Section 14.2:
Symmetric Key Distribution Using Asymmetric Encryption

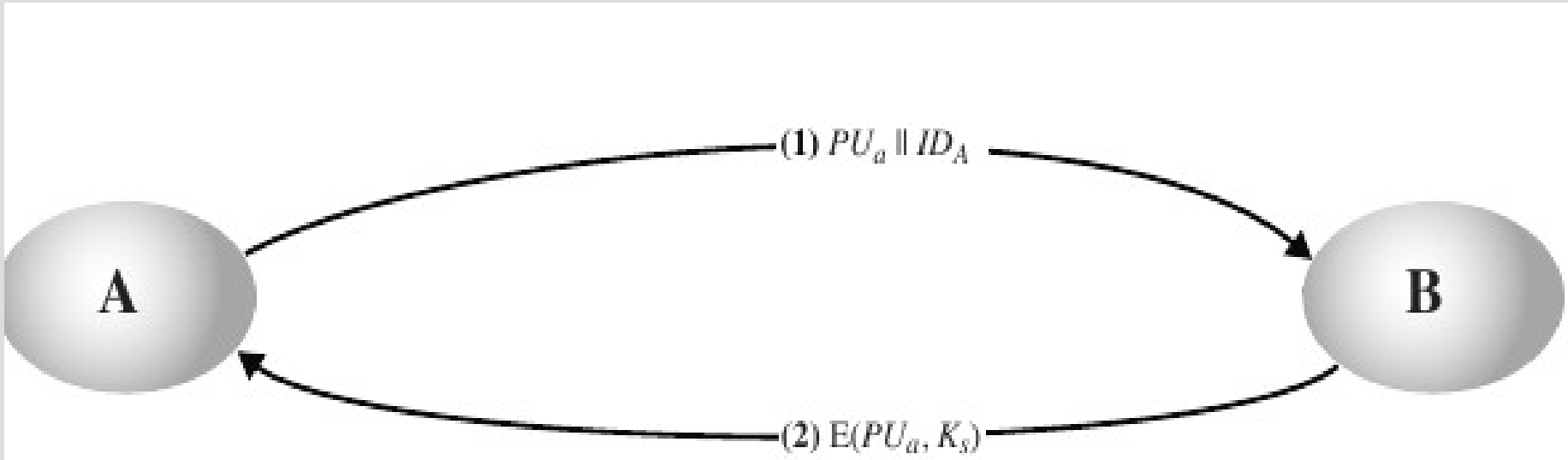


Figure 14.7 Simple Use of Public-Key Encryption to Establish a Session Key

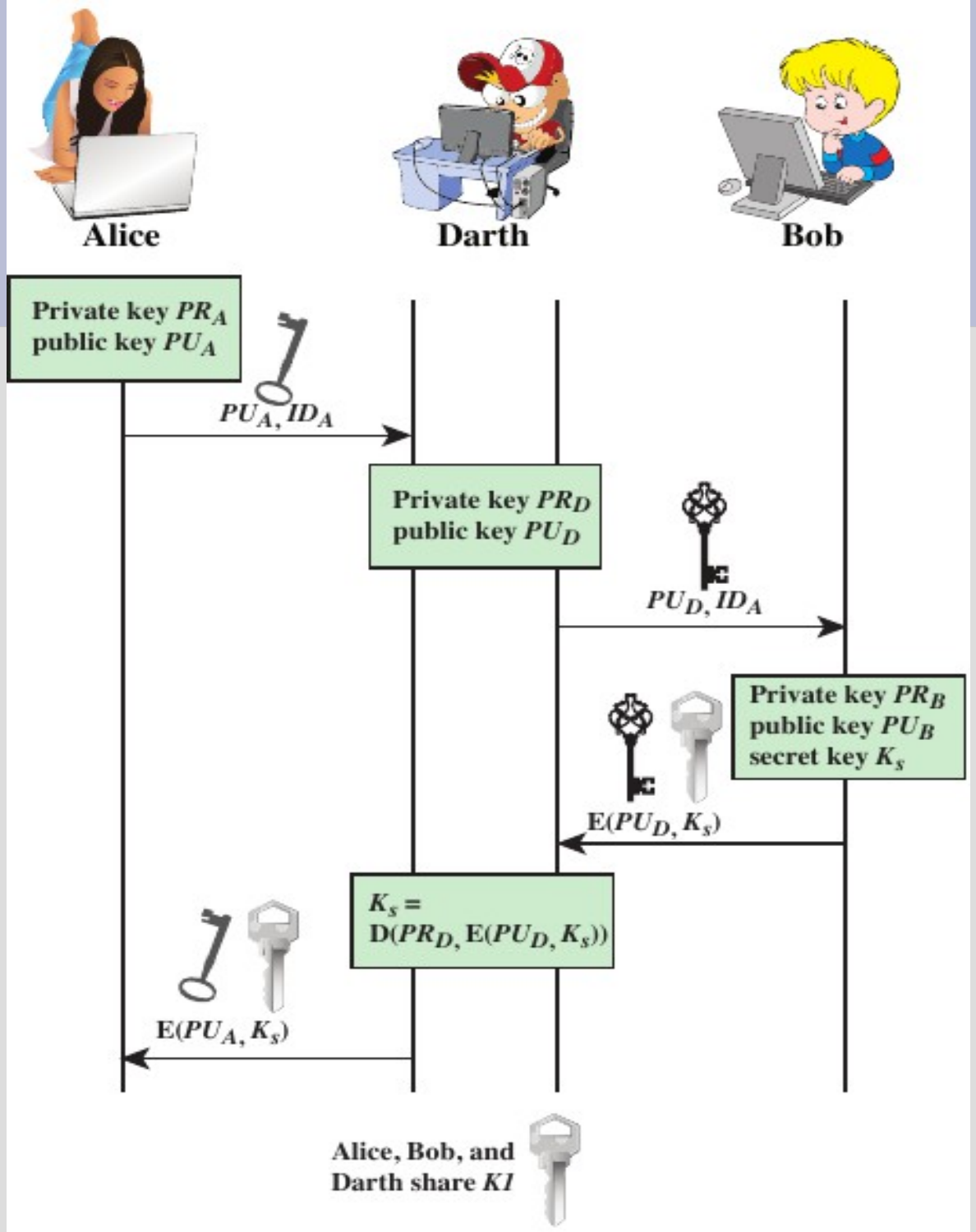


Figure 14.8 Another Man-in-the-Middle Attack

Secret Key Distribution with Confidentiality and Authentication

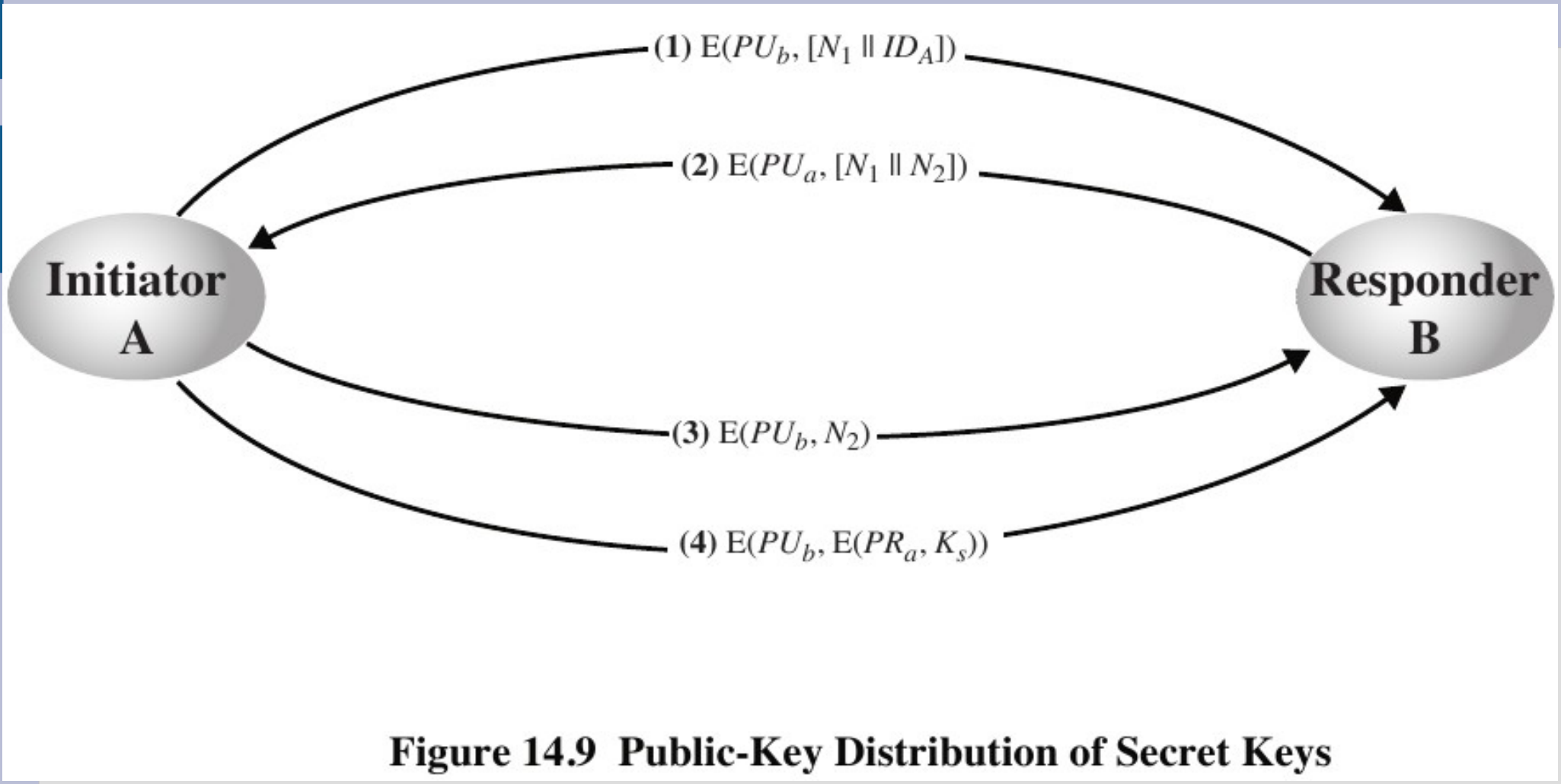


Figure 14.9 Public-Key Distribution of Secret Keys

Protocol Reference: Using Encryption for Authentication in Large Networks of Computers by Needham and Schroeder

Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR by Lowe

Section 14.3: Distribution of Public Keys

Figure 14.10 Uncontrolled Public Key Distribution

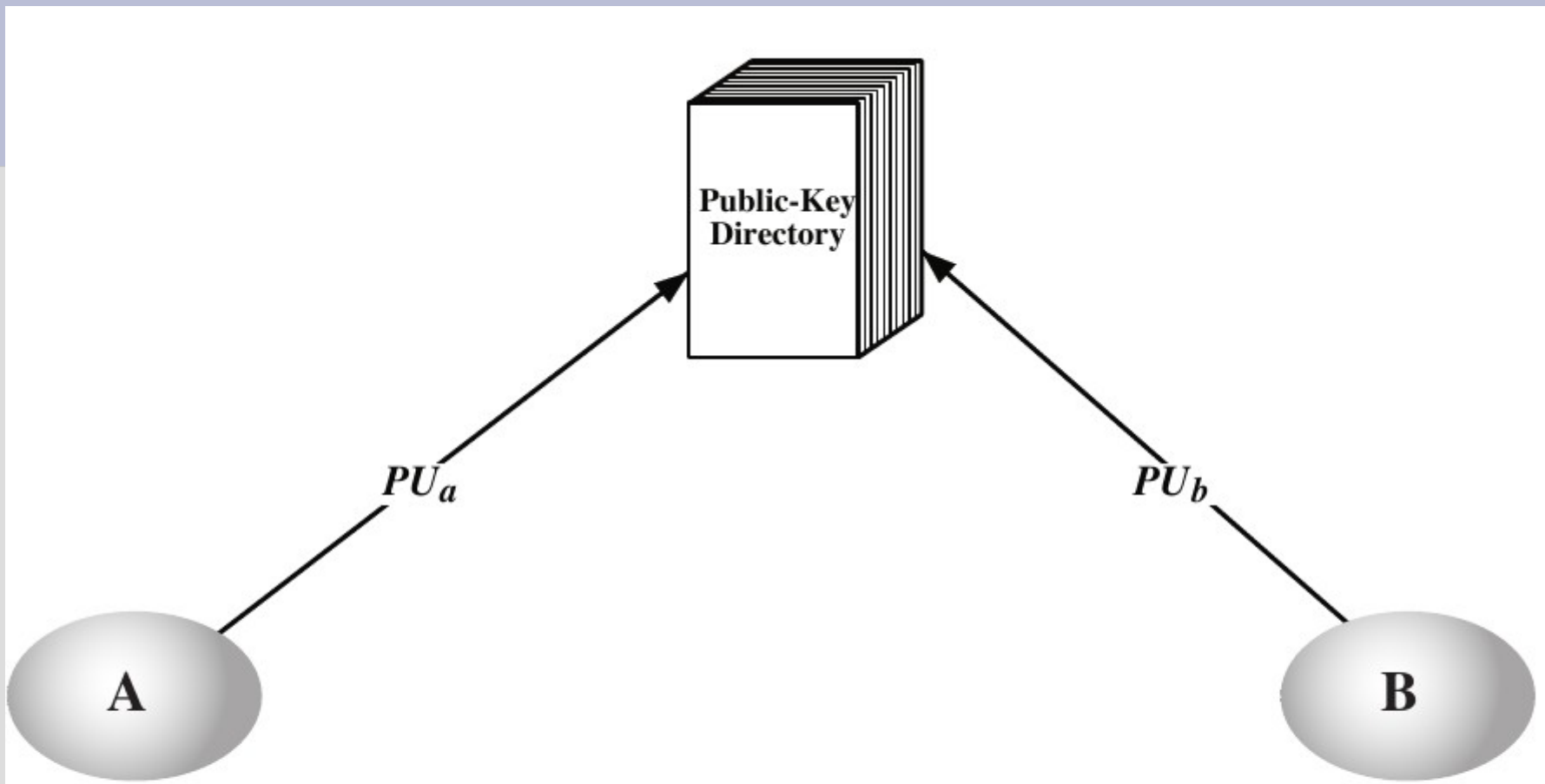


Figure 14.11 Public Key Publication

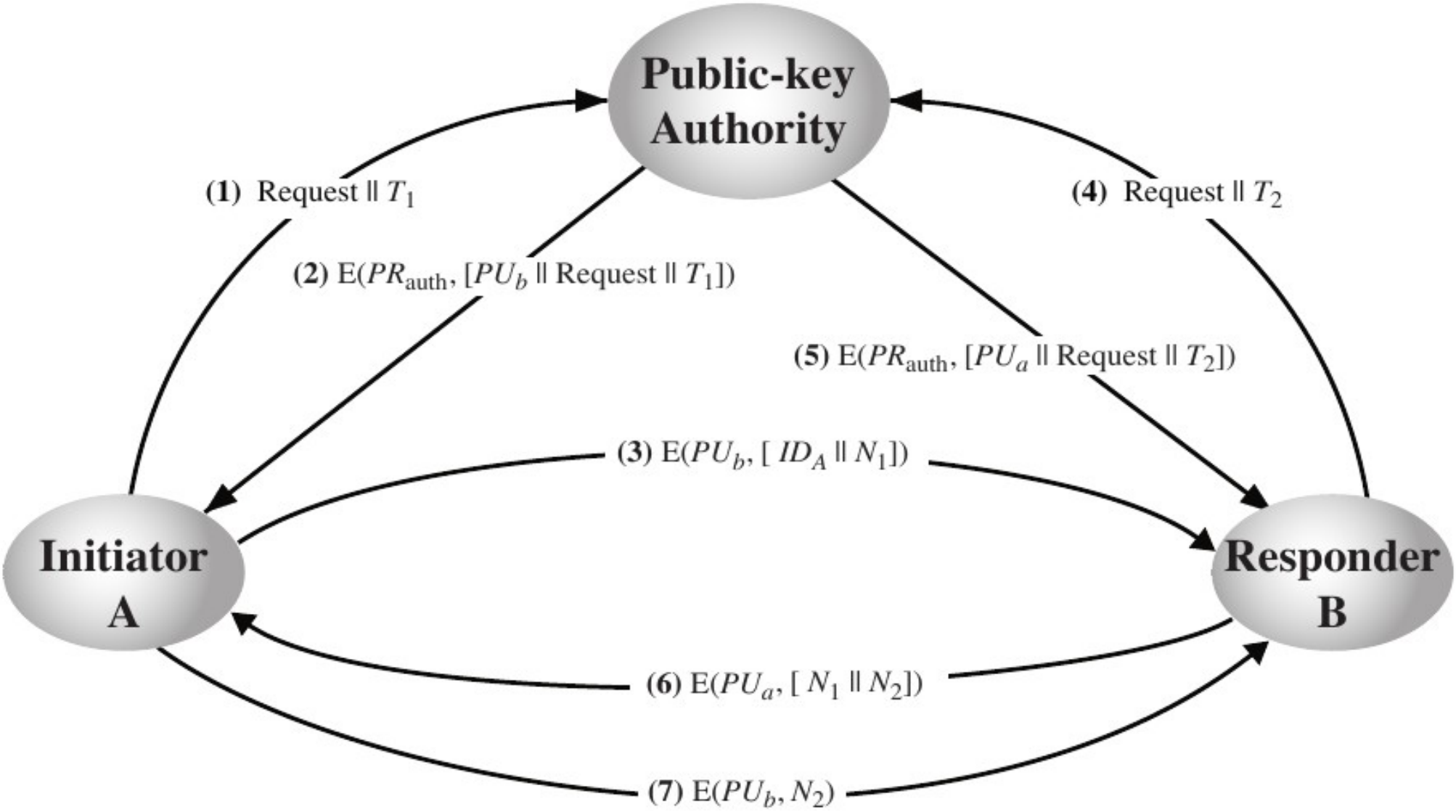


Figure 14.12 Public-Key Distribution Scenario

Section 14.4:
X.509 Sertificates

X.509 Authentication Service

- X.509 is part of X.500 series which defines a directory service
- 1988, V2-1993, V3-1995
- Based on public-key cryptography and digital signatures
- Defines a framework for the provision of authentication services
- Repository of public key certificates
- Used in S/MIME, IPSec, SSL and SET

X.509 Authentication Service

- Distributed set of servers that maintains a database about users.
- Each certificate contains the *public key of a user* and is signed with the *private key of a trusted certification authority (CA).*
- *A certificate is associated with each user*
- Is used in S/MIME, IP Security, SSL/TLS and SET.
- RSA is recommended to use.

