COMP 3355
Public Key Infrastructure

K.P. Chow
University of Hong Kong

Public Key Cryptography

- For Alice and Bob from different enterprise and never have met, without sharing any secrets
  - Alice can sign statements that Bob can verify
  - Bob can encrypt things for Alice
  - Alice and Bob can authenticate each other

  - Suppose Bob gets a message with a signature, and a public key
    - Bob can conclude that the signer of that message knew the private key matching the public key
    - So, what's the problem?

How can Bob conclude that the message was signed by **Alice**?

Binding the Public Key with an Identity

- How can Bob believe that a binding exists between the public key and the identity “Alice”?
- Does Alice know the private key and whether anyone else might have know it as well?

  - We want Bob be able to conclude that, because an entity used the private key, the entity has the identity “Alice”

•We need an “infrastructure”

•**Public Key Infrastructure**

What is Public Key Infrastructure (PKI) ?

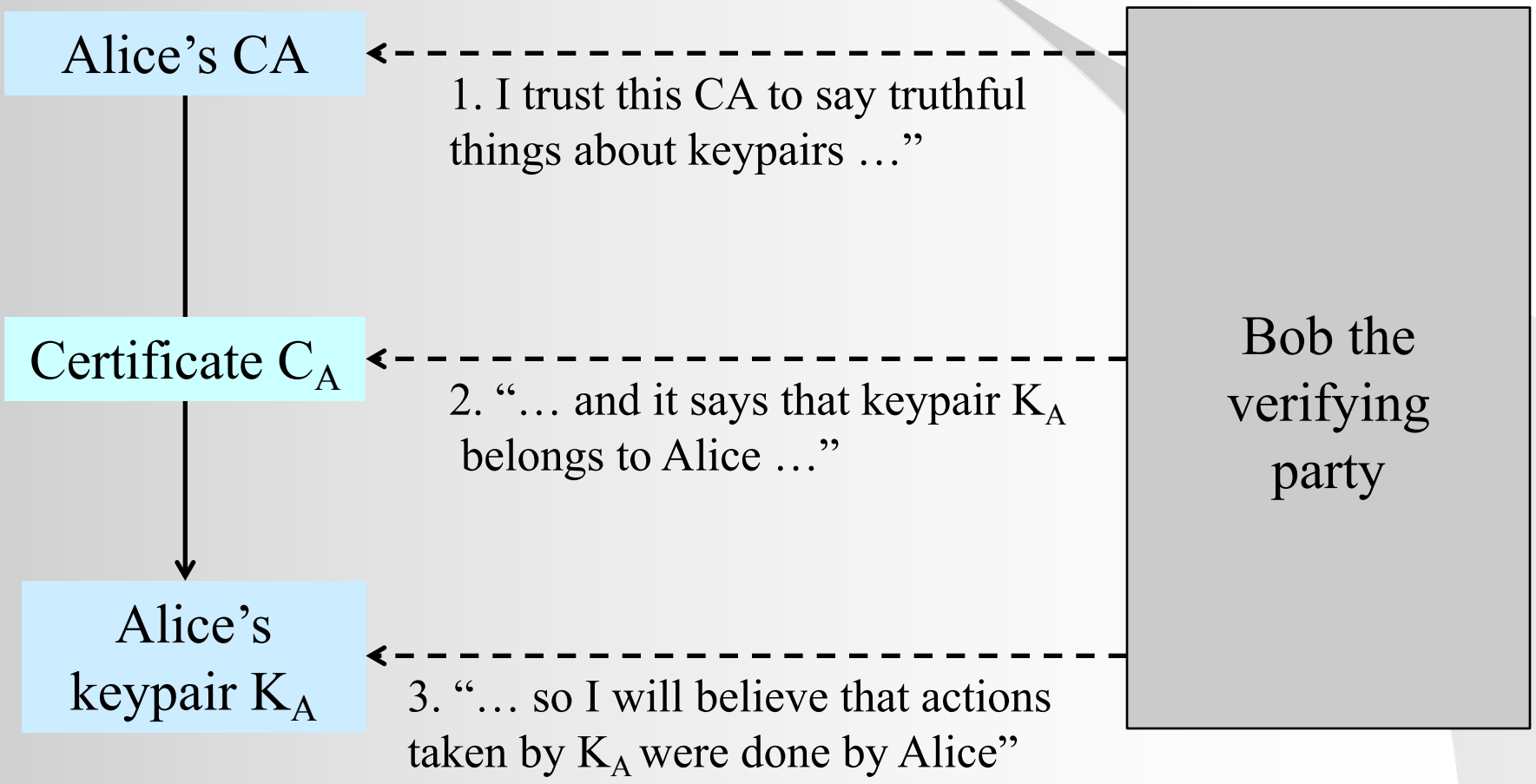
- Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke ***digital certificates***
- PKI is an arrangement that binds public keys with respective user identities by means of a ***certificate authority*** (CA):
  - The binding is established through the registration and issuance process
  - The PKI role that assures this binding is called the Registration Authority (RA): it ensures that the public key is bound to the individual to which it is assigned in a way that ensures non-repudiation
- Law for digital signature, e.g. ETO in Hong Kong

