



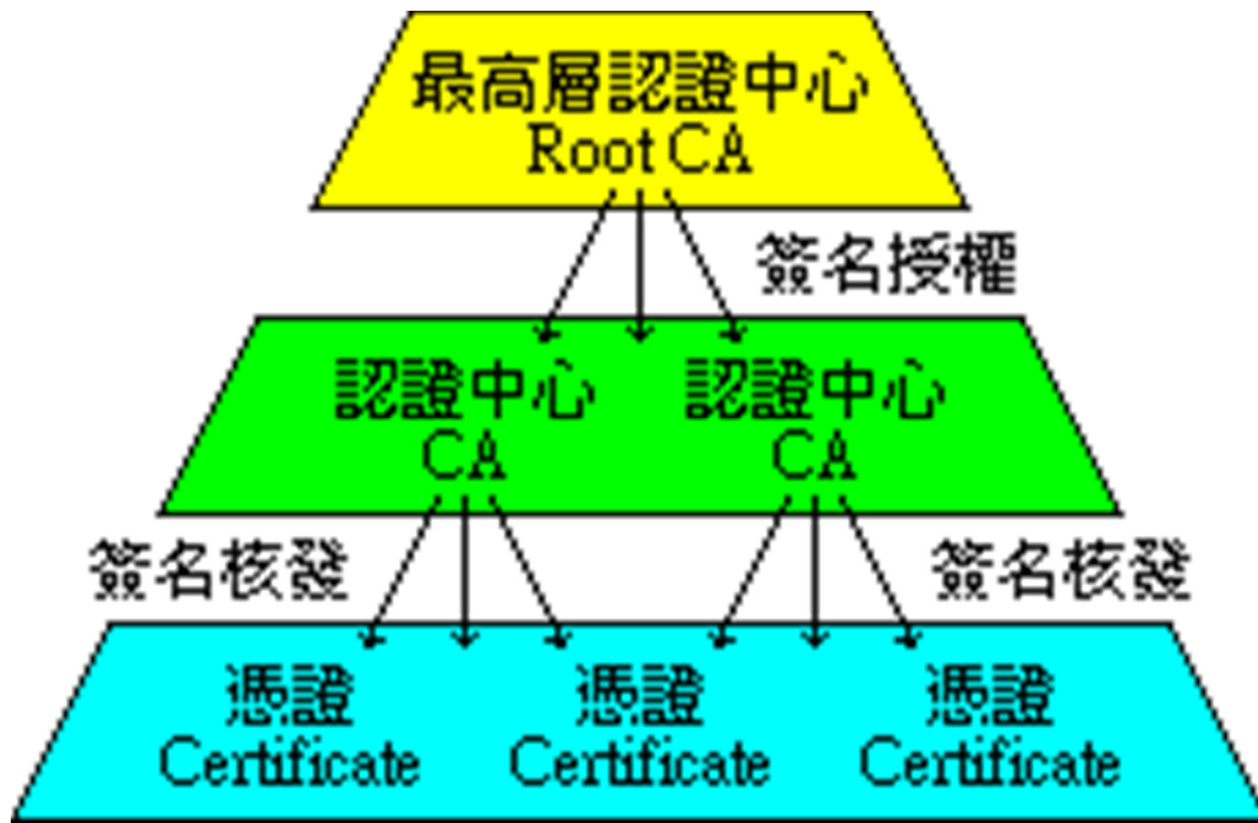
Public-key Infrastructure

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*

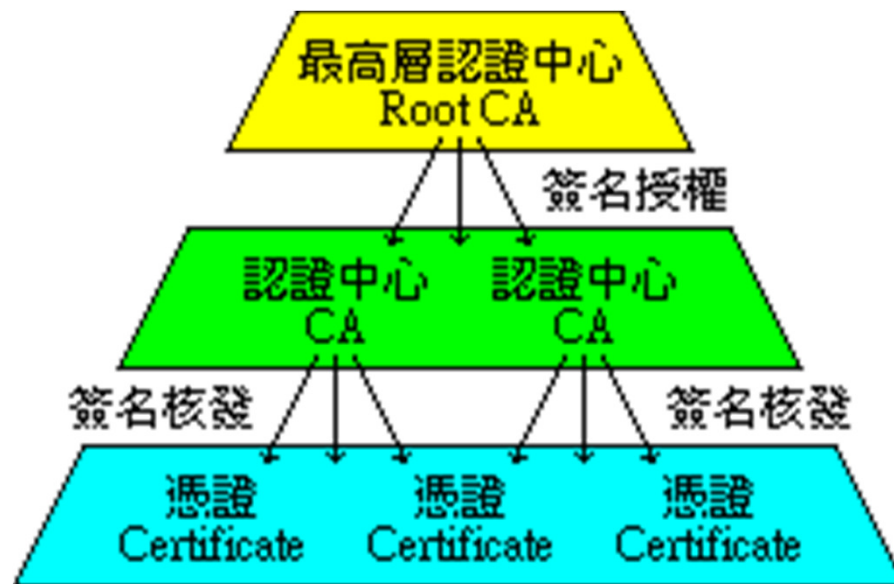
What is a CA ? (2)

