

PKI

Part 1: Public-Key Certificates

Kjell Jørgen Hole
NoWires Research Group
Department of Informatics
University of Bergen

last changed September 21, 2008

Outline

- Introduction to public-key certificates
 - certification paths
 - types of certificates
- Evolution of the X.509 certificate format
 - basic building blocks
 - mandatory contents
 - optional features

Definition of trust

- In general, trust refers to an aspect of a relationship between two entities *A* and *B*
- *A* can be said to “**trust**” *B* when *A* makes the assumption that it knows exactly how *B* will behave
- Note that there is risk associated with trust

3

Sources of trust

- **Direct relationship:** kinship, mateship, contract, multiple prior transactions
- **Direct experience:** prior exposure, a prior transaction
- **Referred trust:** ‘word-of-mouth’, reputation, accreditation
- **Symbols of trust:** brands

4

What is a certificate?

- A **Certificate** is a document generated by a **trusted** third party containing a certified statement
 - everyday examples are driver's licenses, passports, and credit cards
- A **digital certificate** is a collection of electronic data 'digitally signed' by a trusted third party
 - the signing ensures that tampering will be discovered

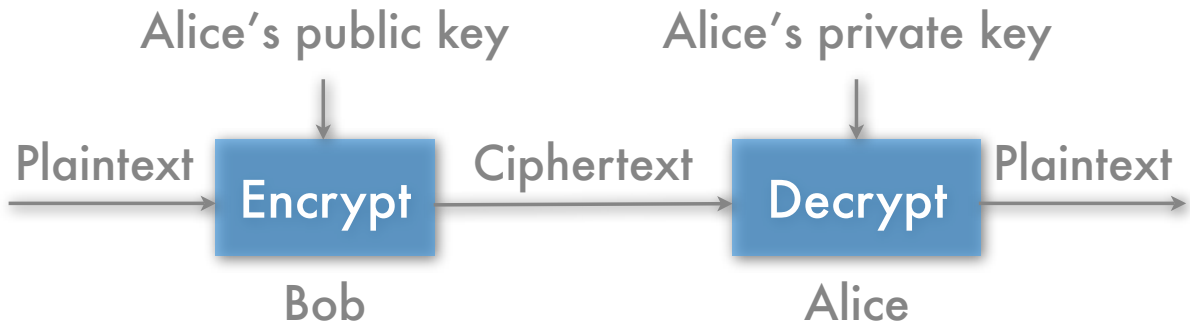
5

Public-key cryptography

- **Public-key cryptography** enables us to realize digital certificates
- Public-key cryptography is based on two keys:
 - **private key**—must be kept secret
 - **public key**—is published
- The two keys are complementary, but the value of the private key cannot be determined from the public key

6

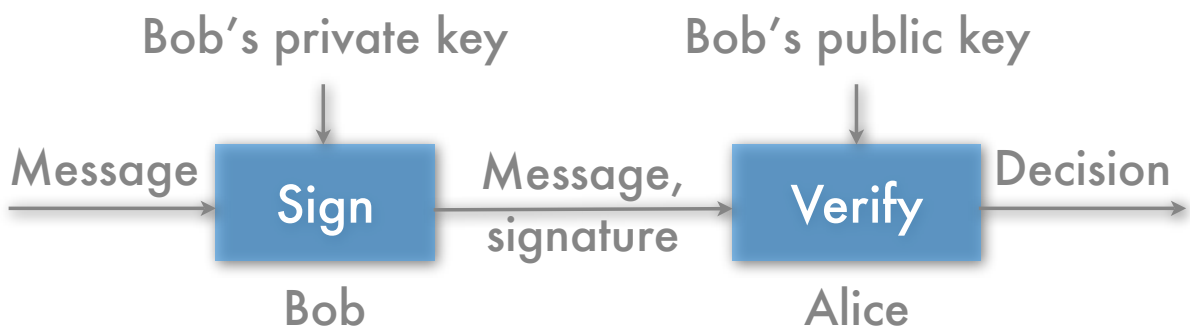
Encryption mode



- When Bob wants to send a **confidential** message to Alice, he uses Alice's public key

7

Signing mode



- When Bob wants to send an **authenticated** message to Alice, he uses his private key to **sign** the message