Do you need a law for encryption???

HKSAR Electronic Transaction Ordinance

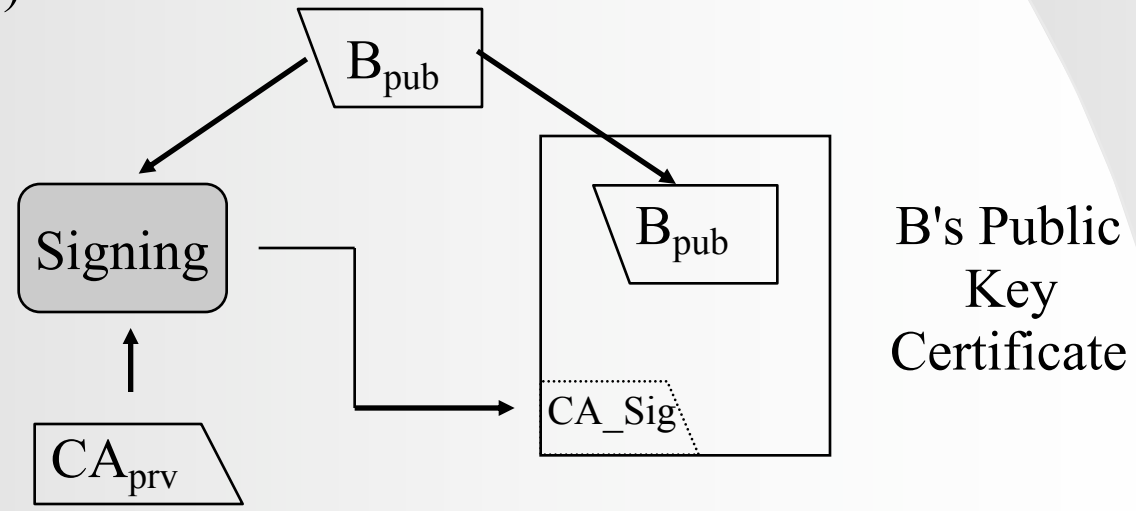
- "asymmetric cryptosystem" means a system capable of generating a secure key pair, consisting of a private key for generating a digital signature and a public key to verify the digital signature
- "digital signature", in relation to an electronic record, means an electronic signature of the signer generated by the transformation of the electronic record using an asymmetric cryptosystem and a hash function such that a person having the initial untransformed electronic record and the signer's public key can determine-
  - (a) whether the transformation was generated using the private key that corresponds to the signer's public key; and
  - (b) whether the initial electronic record has been altered since the transformation was generated;

Basic PKI Architecture



Public Key Certificate (PKC)

- Problems in Public Key Cryptography
  - Private key : users have to keep in secret
  - Public key : make sure everyone can get a correct copy (store in a Public Key Certificate)
- Certification Authority (CA), a trusted third party (e.g. Hong Kong Post CA, VeriSign), says "I, as the CA, certified that B's public key value is 136……., **digitally signed by me**, the CA"
- Needs CA's public key to verify correctness of B's PKC (where to find CA's public key?)



