Lecture Plan

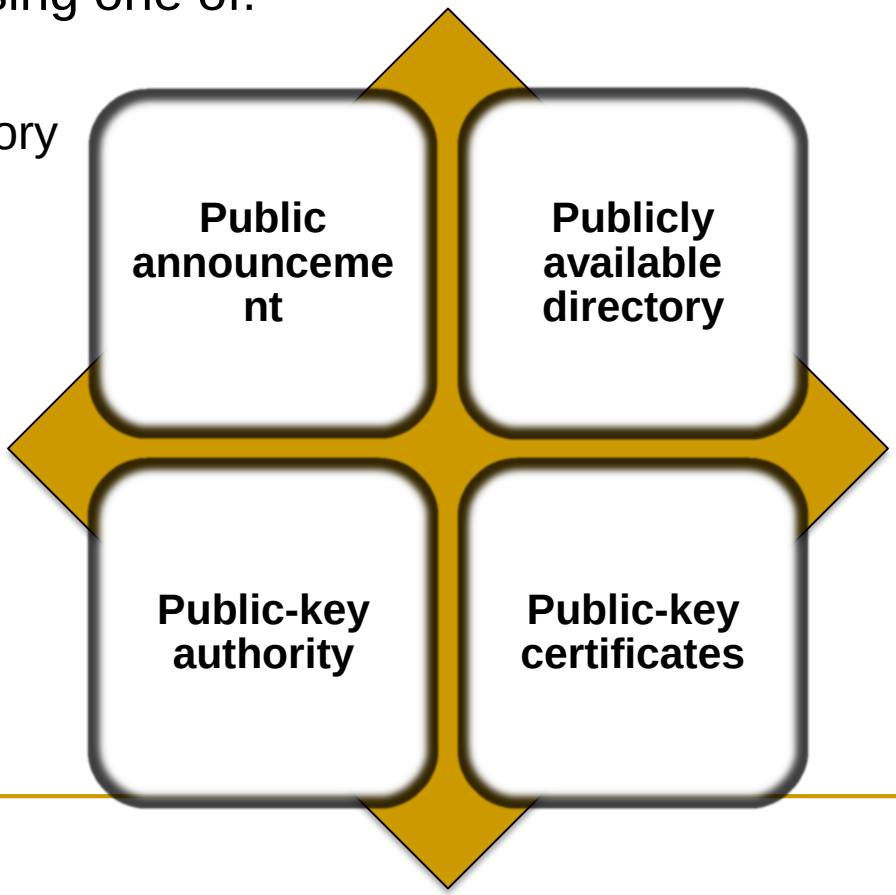
- Distribution of Public Keys
- Public-Key Authority
- Certificate Management and Public Key Infrastructure.

What is the main problem?

- Public key schemes allow two entities to securely communicate over public channel without having to meet in advance.
- How do they obtain public keys?
- Public-Key Authority
- Certificate Management and Public Key Infrastructure.

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
- can be considered as using one of:
  - public announcement
  - publicly available directory
  - public-key authority
  - public-key certificates



Notation

- We use the conventions associated with RSA schemes while explaining public key protocols.
- Public Address: PU Private Address: PR
- Public Key Encryption/Decryption:
  - Encryption: E(PU,M) = C;
  - Decryption: M = E(PR,C)
- Public Key Signature/Verification
- Signing:
  - s = E(PR,M); (M,s) is a signature pair
- Verification
  - M eq E(PU,s)?

NOTE: the notation E(key, message) is used for symmetric key encryption also; the meaning depends on the context.

