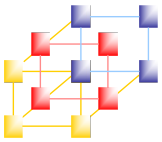
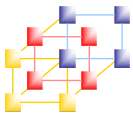
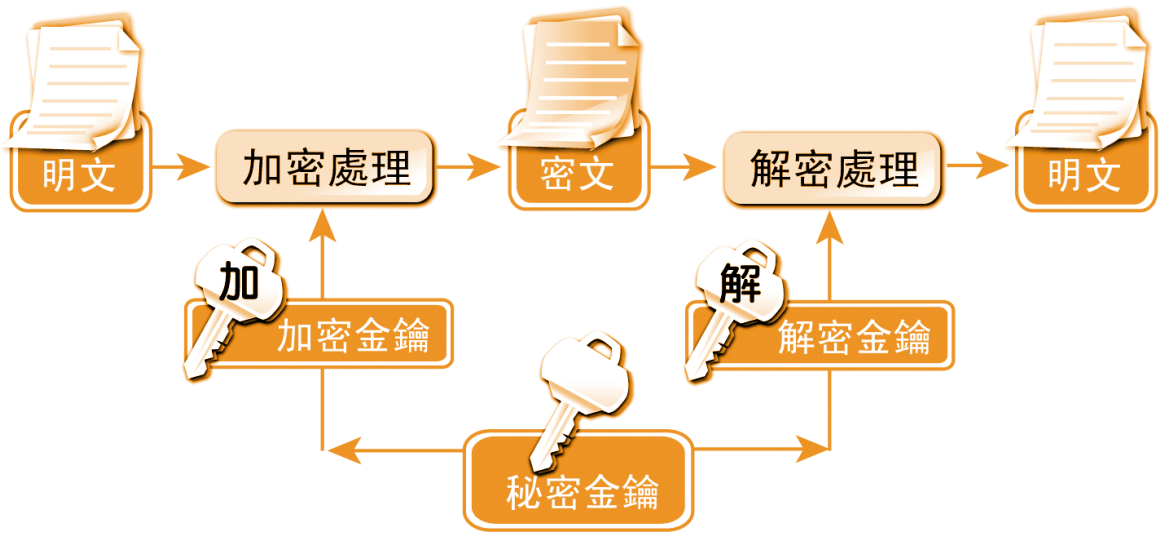


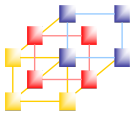
Unit 7
Secure Socket



密碼學基本概念



基本的加解密系統



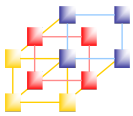
密碼系統的分類

- 對稱性密碼系統(Symmetric Cryptosystems)或秘密金鑰密碼系統(Secret-Key Cryptosystems) 或單金鑰密碼系統(One-Key Cryptosystems)

  加密金鑰及解密金鑰為同一把

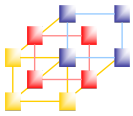
- 非對稱性密碼系統(Asymmetric Cryptosystems)或公開金鑰密碼系統(Public-Key Cryptosystems) 或雙金鑰密碼系統(Two-Key Cryptosystems)

  加密與解密金鑰為不相同的二把金鑰



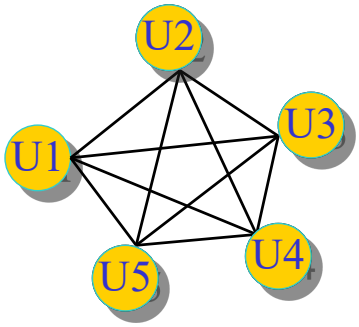
對稱式加解密法

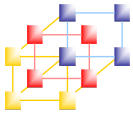
- DES (Data Encryption Standard)
- Triple DES
- AES (Advanced Encryption Standard)
- …



公開金鑰基本概念

- 對稱式密碼系統有金鑰的管理問題
  - 例如要與N個人做秘密通訊，那麼就必須握有N把秘密金鑰
- 為了改善對稱式密碼系統問題，於是便有公開金鑰密碼系統(Public-Key Cryptosystems)的產生