Where is CA's public key? The CA Cert

The CA Cert

CA Certificate: stores CA's public key

e.g. the public key of HK Post

Q: Alice wants to confirm the public key of Bob

CA's public key is 1234

Signed by CA

CA Cert + Bob's PKC

CA's public key is 1234

Signed by CA

Bob's public key is 7890

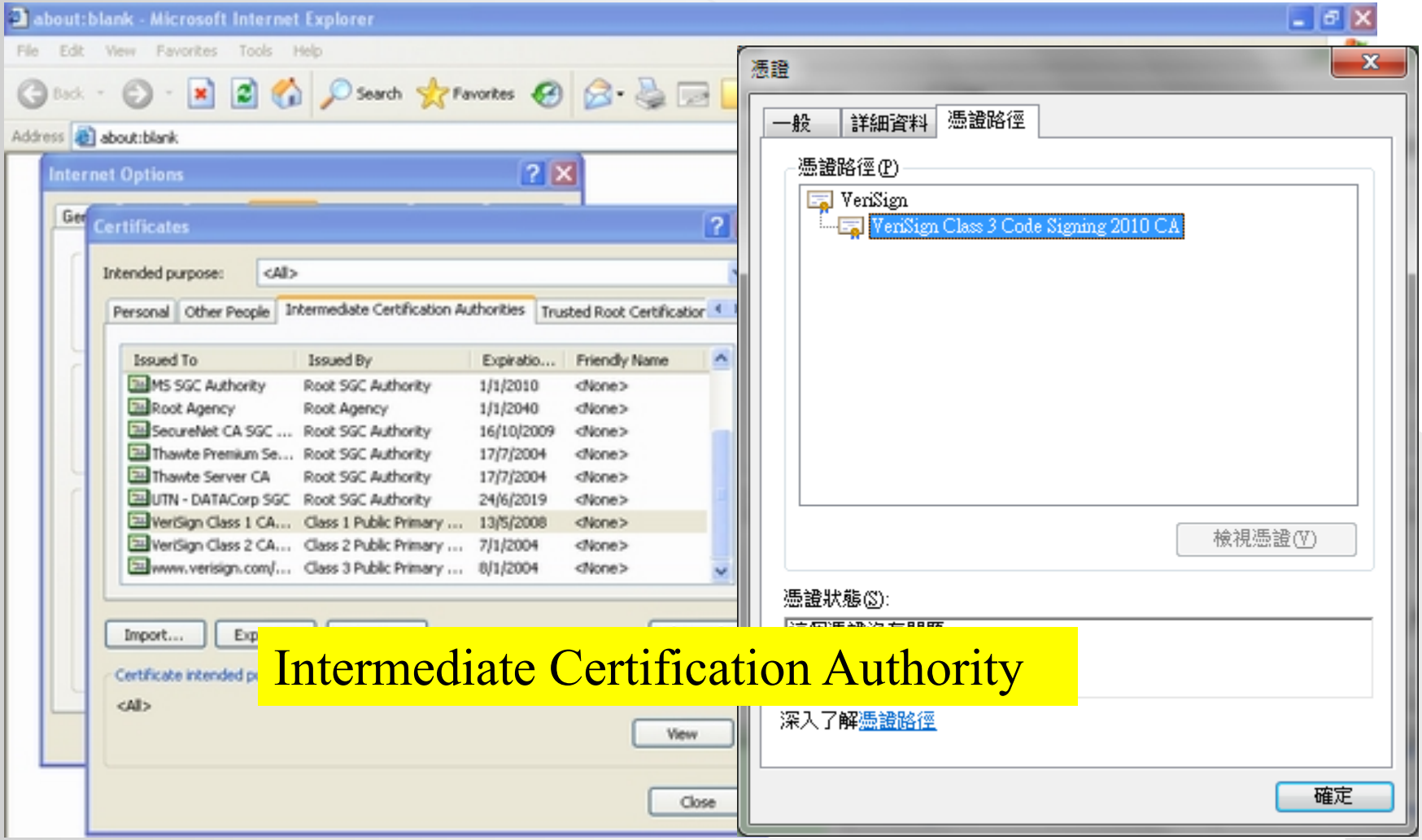
Signed by CA

Alice can confirm Bob's public key is 7890

Root Certificates in IE (A lot!)