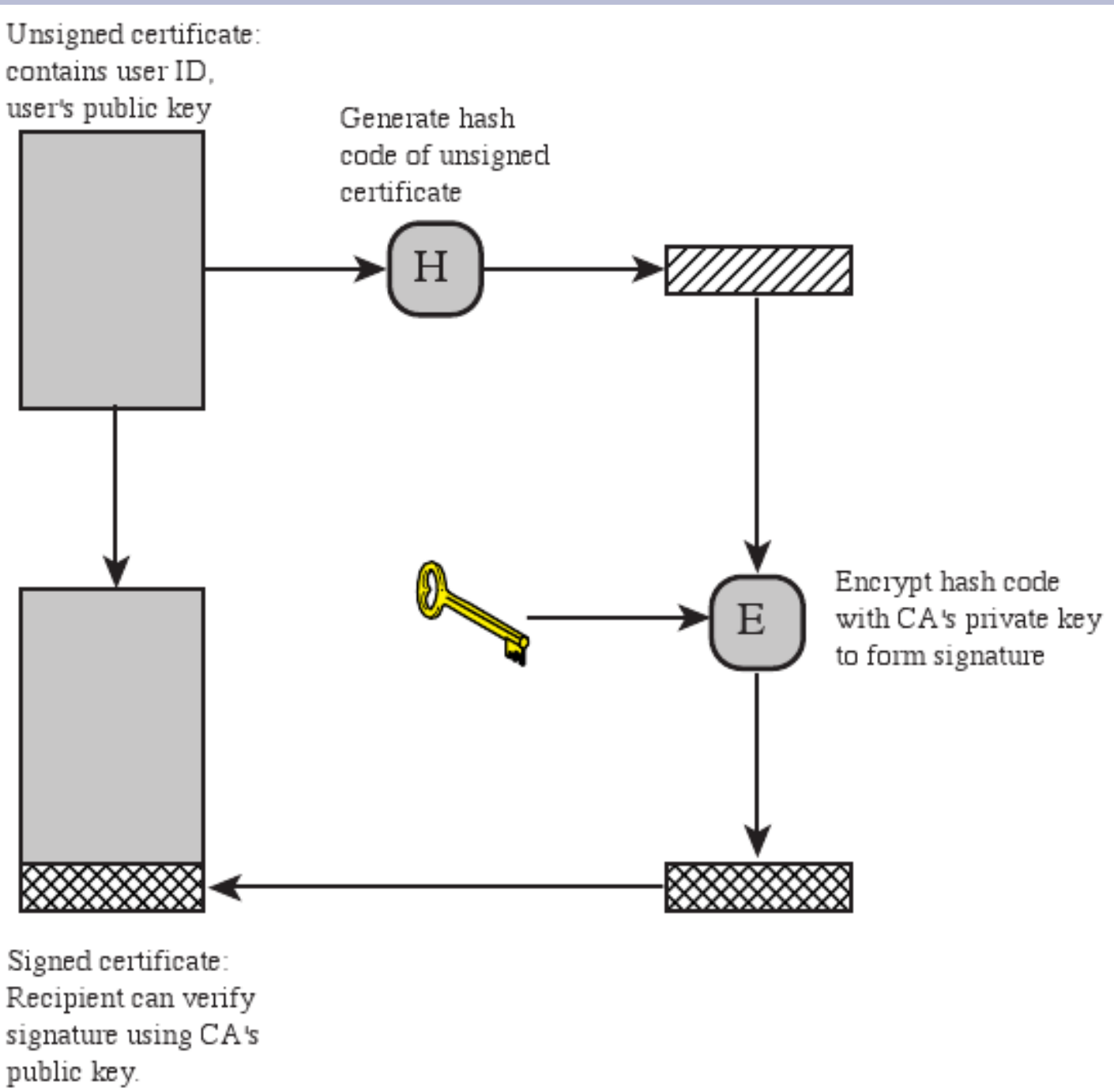


Figure 4.3 Public-Key Certificate Use

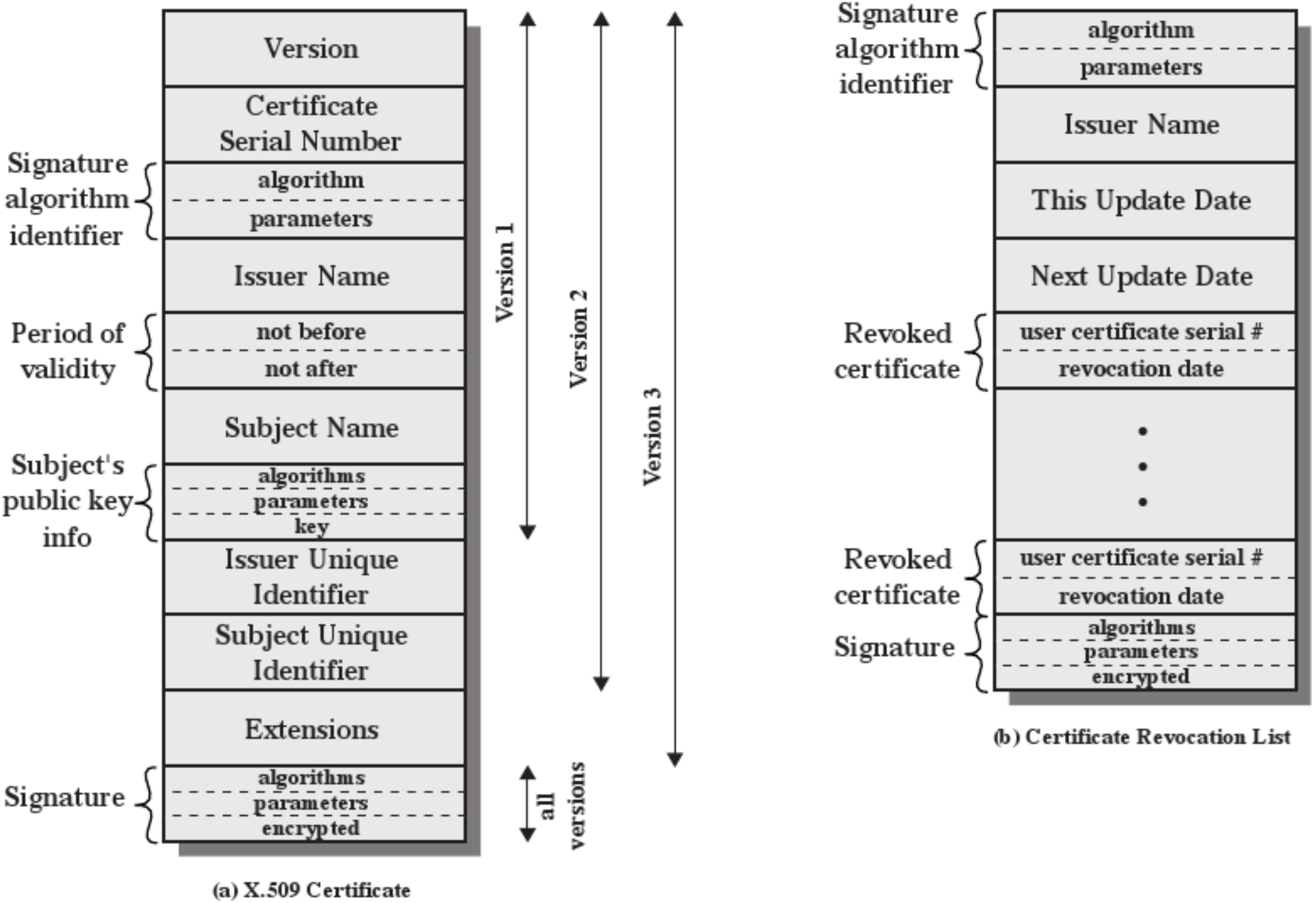


Figure 4.4 X.509 Formats

X.509 Certificate

- **Version**: 1,2 or 3.
- **Serial Number**: Unique number given by the issuing CA
- **Signature algorithm identifier**: The algorithm used to sign the certificate with any associated parameters.
- **Issuer Name**: X.509 name of the CA
- **Period of validity**: The first and the last date that the certificate valid
- **Subject name**: The name of the certificate owner. This certificate certifies the public key of the subject (owner) .
- **Subject's public-key information**: Subject's public key, identifier for the algorithm, key with any associated parameters.

X.509 Certificate

- **Issuer unique identifier**: (optional bit string field for uniquely the issuing CA -X.509 name-)
- **Subject unique identifier**: (optional bit string field for uniquely the subjects -X.509 name-)
- **Extensions**: A set of one or more extension fields added for version 3
- **Signature**: Hash code of other fields that encrypted with the CA's private key. The signature algorithm identifier included.

