# X.509

- 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.

- **Working of X.509 Authentication Service Certificate:**
- The core of the X.509 authentication service is the public key certificate connected to each user. These user certificates are assumed to be produced by some trusted certification authority and positioned in the directory by the user or the certified authority. These directory servers are only used for providing an effortless reachable location for all users so that they can acquire certificates. X.509 standard is built on an IDL known as ASN.1. With the help of Abstract Syntax Notation, the X.509 certificate format uses an associated public and private key pair for encrypting and decrypting a message.
- Once an X.509 certificate is provided to a user by the certified authority, that certificate is attached to it like an identity card. The chances of someone stealing it or losing it are less, unlike other unsecured passwords. With the help of this analogy, it is easier to imagine how this authentication works: the certificate is basically presented like an identity at the resource that requires authentication

# Certificate

- The heart of the X.509 scheme is the public-key certificate associated with each user. These user certificates are assumed to be created by some trusted certifi cation authority (CA) and placed in the directory by the CA or by the user. The directory server itself is not responsible for the creation of public keys or for the certification function: it merely provides an easily accessible location for users to obtain certificates.

# X.509 format

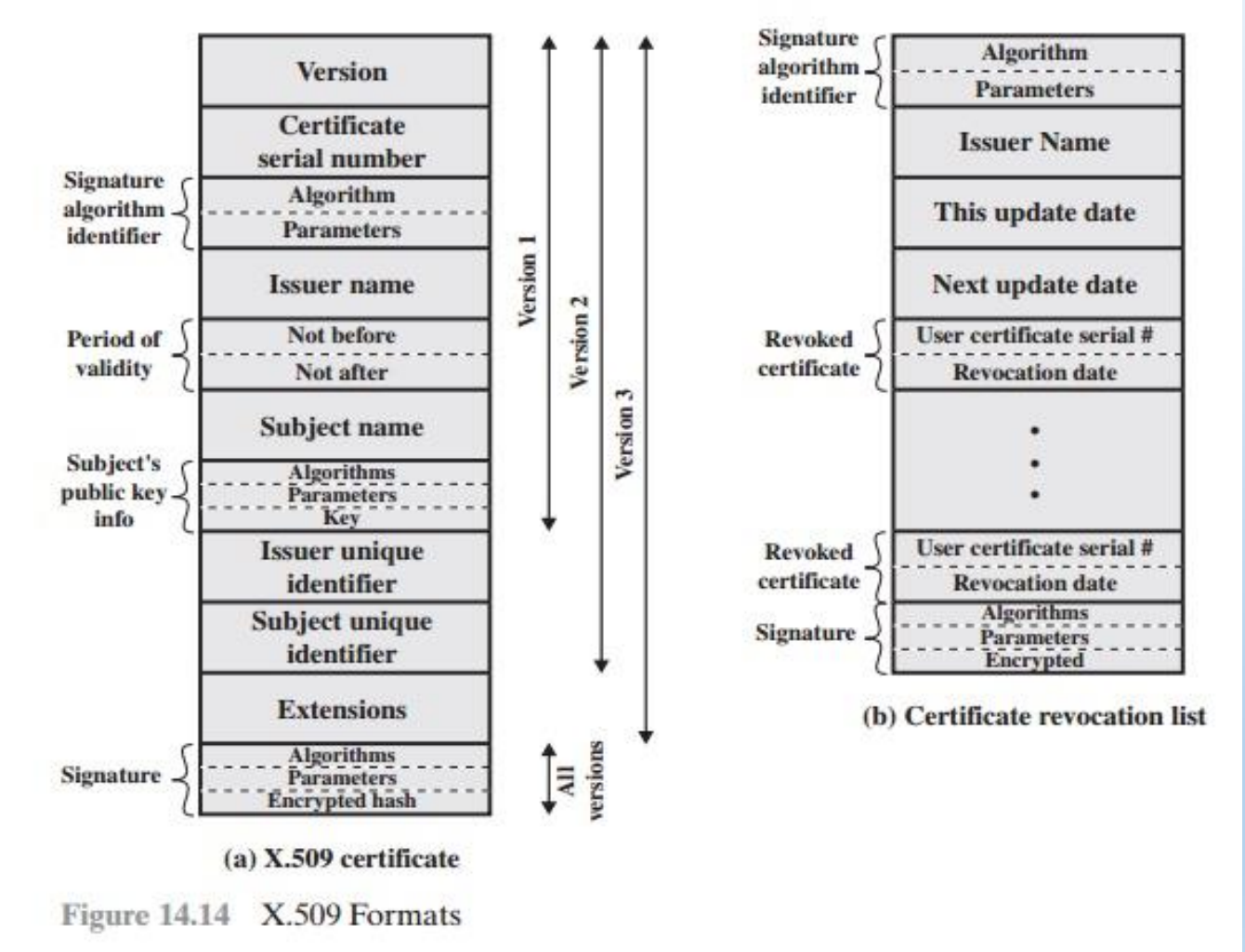



Figure 14.14 X.509 Formats

- Generally, the certificate includes the elements given below:
- **Version number:** It defines the X.509 version that concerns the certificate.
- **Serial number:** It is the unique number that the certified authority issues.
- **Signature Algorithm Identifier:** This is the algorithm that is used for signing the certificate.
- **Issuer name:** Tells about the X.500 name of the certified authority which signed and created the certificate.
- **Period of Validity:** It defines the period for which the certificate is valid.
- **Subject Name:** Tells about the name of the user to whom this certificate has been issued.
- **Subject's public key information:** It defines the subject's public key along with an identifier of the algorithm for which this key is supposed to be used.
- **Extension block:** This field contains additional standard information.
- **Signature:** This field contains the hash code of all other fields which is encrypted by the certified authority private key.

# Obtaining a User's certificate

- Users certificate generated by a CA have the following characteristics:

- Any user with access to the public key of the CA can verily 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 who need for the directory to make special efforts to protect them

- If all users subscribe to the same CA, then there is a common trust of that CA. All user certificates can be placed in the directory for access by all users.
- A user can transmit his or her certificate directly to other users.
- In either case 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.