Public Announcement

- Users distribute public keys to recipients or broadcast to community at large
  - eg. append PGP keys to email messages or post to news groups or email list
- Major weakness is forgery
  - anyone can create a key claiming to be someone else and broadcast it
  - until forgery is discovered can masquerade as claimed user

Figure 14.10 Uncontrolled Public Key Distribution

Publicly Available Directory

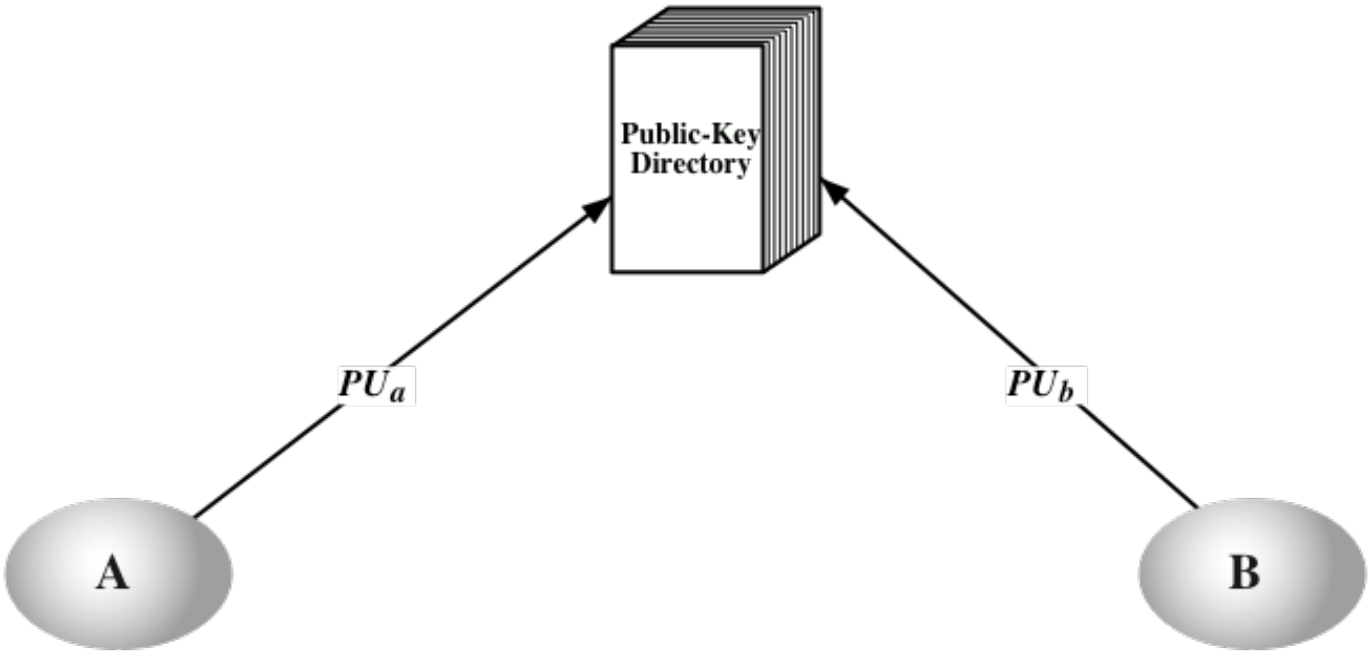


Figure 14.11 Public Key Publication

Publicly Available Directory

- Can obtain greater security by registering keys with a public directory
- Directory must be trusted with properties:
  - contains {name,public-key} entries
  - participants register securely with directory
  - participants can replace key at any time
  - directory is periodically published
  - directory can be accessed electronically
- Still vulnerable to tampering or forgery

Public-Key Authority

Next level of Improvement:

- improve security by tightening control over distribution of keys from directory
- has properties of directory
- and requires users to know public key for the directory
- then users interact with directory to obtain any desired public key securely
  - does require real-time access to directory when keys are needed
  - may be vulnerable to tampering