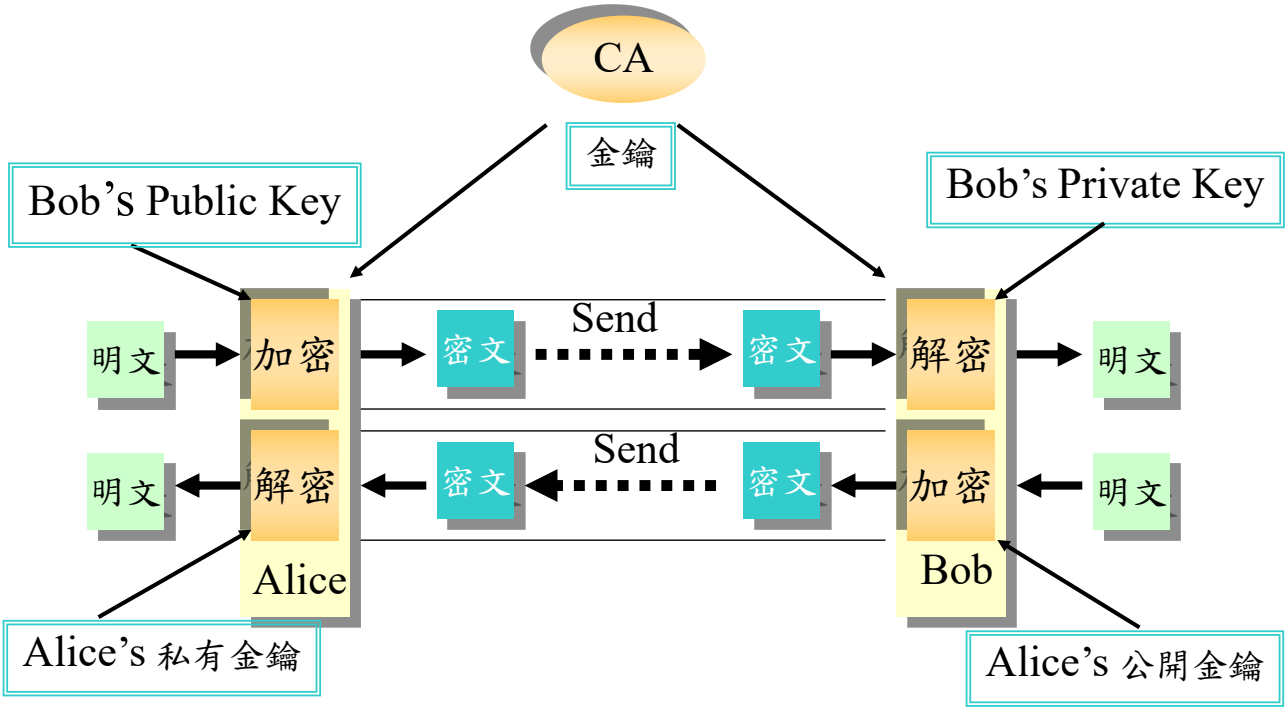




公開金鑰密碼系統

- 著名之公開密碼系統
  - RSA密碼系統
  - ElGamal密碼系統
  - Elliptic Curve Cryptosystem, ECC橢圓曲線的密碼系統
- 公開密碼系統優點
  - 沒有金鑰管理的問題
  - 高安全性
  - 有數位簽章功能
- 公開密碼系統缺點
  - 加解密速度慢

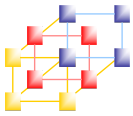
公開金鑰加密系統



RSA 加密法

- 非對稱式密碼系統的一種。
  - 1978年美國麻省理工學院三位教授Rivest、Shamir、Adleman (RSA) 所發展出來的。
- 利用公開金鑰密碼系統作為資料加密的方式，可達到資料加密及數位簽署的功能。

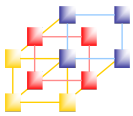
- Encryption
  - RSA 加密演算法，明文加密使用區塊為每次加密的範圍，使用對方公開金鑰 (Public Key) 將明文加密。
- Decryption
  - RSA 解密演算法，必須使用自己的私有金鑰 (Private Key) 才能將密文解出。