Certificate Notation

Y{I} = the signing of I by Y

$$CA\langle\langle A\rangle\rangle = CA\ \{V, SN, AI, CA, T_A, A, A_P\}$$

certificate of user A issued by certification authority CA

encrypted hash code

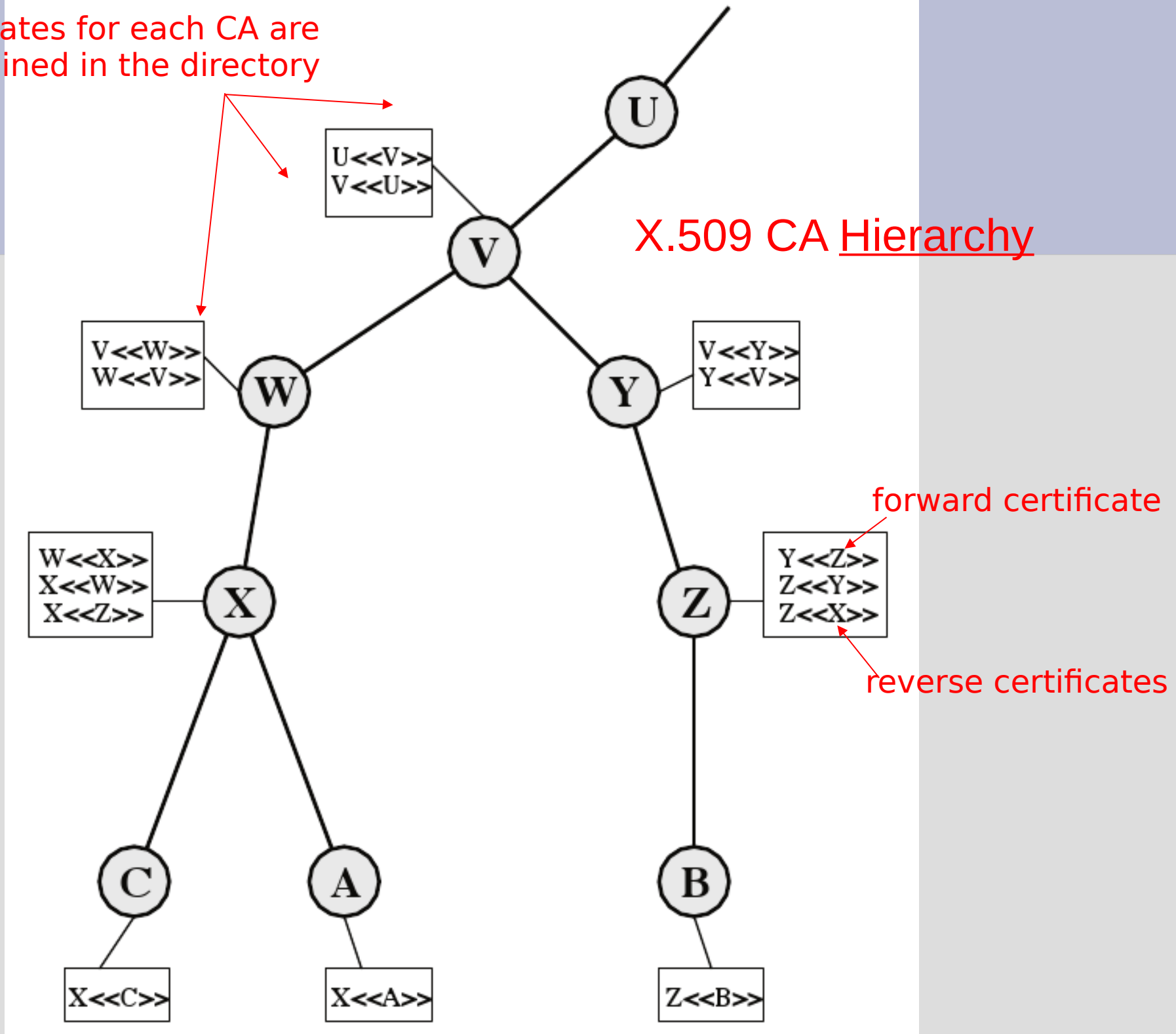
Certificate Characteristics

- If you have the public key of the CA, you can recover the user public key that was certified
- Only the certificate authority can modify the certificate
- Placed in a directory without special protection

Certificate Characteristics

- If all users subscribe to the same CA, then there is common trust of that CA
- User can transmit his certificate directly to others
- Assured messages are secure from eavesdropping and unforgeable
- Not all users can subscribe to the same CA

certificates for each CA are maintained in the directory



X.509 CA Hierarchy

forward certificate

reverse certificates

A Chain of Certificates

- Assume that A (respectively B) has obtained a certificate from CA X_1 (respectively CA X_2). Note that A does not know X_2's public key securely. A needs that key to verify B's certificate issued by X_2.