Public-Key Authority: A simple scenario

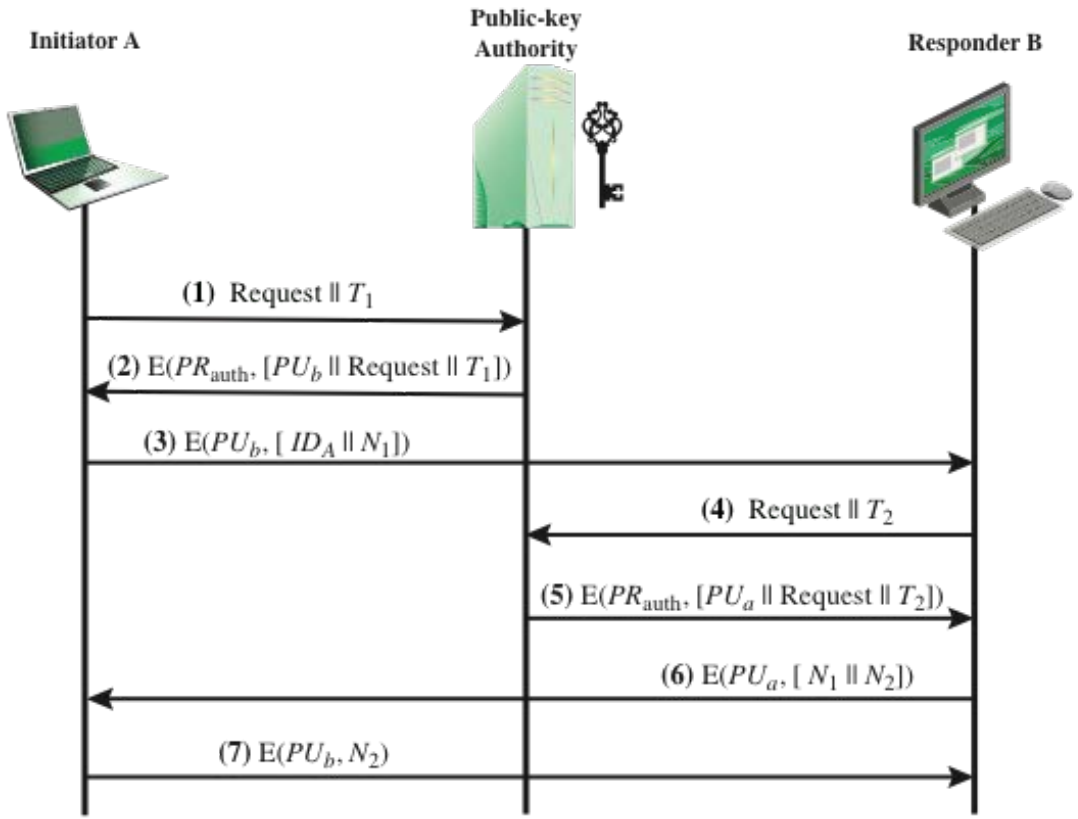
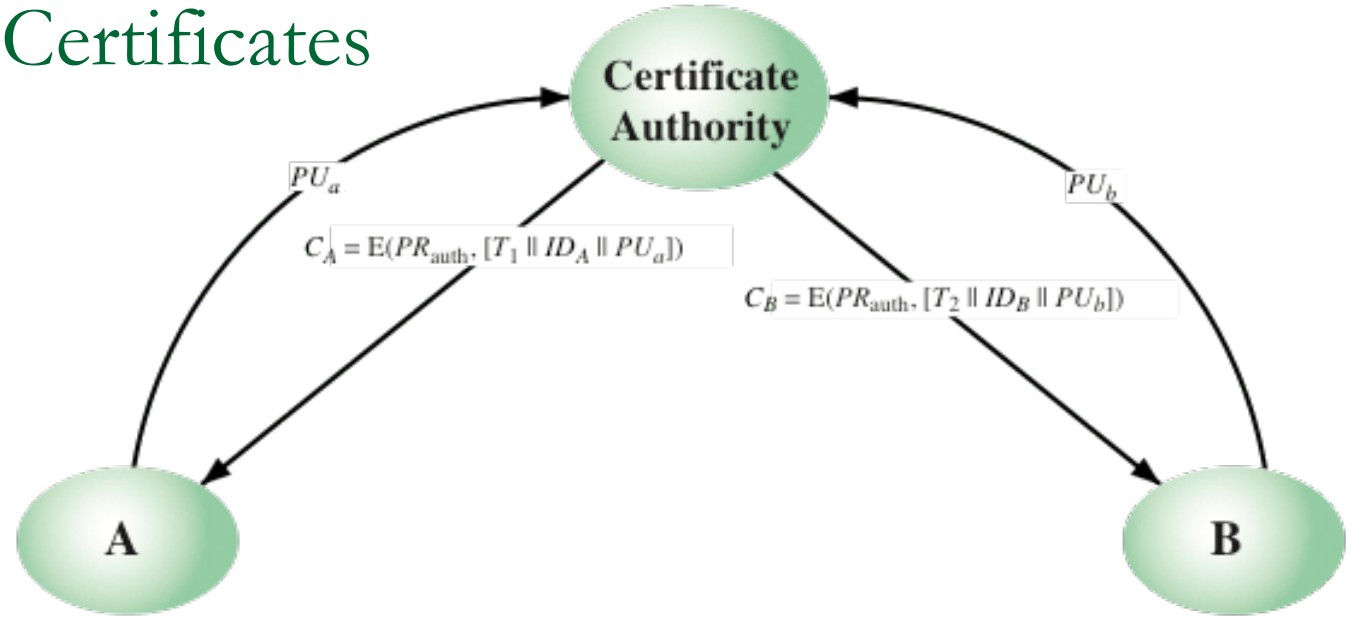