RSA 演算法

- 張三選 2 個大質數 p 和 q (至少100位數)，
  令 N = p ・ q
- 再計算Ø(N)=(p-1)(q-1)，並選一個與Ø(N)互質數 e
  - Ø(N)為Euler's Totient函數，其意為與N互質之個數
- (e, N) 即為張三的公開金鑰
- 加密法為 $C = M^e \bmod N$
- 張三選 1 個數 d，滿足 e ・ d mod Ø(N) = 1
- d 即為張三的解密金鑰(亦稱私有金鑰或祕密金鑰)
- 解密法為 $M = C^d \bmod N$

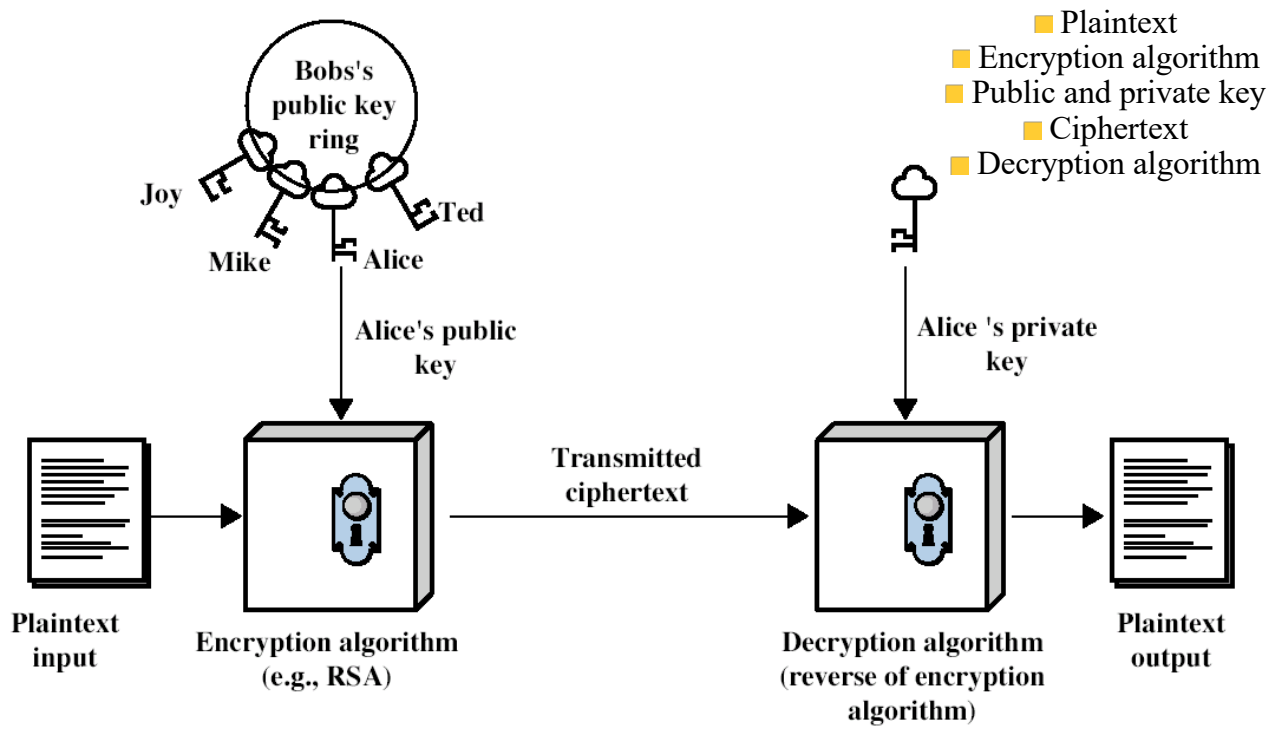
• RSA之安全性取決於質因數分解之困難度
• 要將很大的N因數分解成P跟Q之相乘，是很困難的



