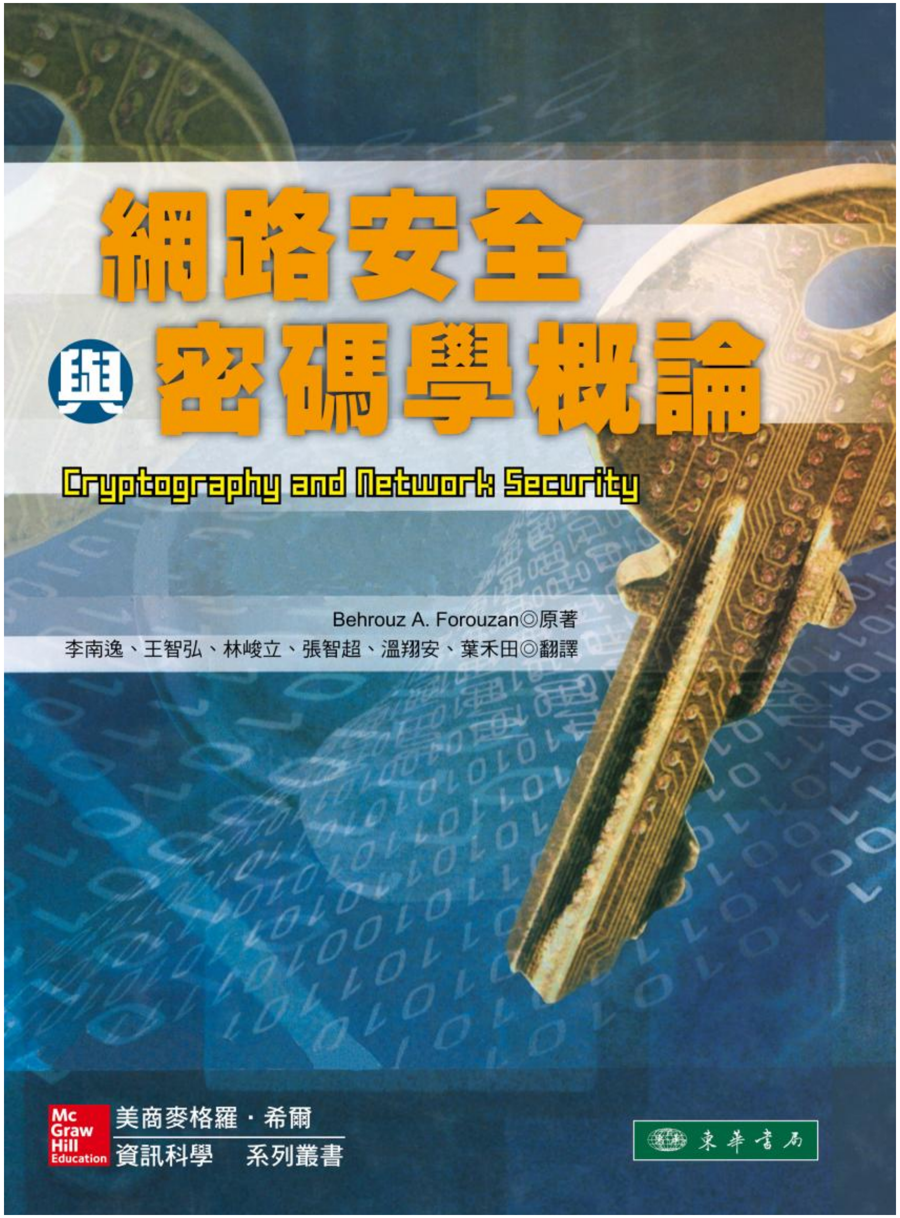
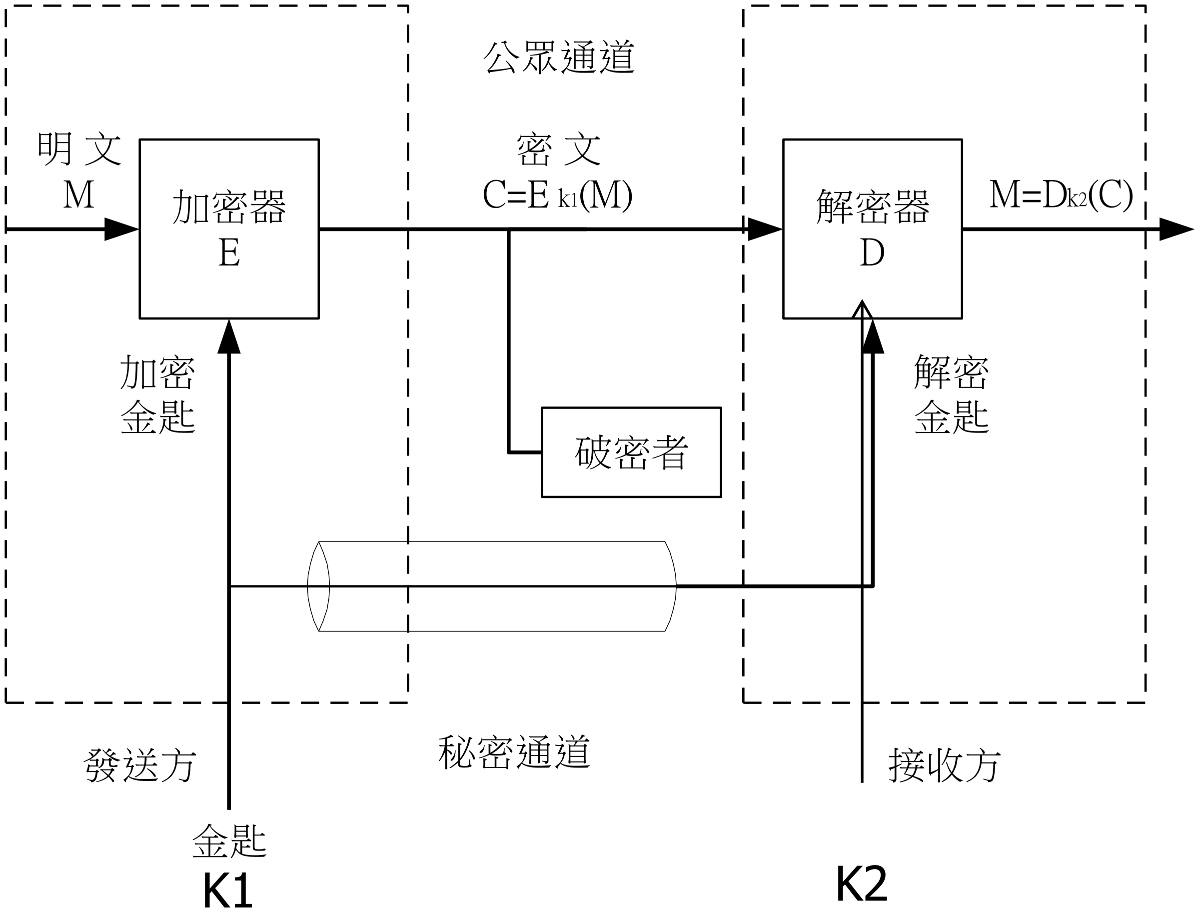


Chapter 10

非對稱式金鑰密碼系統


公眾通道
明 文
M
加密器
E
密 文
C=E k1(M)
解密器
D
M=Dk2(C)
加密
金匙
破密者
解密
金匙
發送方
秘密通道
接收方
金匙
K1
K2