- If there is a large community of users, it may not be practical for all to subscribe to the same CA. Because it is the CA that signs certificates each pump user must have a copy of the CA's own public key to verify signatures.
- With many users, it may be more practical for there to be a number of CAs each of which securely provides its public key to some fraction of the users.

- Now suppose that A has obtained a certificate from certification authority X1 and B has obtained a certificate from CA X2. If A does not securely know the public key of X2, then B's certificate, issued by X2, is useless to A. A can read B’s certificate, but A cannot verify the signature. However, if the two CAs has exchanged their own public keys, the following procedure will enable A B's public key:

1. A obtains from the directory, the certificate of X2, signed by X1. Because A securely knows X1's public key, A can obtain X2's public key from its certificate and verify it by means of X1,'s signature on the certificate
2. A then goes back to the directory and obtains the certificate of B signed by X2 Because A now has a trusted copy of X2's public key, A can verity 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

  X1<<X2>>>X2<<B>>

- In the same fashion, B cam obtain A's public key with the reverse chain

  X2<<X1>>X1<<A>>

- This scheme need not be limited to a chain of two certificates An arbitrarily long path of CAs can be followed to produce a chain.

  X1<<X2>>X2<<X3>>….Xn<<B>>

- In this case, each pair of CAs in the chain (X, X) must have created certificates for each other. All these certificates of CAs by CAs need to appear in the directory, and the user needs to know how they are linked to follow a path to another user's public-key: certificate. X.509 suggests that CAs be arranged in a hierarchy so that navigation is straightforward.

- The directory entry for each CA includes two types of certificates.
- Forward certificates: Certificates of X generated by other CAs.
- Reverse certificates: Certificates generated by X that are the certificates of other CAs.

# Revocation of certificates

- Revocation of Certificates Recall from Figure 1, that each certificate includes period of validity, much like a credit card. Typically, a new certificate is issued just before the expiration of the old one. In addition, 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, including both those issued to users and to other Cas. These lists should also be posted on the directory.
- Each certificate revocation list (CRL) posted to the directory is signed by the issuer and includes the issuer's name, the date the list was created, the date the next CRL is scheduled to be issued, and an entry for each revoked certificate. Each entry consists of the serial number of a certificate and revocation date for that certificate. Because serial numbers are unique within a CA, the serial number sufficient to identify the certificate.
- When a user receives a certificate in a message, the user must determine whether the certificate has been revoked. The user could check the directory each time a certificate is received. To avoid the delays (and possible costs) associated wit directory searches, it is likely that the user would maintain a local cache of certificates and lists of revoked certificates.