Figure 14.12 Public-Key Distribution Scenario

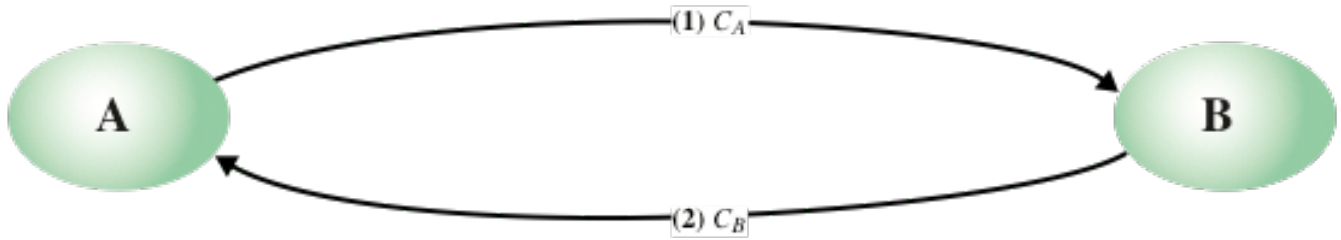
Public-Key Certificates: Another Improvement

- certificates allow key exchange without real-time access to public-key authority
- a certificate binds **identity** to **public key**
  - usually with other info such as period of validity, rights of use etc
- with all contents **signed** by a trusted Public-Key or Certificate Authority (CA)
- can be verified by anyone who knows the public-key authorities public-key