Trusted Root Certification Authority

Public Key Certificate in IE

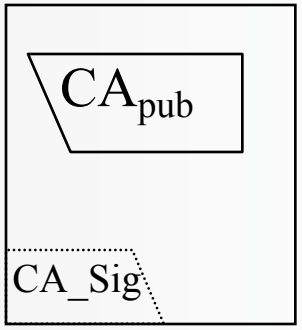


Public Key Certificate (PKC)

- To associate a public-key value to a particular person, device, or entity
- Used to facilitate distribution of public keys
  - User keeps his own private key
  - User keeps his own PKC
  - CA keeps all PKCs for all users
- Each PKC contains a public-key, and the certificate is signed by a "Certification Authority" or some "Intermediate Certification Authority", which has confirmed the identity and other attributes of the holder of the corresponding private key
- Assumption: everyone knows how to verify the CA's digital signature (i.e. everyone knows CA's public key)

Root Certificate

- CA's public key is stored in a special PKC, called "root" cert of the CA
- **Self-signed**
- **Assumption: root certificates are correct!!!**
- How many pre-installed root PKC can you find in the IE or Chrome browser?



CA's 'root' Public Key Certificate

CA's Public Key

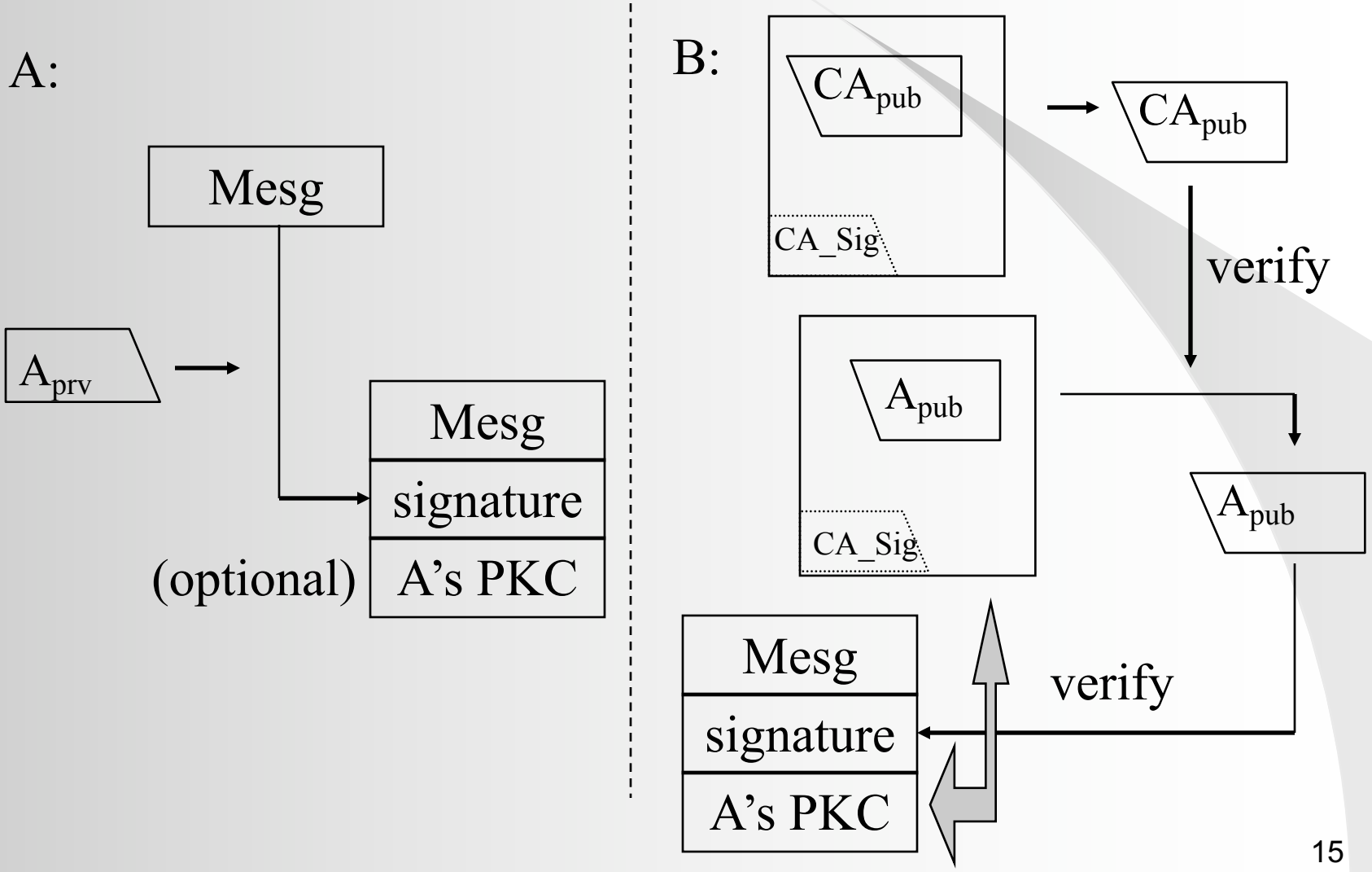
- Certificates can be distributed without any protection, why?
  - No confidentiality needed
  - The certificate contains a digital signature which provides authenticity and integrity
- A large population of users can participate in a PKI system, every user only need to know CA's public key
- The public key (as thus private key) of the CA is extremely important. Usually more secure than a normal user key (e.g. with a longer key length like 2048-bit RSA while normal user key length is 1024-bit)
- Responsible for subject authentication: verification of user id before issuing the PKC