Secret key cryptosystems vs Public key Cryptosystems

- If k1=k2 ➔ symmetric key cryptosystem (對稱), one-key system, secret key system (私密密碼系統)
  - 例子: DES, AES, webmail ID/ password, GSM pin
  - Authentication, privacy, integrity
  - 缺點: 金鑰分配, 金鑰管理, 無法達成不可否認性
- If k1≠k2 ➔ asymmetric key cryptosystem (非對稱式), two-key system, public key system
  - 例子: RSA, Elgamal, D-H key
  - Authentication, privacy, integrity, non-repudiation
  - 缺點: 速度慢

秘密金匙密碼系統具有下列缺點：

1.收發雙方如何獲得其加密金匙及解密金匙，這個問題稱為金匙分配 **(*Key distribution*)**

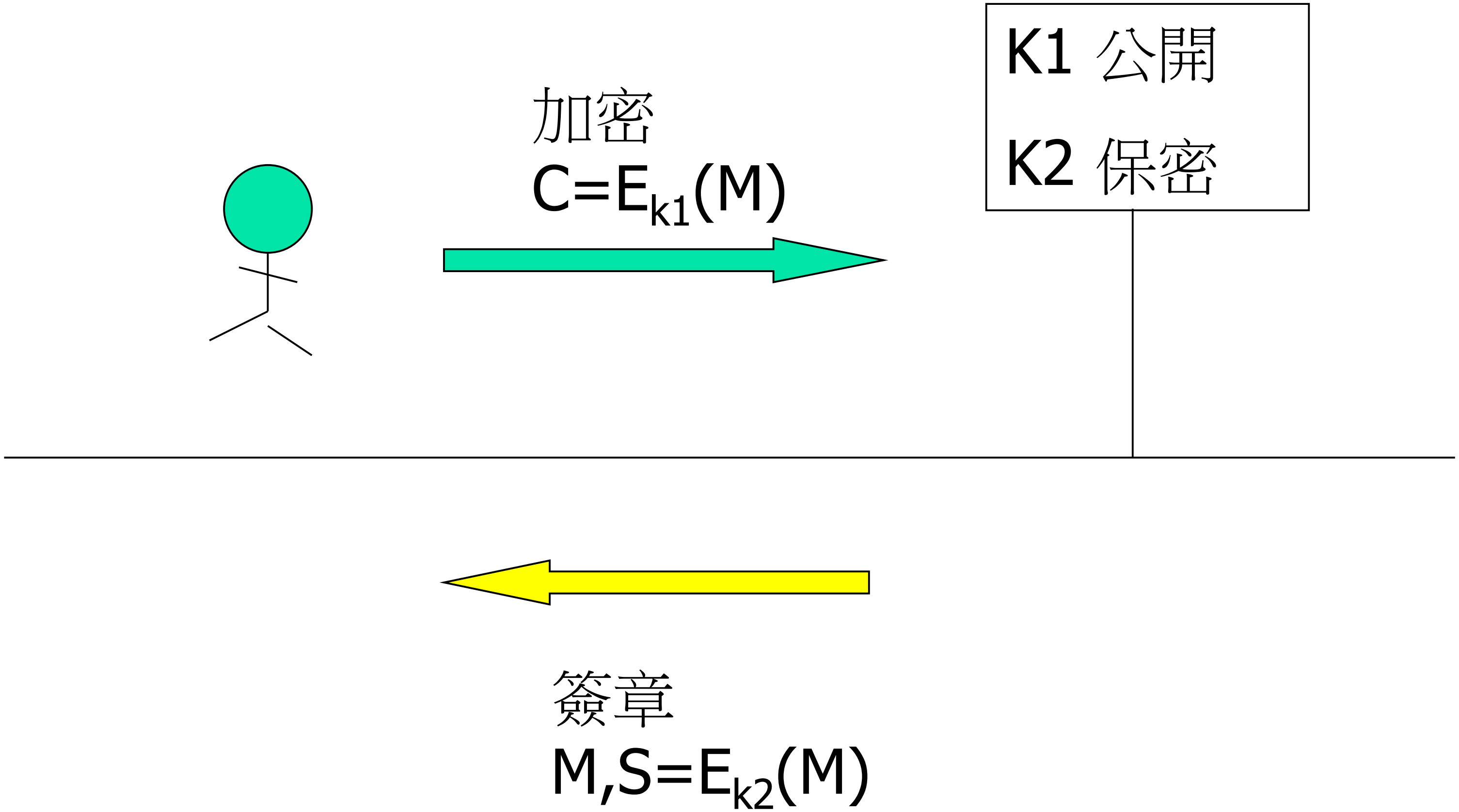
2.. 金匙的數目太大：

若網路中有n人，則每一人必須擁有n-1把金匙。網路中共需有{n(n-1)}/2把不同的金匙。當n等於1000時，每人須保管999把金匙。網路中，共需有499500把不同的金匙。如何管理這麼多的金匙，也是一大問題。

3. 無法達到不可否認性服務

Class Challenge: why?

