Public-key Infrastructure

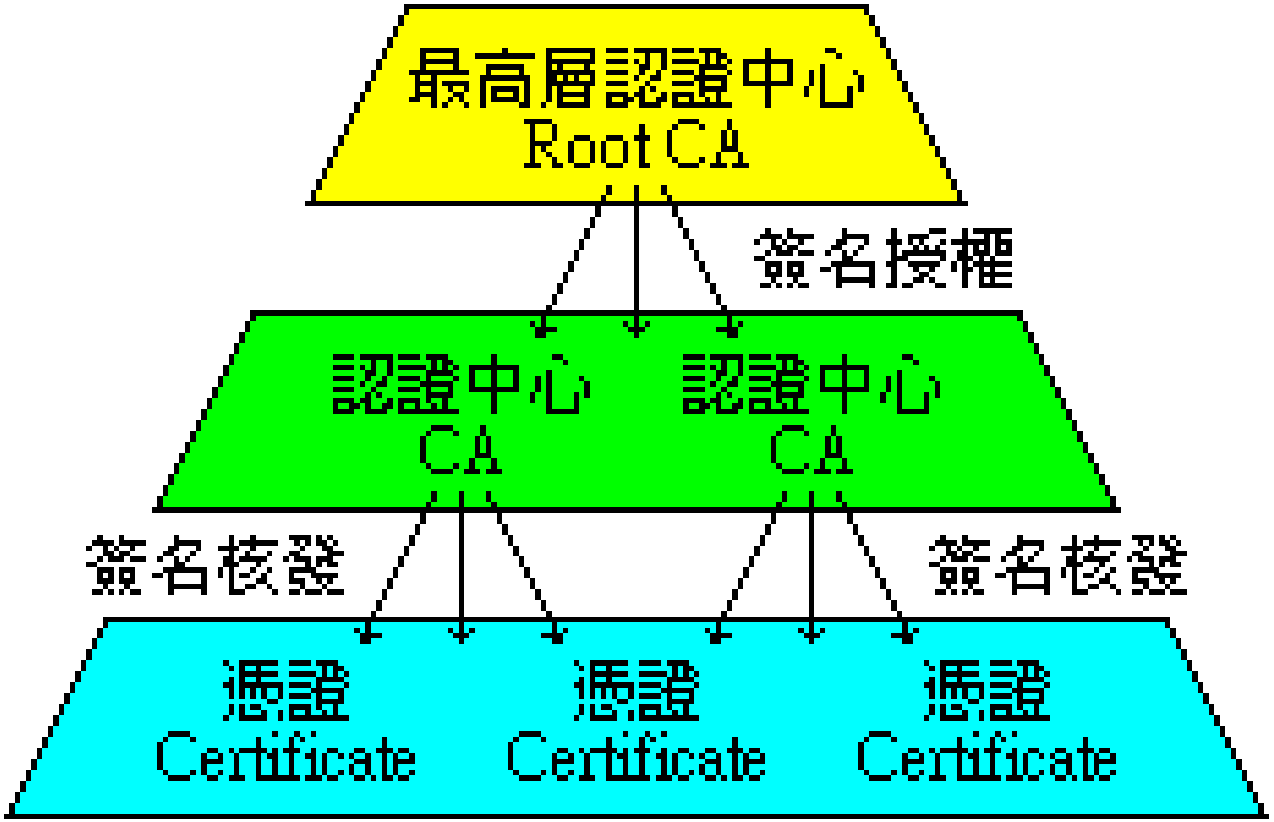
Public-key Infrastructure

- A set of hardware, software, people, policies, and procedures.
- To create, manage, distribute, use, store, and revoke digital certificates.

- Encryption, authentication, signature
- Bootstrapping secure communication protocols.

CA: Certificate Authority (1)

- In God We Trust



CA: Certificate Authority (2)

- Certificate
  - Contains data of the owner, such as Company Name, Server Name, Name, Email, Address,...
  - Public key of the owner.
  - Followed by some digital signatures.
    - Sign for the certificate.
  - In X.509
    - A certificate is signed by a CA.
    - To verify the correctness of the certificate, check the signature of CA.

