

X.509 CERTIFICATE

What is it?

- X.509 is a digital certificate that is built on top of a widely trusted standard known as ITU or International Telecommunication Union X.509 standard, in which the format of PKI certificates is defined.
- X.509 digital certificate is a certificate-based authentication security framework that can be used for providing secure transaction processing and private information.
- These are primarily used for handling the security and identity in computer networking and internet-based communications.
- X.509 is based on the use of public-key cryptography and digital signatures.



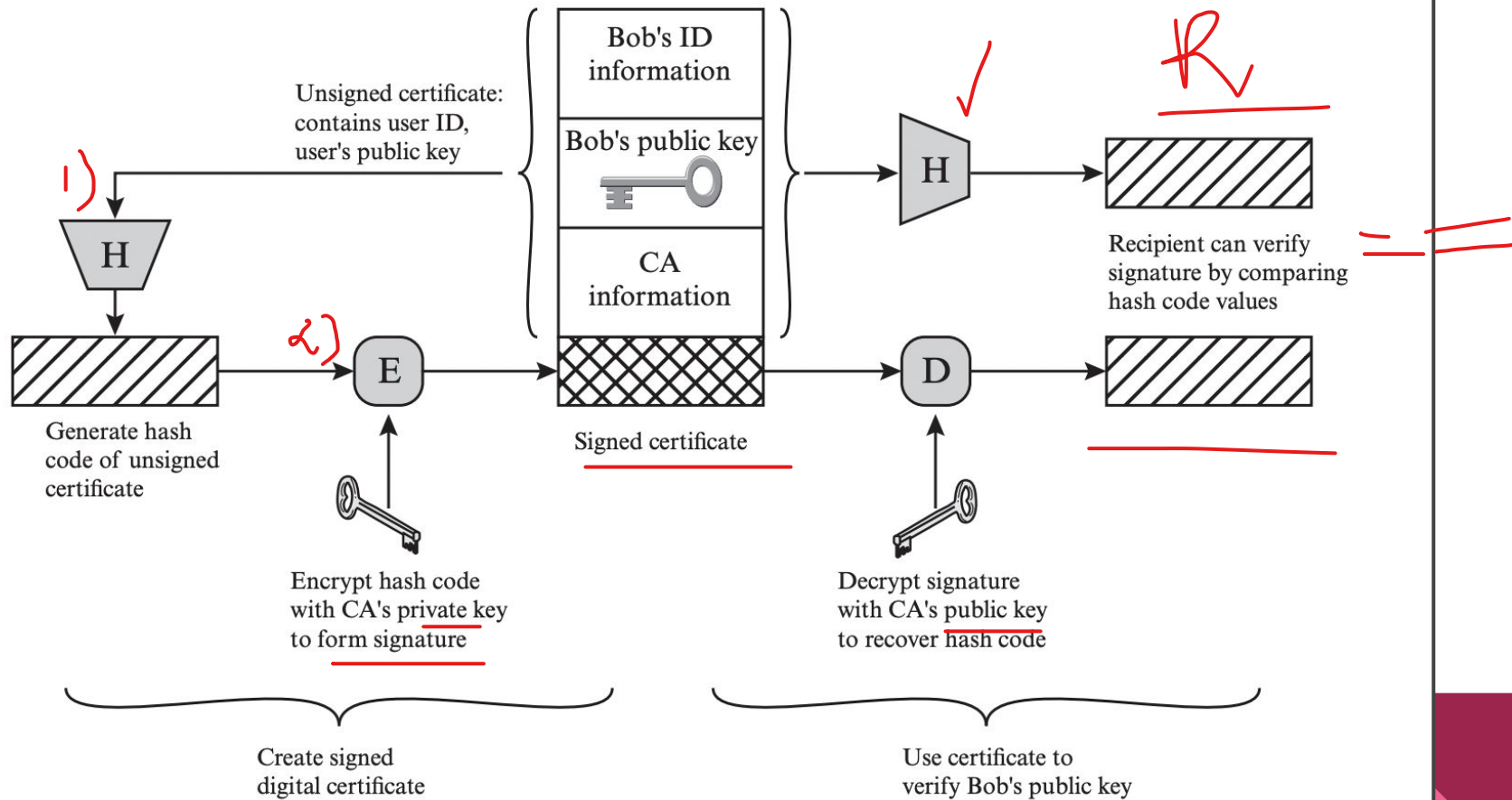
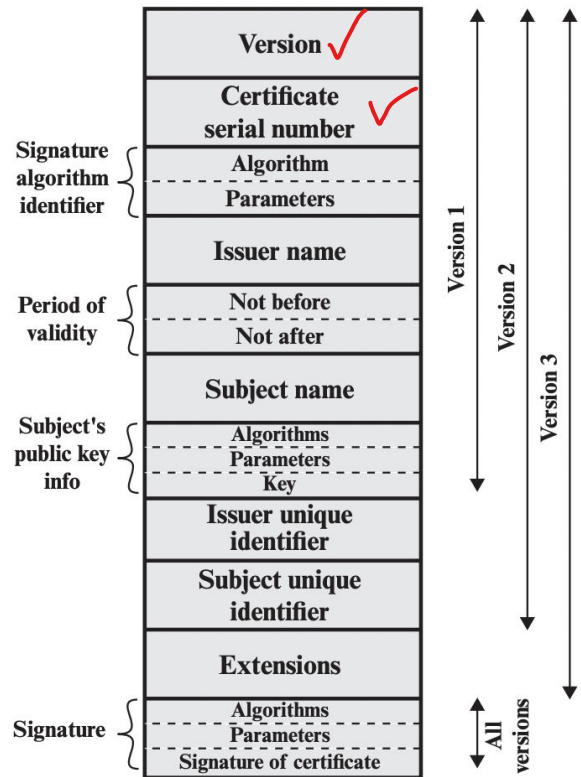