Public-Key Certificates



(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

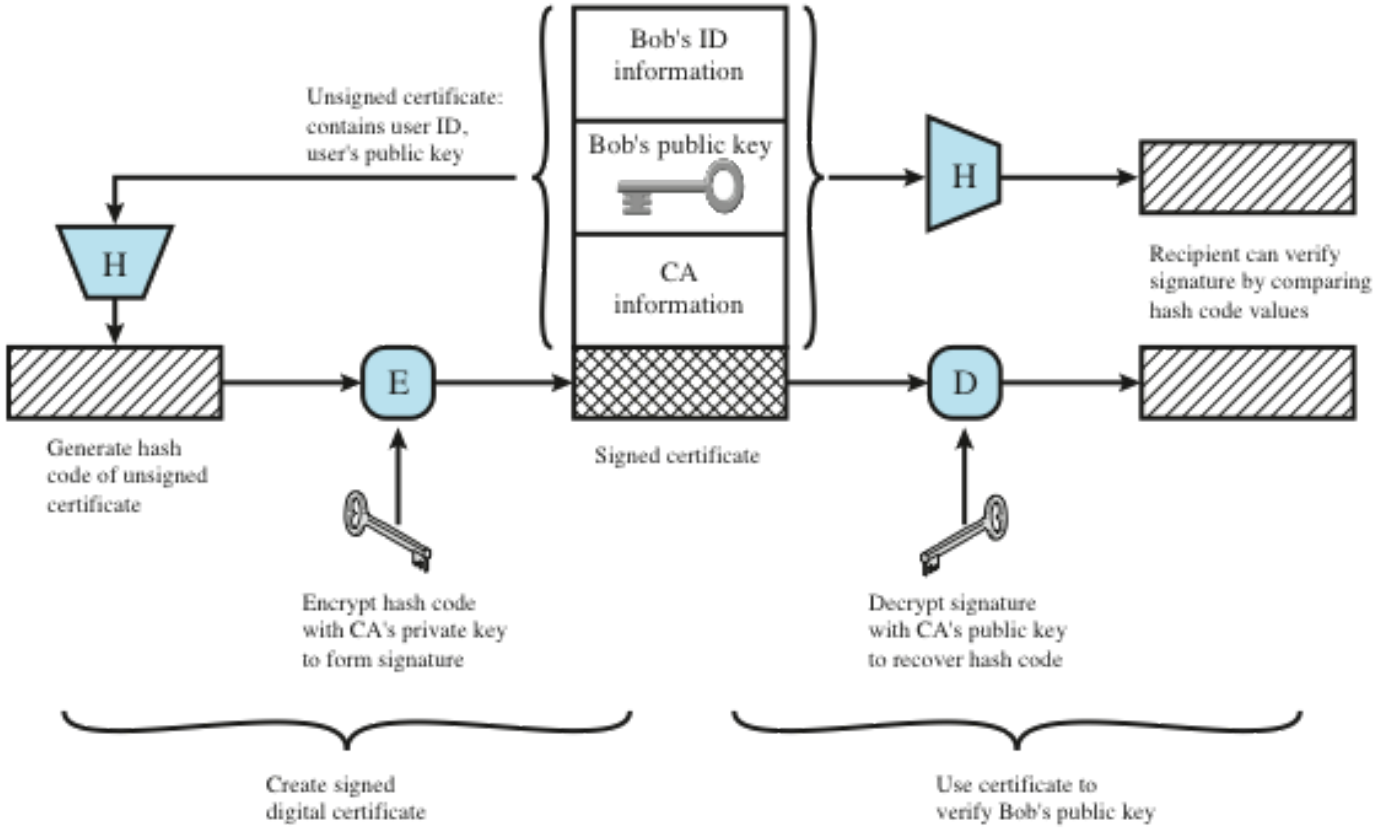
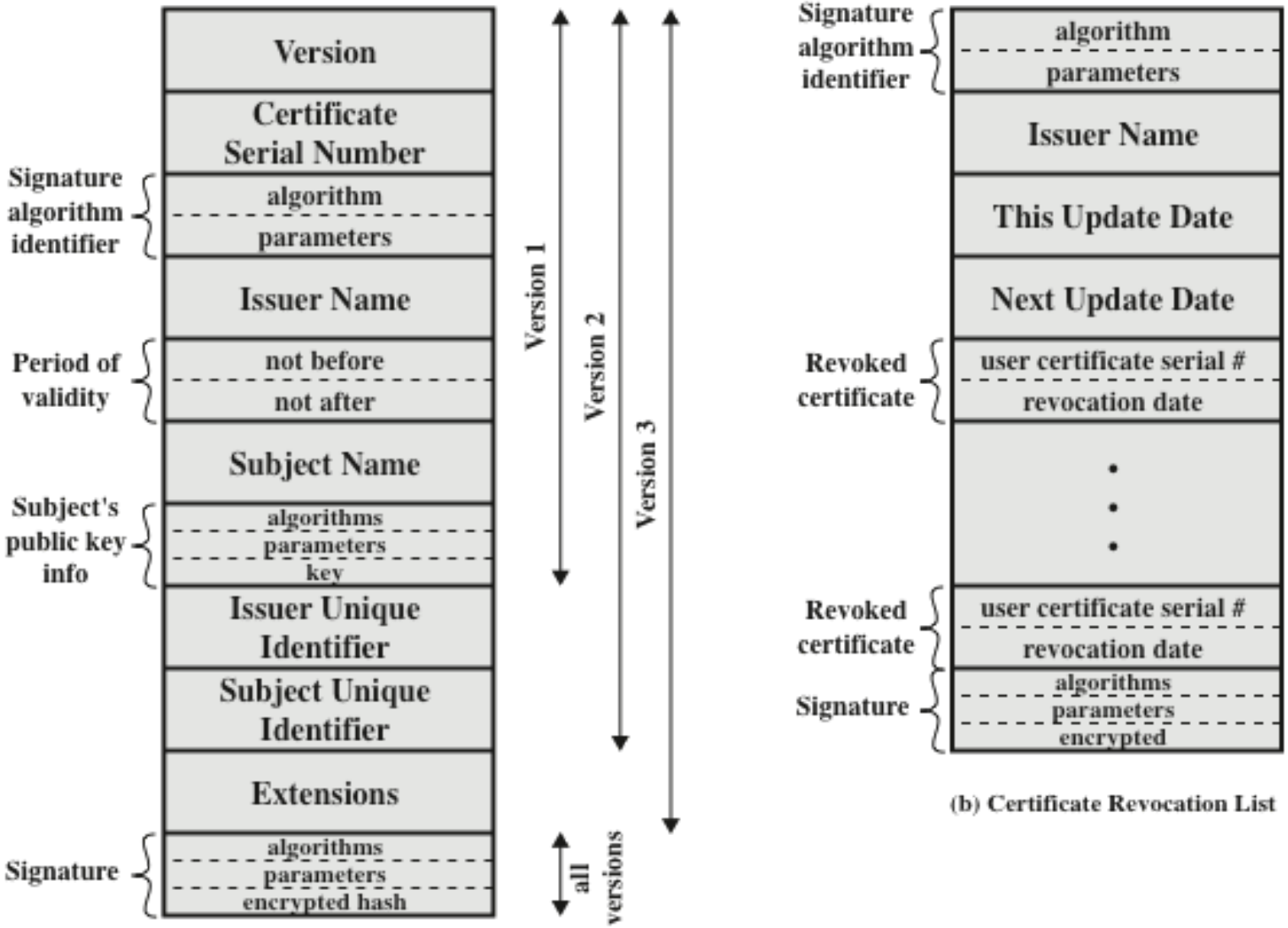