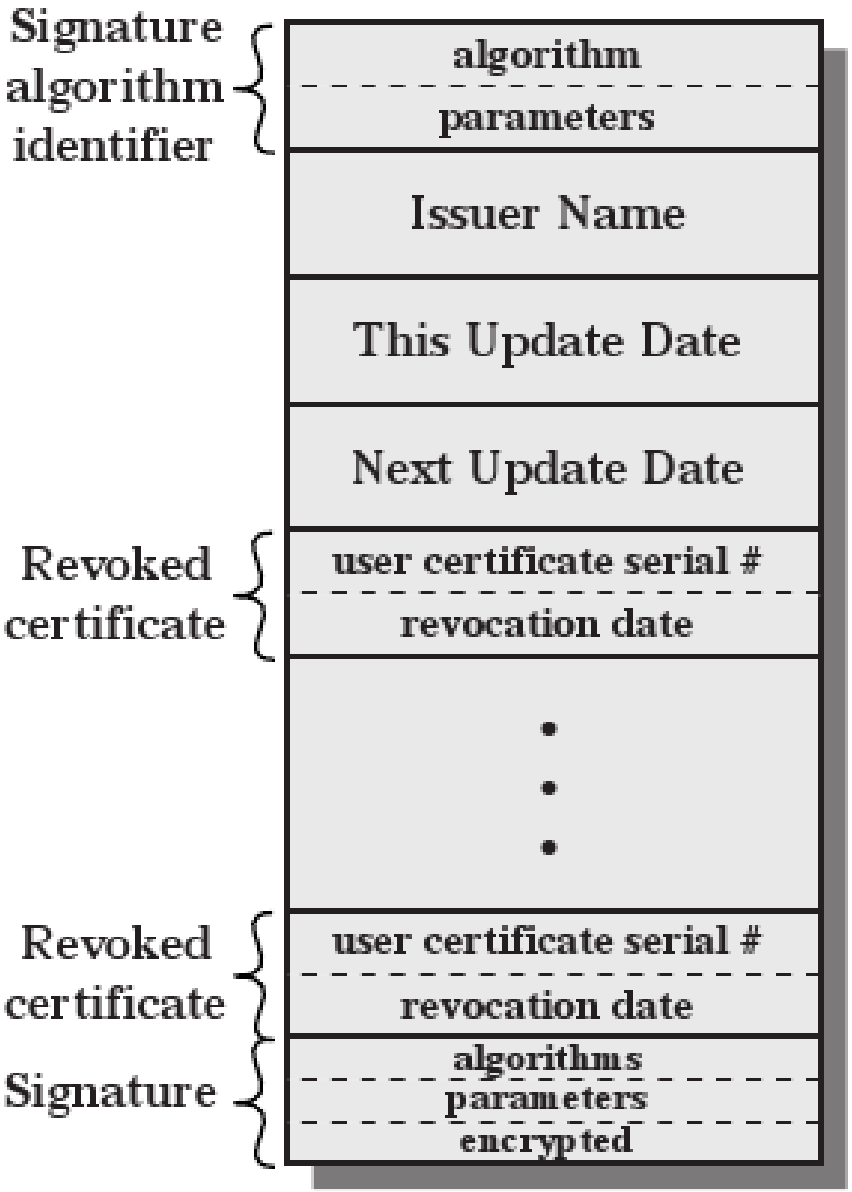
- **Solution**: Two CAs, X_1 and X_2 exchange their own public keys securely (sign each other certificate) and A has used a chain of certificates to obtain B's public key. Considering X.509 notation:
  - X_1 <<X_2 >> X_2 <<B>>
  - Similarly B has the following chain:
    X_2 <<X_1 >> X_1 <<A>>

    Homework: Study the example in page 117 (See Figure 4.5)

Revocation of Certificates

- Certificates have a period of validity
- Certificates can also be revoked because:
  - user's key is compromised
  - user no longer certified by CA
  - CA's certificate is assumed to be compromised
- CA maintains a list of revoked certificates but not expired certificated issued by that CA, and post this list on the directory.

Certificate Revocation List (CRL)



(b) Certificate Revocation List

Revocation of Certificates

- CA (issuer) signs each certificate revocation list (CRL) posted to the directory.

- Each CRL includes
  - The issuer's name
  - The date the list was created
  - The date the next CRL is scheduled
  - For each revoked certificate, there is an entry that contains serial number of a certificate and its revocation date. This serial number is unique for CA.

- When user receives a certificate, user should check the directory for CRL of CA that is the issuer of this certificate.

Revocation of Certificates
Online Certificate Status Protocol (OCSP)

- The Online Certificate Status Protocol (OCSP) is an Internet protocol used for obtaining the revocation status of an X.509 digital certificate (Wikipedia).
- It was created as an alternative to CRLs, specifically addressing certain problems associated with using CRLs in a public key infrastructure (PKI).

- Compared to CRLs:
  - Since an OCSP response contains less information than a typical CRL, OCSP can feasibly provide more **timely** information regarding the revocation status of a certificate without burdening the network.
  - CRLs updated in fix time interval (daily, weekly etc.)
  - CRLs may be seen as analogous to a credit card company's "bad customer list" – an unnecessary public exposure.

Basic PKI implementation using Online Certificate Status Protocol (OCSP)

http://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol

1. Alice and Bob have public key certificates issued by the Certificate Authority (CA).
2. Alice wishes to perform a transaction with Bob and sends him her public key certificate.
3. Bob, concerned that Alice's private key may have been compromised, creates an 'OCSP request' that contains Alice's certificate serial number and sends it to CA.
4. CA's OCSP responder looks up the revocation status of Alice's certificate (using the certificate serial number Bob informed) in his own CA database. If Alice's private key had been compromised, this is the only trusted location at which the fact would be recorded.
5. CA's OCSP responder confirms that Alice's certificate is still OK, and returns a signed, successful 'OCSP response' to Bob.
6. Bob cryptographically verifies the signed response (He has CA's public key on-hand – CA is a trusted responder) and ensures that it was produced recently.
7. Bob completes the transaction with Alice.

Basic PKI implementation using Online Certificate Status Protocol (OCSP)

http://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol

- The Online Certificate Status Protocol (OCSP) is is described in RFC 2560 (X.509 Internet Public Key Infrastructure OCSP) as an Internet protocol.

- Browser Support
  - Internet Explorer starting with version 7 on Windows Vista (not XP) supports OCSP checking
  - All versions of Firefox support OCSP checking. Firefox 3 enables OCSP checking by default.
  - Safari on Mac OS X supports OCSP checking.
  - Later versions of Opera support OCSP checking.

Authentication Procedures

- X.509 includes three authentication procedures making use of public key signatures
- Intended for a variety of applications
- Assumes two parties know each other's public key

X.509 Version 3 Requirements

- Subject field needs to convey more information about the key owner
- Subject field needs more info for applications: IP address, URL
- Indicate security policy information (IPSec)
- Set constraints on certificate applicability – limit damage from faulty CA
- Identify separately different keys used by the same owner at different times – key life cycle management

X.509 Version 3 Extensions

- Added optional extensions rather than fixed fields. Each extension consists of
  - `extension id,`
  - `criticality indicator`
  - `extension value`

- The **criticality indicator** is used to indicate whether an extension can be safely ignored or not (this is needed by application recognizes version 2)
- **Three main categories:**
  - Key and policy information
  - Certificate subject and issuer attributes – *alternative names*
  - Certification Path Constraints - *restrictions*

X.509 Version 3 Extensions

- **Key and policy information** : These extensions are used to convey **additional** information about the subject and issuer keys, plus indicators of certificate policy. Such policy can be used to a set of applications with common security requirements.

- **Certificate subject and issuer attributes:** These extensions are used to convey **additional** information about
  - A certificate subject or
  - A certificate issuer.
- **Certification Path Constraints**: These extensions permit constraints specifications to be included in certificates issued for CAs by other CAs (considering certificate chains).

Public-Key Infrastructure (PKI)

RFC 2822 define Public-Key Infrastructure (PKI) as the set of hardware, software, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography.

Main objectives for developing **PKI** is to enable Secure, convenient and efficient obtaining of public keys.

IETF PKI X.509 **(PKIX)** working group (http://www.ietf.org/html.charters/pkix-charter.html) setting up a formal (and generic) model based on X.509 in order to use PKI architecture on the Internet.