Figure 14.14 X.509 Public-Key Certificate Use

Overall Scheme for generating certificate

- The certificate for Bob's public key includes unique identifying information for Bob, Bob's public key, and identifying information about the CA, plus other information as explained subsequently.
- This information is then signed by computing a hash value of the information and generating a digital signature using the hash value and the CA's private key.
- X.509 indicates that the signature is formed by encrypting the hash value.





(a) X.509 certificate

Figure 14.15 X.509 Formats 1 2 3

- **Version:** Differentiates among successive versions of the certificate format; the default is version 1. If the *issuer unique identifier* or *subject unique identifier* are present, the value must be version 2. If one or more extensions are present, the version must be version 3. Although the X.509 specification is currently at version 7, no changes have been made to the fields that make up the certificate since version 3.
- **Serial number:** An integer value unique within the issuing CA that is unambiguously associated with this certificate.
- **Signature algorithm identifier:** The algorithm used to sign the certificate together with any associated parameters. Because this information is repeated in the signature field at the end of the certificate, this field has little, if any, utility.
- **Issuer name:** X.500 name of the CA that created and signed this certificate.
- **Period of validity:** Consists of two dates: the first and last on which the certificate is valid.
- **Subject name:** The name of the user to whom this certificate refers. That is, this certificate certifies the public key of the subject who holds the corresponding private key.
- **Subject's public-key information:** The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.
- **Issuer unique identifier:** An optional-bit string field used to identify uniquely the issuing CA in the event the X.500 name has been reused for different entities.
- **Subject unique identifier:** An optional-bit string field used to identify uniquely the subject in the event the X.500 name has been reused for different entities.
- **Extensions:** A set of one or more extension fields. Extensions were added in version 3 and are discussed later in this section.
- **Signature:** Covers all of the other fields of the certificate. One component of this field is the digital signature applied to the other fields of the certificate. This field includes the signature algorithm identifier.

The standard uses the following notation to define a certificate:

$$\text{CA} \ll A \gg = \text{CA} \{ \text{V}, \text{SN}, \text{AI}, \text{CA}, \text{UCA}, \text{A}, \text{UA}, \text{Ap}, \text{T}^{\text{A}} \}$$

where

/ $Y \ll X \gg$ = the certificate of user X issued by certification authority Y

$Y \{I\}$ = the signing of I by Y. It consists of I with an encrypted hash code appended

V = version of the certificate

SN = serial number of the certificate

AI = identifier of the algorithm used to sign the certificate

CA = name of certificate authority

UCA = optional unique identifier of the CA

A = name of user A

UA = optional unique identifier of the user A

Ap = public key of user A

T^A = period of validity of the certificate

Obtaining a user's certificate

User A

User B

Features of certificates:

