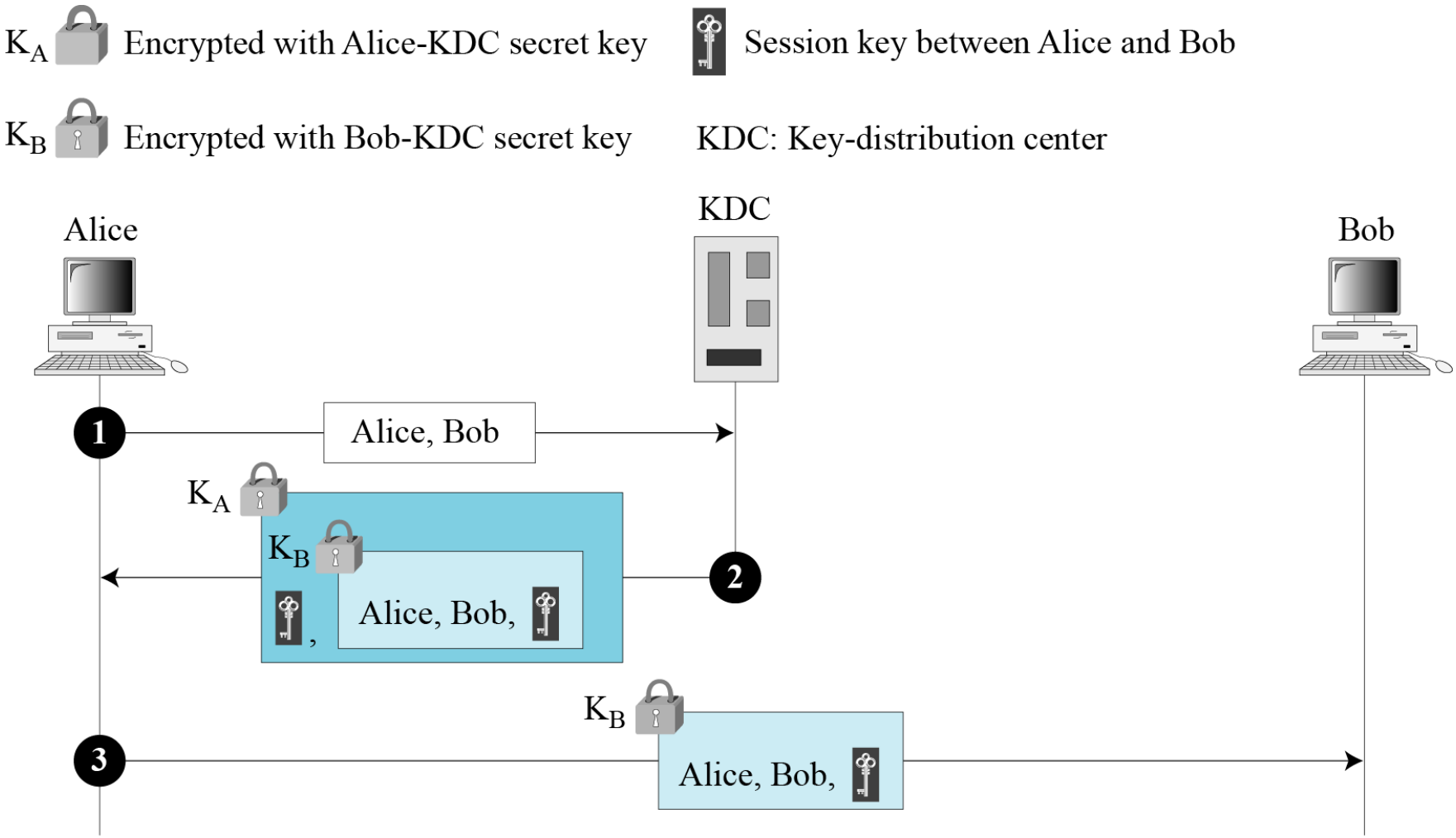
Note

A session symmetric key between two parties is used only once.