Section 14.5:
Public Key Infrastructure

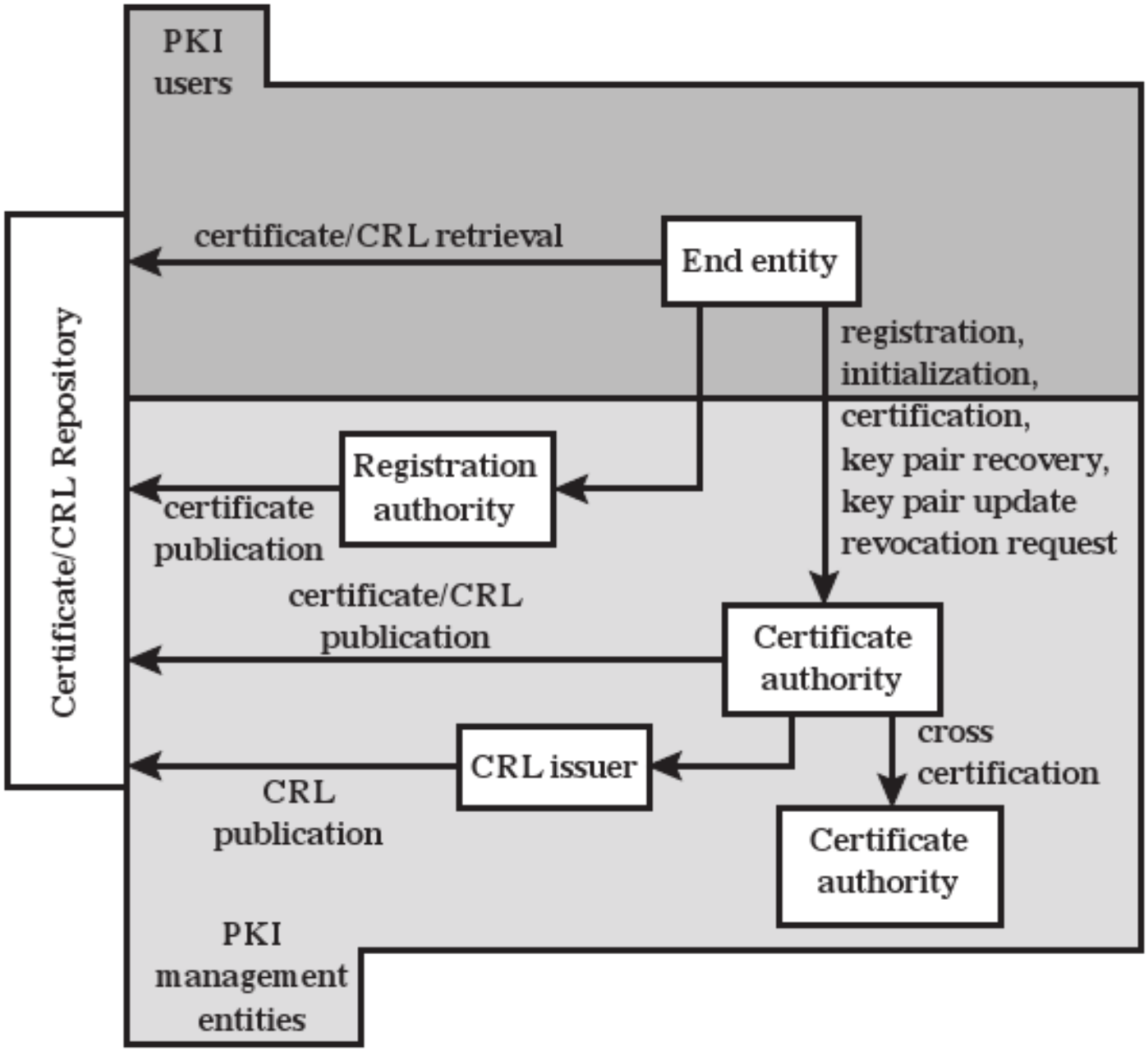


Figure 4.7 PKIX Architectural Model

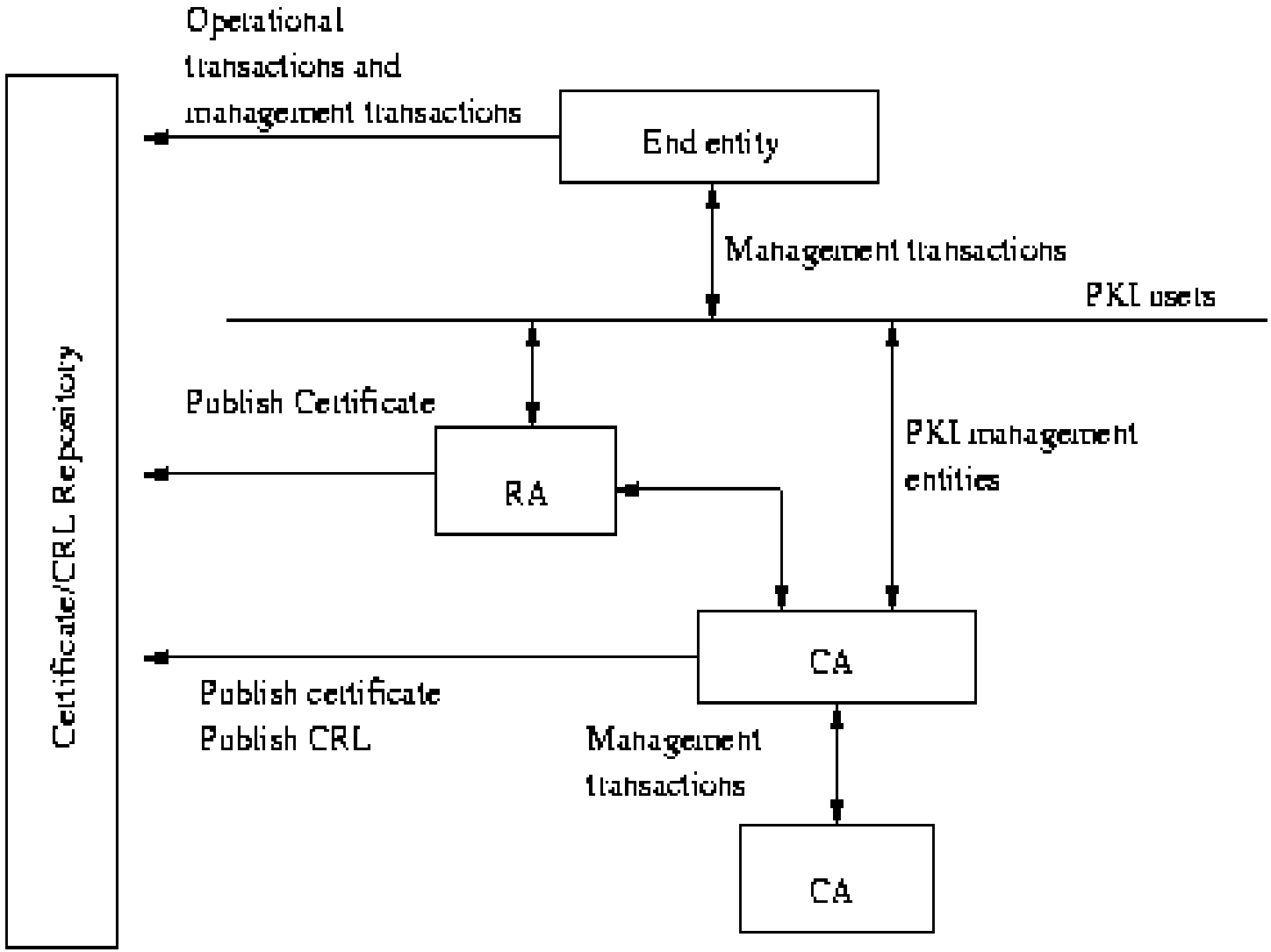
Public-Key Infrastructure (PKI)
PKIX Architectural Model

- This model key elements
  - End entity
  - Certification Authority (CA)
  - Registration Authority (RA)
  - CRL Issuer
  - Repository

Public-Key Infrastructure (PKI)
PKIX Management Functions

- Registration
- Initialization
- Certification
- Key pair recovery
- Key pair update
- Revocation request
- Cross Certification