□ Root CA (最高層認證中心)

- In Micro\$oft: 「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



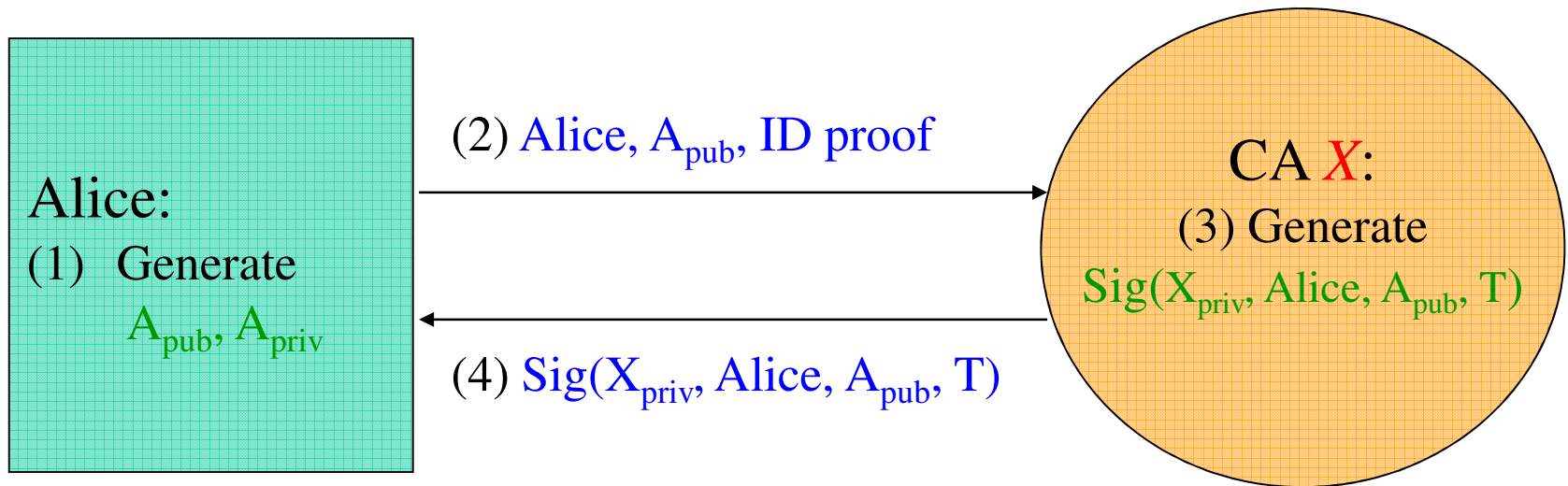
□ Cost of certificate

- HiTrust : NT \$**30,000** / per year / per host
- 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 $\text{Sig}(X_{\text{priv}}, A, A_{\text{pub}})$ by the CA X
 - The expiration date
 - Applications
 - Encryption / Signature

Certificate (2)



$$\text{Cert}_{A,X} = [\text{Alice}, A_{pub}, \text{Sig}(X_{priv}, \text{Alice}, A_{pub}, T)]$$

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 ports
 - security/openssl

ports/Mk/bsd.openssl.mk

```
# WITH_OPENSSL_BASE=yes - Use the version in the base system.
# WITH_OPENSSL_PORT=yes - Use the port, even if base is up to date
# WITH_OPENSSL_BETA=yes - Use a snapshot of recent openssl
# WITH_OPENSSL_STABLE=yes - Use an older openssl version
.....
# if no preference was set, check for an installed base version
# but give an installed port preference over it.
.if !defined(WITH_OPENSSL_BASE) && \
    !defined(WITH_OPENSSL_BETA) && \
    !defined(WITH_OPENSSL_PORT) && \
    !defined(WITH_OPENSSL_STABLE) && \
    !exists(${DESTDIR}/${LOCALBASE}/lib/libcrypto.so) && \
    exists(${DESTDIR}/usr/include/openssl/opensslv.h)
WITH_OPENSSL_BASE=yes
.endif
```

A decorative graphic on the left side of the slide, consisting of a vertical blue bar with a gradient from dark blue at the top to light blue at the bottom, and a horizontal blue bar at the top left corner.