Figure 14.14 Public-Key Certificate Use

X.509 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

X.509 Certificates



(a) X.509 Certificate

(b) Certificate Revocation List

Obtaining a Certificate

User certificates generated by a CA have the following characteristics:

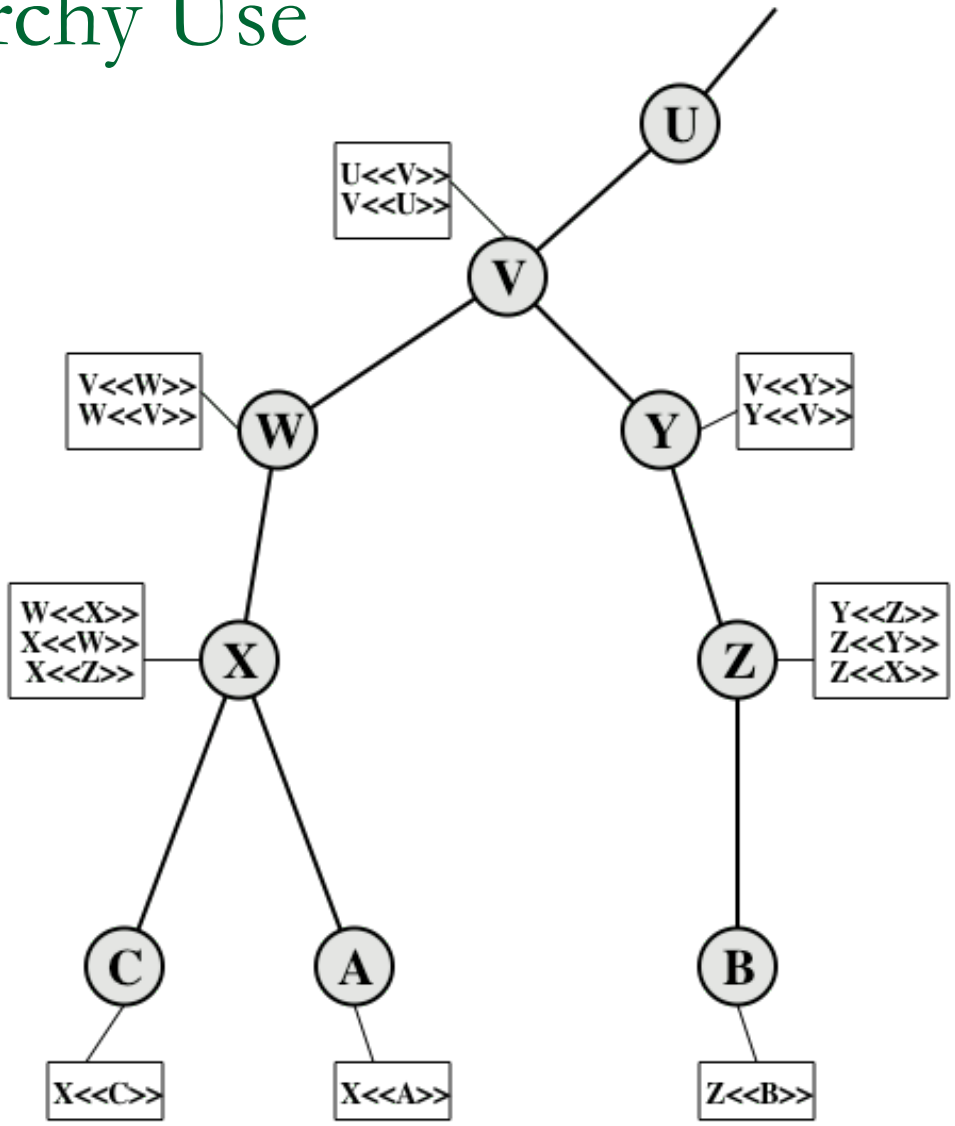
- Any user with access to the public key of the CA can verify the user public key that was certified
- No party other than the certification authority can modify the certificate without this being detected