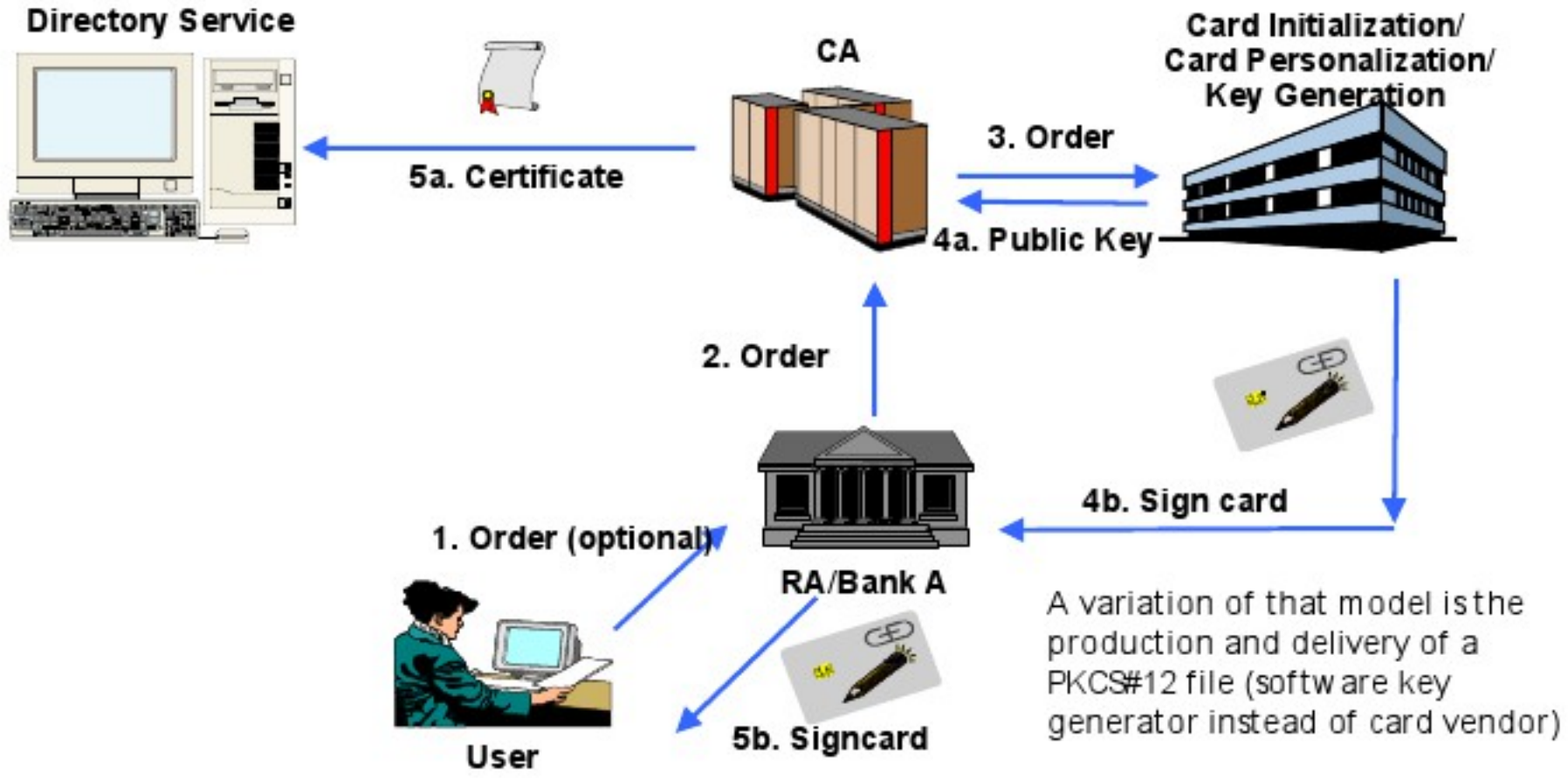
- PKIX Management Protocols: RFC 2510 defines the certificate management protocols (CMP) which describes each management functions explicitly.



Reference: http://ospkibook.sourceforge.net/docs/OSPKI-2.4.6/OSPKI/pkix-overview.htm

Three typical scenarios of PKI implementations

1.1 Workflow Model A: "Bank" Scenario



http://www.europepki.org/getFile.php?rubr=archive&doc=27

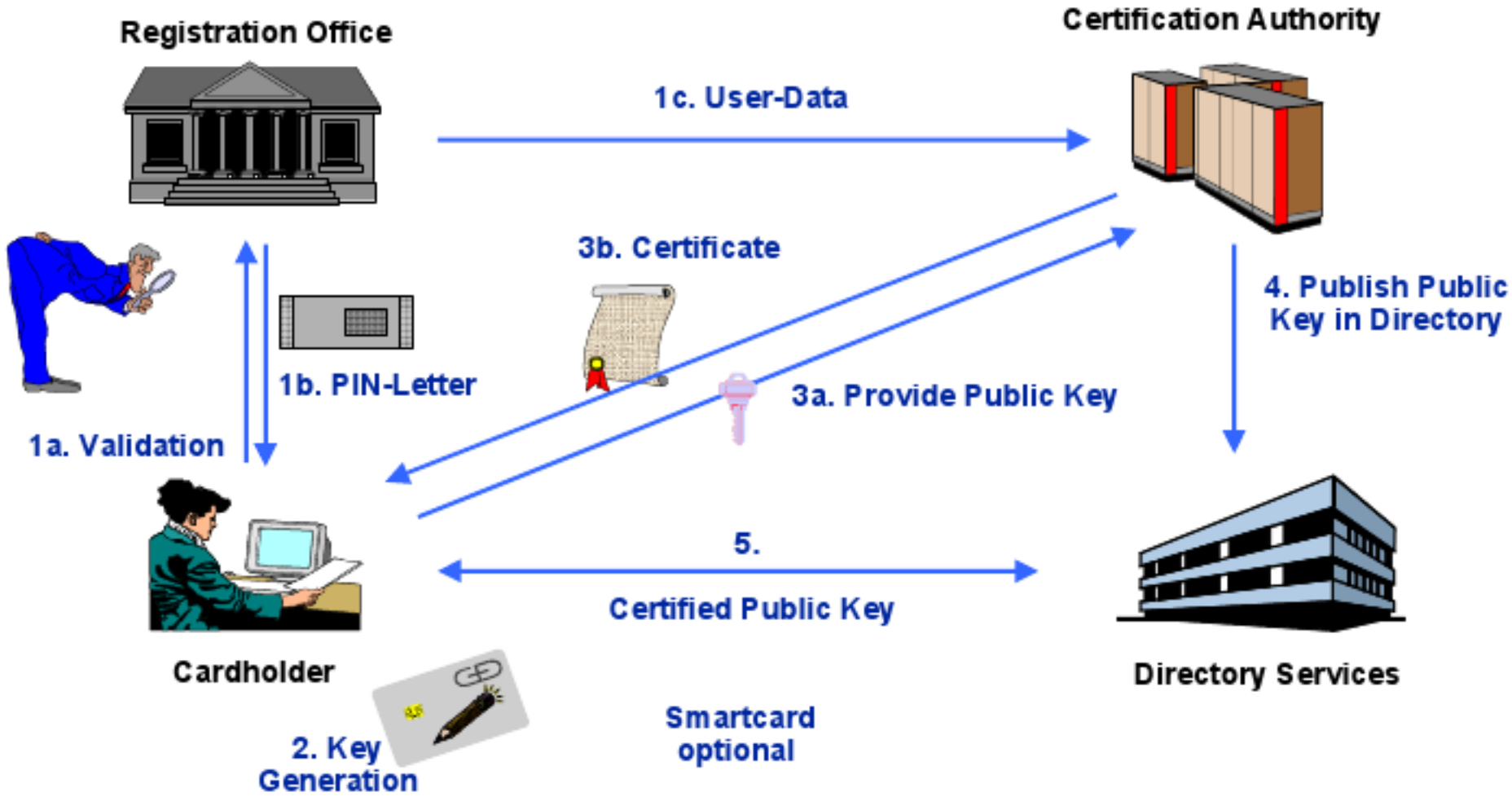
1.1 Workflow Model A: "Bank" Scenario

“ ... A banking card scenario is an attempt to combine ordinary and existing account creation procedures typically found in banks with the issuance of a smartcard containing certified key pairs. A user registers at a bank (creates an account) and applies for a smartcard. The data is forwarded to the CA operated by the bank which orders a smartcard from the card vendor. The card vendor does produce the card and the keys and delivers the public key to the CA, which certifies the key and distribute the certificate to the directory, and the card to the RA for delivery to the user.