8

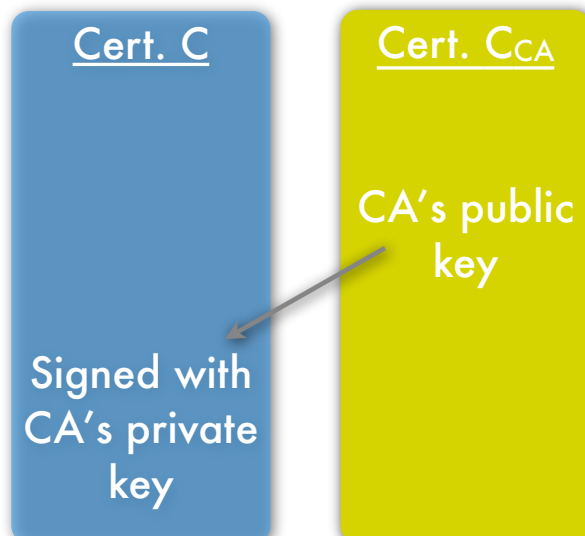
Public-key certificate

- A **public-key certificate** contains an entity's public key
- The certificate is generated by a trusted third party, called a **Certification Authority (CA)**
- The CA has verified the identity of the entity holding the private key corresponding to the public key in the certificate

9

Certificate validation

- The public-key certificate, denoted C , is digitally signed with the CA's private key
- To validate C , the user must first get hold of the CA's certificate, C_{CA}
- The public key in C_{CA} is then used to verify the digital signature of C



10

So far we have

A CA digitally signs a public-key certificate to guarantee its content and to enable detection of tampering

The certificate binds a public key to the name of the entity with the private key

11

Lack of trust

- **Problem:**
 - Alice communicates with users whose certificates are issued by different CAs
 - Alice does not trust all these CAs
- **Solution:**
 1. CAs issue certificates to other CAs
 2. Alice extends a path of certificates until she reaches a CA she trusts

Certification path

- A CA is denoted a **trust point** (or trust anchor) if an end entity trusts the CA's public key
- A **certification path** is a chain of certificates:
 - the issuer of the first certificate is a trust point
 - the subject of the last certificate is the end entity
- A certification path is constructed to verify the certificate of the end entity

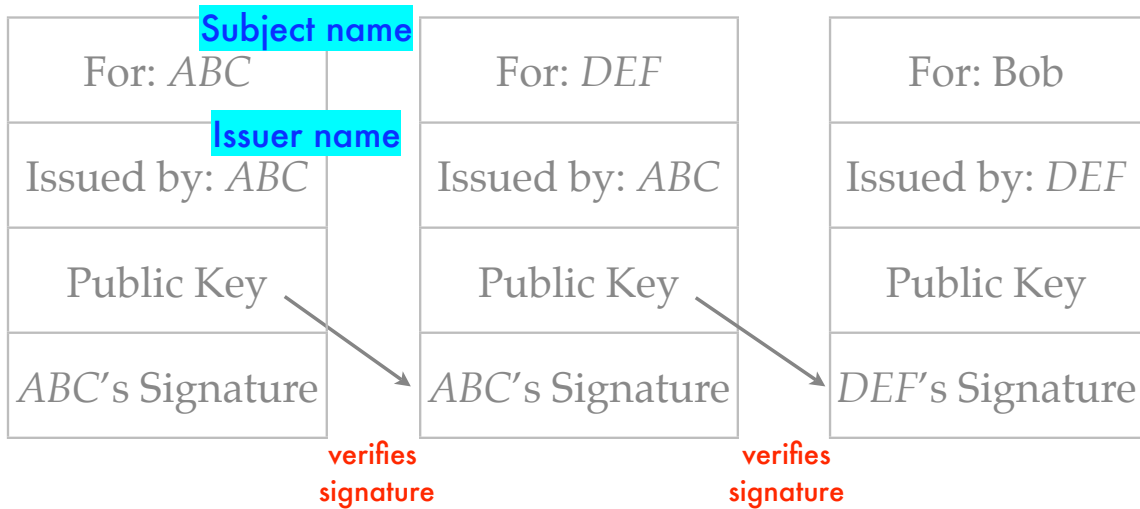
13

Two-step path processing

1. **Path construction** involves collecting all certificates needed to form a complete path from an end entity to a trust point
2. **Path validation** involves examining each certificate in the path in turn, verifying the digital signature, examining validity period, checking revocation status, looking at policies, key usage restrictions, and so on