- 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

CA Hierarchy

- if both users share a common CA then they are assumed to know its public key
- otherwise CA's must form a hierarchy
- use certificates linking members of hierarchy to validate other CA's
  - each CA has certificates for clients (forward) and parent (backward)
- each client trusts parents certificates
- enable verification of any certificate from one CA by users of all other CAs in hierarchy

CA Hierarchy Use



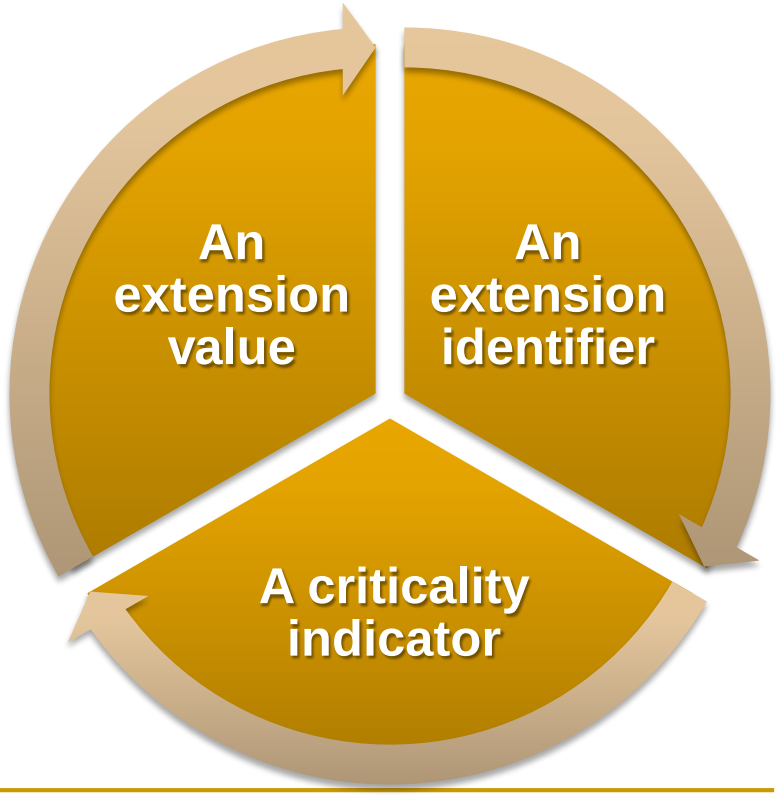
Certificate Revocation

- 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



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

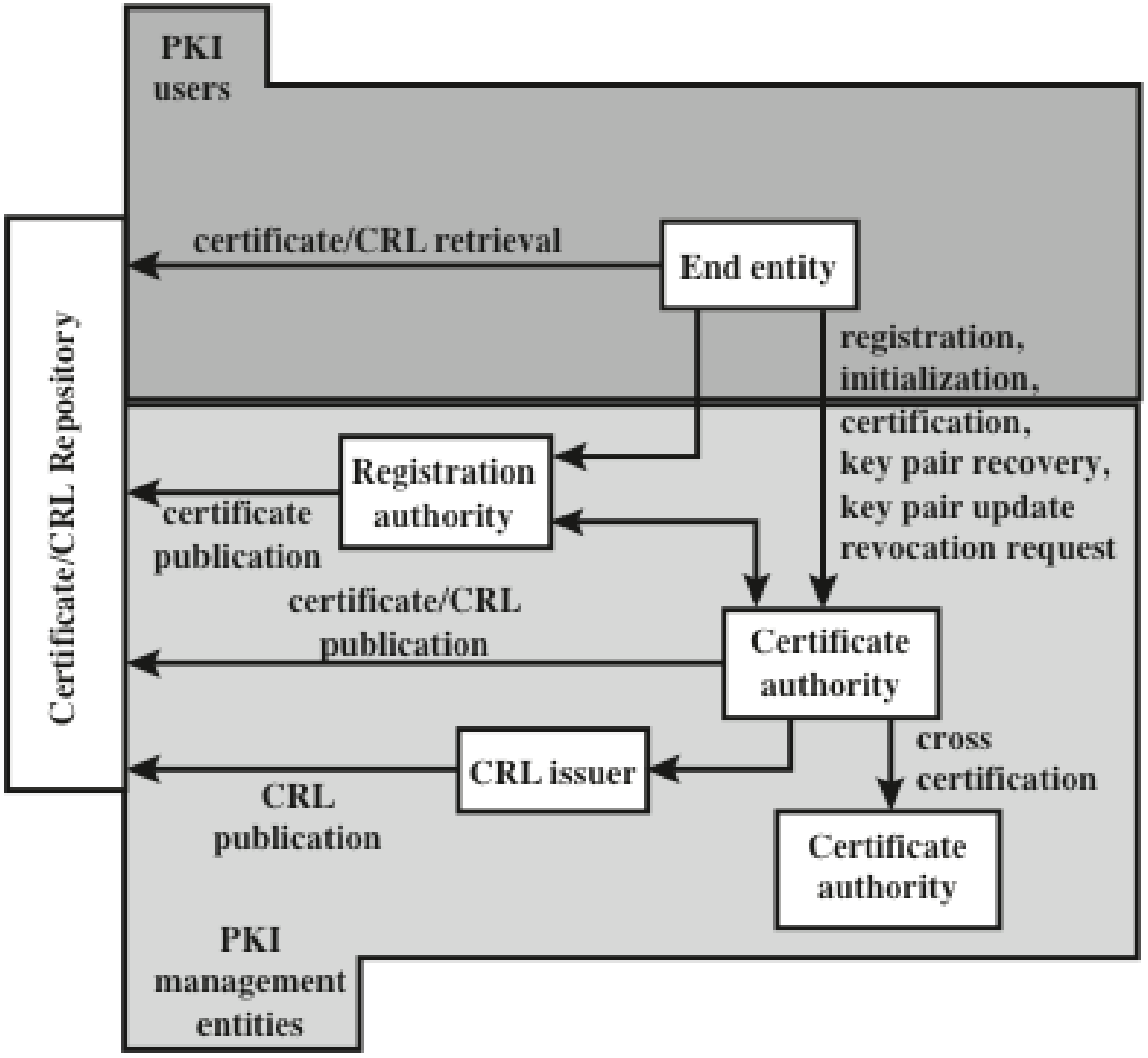


Figure 14.17 PKIX Architectural Model

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

- We Have considered:
  - distribution of public keys
    - announcement, directory, Public Key authrority, Certificate Authority
  - X.509 authentication and certificates
  - public key infrastructure (PKIX)