# Authentication Procedure

- It includes 3 alternative authentication prosedures.
- One-Way Authentication: involves a single transfer of information from one user (A) to another (B), and establishes the following:

1.The identity of A and that the message was generated by A.

2.That the message was intended for B.

3.The integrity and originality (it has not been sent multiple times) of the message

- Note that only the identity of the initiating entity is verified in this process, not that of the responding entity.

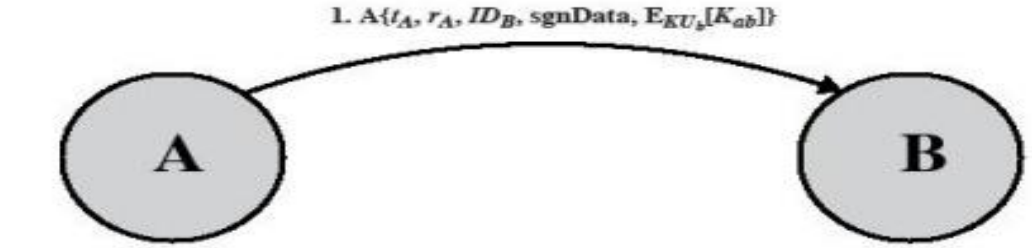
- Two-Way Authentication: In addition to the three elements just listed, two authentication establishes the following elements:

4. The identity of B and that the reply message was generated by B.

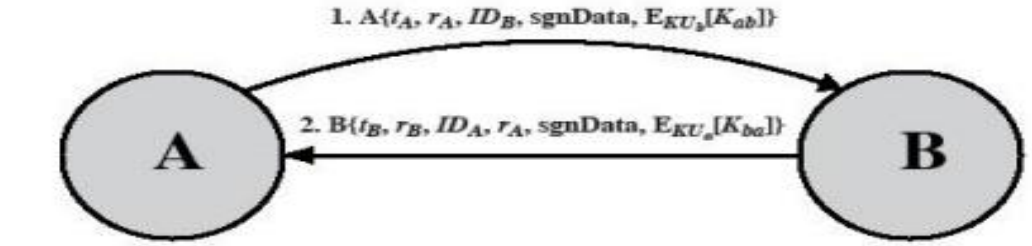
5. That the message was intended for A

6. The integrity and originality of the reply.

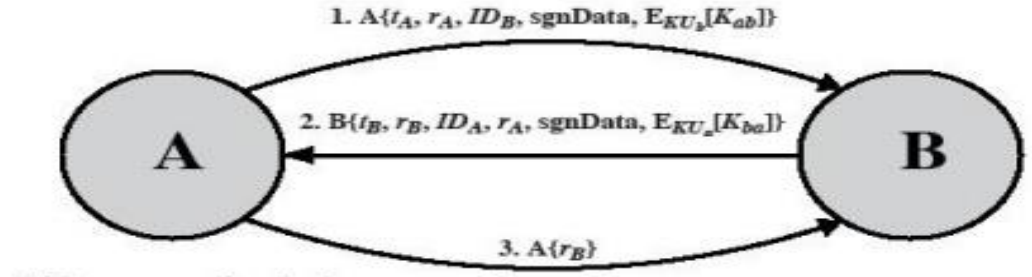
- Two way authentication thus permits both parties in a communication to verify identity of the other.
- Three way Authentication :In three-way authentication, a final message from A to B is included, which contains a signed copy of the nonce $r_B$. The intent of this design is that Timestamps need not be checked. Because both nonces are echoed by the other side, each side can check the returned nonce to detect replay attacks.




(a) One-way authentication




(b) Two-way authentication




(c) Three-way authentication

**Fig.4.4.1 X509 Strong Authentication Procedures**

**Applications of X.509 Authentication Service Certificate:**

- Many protocols depend on X.509 and it has many applications, some of them are given below:
- Document signing and Digital signature
- Web server security with the help of Transport Layer Security (TLS)/Secure Sockets Layer (SSL) certificates
- Email certificates
- Code signing
- Secure Shell Protocol (SSH) keys
- Digital Identities