If the certificate is to be stored on the card as well, two options are currently used: either the card production and key generation phase is done in advance, the CA has a list of public keys already available at the card vendor and delivers the certificate with the final card order to the card vendor, or the certificate is downloaded to the card at the RA or by the user.”.

REFERENCE: http://www.europepki.org/getFile.php?rubr=archive&doc=27

1.2 Workflow Model B: "Online" Scenario



REFERENCE: http://www.europepki.org/getFile.php?rubr=archive&doc=27

1.2 Workflow Model B: "Online" Scenario

"In this scenario the key generation happens at the user. The sequence of steps is:

1. Registration: the user applies for a card, delivers his information which gets verified and forwarded to the CA. The user gets an empty card and an initial one-time password in a security envelope from the RA.

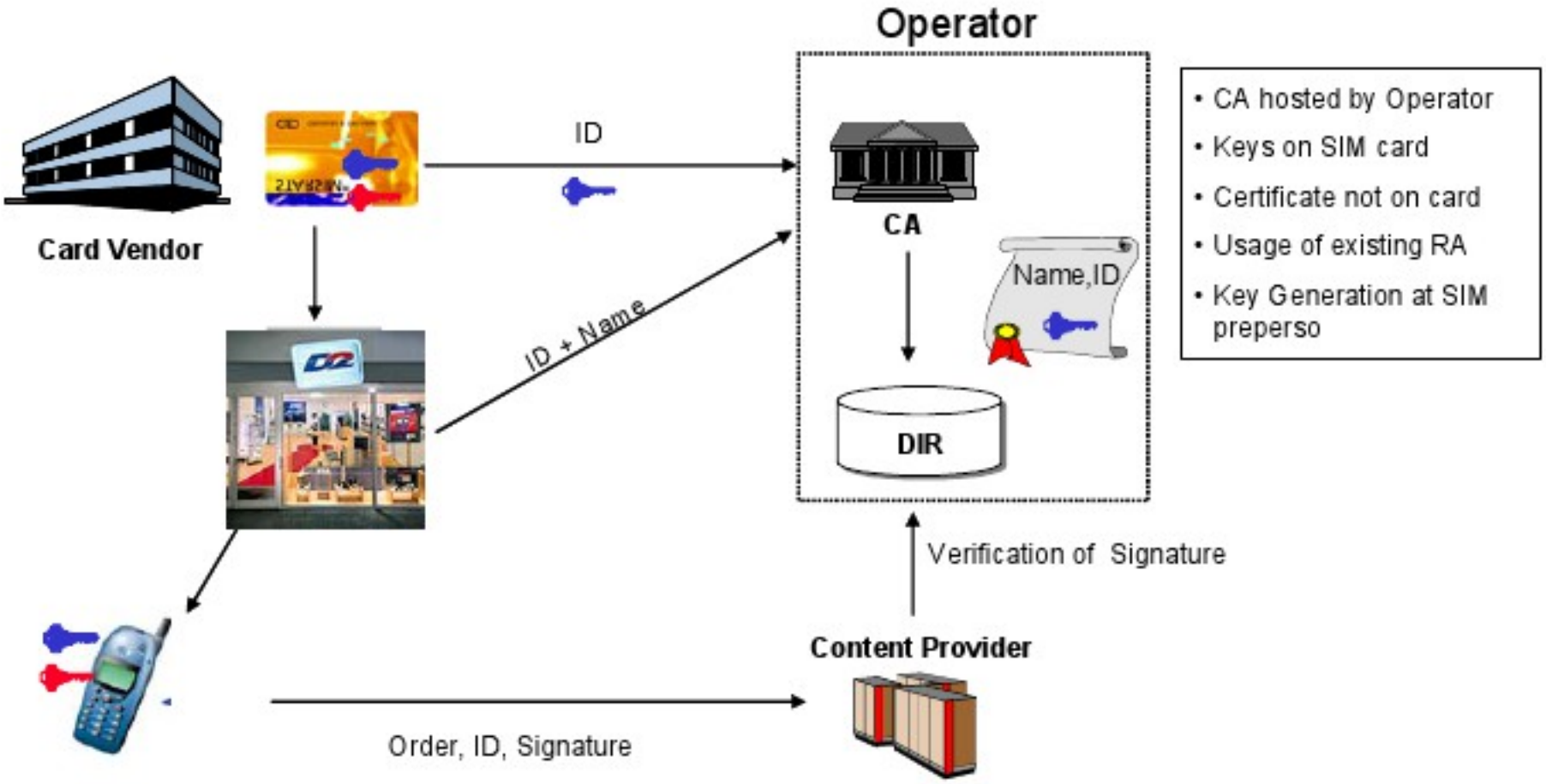
2. Key generation: the user generates keys and stores them on the card

3. Certification request: the user sends the public key along with the one-time password online to the CA. The password establishes a relationship between the key and the user information delivered by the RA to the CA. A certificate with this information is created and returned to the user and published in a directory.

This scenario is capable to support any type of key storage media, including keys stored in files (soft keys). It is the scenario typically deployed with internet applications like browsers and e-mail clients and is supported by their built-in functionality. It is the basic model for the PKI standardisation activities conducted by the IETF and its PKIX working group. The initial one-time password is called the "IAK" in the PKIX model."

REFERENCE: http://www.europepki.org/getFile.php?rubr=archive&doc=27

1.3 Workflow Model C: "Mobile" Scenario



http://www.europepki.org/getFile.php?rubr=archive&doc=27

1.3 Workflow Model C: "Mobile" Scenario

“ The mobile scenario is close to the banking scenario. Again it is an attempt to combine ordinary user enrolment procedures, this time those found at mobile network operators, with delivery of keys and certificates to enable PKI-functionality. Key generation is done centrally during card production. The difference between this scenario and the banking scenario is the change in the sequence of steps “card distribution” and “certificate production”. Whereas in the banking scenario the intended card holder is known during “card production”, this is not true for the mobile scenario where cards are assigned to cardholders at the card distribution points. Thus “certificate production” has to be delayed until a card is given to a cardholder.

In this scenario a CA, probably a mobile phone operator, orders a batch of cards (SIMs) from a card vendor. The cards are equipped with key pairs by the card vendor and delivered to card distribution points (operator outlets where mobile phone contracts are sold). The CA receives information on cards and public keys delivered to the distribution points.

Once a new user shows up at a distribution point, he/she is assigned a card. The CA gets informed on the personal information of the cardholder and the card he/she did receive, produces a certificate for the now assigned key(s) and publish them in its directory.

In contrary to other scenarios the certificate is not distributed to the cardholder. If the cardholder needs to see his own certificate he/she has to perform a directory lookup.”

REFERENCE: http://www.europepki.org/getFile.php?rubr=archive&doc=27

Visit the following web sites for such applications in Turkey:

http://www.avea.com.tr/web/Servisler/Islemler/Mobil_Yasam/AveaMobilimza

http://www.turkcell.com.tr/bireysel/servisler/Sayfalar/turkcell-mobil-imza.aspx

Important References

- X.509 (From Wikipedia, the free encyclopedia): http://en.wikipedia.org/wiki/X.509
- Public key infrastructure (From Wikipedia, the free encyclopedia): http://en.wikipedia.org/wiki/Public_key_infrastructure
- Açık Anahtar Altyapısı (PKI) Eğitim Kitabı : http://www.kamusm.gov.tr/dosyalar/kitaplar/aaa/
- ODTÜ - UYGULAMALI MATEMATİK ENSTİTÜSÜ, Açık Anahtar Altyapısı Araştırma Grubu : http://www.pki.iam.metu.edu.tr/
- Deliverables by the EuropePKI consortium: http://www.europepki.org/php/deliverables.php

Turkish Electronic Signature Law / Qualified Electronic Certificate Service Providers

- Turkish Electronic Signature Law (Elektronik imza Kanunu)

  http://www.resmigazete.gov.tr/eskiler/2004/01/20040123.htm#1
- Qualified Electronic Certificate Service Providers (Nitelikli (yasal gerçerliliği olan) Elektronik İmza Hizmet Sağlayıcılar):
  - http://www.turktrust.com.tr/en/index.html
  - http://www.e-guven.com/en/about_E-GUVEN.html
  - http://e-tugra.com.tr
  - http://www.kamusm.gov.tr/