15.1.2 Continued

A Simple Protocol Using a KDC

Figure 15.4 *First approach using KDC*



15.1.2 Continued

Needham-Schroeder Protocol

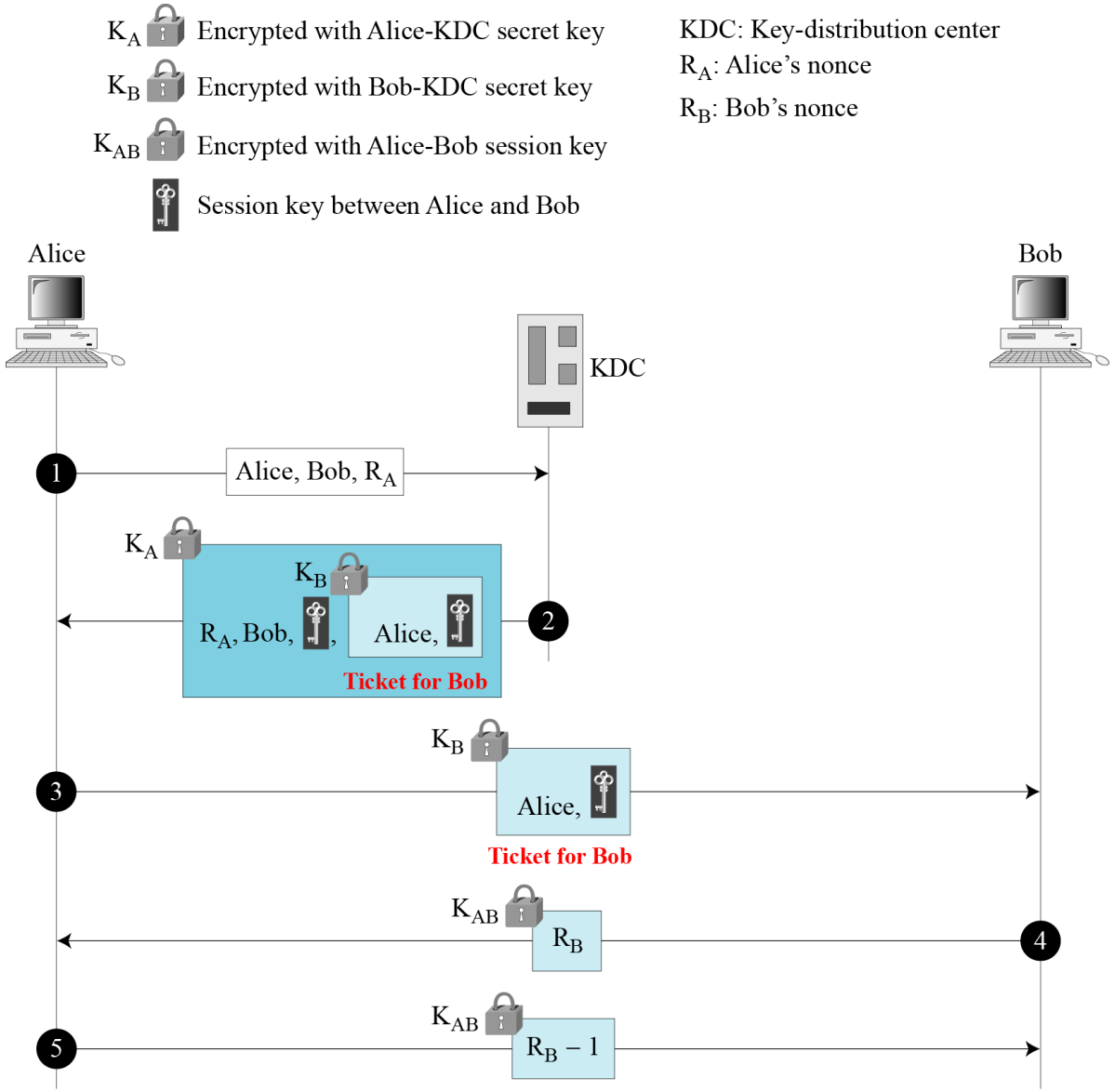


Figure 15.5

Needham-Schroeder protocol

15.1.2 Continued

Otway-Rees Protocol

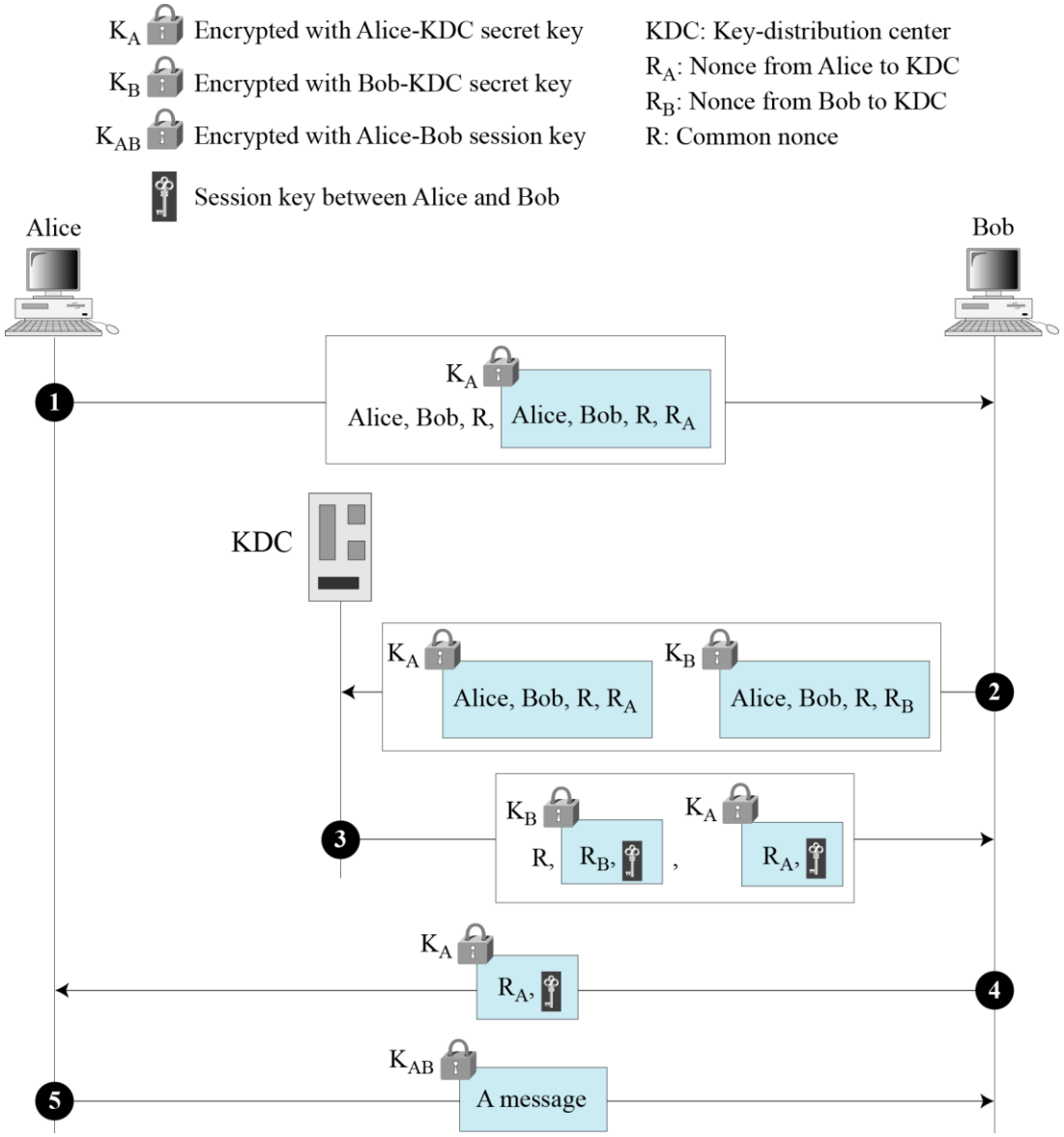


Figure 15.6

Otway-Rees protocol

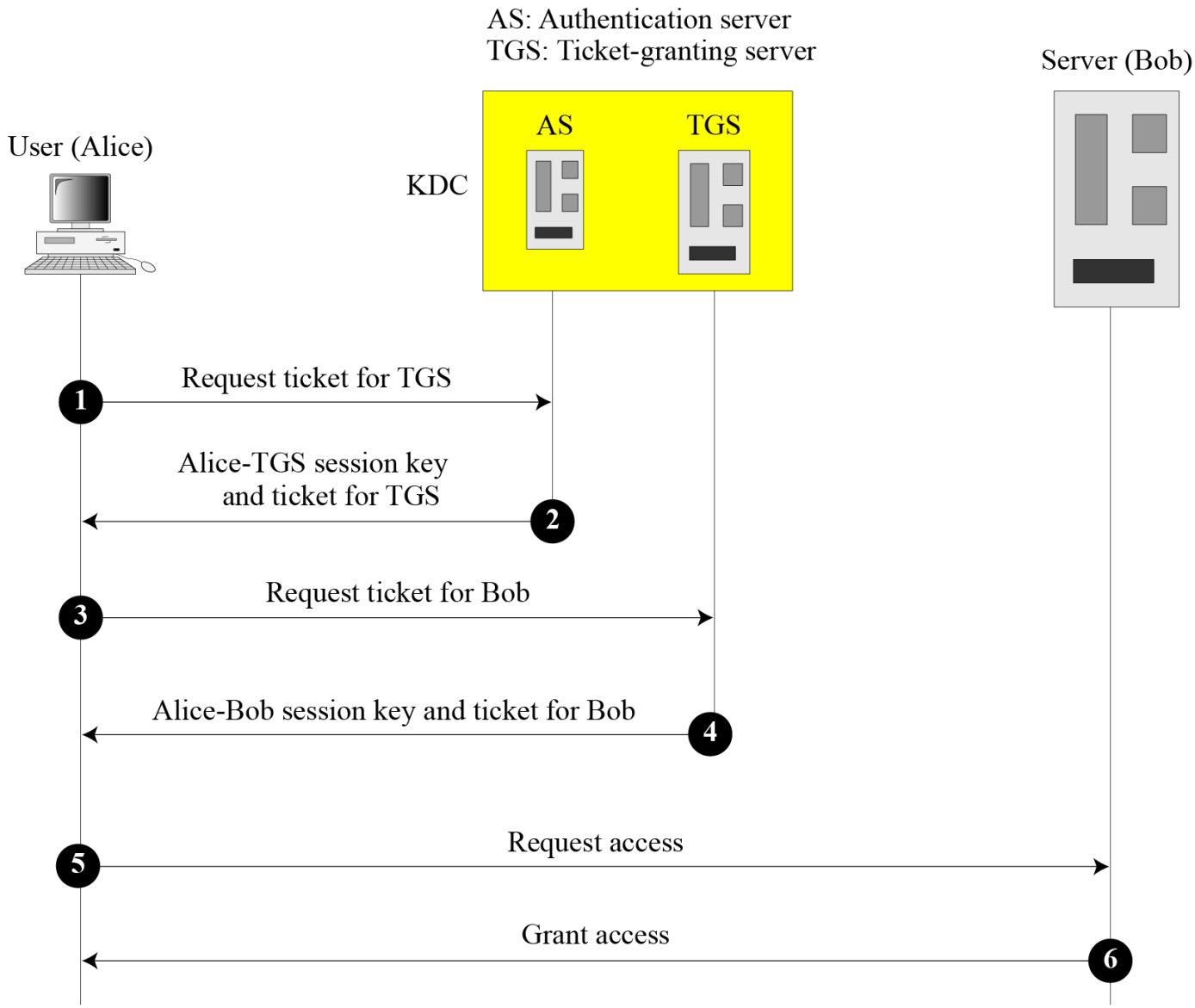
15-2 KERBEROS

Kerberos is an authentication protocol, and at the same time a KDC, that has become very popular. Several systems, including Windows 2000, use Kerberos. Originally designed at MIT, it has gone through several versions.

Topics discussed in this section:

15.2.1 Servers

Figure 15.7 *Kerberos servers*



15.2.1 Continued

Authentication Server (AS)

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

Ticket-Granting Server (TGS)