Verifying Signed Message with PKC issued by CA

- Each user in the communication system should possess (at least) one certificate from the CA
- Each certificate contains a public-key value and information that uniquely identifies the certificate's "subject" (a.k.a. a "subscriber" of the CA)
- When B receives a message from A signed by A's private key:
  1. B gets the public key of CA (in the CA certificate)
  2. B gets A's PKC (inside the message from A, a directory service, or …)
  3. B verifies A's PKC by CA's public key, then extracts A's public key from the PKC
  4. B can verify A's digital signature, by A's public key

  – B is called a "relying party"

Certificate usage in digital signature (from A to B)

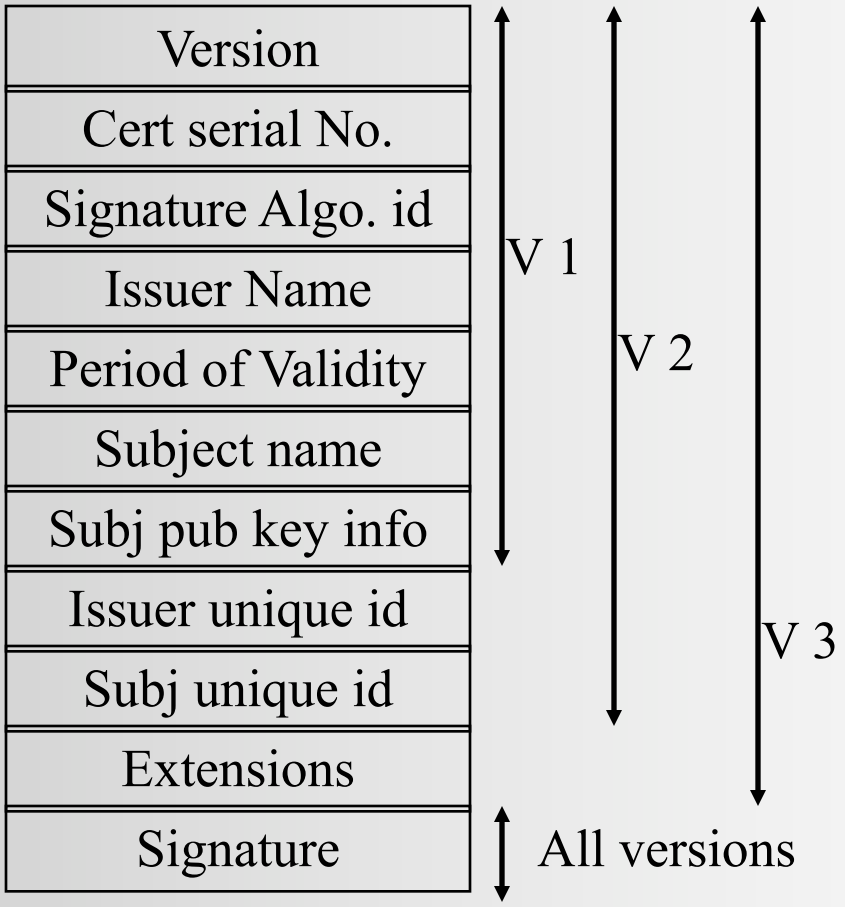


Encrypted Message

- When A wants to send an encrypted message to B:
  1. A gets the public key of CA
  2. A gets B's PKC
  3. A verifies B's PKC by CA's public key, then extracts B's public key
  4. A encrypts the message using B's public key

- When B's receives the message from A
  - B' decrypts the message using B's private key

ITU-T X.509 PKC format



Issuer : the CA

Subject : owner of the public key

Signature : the digital signature by CA

Signature Algo. Id : algo used in “signature”

** Uses ASN.1 (Abstract Syntax Notation 1) encoding*

Validity Period

- A certificate has a life time (just as keys)
- A certificate contains start date/time and expiration date/time
- Expired certificate are only used to verify signature on a old document (e.g. for auditing purpose)
- A new certificate should be issued to the subscriber when his/her old certificate is expired
- In event of suspected key compromise, a new certificate should be issued, and the old certificate should be “revoked” prior to its expiry date

Verifying digital signature

Checking we need to do:

- Verify the digital signature with signer's public key in signer's PKC
- Verify signer's PKC using CA's public key in the root cert
- Confirm the signer's PKC is not yet expired

Key Management

- Major issue in PKI implementation
- Key questions:
  - Who is responsible to generate the key?
  - Should we backup the key?
  - Where the PKC should be stored?
  - Certificate revocation?

Key Generation (1)

- Basic approach
  - Alice generates her own key pair
  - The CA signs a statement about Alice's public key
  - Bob knows the CA's public key
- Any problem?

  Neither Bob nor the CA know the key pair really belongs to Alice

- Improved approach
  - CA generates the key pair for Alice and then issues the key pair to Alice

  The CA knows the key pair belongs to Alice and worry that CA may store or disclose the private key

Key Generation (2)

- Another improved approach
  - Alice generates her own key pair
  - Alice brings the public key to the CA
  - The CA signs a statement about Alice's public key

Alice can be sure that CA doesn't know her private key. CA might worry that Alice doesn't really know the private key

- How about
  - Alice generates her own key pair
  - Alice brings the public key to the CA and some signed request showing that she knows the private key

Known as "Certificate Request" in the PKI

Common Systems in Key-pair Generation

- Key-pair holder system
  - private key owner performs the generation
  - private key never goes to other places
  - best for non-repudiation requirement
- Central system
  - private key has to be transported to the owner
  - key-pair generated in central Registration Authority (RA) associated with CA, RA generates a key on behalf of an end-user
  - central site specialized in key generation, more resources, better algorithm, stronger control, etc.
  - related functions like key archival can provided