14

Certification path



- Alice has *ABC's* public key
- *ABC's* certificate is the trust point

Example explanation

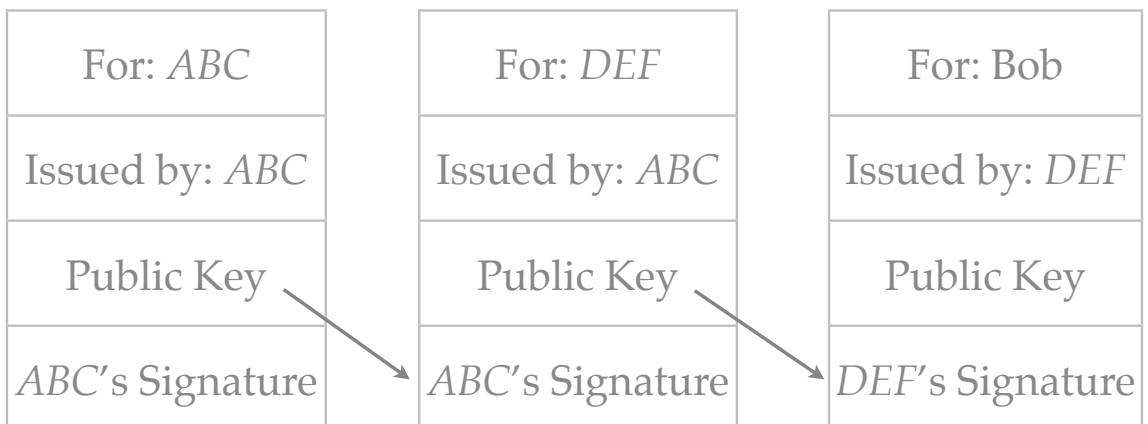
- Alice trusts the CA denoted *ABC*
 - she has *ABC's* public key
- *ABC* has issued a certificate to a CA denoted *DEF*
- Alice receives certificate from Bob signed by the private key of *DEF*
- To verify Bob's certificate Alice follows the certification path in the figure

Three types of certificates

1. **User certificates** for entities that are not CAs
2. **CA certificates** issued to CAs
 - part of certification paths
3. **Self-issued certificates**, or **root certificates**, are a special class of CA certificates where the issuer and the subject name are the same
 - used to distribute public-keys and to establish trust points

17

Example revisited



**Self-issued
(or root)
certificate**

CA certificate

User certificate

18

X.500 names

- **Distinguished Name (DN)**
 - X.500 hierarchical naming system
 - ordered list of naming attributes
 - the most common attributes are:
 - **c**=country, **o**=organization,
ou=organizational unit, **l**=locality,
cn=common name

`c=NO; o=UiB; ou=Department of Informatics; cn=Kjell Jørgen Hole`

19

X.509 certificate evolution

- **Version 1:**
 - X.509 certificate specified in document CCITT Recommendation X.509
 - X.509 first published in 1988
- **Version 2:**
 - introduced in 1993
 - addressed the problem of reuse of X.500 names

20

X.509 Version 3

- **Version 3** introduced certificate extensions
 - extension fields are used to include information not supported by basic certificate fields
- Only Version 3 certificates are considered during the remainder of the lecture series

certificate = X.509 Version 3 public-key certificate

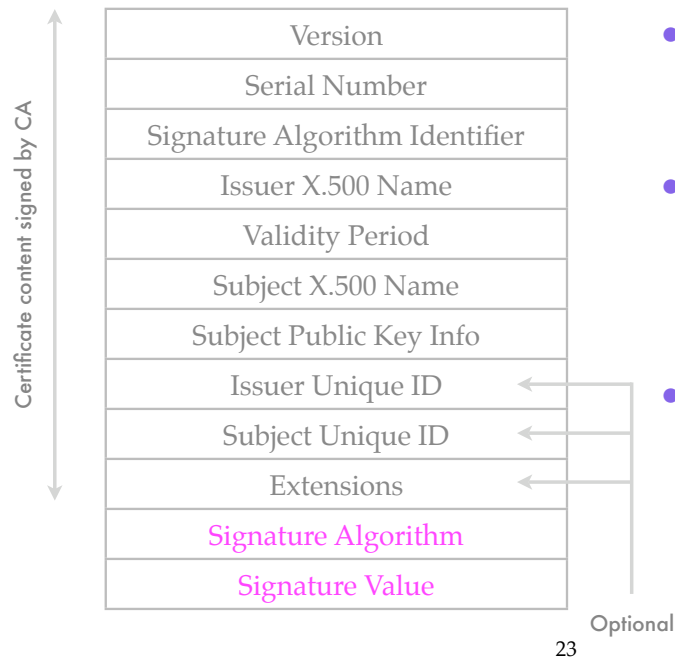
21

Naming in X.509 Version 3

- Version 3 is not restricted to the X.500 naming system
- Any entity can be identified by one or more names of a variety of different forms:
 - Internet e-mail address
 - Internet domain name
 - Uniform Resource Locator (URI)
 - IP address

22

X.509 certificate structure



- The table shows the certificate **fields**
- The content of the certificate is given by the 10 first fields
- The two last fields ensure that it is possible to detect tampering

Explanation of fields

- **Version:** indicator of Version 1, 2, or 3 format
- **Serial number:** unique identifying number for certificate, assigned by issuing CA
- **Signature Algorithm Identifier:** algorithm identifier of the digital signature algorithm used by the CA to sign certificate
- copy of the signature algorithm field protected by the digital signature

Explanations ...