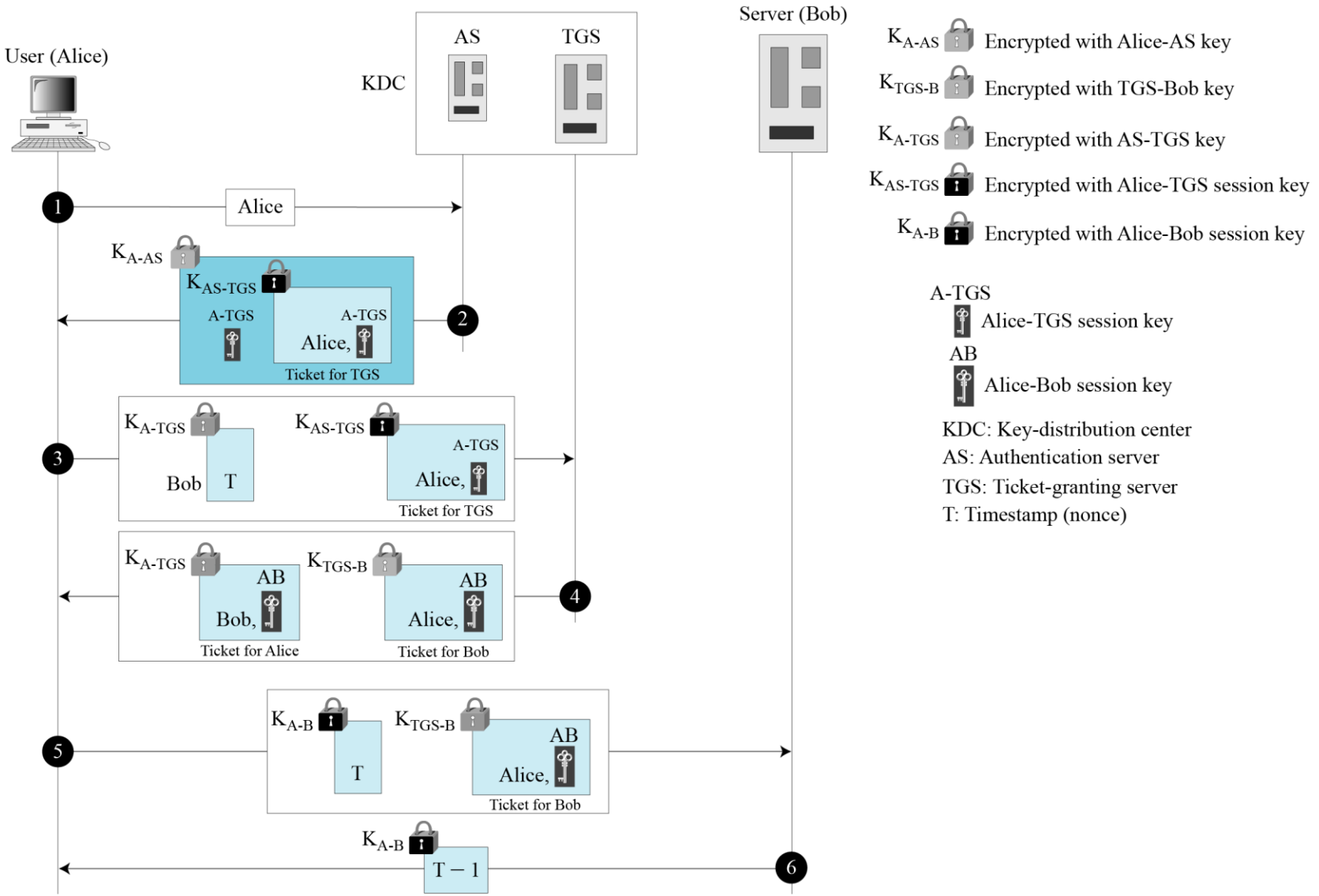
The ticket-granting server (TGS) issues a ticket for the real server (Bob).

Real Server

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

15.2.2 Operation

Figure 15.8 *Kerberos example*



15.2.3 Using Different Servers

Note that if Alice needs to receive services from different servers, she need repeat only the last four steps.

15.2.4 Kerberos Version 5

The minor differences between version 4 and version 5 are briefly listed below:

1) *Version 5 has a longer ticket lifetime.*
2) *Version 5 allows tickets to be renewed.*
3) *Version 5 can accept any symmetric-key algorithm.*
4) *Version 5 uses a different protocol for describing data types.*
5) *Version 5 has more overhead than version 4.*

15.2.5 Realms

Kerberos allows the global distribution of ASs and TGSs, with each system called a realm. A user may get a ticket for a local server or a remote server.

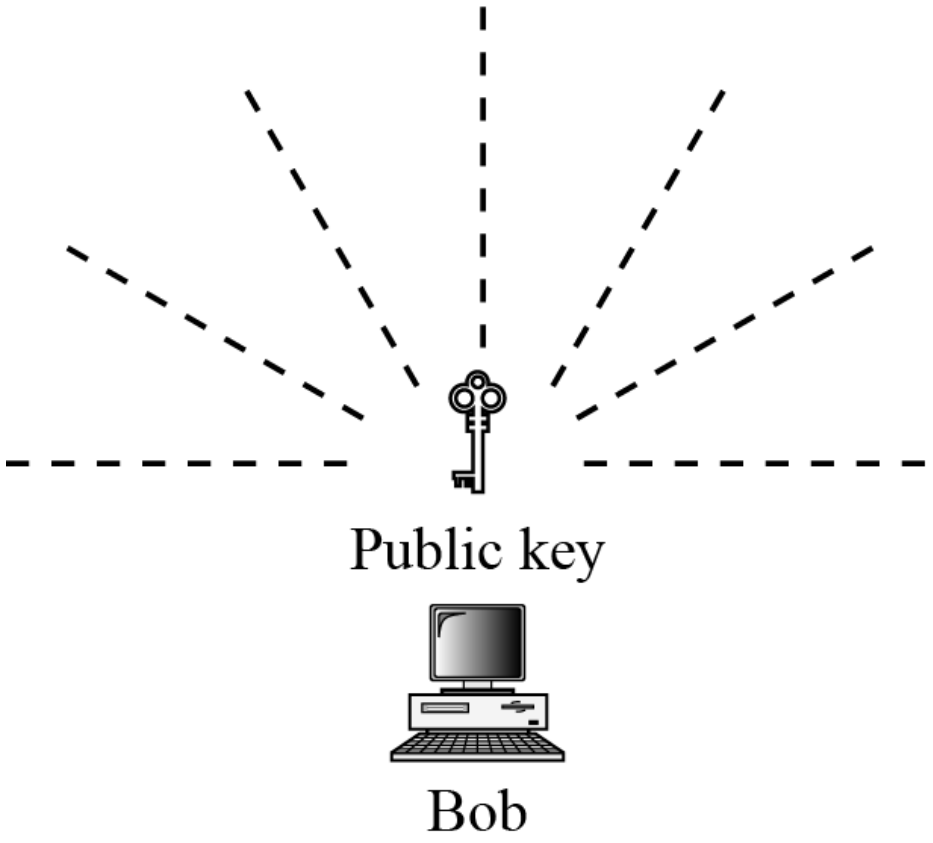
15-4 PUBLIC-KEY DISTRIBUTION

In asymmetric-key cryptography, people do not need to know a symmetric shared key; everyone shields a private key and advertises a public key.