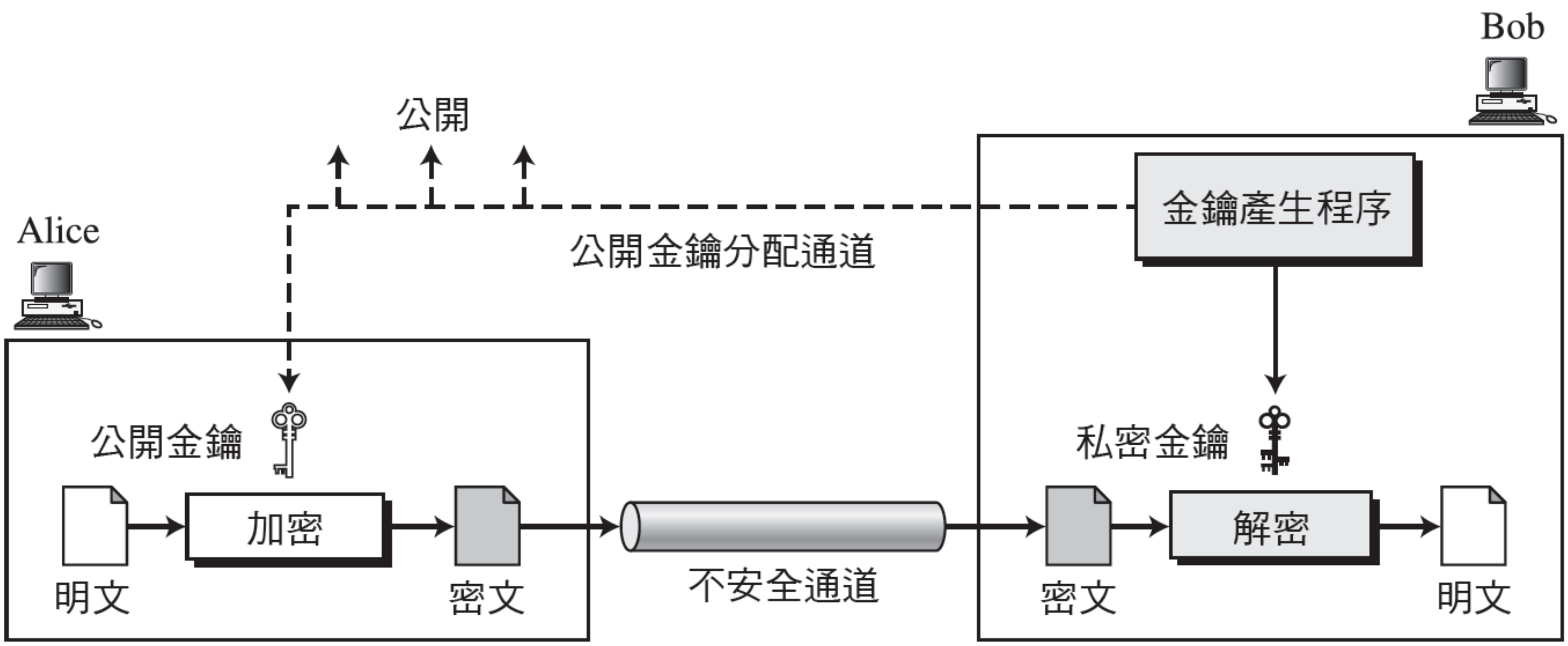
Concept of Public key cryptosystem



10.1.1 金鑰

- 非對稱式金鑰密碼學使用兩把不同的金鑰：一把為私密金鑰（private key），一把為公開金鑰（public key）。

圖 10.2 非對稱式金鑰密碼系統的一般概念➔ 加密

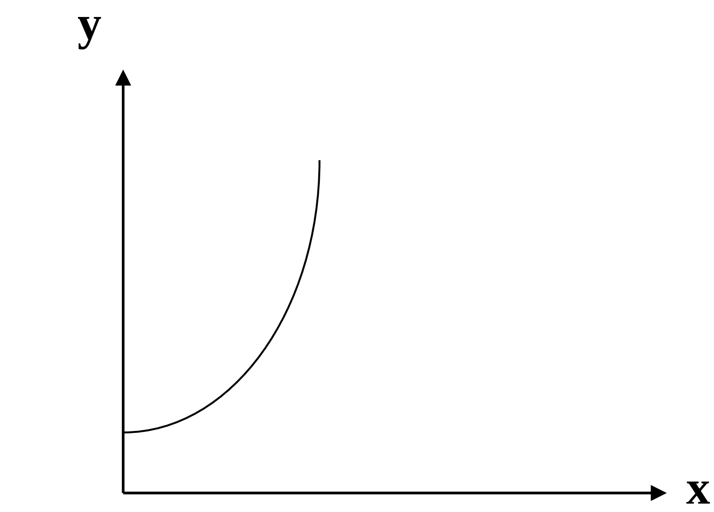


10.1.2 一般概念

- 明文／密文
  - 非對稱式金鑰密碼學將明文與密文當成整數。
- 加密／解密
  - 密文表示為 $C = \boldsymbol{f}\,(K_{公開}, P)$
  - 明文為 $P = \boldsymbol{g}\,(K_{私密}, C)$。
  - Question: 我們使用電腦時 那些情境這樣用?

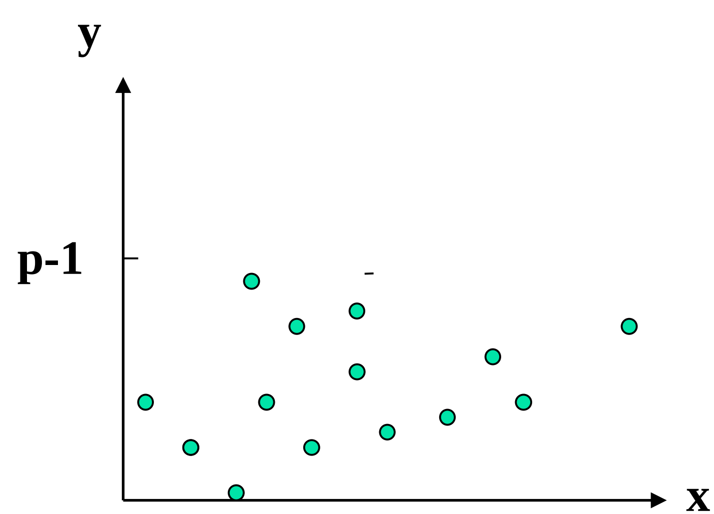
Discrete Logarithm Problem (DLP)

- **$x \rightarrow y=f(x)=g^x$:指數函數 (exponential function)**
- **$y=f(x)=g^x \rightarrow x$: 對數函數 (Logarithmic Function)**



$y=f(x)=g^x$ mod p, g, p ➔ x : 離散對數函數(Discrete Logarithmic Problem:DLP)

➔ NP problem



Factor Problem (FAC)

- **Given *N*=*p**q*, where *p* and *q* are large primes**

➔ It is hard (infeasible) to find *p* or *q*

Question: List and compare the time complexities of linear search problem, binary search problem & factoring problems or logarithm problem. Substitute the parameter N with different numbers; for examples, n=1, 10, 100, 1000, 10000

RSA 系統

Key Generation

Select p, q	p and q both prime
Calculate $n = p \times q$	
Calculate $\phi(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \bmod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

RSA 加密系統

Browser

加密 m

$1.\ \mathrm{C} = m^{e} \bmod n$

1. Get public key

2. C

Server

$PublicKey : (n, e)$

$\mathrm{Private\ key} : (p, q, d),$

$\mathrm{where\ e \cdot d \bmod \phi(n) = 1.}$

$\mathrm{that\ is,\ e \cdot d = k \cdot \phi(n) + 1.}$

$$3.\text{解密 } C^{d} = (m^{e})^{d} = m^{ed}$$

$$= m^{k\phi(n)+1} = m \bmod n$$

1. Security

Based on FAC problem

RSA 加密系統－ 例子

Browser

加密 $m = 2$

$1.\ C = m^e \bmod n$

$C = 2^3 \bmod 15 = 8$

1. Get public key

2. C=8

Server

$PublicKey:(n = 3*5, e = 3)$

$\Pr ivate \text{ key}:(p = 3, q = 5, d = 3),$

$\text{where } e \cdot d \bmod \phi(n) = 1$

$3*3 \bmod 8 = 1$

$\text{that is, } e \cdot d = k \cdot \phi(n) + 1.$

3.解密 $C^d = (m^e)^d = m^{ed}$

$= 8^3 = 64*8 = 4*8 = 32 = 2$

1. **Security**

Based on FAC problem

演算法10.2 RSA 金鑰產生

RSA_Key_Generation

{

Select two large primes p and q such that $p \neq q$.

$n \leftarrow p \times q$

$\phi(n) \leftarrow (p-1) \times (q-1)$

Select e such that $1 < e < \phi(n)$ and e is coprime to $\phi(n)$

$d \leftarrow e^{-1} \bmod \phi(n)$ //d是e在模$\phi(n)$底下的乘法反元素

Public_key $\leftarrow$ (e, n) //公開宣布

Private_key $\leftarrow d$ //保持祕密

return Public_key and Private_key

}

演算法10.3 RSA 加密

```
RSA_Encryption (P, e, n)                    // P 是在 Z_n 下的明文且 P < n
{
    C ← Fast_Exponentiation (P, e, n)       //計算 (P^e mod n)
    return C
}
```

10.2.1 程序 (續)

- 以前標準，現在

注意

在 RSA 中，*p* 及 *q* 必須至少為 512 位元；*n* 必須至少為 1024 位元。

最新Update *p* 及 *q* 必須至少為 1024 位元；*n* 必須至少為 2048 位元

演算法10.4 RSA 解密

```
RSA_Decryption (C, d, n)                          //C是在Zn下的密文
{
    P ← Fast_Exponentiation (C, d, n)             //計算(C^d mod n)
    return P
}
```

10.2.1 程序 (續)