CA: Certificate Authority (3)

- Certificate Authority (CA)
  - “憑證授權” in Windows CHT version.
  - In X.509, it is itself a certificate.
    - The data of CA.
    - To sign certificates for others.
  - Each CA contains a signature of Root CA.
  - To verify a valid certificate
    - Check the signature of Root CA in the certificate of CA.
    - Check the signature of CA in this certificate.
  - Reference: http://www.imacat.idv.tw/tech/sslcerts.html

What is a CA ? (1)

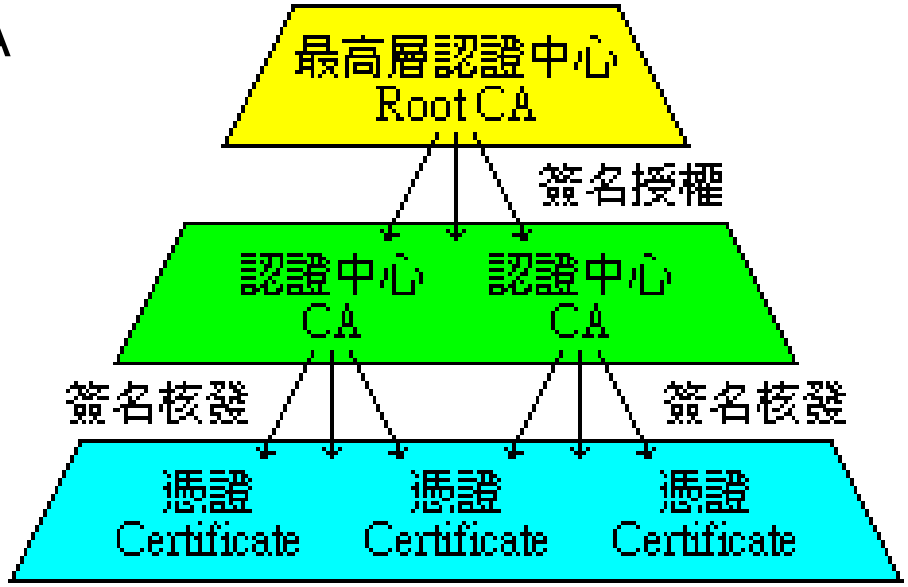
- *Certificate Authority* (認證中心)
- Trusted server which signs certificates
- One private key and relative public key
- Tree structure of X.509
  - *Root CA*

What is a CA ? (2)

- Root CA (最高層認證中心)
  - In Microsoft:「根目錄授權憑證」
  - Root CA do not sign the certificates for users.
    - Authorize CA to sign the certificates for users, instead.
  - Root CA signs for itself.
    - It is in the sky.
  - To trust Root CA
    - Install the certificate of Root CA via secure channel.

What is a CA ? (3)