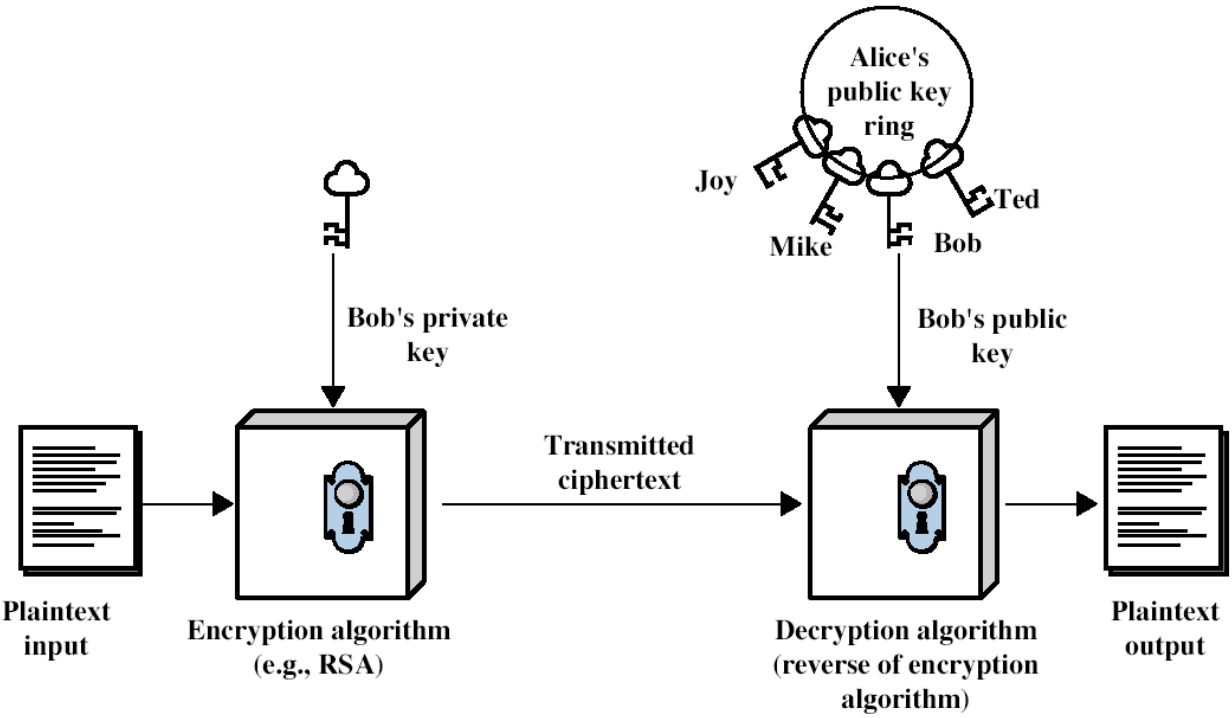
RSA 演算法- 例子

- 張三選 p = 3 ， q = 11
  此時 N = p ・ q = 3 x 11 = 33
- 張三選出一個與 (p-1) x (q-1) = (3-1)(11-1) = 20互質數 e = 3
- (e, N) = (3,33) 即為張三的公開金鑰
- 張三選一個數 d = 7 當作解密金鑰，
  滿足 e ・ d ≡ 1 mod 20 (7 x 3 ≡ 1 mod 20)
- 令明文 M = 19
  - 加密 : $C = M^e \bmod N = 19^3 \bmod 33 = 28$
  - 解密 : $M = C^d \bmod N = 28^7 \bmod 33 = 19$