- RSA的證明

如果$n = p \times q$，$a < n$，而且k是一個整數，則$a^{k \times \phi(n)+1} \equiv a \pmod{n}$。

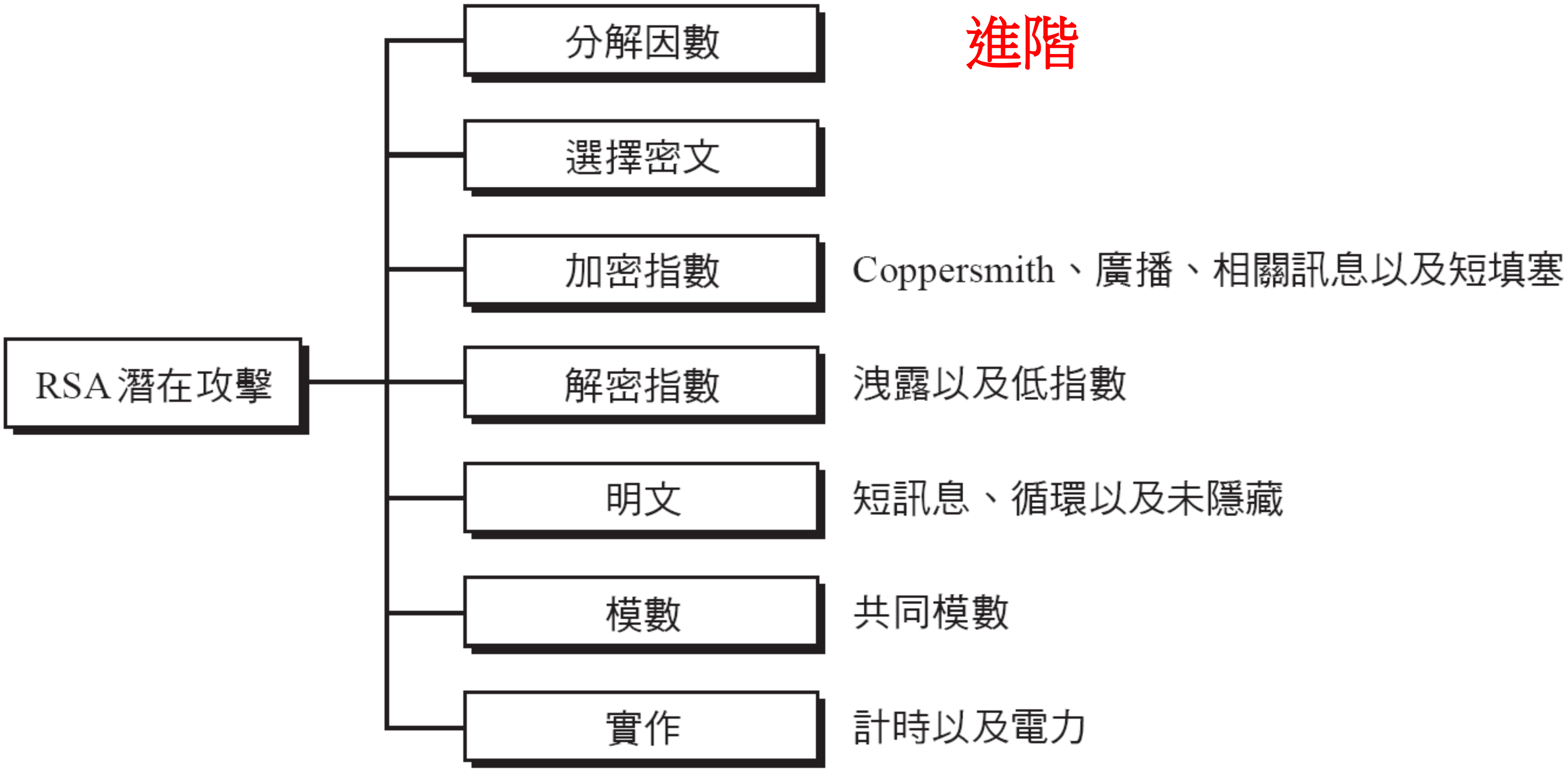
$P_1 = C^d \bmod n = (P^e \bmod n)^d \bmod n = P^{ed} \bmod n$

$ed = k\,\phi(n) + 1$ //d與e在模$\phi(n)$底下互為乘法反元素

$P_1 = P^{ed} \bmod n \rightarrow P_1 = P^{k\phi(n)+1} \bmod n$

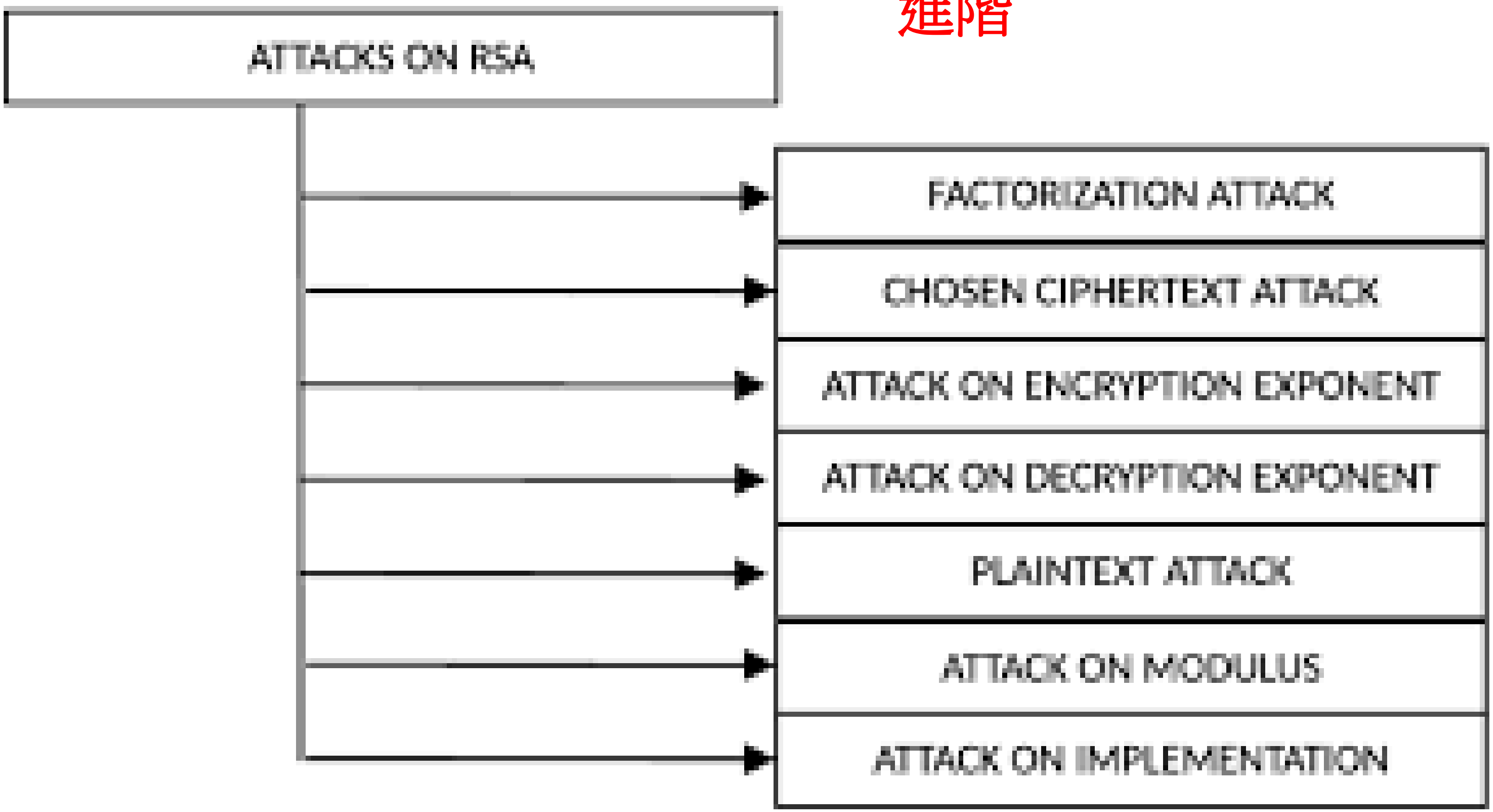
$P_1 = P^{k\phi(n)+1} \bmod n = P \bmod n$ //尤拉定理（第二種版本）