Topics discussed in this section:

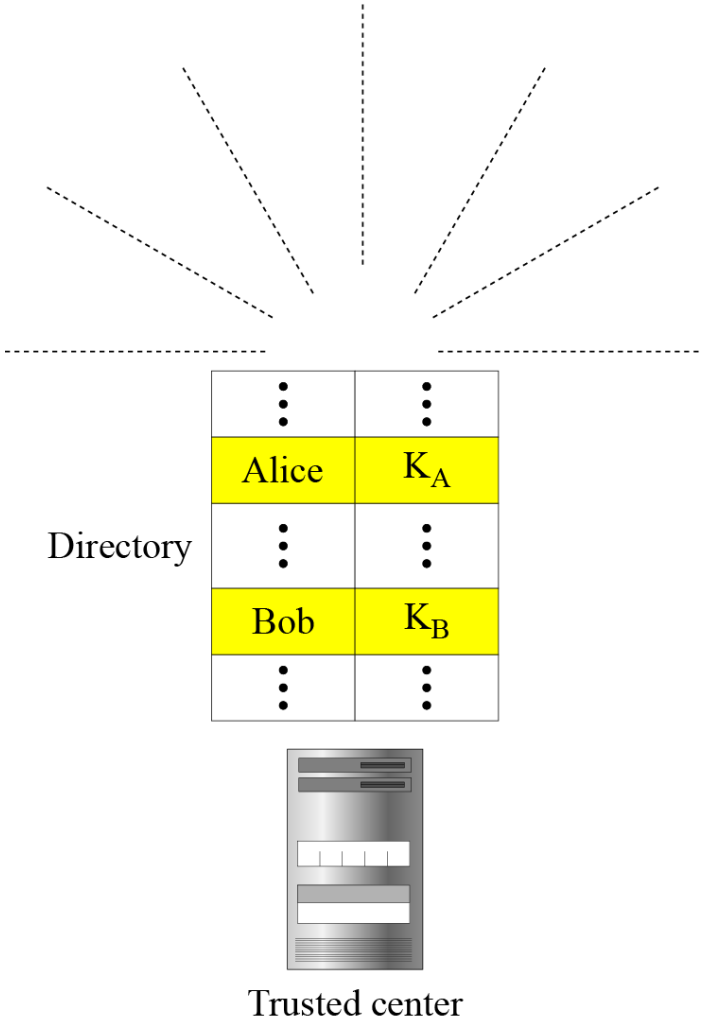
15.4.1 Public Announcement

Figure 15.13 *Announcing a public key*



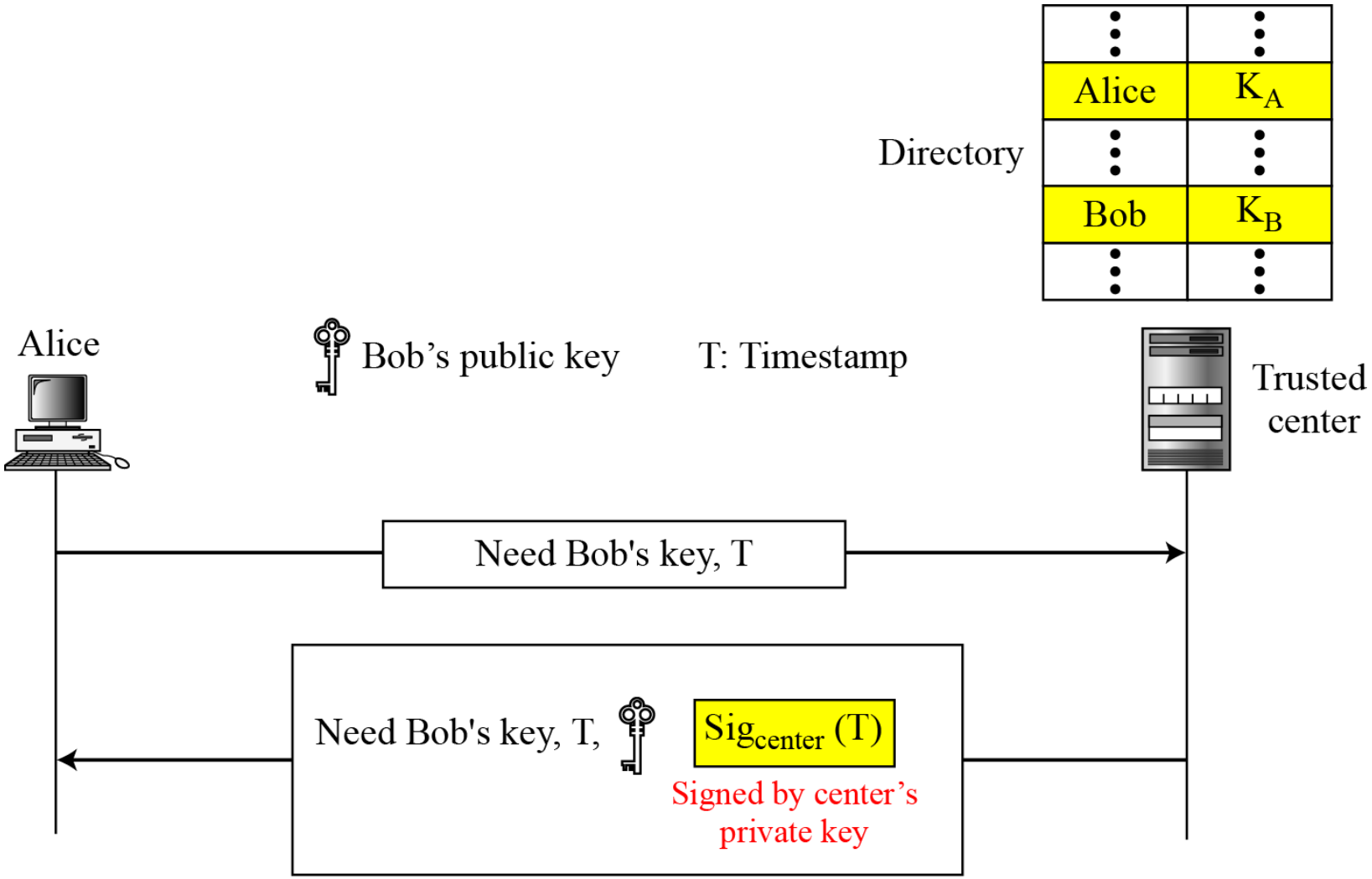
15.4.2 Trusted Center

Figure 15.14 *Trusted center*



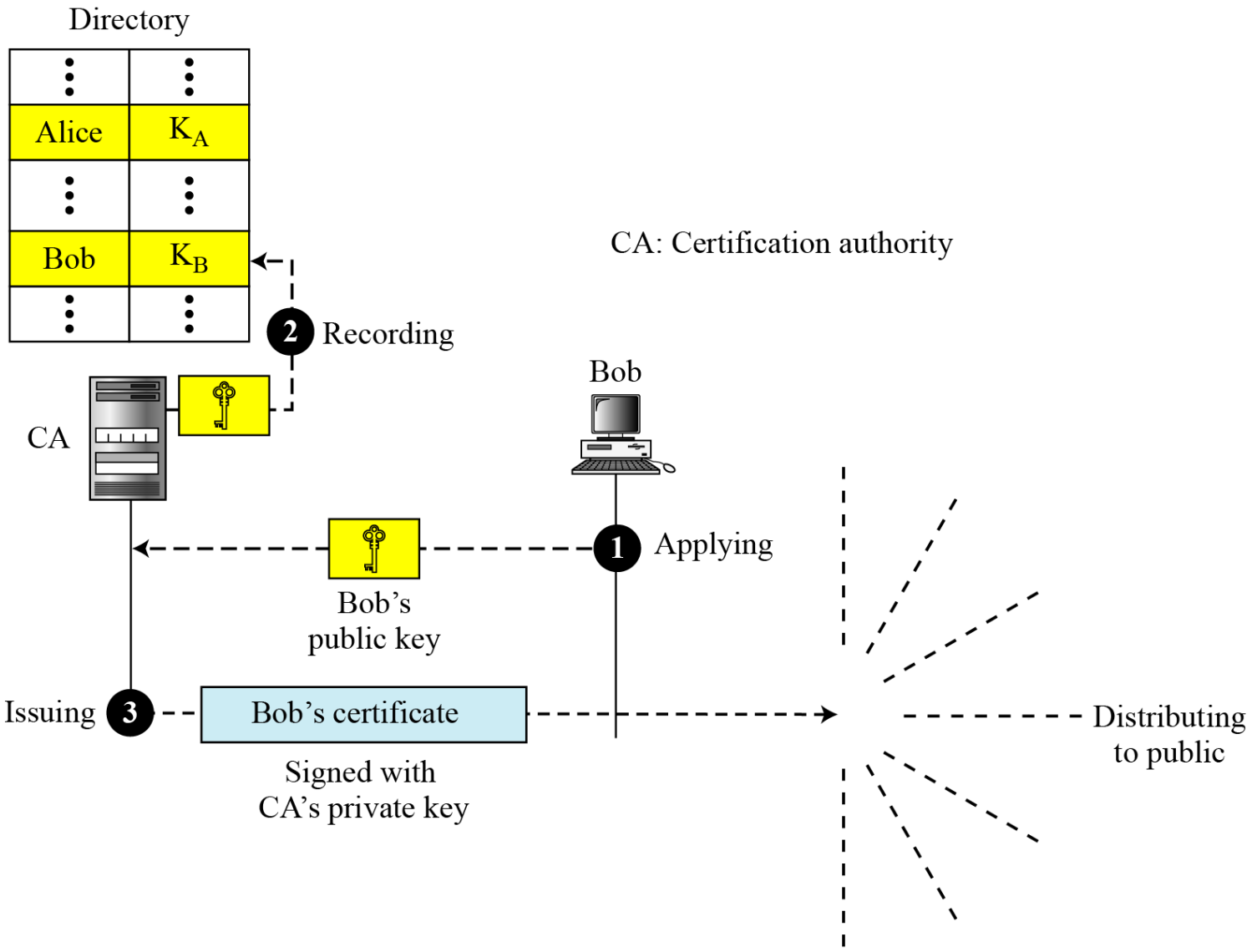