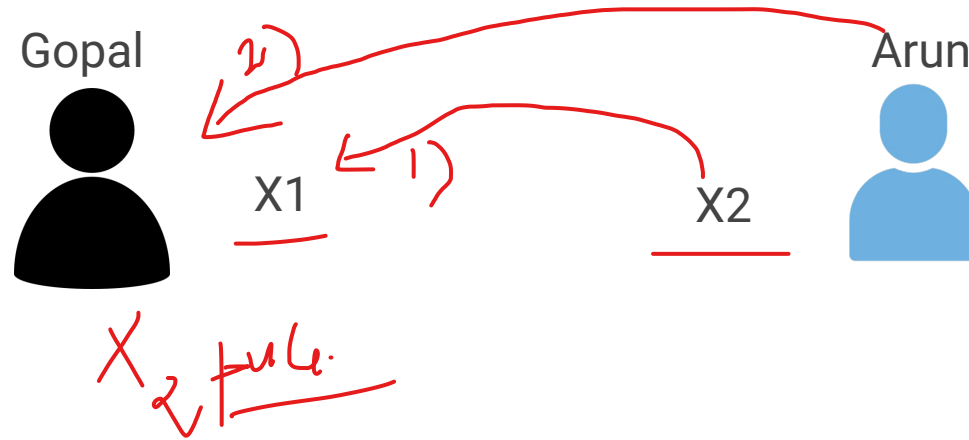
- Any user with access to the public key of the CA can verify the user public key that was certified.
- No party other than the certification authority can modify the certificate without this being detected.
- Because certificates are unforgeable, they can be placed in a directory without the need for the directory to make special efforts to protect them

1) All Users in same CA: T

All user certificates can be placed in the directory for access by all users. In addition, a user can transmit his or her certificate directly to other users.

In either case, 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

Different User in Different CA



How Gopal can check Arun's certificate?

1. Gopal has every CA's public key (Expensive and risky)
2. Use X2 certificate signed by X1 and decrypt it => get X2 public key
3. Use result of step 2 to verify arun's certificate

Now suppose that A has obtained a certificate from certification authority X_1 and B has obtained a certificate from CA X_2 . If A does not securely know the public key of X_2 , then B's certificate, issued by X_2 , is useless to A. A can read B's certificate, but A cannot verify the signature. However, if the two CAs have securely exchanged their own public keys, the following procedure will enable A to obtain B's public key.

Step 1 A obtains from the directory the certificate of X_2 signed by X_1 . Because A securely knows X_1 's public key, A can obtain X_2 's public key from its certificate and verify it by means of X_1 's signature on the certificate.

Step 2 A then goes back to the directory and obtains the certificate of B signed by X_2 . Because A now has a trusted copy of X_2 's public key, A can verify the signature and securely obtain B's public key.

A has used a chain of certificates to obtain B's public key. In the notation of X.509, this chain is expressed as

$$X_1 \ll X_2 \gg X_2 \ll B \gg$$

In the same fashion, B can obtain A's public key with the reverse chain:

$$X_2 \ll X_1 \gg X_1 \ll A \gg$$

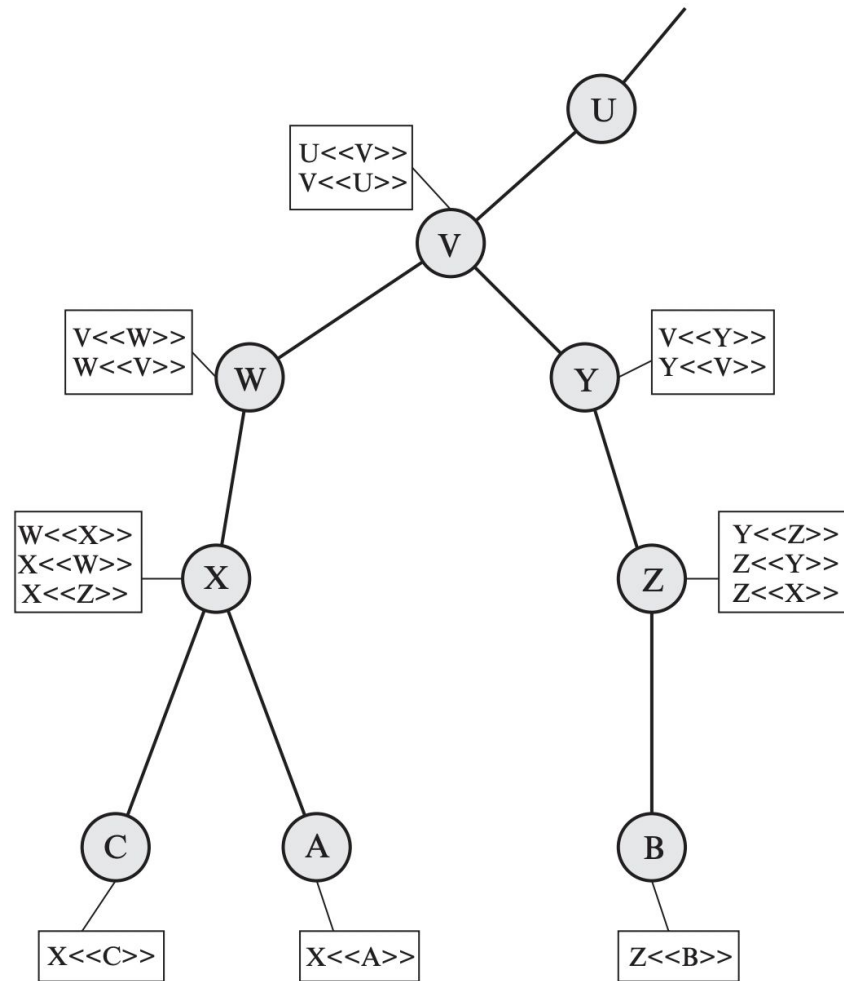


Figure 14.16 X.509 Hierarchy: A Hypothetical Example

Revocation of Certificates

1. The user's private key is assumed to be compromised.
2. The user is no longer certified by this CA. Reasons for this include that the subject's name has changed, the certificate is superseded, or the certificate was not issued in conformance with the CA's policies.
3. 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, including both those issued to users and to other CAs. These lists should also be posted on the directory.

Signature algorithm identifier	{	Algorithm
		Parameters
		Issuer name
		This update date
Revoked certificate	{	Next update date
		User certificate serial #
	{	Revocation date
		• • •
Revoked certificate	{	User certificate serial #
		Revocation date
Signature	{	Algorithms
		Parameters
		Signature of certificate

(b) Certificate revocation list

Drawbacks of Version 2

1. The subject field is inadequate to convey the identity of a key owner to a public-key user. X.509 names may be relatively short and lacking in obvious identification details that may be needed by the user.
2. The subject field is also inadequate for many applications, which typically recognize entities by an Internet email address, a URL, or some other Internet-related identification.
3. There is a need to indicate security policy information. This enables a security application or function, such as IPsec, to relate an X.509 certificate to a given policy.
4. There is a need to limit the damage that can result from a faulty or malicious CA by setting constraints on the applicability of a particular certificate.
5. It is important to be able to identify different keys used by the same owner at different times. This feature supports key lifecycle management: in particular, the ability to update key pairs for users and CAs on a regular basis or under exceptional circumstances.


Extended Fields- Key & Policy Info

- **Authority key identifier:** Identifies the public key to be used to verify the signature on this certificate or CRL. Enables distinct keys of the same CA to be differentiated. One use of this field is to handle CA key pair updating.
- **Subject key identifier:** Identifies the public key being certified. Useful for subject key pair updating. Also, a subject may have multiple key pairs and, correspondingly, different certificates for different purposes (e.g., digital signature and encryption key agreement).
- **Key usage:** Indicates a restriction imposed as to the purposes for which, and the policies under which, the certified public key may be used. May indicate one or more of the following: digital signature, nonrepudiation, key encryption, data encryption, key agreement, CA signature verification on certificates, CA signature verification on CRLs.
- **Private-key usage period:** Indicates the period of use of the private key corresponding to the public key. Typically, the private key is used over a different period from the validity of the public key. For example, with digital signature keys, the usage period for the signing private key is typically shorter than that for the verifying public key.
- **Certificate policies:** Certificates may be used in environments where multiple policies apply. This extension lists policies that the certificate is recognized as supporting, together with optional qualifier information.
- **Policy mappings:** Used only in certificates for CAs issued by other CAs. Policy mappings allow an issuing CA to indicate that one or more of that issuer's policies can be considered equivalent to another policy used in the subject CA's domain.

Subject & Issuer Attributes

- **Subject alternative name:** Contains one or more alternative names, using any of a variety of forms. This field is important for supporting certain applications, such as electronic mail, EDI, and IPsec, which may employ their own name forms.
- **Issuer alternative name:** Contains one or more alternative names, using any of a variety of forms.
- **Subject directory attributes:** Conveys any desired X.500 directory attribute values for the subject of this certificate.

Certificate Path Constraints

- **Basic constraints:** Indicates if the subject may act as a CA. If so, a certification path length constraint may be specified.
 - **Name constraints:** Indicates a name space within which all subject names in subsequent certificates in a certification path must be located.
 - **Policy constraints:** Specifies constraints that may require explicit certificate policy identification or inhibit policy mapping for the remainder of the certification path.
- 

Thank You