- Tree structure of CA

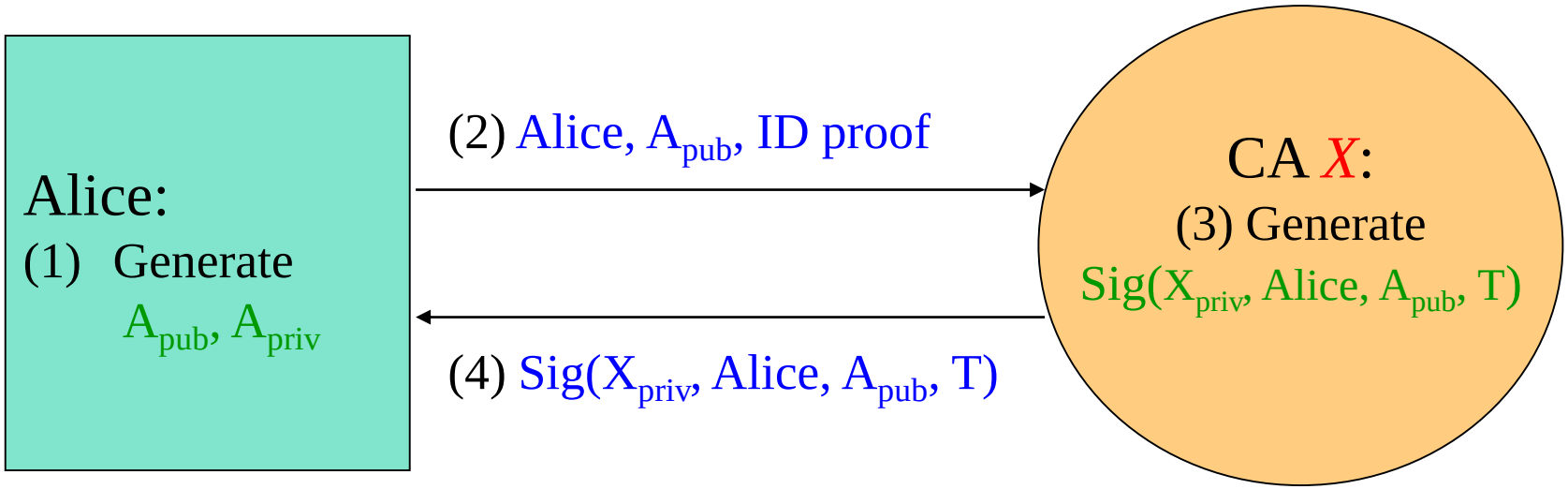


- Cost of certificate
  - HiTrust : NT *$30,000* / per year / per domain
  - TWCA: NT *$30,000* / per year / per domain
  - GoGetSSL (Comodo): USD 7.85 / per year / per domain
  - GoGetSSL (Comodo Wildcard): USD 70.95 / per year
  - Myself : NT *$0*

Certificate (1)

- Digital Certificate, Public-key Certificate, Network Identity
- A certificate is issued by a CA *X*
- A certificate of a user A consists:
  - The name of the issuer CA *X*
  - His/her public key A_{pub}
  - The signature $Sig(X_{priv}, A, A_{pub})$ by the CA *X*
  - The expiration date
  - Applications
    - Encryption / Signature

Certificate (2)



$Cert_{A,X}$=[Alice, A_{pub}, Sig(X_{priv}, Alice, A_{pub}, T)]

Note: CA does not know A_{priv}

Certificate (3)

- Guarantee of CA and certificate
  - Guarantee the public key is of *someone*
  - *Someone* is not guaranteed to be *safe*
- Security of transmitting DATA
  - Transmit *session key* first
    - *Public-key cryptosystem*
  - Transmit DATA by session key
    - *Symmetric-key cryptosystem*

OpenSSL

OpenSSL

- http://www.openssl.org/
- In system
  - /usr/src/crypto/openssl
- In pkg
  - openssl

Example: Apache SSL settings

Example: Apache SSL settings – Flow

- Flow
  - Generate random seed
  - Generate RootCA
    - Generate private key of RootCA
    - Fill the Request of Certificate.
    - Sign the certificate itself.
  - Generate certificate of Web Server
    - Generate private key of Web Server
    - Fill the Request of certificate
    - Sign the certificate using RootCA
  - Modify apache configuration ➔ restart apache

Example: Apache SSL settings – Generate random seed

- openssl rand -out rnd-file num

  % openssl rand -out /etc/ssl/RootCA/private/.rnd 1024

- chmod go-rwx rnd-file

  % chmod go-rwx /etc/ssl/RootCA/private/.rnd

Example: Apache SSL settings – Certificate Authority (8)

- Include etc/apache22/extra/httpd-ssl.conf

```
SSLEngine on
SSLHonorCipherOrder on
SSLCompression off
SSLSessionTickets Off
SSLCipherSuite SSLCipherSuite "ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-
SHA384:ECDHE-RSA-AES256-SHA:DHE-RSA-AES256-SHA:AES256-GCM-SHA384"
SSLCertificateFile /etc/ssl/nasa/nasa.crt.pem
SSLCertificateKeyFile /etc/ssl/nasa/private/nasa.key.pem

# OCSP Stapling, only in httpd 2.3.3 and later
  SSLUseStapling on
  SSLStaplingResponderTimeout 5
  SSLStaplingReturnResponderErrors off

# HSTS (mod_headers is required) (15768000 seconds = 6 months)
  Header always add Strict-Transport-Security "max-age=15768000; preload"
# Public Key Pinning (HPKP)
##Header set Public-Key-Pins "pin-
sha256=\"klO23nT2ehFDXCfx3eHTDRESMz3asj1muO+4aIdjiuY=\"; pin-
sha256=\"633lt352PKRXbOwf4xSEa1M517scpD3l5f79xMD9r9Q=\"; max-age=2592000;
includeSubDomains"
```

Appendix: PGP

PGP

- Pretty Good Privacy
- Public key system
  - Encryption
  - Signature
- security/gnupg
- Will talk more in Network Administration
- Ref: http://security.nknu.edu.tw/textbook/chap15.pdf