- **Issuer:** X.500 name of the issuing CA
- **Validity Period:** Start and expiration dates
- **Subject Name:** X.500 name, i.e. DN, of the holder of the private key for which the corresponding public key is being certified
- **Subject-public-key information:** value of public key and an identifier for the algorithm to be used with the key

25

Unique ID fields

- **Issuer Unique ID and Subject Unique ID:** optional bit strings used to make the names unambiguous in the event that the same names have been reassigned to different entities
 - difficult to manage, easy to overlook
 - fields should not be used

26

Digital signature algorithms

Public key algorithm	Hash function	Algorithm identifier
RSA	MD5	md5WithRSAEncryption
RSA	SHA-1	sha1WithRSAEncryption
DSA	SHA-1	id-dsa-with-sha1
ECDSA	SHA-1	ecdsa-with-sha1

- Common digital signature algorithms

27

Signature value

- **Signature value:** the field contains the value of the signature
- value is encoded as a bit string using conventions defined for the given signature algorithm
- Generated by the CA using the private key
- Anyone can verify the signature using the CA's public key

28

Optional extensions

- **Extensions:** contains one or more certificate extensions. Each extension includes
 - an extension identifier
 - a **criticality flag** (critical / noncritical)
 - an extension value
- When the criticality flag is set the extension must be processed and understood, or the certificate is not to be used

29

Criticality

- Noncritical extensions facilitate certificate sharing between different applications and graceful migration
- Critical extensions cause interoperability problems and should be avoided except to address security concerns

30

Standard vs. private extensions

- Extensions allow a CA to include information not supported by the basic certificate content
- It is possible to define private extensions, however, they should be avoided to achieve interoperability
- Only standard extensions are discussed in the following

31

Classification of extensions

- There are five different groups of extensions:
 1. **Subject ex:** is Bob a CA or an end entity?
 2. **Name ex:** Are `alice@fox.com` and `c=US; o=Fox Consulting; cn=Alice Adams` the same person?
 3. **Key ex:** Can this public key be used for key transport? verify digital signature?

32

4. **Policy information ex:** Can I trust Alice's certificate? Is it appropriate for large value transactions?

5. **Additional info ex:** Where can I find

- certificates issued to a certain identity?
- certification revocation lists issued by some given CAs?

33

Name extensions

- Initially, the only name form available was DN
- X.509 Version 3 has naming extensions called:
 - **the subject alternative names**—mail addresses or DNS names for humans; URLs and DNS names for computers
 - **the issuer alternative names**—list of general names, e.g. CA's mail address

34

Policy extensions

- In early implementations, each CA issued certificates under only one (implicit) policy
 - it is clearly inefficient to deploy a separate CA for every policy
- Two standard policy extensions exist to solve this problem:
 - certificate policies extension
 - policy mapping extension

Certificate policies extension

- The certificate policies extension in a CA certificate indicates the policies under which the CA operates
- In an end entity certificate, the extension indicates the policy (policies) under which the certificate was issued
- A globally unique Object Identifier (OID) in the extension identifies a certain certificate policy

Certificate policies ...

- A OID may indicate that the certificate
 - is to be used with a particular application
 - has a certain relative quality (very good, good, or mediocre)
- Note that since the same certificate might be used for multiple applications, several policies might be identified in one certificate

37

Policy mapping extension

- The policy mapping extension is used in CA certificates to translate policy information between two policy domains
- Policy mapping translates remote policy OIDs into local policy OIDs known to the certificate user
- The extension contains a list of one or more OID pairs

38

Additional info extensions

- **CRL distribution points**—information on where and how to find one or more CRLs
- **Authority information access**—information on how to access CA information

39

Summary

- A public-key certificate binds a public key to (the name of) an entity. This entity has the private key corresponding to the public key
- X.509 Version 3 certificates dominates on the Internet
- a certificate consists of many fields, some of which are optional
- the certificate content is signed with the CA's private key

40

Summary

- There are three types of certificates:
 - user certificates
 - CA certificates
 - self-issued (or root) certificates

41

Sources

- C. Adams and S. Lloyd, [Understanding PKI](#), 2nd Edition, Addison Wesley, 2003
- W. Ford and M. S. Baum, [Secure Electronic Commerce](#), 2nd Edition, Prentice Hall, 2001
- R. Housley and T. Polk, [Planning for PKI](#), Wiley, 2001

42