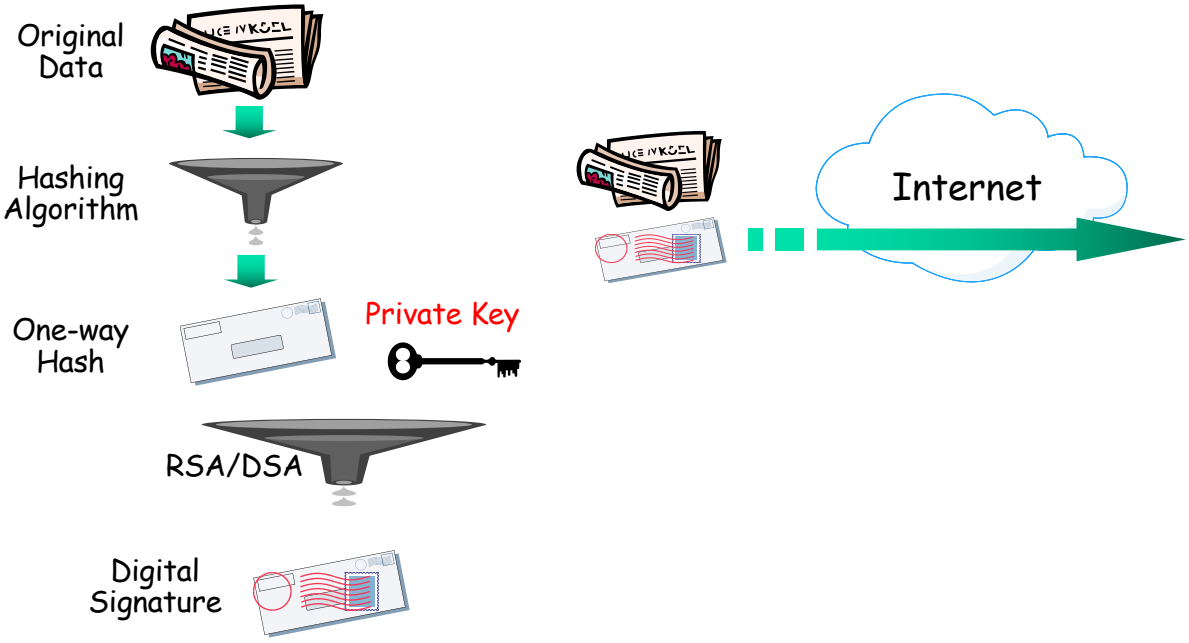
Public-Key Cryptography -- Encryption



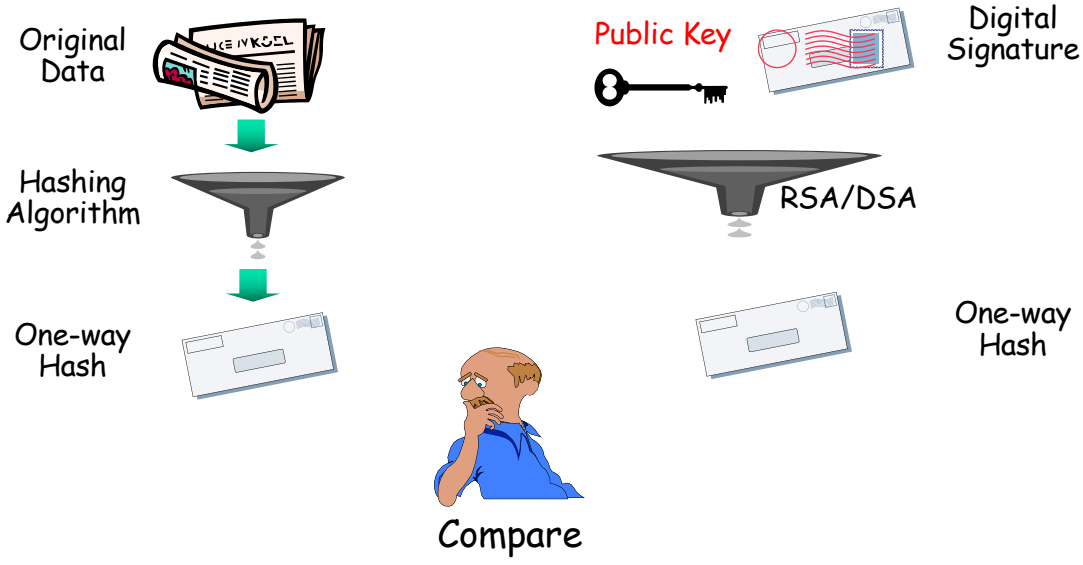
Public-Key Cryptography -- Authentication