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 – Generate private key of RootCA

❑ `openssl genrsa -des3 -rand rnd-file -out rootca-key-file num`

`% openssl genrsa -des3 -rand /etc/ssl/RootCA/private/.rnd \
-out /etc/ssl/RootCA/private/rootca.key.pem 2048`

- Note: phrase are asked (something like password)

❑ `chmod go-rwx rootca-key-file`

`% chmod go-rwx /etc/ssl/RootCA/private/rootca.key.pem`

Example: Apache SSL settings – Fill the Request of Certificate

❑ `openssl req -new -key rootca-key-file -out rootca-req-file`

```
% openssl req -new -key /etc/ssl/RootCA/private/rootca.key.pem \  
-out /etc/ssl/RootCA/private/rootca.req.pem
```

❑ `chmod go-rwx rootca-req-file`

```
% chmod go-rwx /etc/ssl/RootCA/private/rootca.req.pem
```

Enter pass phrase for rootca-key-file:

```
Country Name (2 letter code) [AU]:TW  
State or Province Name (full name) [Some-State]:Taiwan  
Locality Name (eg, city) []:HsinChu  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NCTU  
Organizational Unit Name (eg, section) []:CS  
Common Name (eg, YOUR name) []:nasa.cs.nctu.edu.tw  
Email Address []:liuyh@cs.nctu.edu.tw
```

A challenge password []: **(No need , Enter please)**

An optional company name []: **(Enter please)**

Example: Apache SSL settings – Sign the certificate itself

- ❑ `openssl x509 -req -days num -sha1 -extfile path_of_openssl.cnf -extensions v3_ca -signkey rootca-key-file -in rootca-req-file -out rootca-crt-file`

```
% openssl x509 -req -days 5109 -sha1 -extfile /etc/ssl/openssl.cnf  
-extensions v3_ca -signkey /etc/ssl/RootCA/private/rootca.key.pem  
-in /etc/ssl/RootCA/private/rootca.req.pem -out  
/etc/ssl/RootCA/private/rootca.crt.pem
```

- ❑ `rm -f rootca-req-file`

```
%rm -f /etc/ssl/RootCA/private/rootca.req.pem
```

- ❑ `chmod go-rwx rootca-crt-file`

```
%chmod go-rwx /etc/ssl/RootCA/private/rootca.crt.pem
```

Example: Apache SSL settings – Generate private key of Web Server

❑ `openssl genrsa -out host-key-file num`

`%openssl genrsa -out /etc/ssl/nasa/private/nasa.key.pem 2048`

❑ `chmod go-rwx host-key-file`

`%chmod go-rwx /etc/ssl/nasa/private/nasa.key.pem`

Example: Apache SSL settings – Fill the Request of Certificate

- ❑ `openssl req -new -key host-key-file -out host-req-file`
`% openssl req -new -key /etc/ssl/nasa/private/nasa.key.pem -out /etc/ssl/nasa/private/nasa.req.pem`
- ❑ `chmod go-rwx host-req-file`
`% chmod go-rwx /etc/ssl/nasa/private/nasa.req.pem`

Example: Apache SSL settings – Sign the certificate using RootCA

❑ Transmit host-req-file to Root CA, and do following steps in RootCA

- openssl x509 -req -days num -sha1 -extfile path_of_openssl.cnf -extensions v3_ca -CA rootca-crt-file -CAkey rootca-key-file -CAserial rootca-srl-file -CAcreateserial -in host-req-file -out host-crt-file

```
% openssl x509 -req -days 365 -sha1 -extfile /etc/ssl/openssl.cnf  
-extensions v3_ca -CA /etc/ssl/RootCA/private/rootca.crt.pem  
-CAkey /etc/ssl/RootCA/private/rootca.key.pem  
-CAserial /etc/ssl/RootCA/private/rootca.srl -CAcreateserial  
-in /etc/ssl/nasa/private/nasa.req.pem  
-out /etc/ssl/nasa/private/nasa.crt.pem
```

- rm -f host-req-file (in both RootCA and Web Server)
% rm -f /etc/ssl/nasa/private/nasa.req.pem
- Transmit host-crt-file back to Web Server

Example: Apache SSL settings – Certificate Authority (8)

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

```
##  
## SSL Virtual Host Context  
##  
<VirtualHost _default_:443>  
# General setup for the virtual host  
DocumentRoot /home/wwwadm/data  
<Directory "/home/wwwadm/data">  
    Options Indexes FollowSymLinks  
    AllowOverride All  
    Order allow,deny  
    Allow from all  
</Directory>  
ServerName nasa.cs.nctu.edu.tw:443  
ServerAdmin liuyh@nasa.cs.nctu.edu.tw  
ErrorLog /var/log/httpd/nasa.cs-error.log  
CustomLog /var/log/httpd/nasa.cs-access.log common  
  
SSLEngine on  
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:!SSLv2:+EXP:+eNULL  
SSLCertificateFile /etc/ssl/nasa/nasa.crt.pem  
SSLCertificateKeyFile /etc/ssl/nasa/private/nasa.key.pem
```


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>