15.4.3 Controlled Trusted Center

Figure 15.15 *Controlled trusted center*



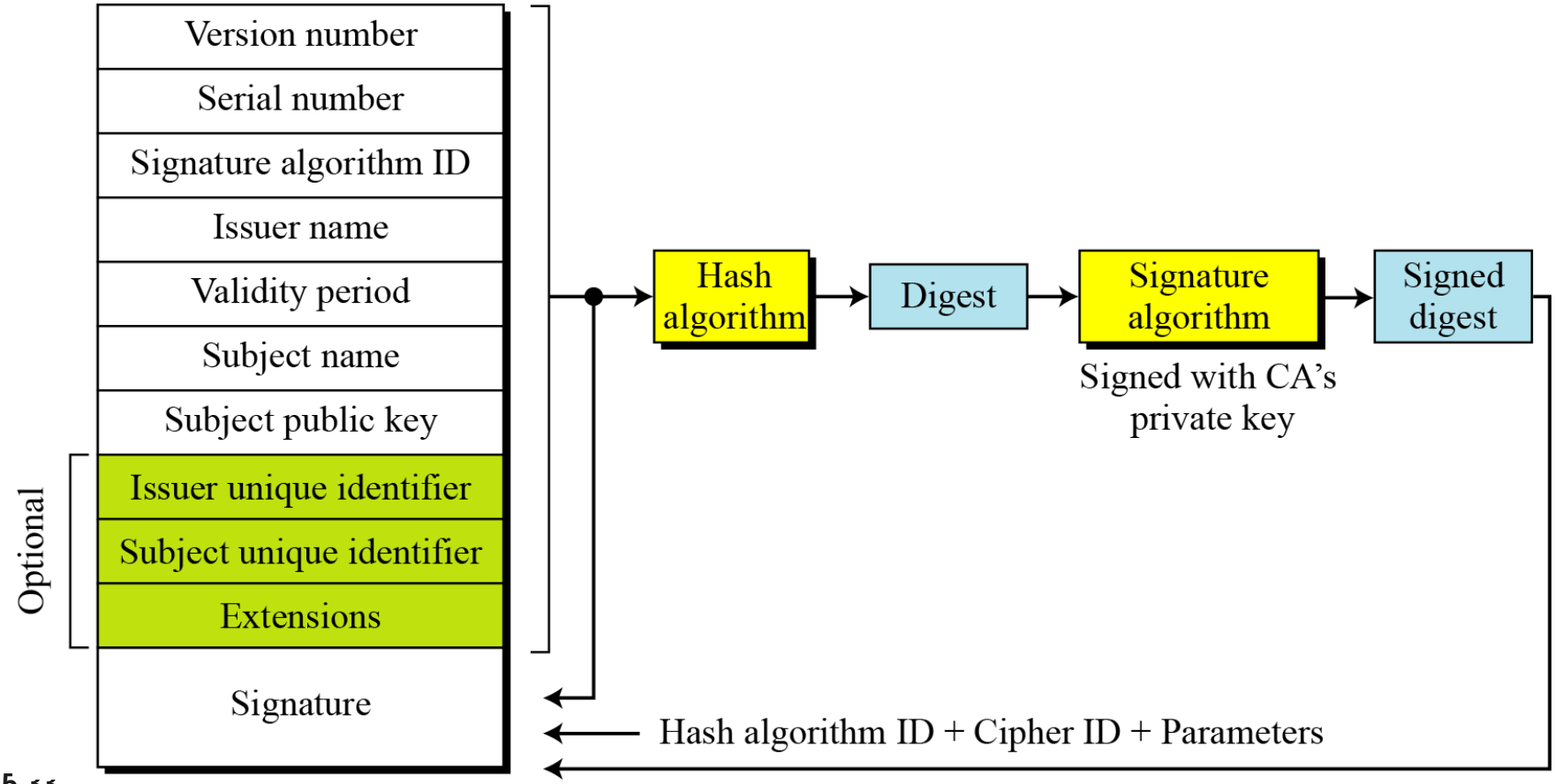
15.4.4 Certification Authority

Figure 15.16 *Certification authority*



Certificate

Figure 15.17 shows the format of a certificate.



15.4.5 Continued

Certificate Renewal