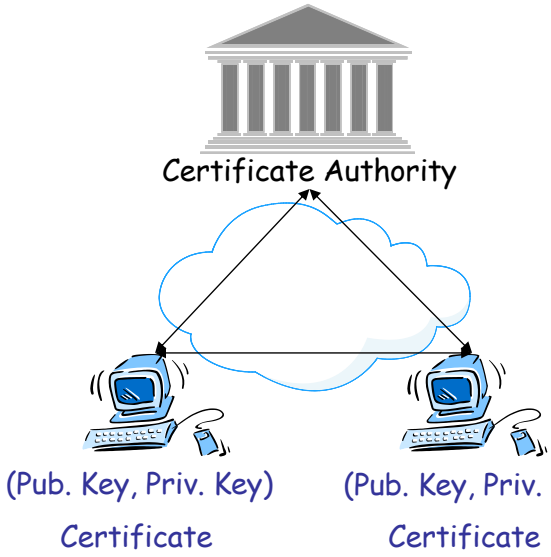
Digital Signature -- Sender



Digital Signature -- Receiver

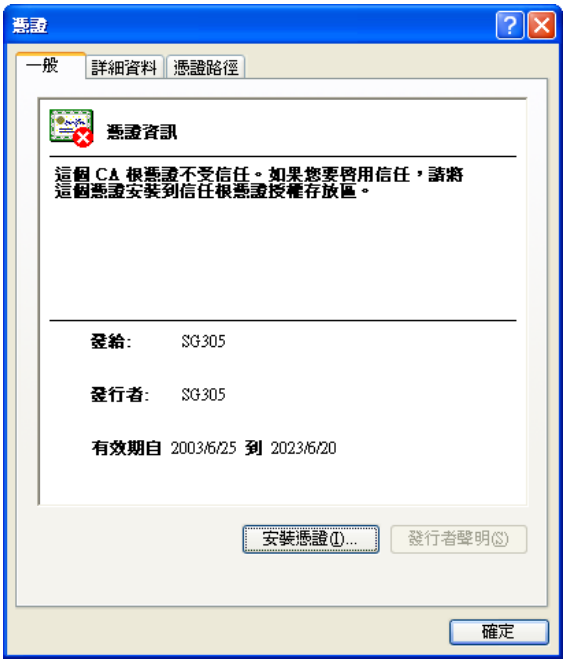


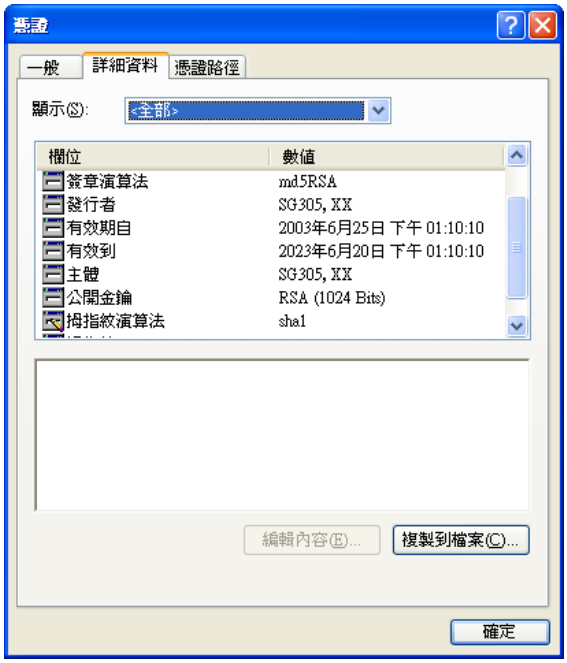
Public-Key Infrastructure

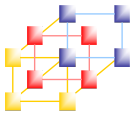


- X.509 (ITU-T)
  - Directory service
  - Authentication Framework
  - Lightweight Directory Access Protocol (LDAP; RFC1777)
- Certification Revocation List (CRL)
  - Lightweight Directory Access Protocol (LDAP; RFC1777)
  - Online Certificate Status Protocol (OCSP;RFC2560)