圖10.8 RSA 潛在攻擊的分類



Potential Attacks on RSA

進階



RSA 簽章系統 (RSA digital Signature)

Browser

Server

$$\boxed{\begin{array}{l} PublicKey:(n,e) \\ \text{Private key}:(p,q,d), \\ \text{where } e\cdot d \bmod \phi(n)=1. \\ \text{that is, } e\cdot d=k\cdot\phi(n)+1. \end{array}}$$

$\boxed{\text{Get the public key}}$

1b. Get public key & (S,m)

$\longleftarrow$

1a.簽章 $m^d = S \bmod n$

2.驗證 (m,S)

$$S^e \stackrel{?}{=} m \mod n$$

$$(m^d)^e \stackrel{?}{=} m \bmod n$$

RSA 簽章系統—例子（Example）

Browser

Server

Get the public key

1b. Get public key
(*S=13,m=7*)

$$PublicKey:(n=15, e=3)$$
$$\Pr ivate\ \mathrm{key}:(p, q, d=3),$$
$$\text{where } e \cdot d \bmod \phi(n)=1.$$
$$\text{that is, } e \cdot d = k \cdot \phi(n)+1.$$

1*a*.簽章 $m^d = S \bmod n$

$$7^3 = 49*7 = 4*7 = 28 = 13$$

2.驗證 (m, S)

$$S^e \stackrel{?}{=} m \mod n$$

$$(13)^3 \stackrel{?}{=} 7 \mod 15$$

$$169*13 = 4*13 = 52 = 7$$

RSA 數位簽署系統安全 (Security of RSA signature)

1. Security

Based on FAC problem

2. Forgability

a) first choose S, then compute $m=S^e \bmod n$
b) Of course, $S^e=m \bmod n$

3. Practically, hash function is applied on *m* before computing *S*

改良RSA 數位簽署系統 (Improved RSA Signature)

簽署:

1. Given m

2. Compute $S = h(m)^d \bmod n$

$\xrightarrow{(m, S)}$

驗證:

1. Given (m, S)

2. Verify $S^e \stackrel{?}{=} h(m) \bmod n$

- 簽章前先套用hash function 的優點 (Advs of hashing before signing)

1. 防止 偽造攻擊 (deter forgery)
2. 降低計算量及簽章大小 (reduce computations & size of the signature)

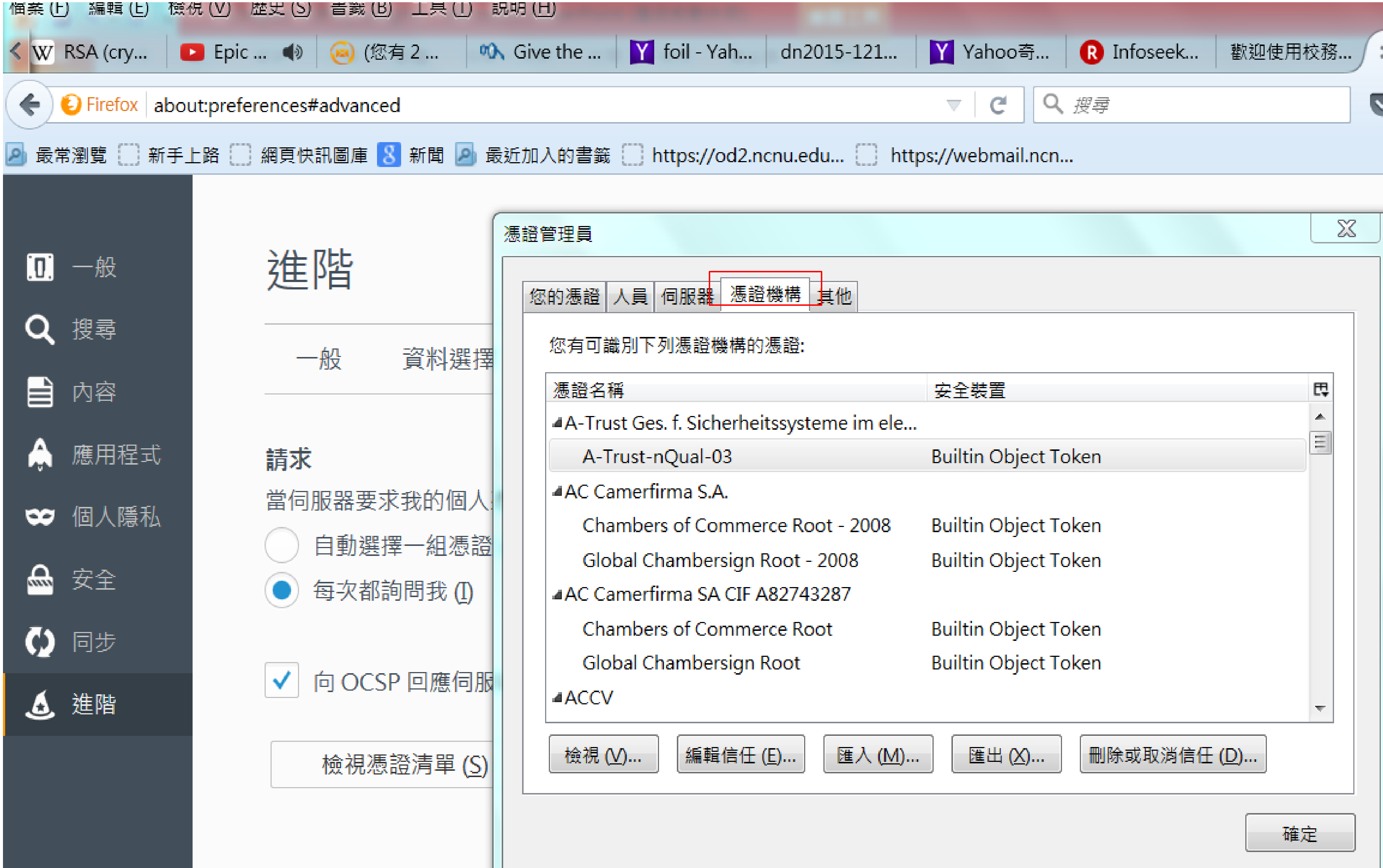
記住: 所有真在在用的數位簽章 都有套用雜湊函數 但原理說明時去掉較簡化,易說明 (please note that we apply hashing before signing in all practical applications; but, in many books, we eliminate hashing for simplicity)

RSA in 實務 (Some RSA standards)