Each certificate has a period of validity. If there is no problem with the certificate, the CA issues a new certificate before the old one expires.

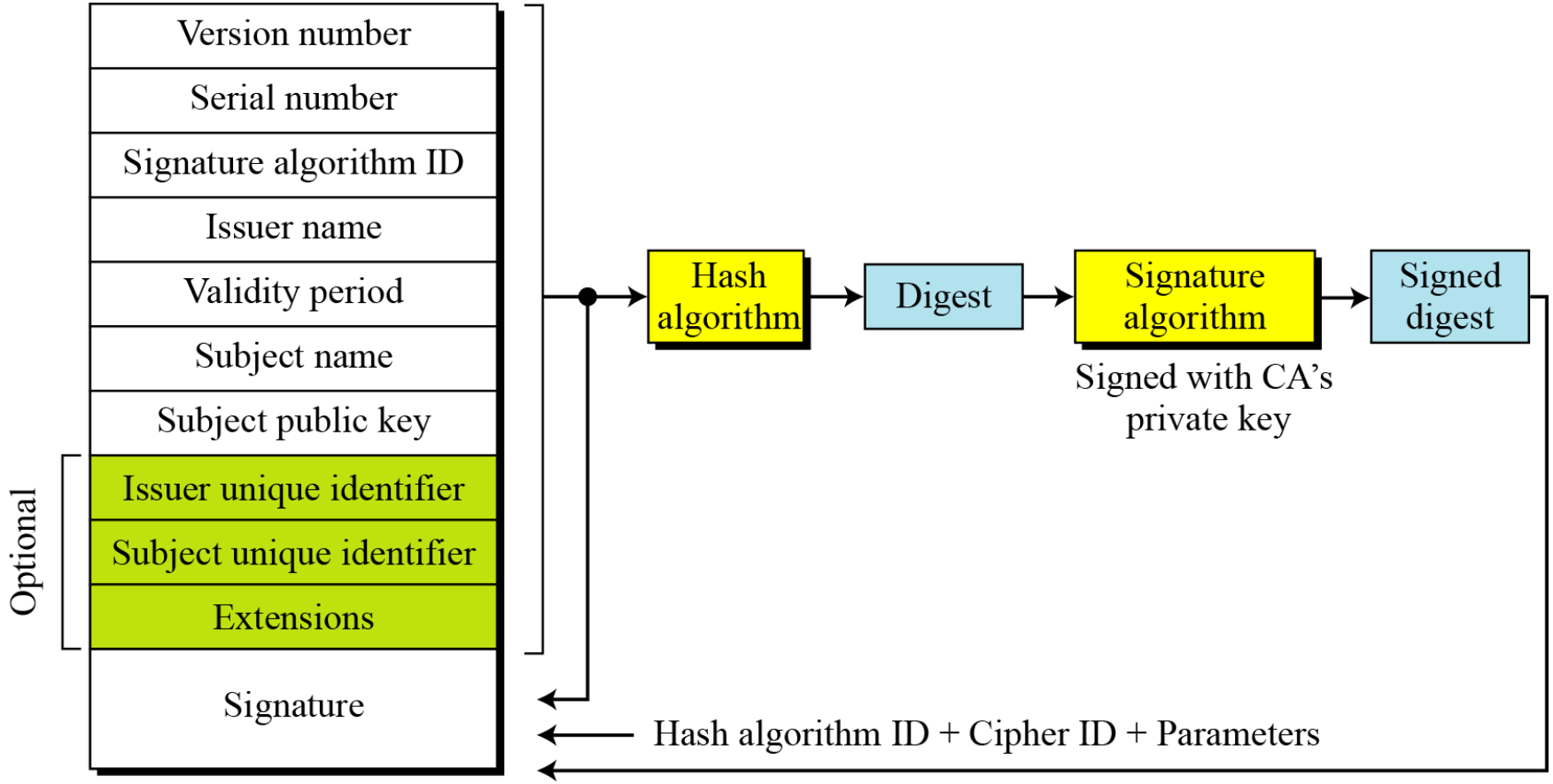
Certificate Renewal

In some cases a certificate must be revoked before its expiration.