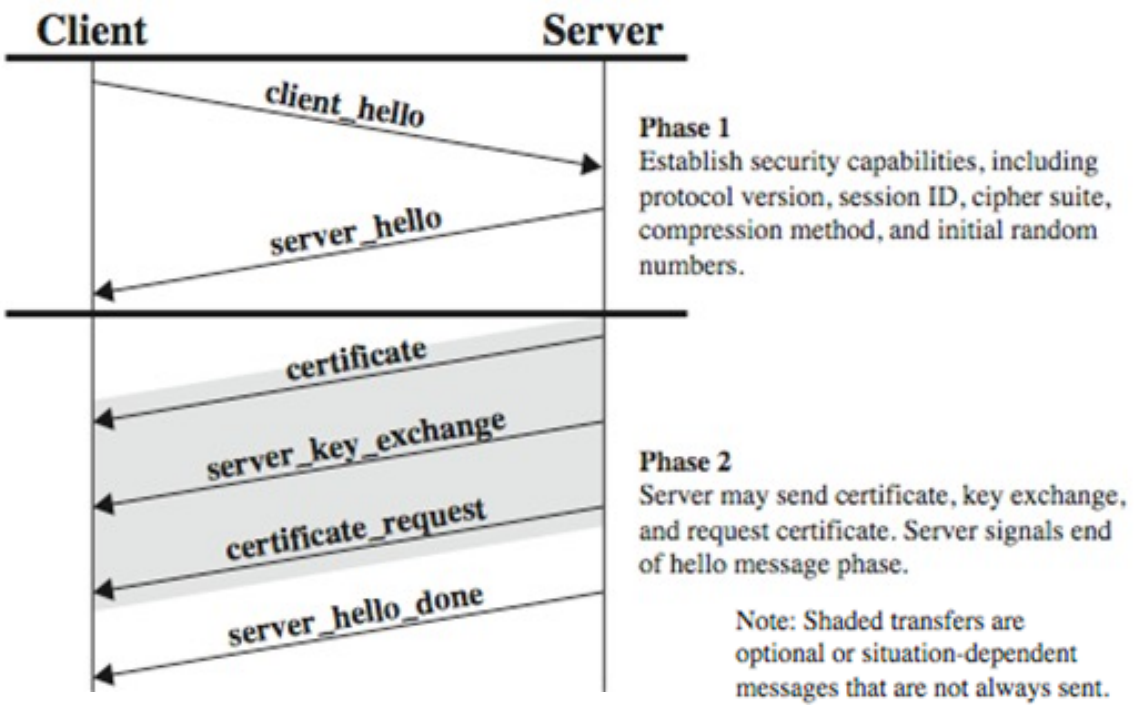
Certificate

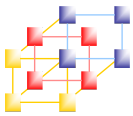




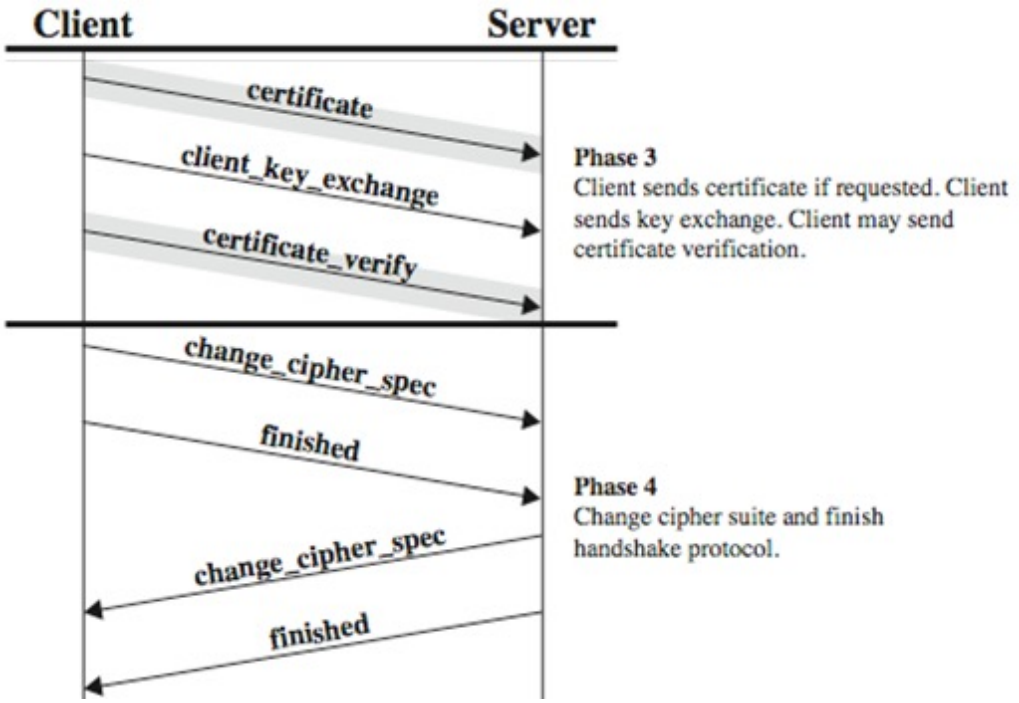


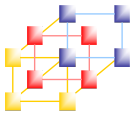
SSL Handshake Protocol (1/2)





SSL Handshake Protocol (2/2)





Create ssl.conf

Install openssl
https://slproweb.com/products/Win32OpenSSL.html

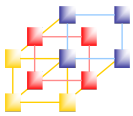
```
[req]
prompt = yes
default_md = sha256
default_bits = 2048
distinguished_name = req_distinguished_name
x509_extensions = v3_req

[req_distinguished_name]
countryName                     = Country Name (2 letter code)
stateOrProvinceName             = State or Province Name (full name)
localityName                    = Locality Name (eg, city)
0.organizationName              = Organization Name (eg, company)
organizationalUnitName          = Organizational Unit Name (eg, section)
commonName                      = Common Name (e.g. server FQDN or YOUR name)
emailAddress                    = Email Address

[v3_req]
subjectAltName = @alt_names

[alt_names]
DNS.1 = *.localhost
DNS.2 = localhost
IP.1 = 192.168.2.100
IP.2 = 127.0.0.1
```

設定檔的 [alt_names] 區段，是用來設定 SSL 憑證的域名要設定幾組域名都可以，建議可以把可能會用到的本機域名 (localhost) 或是區域網路的 IP 地址都加上去，以便後續進行遠端連線測試。



Create Certificate

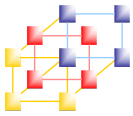
```
openssl req -newkey rsa:2048 -nodes -keyout server.key -x509 -days 365 -out server.cer -config ssl.conf
```

```