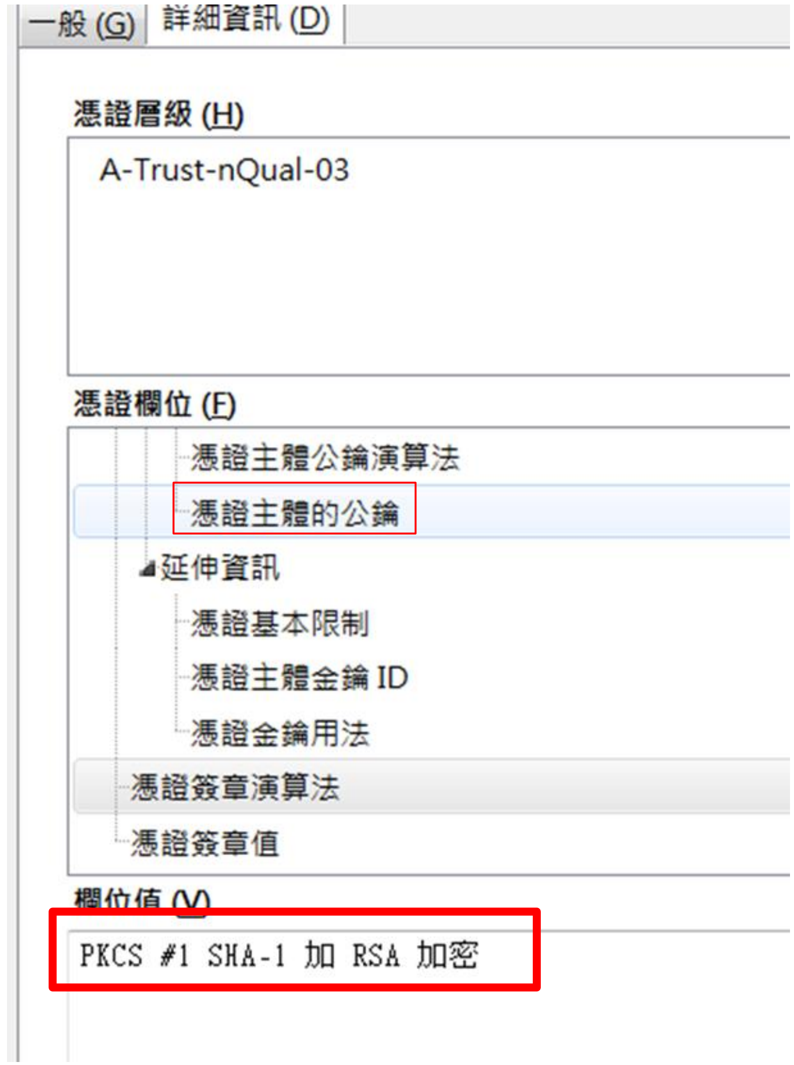
- RSA → PKCS#1 → Optimal Asymmetric Encryption Padding (OAEP)

  → Probabilistic Signature Scheme for RSA (RSA-PSS).
- https://en.wikipedia.org/wiki/RSA_%28cryptosystem%29
- http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-rsa-cryptography-standard.htm
- RSA demo http://logos.cs.uic.edu/340%20notes/rsa.html
- [Day27] 非對稱金鑰加密系統(RSA) windows內建憑證工具 https://ithelp.ithome.com.tw/articles/10188824

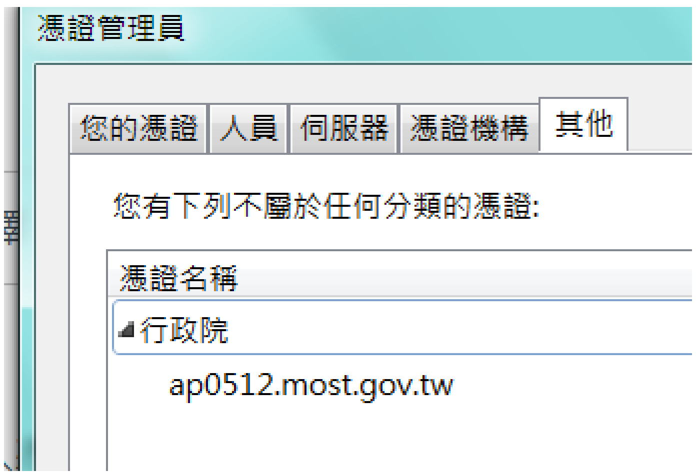
RSA in 實務 (RSA Certificate in Browser)

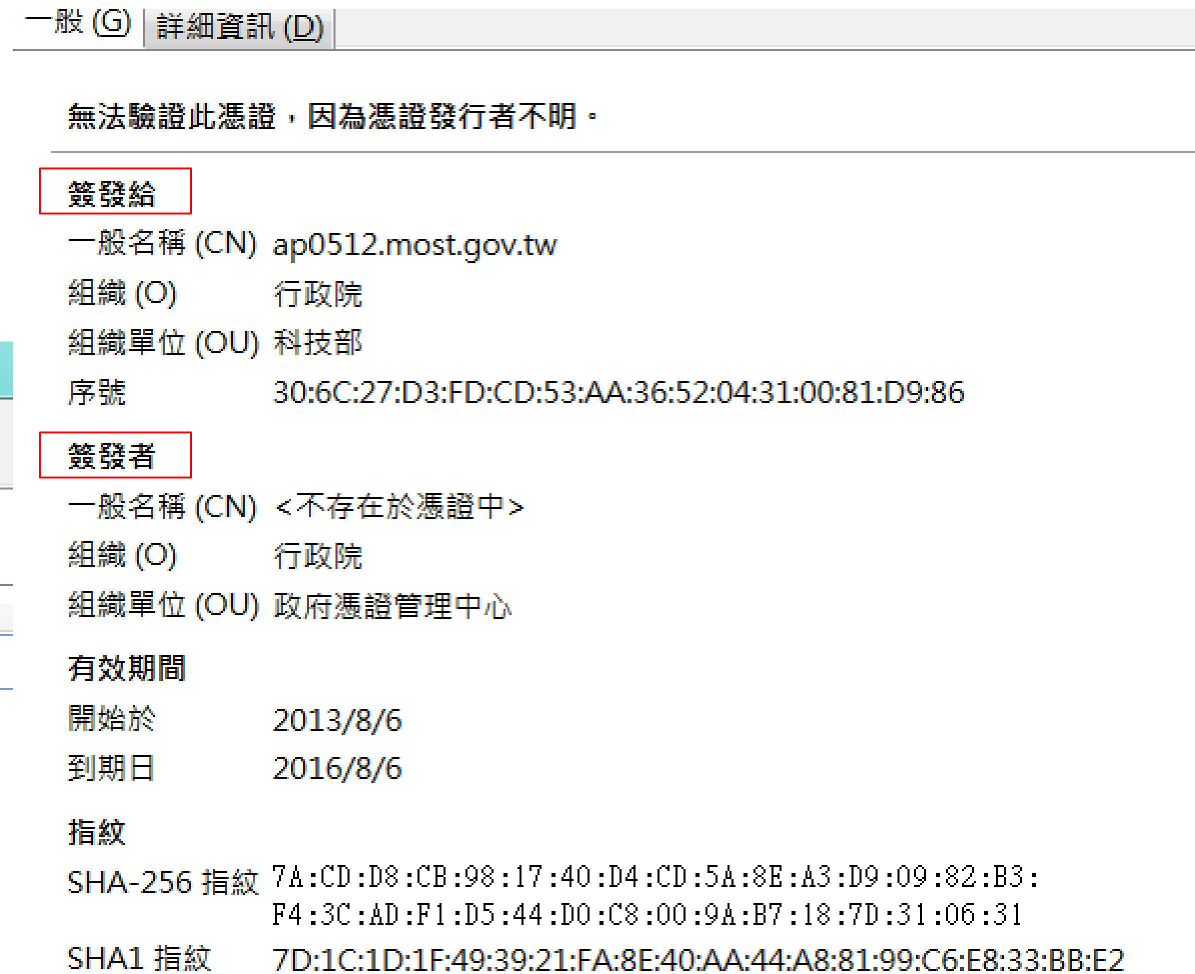


RSA in 實務 (RSA Certificate in Browser)