Certificate generation steps

- CA is presented with the required certificate content information from the “applicant”
- CA verifies the accuracy of the information
- The certificate is created, and signed by a signing device using the CA’s private key
- A copy of the certificate is sent to the subscriber
- A copy of certificate may be submitted to some directory services
- CA records the certificate generation process in the audit journal

Registration Authorities (RA)

- Supporting role to CA: support subject authentication
- RA does not issue certificates
- Key functions of RA:
  - Registering, de-registering, and changing attributes of subscribers
  - Authenticating subscribers
  - Authorizing requests for key-pair or certificate generation, or recovering backed up keys
  - Accepting and authorizing requests for certificate suspension or revocation
  - physically delivering tokens to the authorized users, and recovering obsolete tokens from users

Private Key Protection

- Stored in a tamper-resistant hardware token, e.g. smart card, PCMCIA card.
- Stored in encrypted date file, with authentication control by PIN, the encryption key is derived from the PIN
- Hardware token is more expensive and more secure

Backup of Digital Signature Key-pairs

- No party other than the holder of private key should be able to access the private key (in ANSI X9.57, it requires the private key be NEVER leave the device it is stored)
- No backup or archival is needed for a digital signature private key. In fact it is better to ensure that the digital signature private key is destroyed securely after its life time.
- The public key certificate together with the stored public key for digital signature verification should be properly archived

Backup of Encryption Key-pairs

- **Decryption** private key should be archived properly ✗
- Public key certificates are usually archived

How certificates are distributed?

- Certificates are digitally signed
- Directory Service
  - A data base of (valid and update) certificates
  - ISO standard: X.500
  - Proprietary directory service such as Microsoft Exchange, Lotus Notes, Novell NDS
  - Internet Lightweight Directory Access Protocol (LDAP)
- Certificates can be distributed through insecure channels such as email (S/MIME and MOSS) or WWW
  - Protected by the digital signature of CA

Certificate Revocation

- Reasons for revocation:
  - detected or suspected key compromise
  - change of data (e.g. subject name)
  - change of relationship between subject and CA (e.g. employee quitting a job from an organization which uses the current CA)
- Who can revoke?
  - The subject
  - The CA
  - An authorized third party

Certificate Revocation List (CRLs)

- A time-stamped list of revoked certs, digitally signed by the CA, available to all users
- Each revoked cert is identified by a certificate serial number
- Users of public key certificates should checks a “suitably-recent” CRL
  - Problem: what is “suitably-recent”?
- CRL will not contain certificates that are expired
- CRLs are distributed regularly, e.g. hourly, daily, etc.
- CRLs are digitally signed, thus can be sent via unprotected channels

X.509 CRL format

- Version : 1 or 2, v2 contains CRL entry extension and CRL extension
- Signature Algo ID : indicator of algorithm signing this CRL
- Issuer : the creator of this CRL, most likely the CA
- This update : date/time of issue of this CRL
- Next update : date/time of issue of next CRL
- List of revoked certs:
  - user certificate : cert serial number
  - revocation date
  - CRL entry extension : additional info

Digital signature verification with CRL

- When A sends B a message signed by A's private key:
  1. B gets A's PKC (from A, a directory service, or others)
  2. B gets the public key of CA
  3. B gets the CRL
  4. *B checks A's cert is not revoked (use the CRL)*
  5. B verifies A's PKC by CA's public key, then extracting A's public key from A's PKC
  6. B verifies A's digital signature, by A's public key

Verifying digital signature

Checking we need to do:

- Verify the digital signature with signer's public key in signer's PKC
- Verify signer's PKC using CA's public key in the root cert
- Confirm the signer's PKC is not yet expired
- Check the signer's PKC is not yet revoked (using the CRL)
- Check the digital signature of the CRL

Certification Path

- The *certificate chain* is a list of certificates used to authenticate an entity
- The chain begins with the certificate of that entity, and each certificate in the chain is signed by the entity identified by the next certificate in the chain
- The chain terminates with a root CA certificate: a certificate signed by the CA itself
- The signatures of all certificates in the chain must be verified until the root CA certificate is reached

Example Certification Path