Delta Revocation

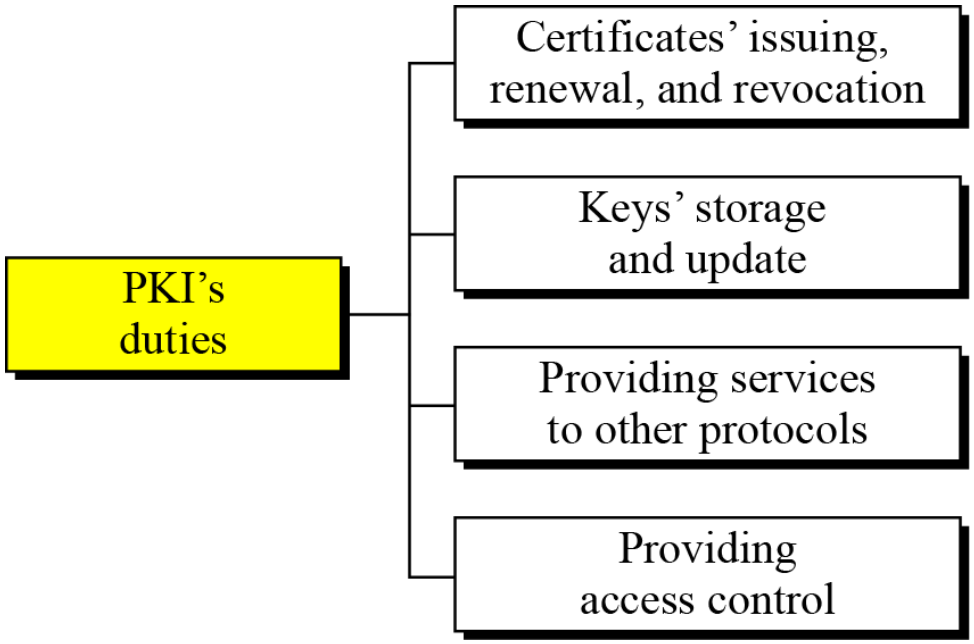
To make revocation more efficient, the delta certificate revocation list (delta CRL) has been introduced.

Figure 15.17 *Certificate revocation format*



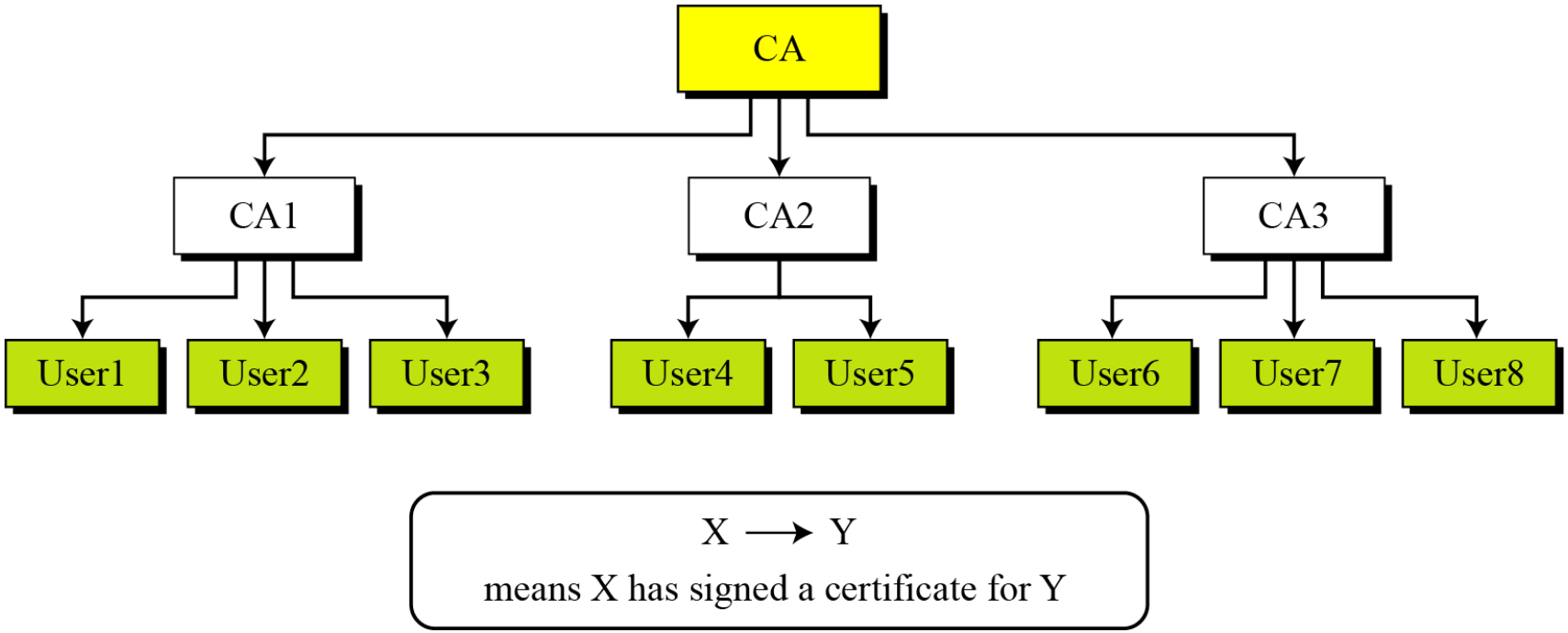
15.4.6 Public-Key Infrastructures (PKI)

Figure 15.19 *Some duties of a PKI*



Trust Model

Figure 15.20 *PKI hierarchical model*



15.4.6 Continued

Example 15.3

Show how User1, knowing only the public key of the CA (the root), can obtain a verified copy of User3's public key.

Solution

User3 sends a chain of certificates, CA<<CA1>> and CA1<<User3>>, to User1.

a. User1 validates CA<<CA1>> using the public key of CA.
b. User1 extracts the public key of CA1 from CA<<CA1>>.
c. User1 validates CA1<<User3>> using the public key of CA1.
d. User1 extracts the public key of User 3 from CA1<<User3>>.