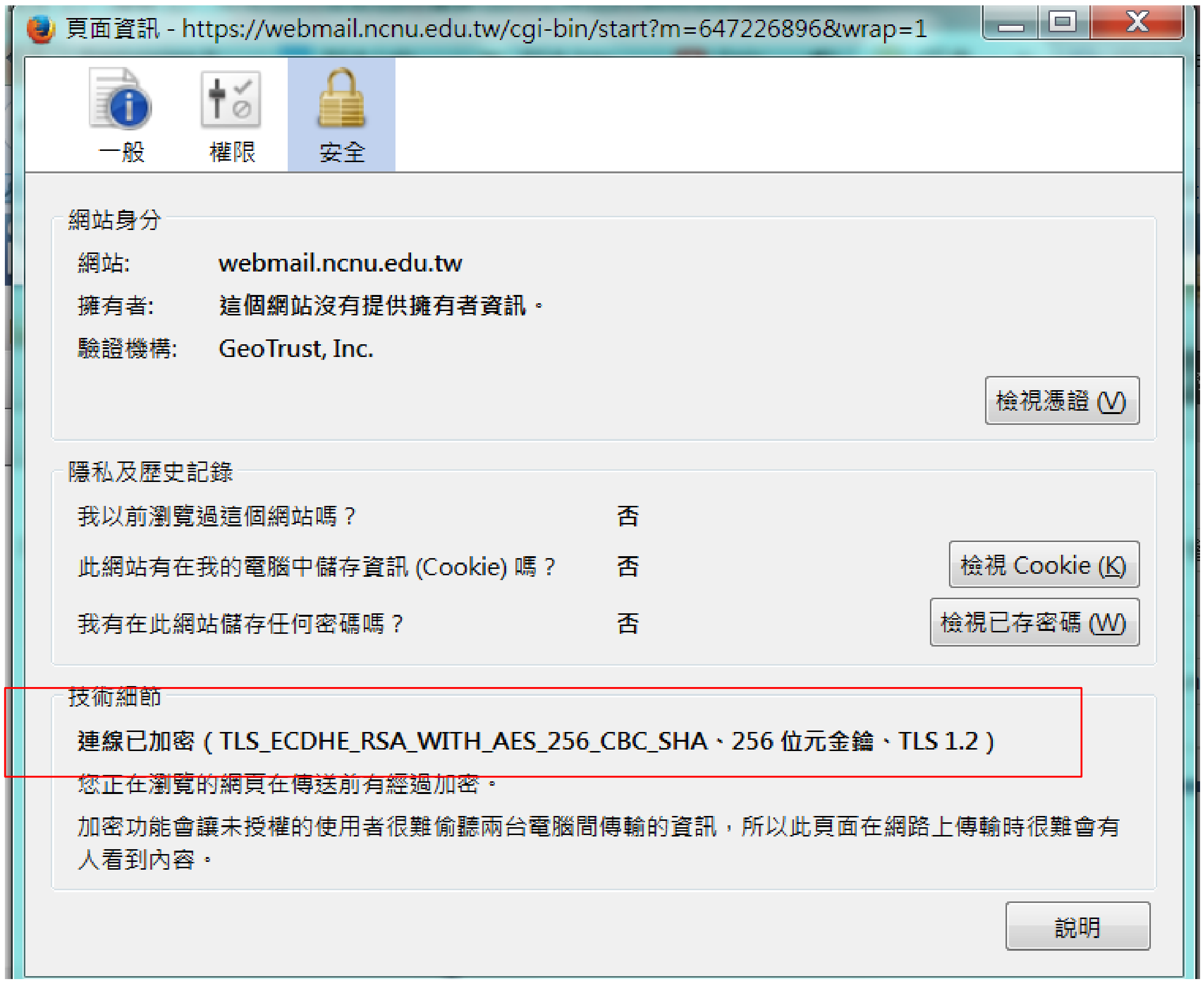


RSA in 實務 (RSA Certificate in Browser)





RSA in 實務– NCNU email



RSA in 實務– NCNU email

一般 (G) | 詳細資訊 (D)

此憑證已驗證用於下列用途:

- SSL 客戶端憑證
- SSL 伺服器憑證

簽發給

一般名稱 (CN)	*.ncnu.edu.tw
組織 (O)	*.ncnu.edu.tw
組織單位 (OU)	GT19970321
序號	03:23:C3

簽發者

一般名稱 (CN)	RapidSSL CA
組織 (O)	GeoTrust, Inc.
組織單位 (OU)	<不存在於憑證中>

有效期間

開始於	2011/9/13
到期日	2016/9/13

指紋

SHA-256 指紋	00:25:72:C8:A1:4D:7F:09:89:65:AA:71:B9:8F:56:14: 77:2D:A5:CD:3F:61:26:70:E0:AD:CC:39:E3:D5:6B:79
SHA1 指紋	05:30:9E:9F:BD:03:07:5A:6D:1C:03:53:4E:73:36:75:11:34:0D:1D

1. RSA及Rabin數位簽署為確定式簽署方式，即一個明文對應一個簽署文。(Both RSA and Rabin are deterministic signature; that is, each plaintext has one specific corresponding signature)

2. 在1985年, ElGamal 提出一種機率式的簽署方式，對於每一明文*m*，可有許多的合法簽署文。(Elgamal is a probabilistic signature; each plaintext could have several signatures)

3. ElGamal 簽署之安全性係基於解離散對數之困難度上。(Elgamal is based on DLP)

ElGamal 密碼系統 (Elgamal Cryptosystem)

- 除了 RSA 與 Rabin，另外一個公開金鑰密碼系統是 ElGamal 密碼系統(ElGamal cryptosystem)，根據發明者 Taher ElGamal 命名。ElGamal 是基於我們在第九章所討論過的離散對數問題。
- 本節討論主題
  - ElGamal密碼系統
  - 程序
  - 證明
  - 分析
  - ElGamal 的安全性
  - 應用

ElGamal 密碼系統 (Elgamal Cryptosystem)

