# Agenda

• Key distribution using asymmetric encryption

* Public-key certificates
* Public-key distribution of secret keys
* Certification Authority and X.509

# Public-key certification

• motivation: Trudy plays pizza prank on Bob

* Trudy creates e-mail order:

*Dear Pizza Store, Please deliver to me four pepperoni pizzas. Thank you, Bob*

* Trudy signs order with her private key
* Trudy sends order to Pizza Store
* Trudy sends to Pizza Store her public key, but says it’s

Bob’s public key

* Pizza Store verifies signature; then delivers four pepperoni pizzas to Bob
* Bob doesn’t even like pepperoni

# Certification authorities

*certification authority (CA):* binds public key to particular entity, E.

• E (person, router) registers its public key with CA.

* E provides “proof of identity” to CA.
* CA creates certificate binding E to its public key.
* certificate containing E’s public key digitally signed by CA – CA says “this is E’s public key”



# Certification authorities

* when Alice wants Bob’s public key:
  + gets Bob’s certificate (Bob or elsewhere).
  + apply CA’s public key to Bob’s certificate, get

Bob’s public key

Bob’s

public

+

K B key

|  |
| --- |
| digital  signature  (decrypt) |

certificate for Bob’s public key, signed by CA public key

K

B

+

CA

K

CA

+

6

Public Key Distribution of Secret Key

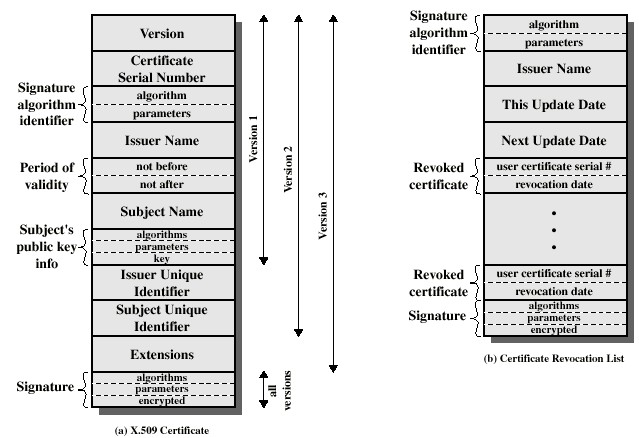
* Prepare a message
* Encrypt that message using conventional encryption using one time session key
* Encrypt the session key using public-key encryption with Alice’s **public key**
* Attach the encrypted session key to the message and send it to Alice
* Only Alice can decrypt the session key
* Bob has obtained Alice’s public key by means of Alice’s **public-key certificate**, must be a valid key

**Note: Important technique used in several protocols**

# 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 CA.
* Is used in S/MIME, IP Security, SSL/TLS and SET.
* RSA is recommended to use but not mandatory.
* Digital Signature is assumed to use Hash algorithm
* Digital Certificate: user’s id, public-key and CA information as input to hash function. Hash is then encrypted with CA’s private key to produce **Digital Certificate**

# X.509 Formats



9

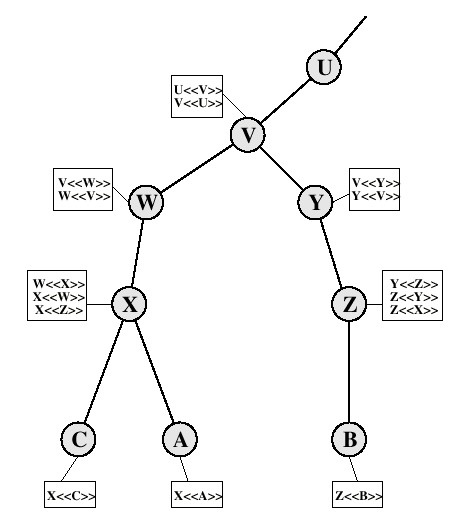
# Distributed Directory

* Users can be registered with a CA and would know its public Key
* Now if A got its certificate from CA X1 and B got it from CA X2.
* If A doesn’t know CA X2’s public key, it can’t trust B’s certificate issued by CA X2.
* However, if the two CA’s have securely exchanged their public keys, then it can work.
  + A obtains the certificate of X2 signed by X1 from directory
  + A knows securely X1’s public key
  + A obtains X2’s public key from its certificate and can verify using X1’s signature on certificate
  + A can now get B’s certificate from CA X2.
  + Since it has trusted public key for CA X2, things work as usual.

Distributed Directory: Certificate Chain

* Notation *Y <<X>>* Certificate of user *X* issued by authority *Y*
* A obtains B’s public key using the following X.509 notation *X1 << X2>> X2 << B>>*
* B obtains A’s public key using the following X.509 notation *X2<< X1>> X1 << A>>*
  + *Arbitrary chain is possible as long as consecutive pair (Xn, Xn+1) of CAs have exchanged certificates securely*

X509 CA Hierarchy



-

-

# Hierarchy of CAs

* Previous figure: Connected Circles hierarchical relationship, boxes shows certificates maintained in each CA’s directory
  + - Forward Certs: Certs of X generated by other

CAs (e.g at circle X, W<<X>>) – PARENT

* + - Reverse Certs: Certs generated by X for others.

(e.g. at circle X, X<<C>> X<<A>>) - CHILD

* A can acquire the following Certs from the directory to establish as certification to B

*X << W>> W <<V>> V <<Y>> Y<<Z>>Z<<B>>*

*(Try to get A’s certificate)*

# Revocation of Certificates

* Reasons for revocation:
  + The users secret key is assumed to be compromised.
  + The user is no longer certified by this CA.
  + The CA’s certificate is assumed to be compromised.
* Client’s may keep a cache every time they check against the revocation list.
* X.509 has a new version 3 with some recommendations for improvement
  + read in your own time if interested

13

# Obtaining a User’s Certificate

* Characteristics of certificates generated by CA:
  + Any user with access to the public key of the CA can recover the user public key that was certified.

oUser can independently calculate hash, decrypt digital certificate using CA’s public key, extract hash and compare if hashes match.

* + No party other than the CA can modify the certificate without this being detected.
* Certificates stored in a Directory server – not part of standard.