Generating a 2048 bit RSA private key
.................+++
..+++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:TW
State or Province Name (full name) []:Taiwan
Locality Name (eg, city) []:Taichung
Organization Name (eg, company) []:FCU
Organizational Unit Name (eg, section) []:IECS
Common Name (e.g. server FQDN or YOUR name) []:Server
Email Address []:server@gmail.com
```

Check Certificat

```
openssl x509 -in server.cer -text -noout
```

1-SSLServer.py
1-SSLClient.py



Simple Secure Socket Program

- Server
  - Create context

```
ctx = ssl.SSLContext(ssl.PROTOCOL_TLS_SERVER)
```

  - Load certificate

```
ctx.load_cert_chain(certfile='server.cer',
    keyfile='server.key')
```

  - Wrap socket

```
sslsocket = ctx.wrap_socket(srvSocket, server_side=True)
```

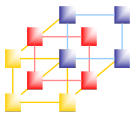
- Client
  - Create context

```
ctx = ssl._create_unverified_context()
```

  - Wrap socket

```
sslsocket = ctx.wrap_socket(clientSocket)
```


2-SSLServer.py
2-SSLClient.py



Verify Certificate

- Server

```
ctx = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
ctx.verify_mode = ssl.CERT_REQUIRED
ctx.load_cert_chain(certfile='server.cer',
    keyfile='server.key')
ctx.load_verify_locations(cafile='client.cer')
```

- Client

```
ctx =
ssl.create_default_context(ssl.Purpose.SERVER_AUTH,
    cafile='server.cer')
ctx.load_cert_chain(certfile='client.cer',
    keyfile='client.key')
```