與**PKDS**系統相同，令本系統中存在一大質數***p***及模***p***之原根***g***。

1.使用者 ***i*** 任選其私有秘密金匙x_i，並求出其公開金匙 $y_i = g^{x_i} \bmod p$

2. 使用者 ***j*** 任選一亂數 ***r*** 並利用使用者 ***i***之 公開金匙求出

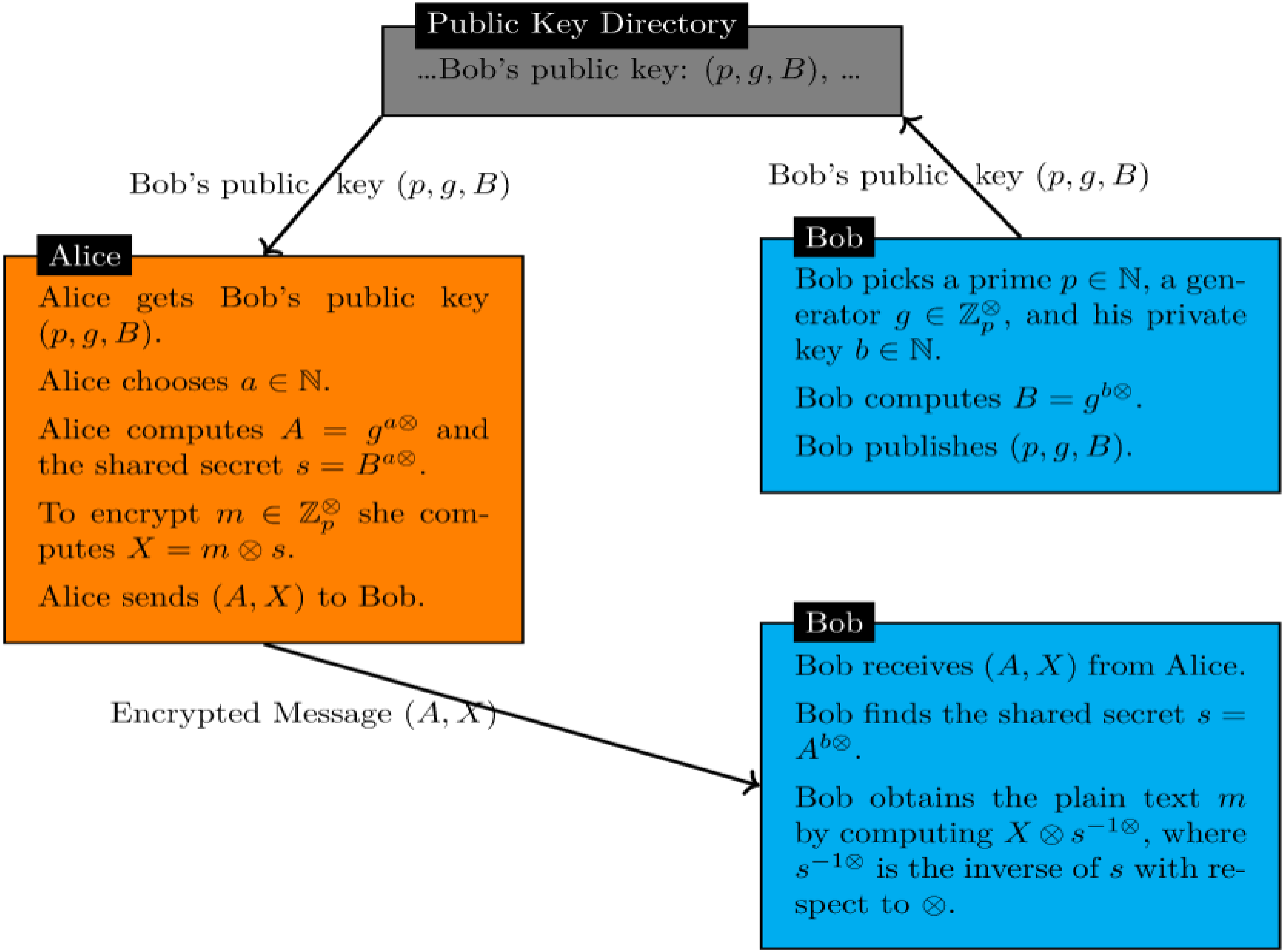
$$C_1 = g^r \bmod p \text{，} \quad C_2 = m y_i^r \bmod p$$

並將密文$C=(C_1, C_2)$送給使用者***i***。

3. 使用者***i***收到密文後，利用其秘密金匙x_i求出

$$C_1^{x_i} = (g^r)^{x_i} = (g^{x_i})^r = y_i^r \bmod p$$

$$C_2(y_i^r)^{-1} = (m y_i^r)(y_i^r)^{-1} = m \bmod p$$


Public Key Directory
...Bob's public key: (p, g, B), ...
Bob's public key (p, g, B)
Bob's public key (p, g, B)
Alice
Alice gets Bob's public key (p, g, B).
Alice chooses $a \in \mathbb{N}$.
Alice computes $A = g^{a\otimes}$ and the shared secret $s = B^{a\otimes}$.
To encrypt $m \in \mathbb{Z}_p^{\otimes}$ she computes $X = m \otimes s$.
Alice sends (A, X) to Bob.
Bob
Bob picks a prime $p \in \mathbb{N}$, a generator $g \in \mathbb{Z}_p^{\otimes}$, and his private key $b \in \mathbb{N}$.
Bob computes $B = g^{b\otimes}$.
Bob publishes (p, g, B).
Encrypted Message (A, X)
Bob
Bob receives (A, X) from Alice.
Bob finds the shared secret $s = A^{b\otimes}$.
Bob obtains the plain text m by computing $X \otimes s^{-1\otimes}$, where $s^{-1\otimes}$ is the inverse of s with respect to $\otimes$.

ElGamal 公開金匙分配系統例子(example)

Receiver public key $Y=2^4 \bmod 11=5$

- *x=4 secret key*

Sender 1
$m=3$
$r=2$, $C1=2^2 \bmod 11=4$,
$C2=3 * Y^2=3 * 5^2=3 * 3=9$

$C1=4, C2=9$

Decrypt message 1
$C1^x=4^4 \bmod 11=2^8$,
$C2*(2^8)^{-1}=9*(2^8)^{-1}=9 * 2^2=36=3$

Sender 2
$m=3$
$r=3$, $C1=2^3 \bmod 11=8$,
$C2=3 * Y^3=3 * 5^3=3 * 4=12=1$

$C1=8, C2=1$

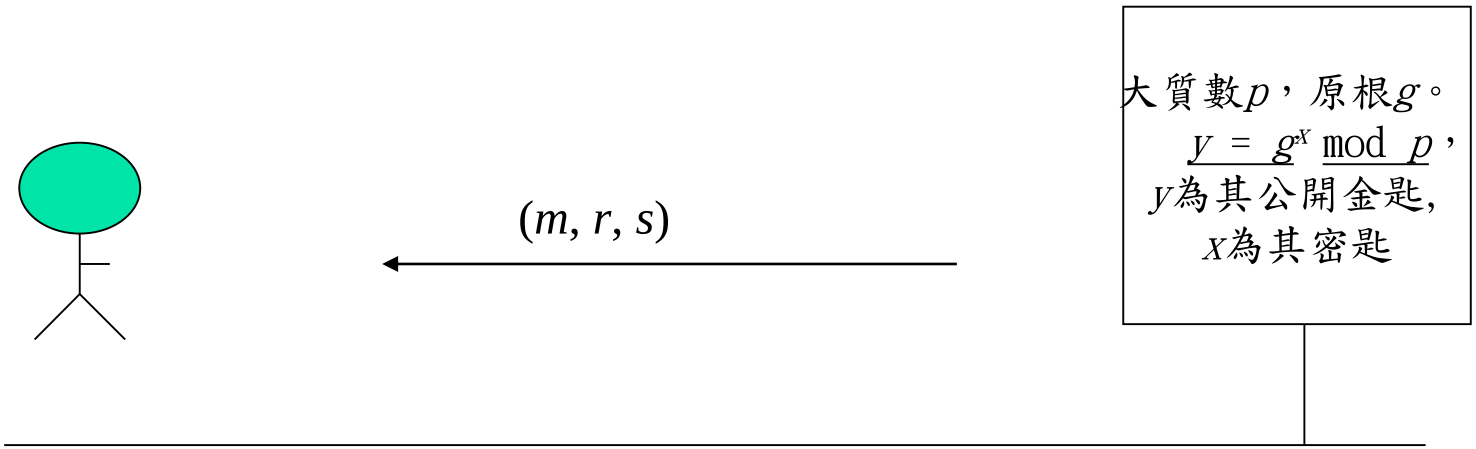
Decrypt message 2
$C1^x=8^4 \bmod 11=2^{12}=4$,
$C2*(4)^{-1}=1*(4)^{-1}=1 * 3=3$

Homework:

1. 以指數函數實現ElGamal公開金匙分配系統，請問當 $p=11$, $g=2$, $y=4$, $x=2$, ($C_1=4$, $C_2=4$) 時 $m=$?

➔ [illegible]

ElGamal數位簽署(Elgamal Digital Signature)



1) 簽署：

對於明文$1 \leqq m \leqq p-1$，

1.1) 先任選一整數k滿足$(k, p-1) = 1$

1.2) 求出 (r, s) 滿足

$r = g^k \bmod p$，及

$s = k^{-1}(m - xr) \bmod p-1$

或 $m = xr + ks \bmod p-1$ 。

2) 驗証：驗証下式是否成立

$g^m = y^r r^s \bmod p$。(注意：$g^m = g^{xr} g^{ks} = g^{xr+ks} \bmod p$)

若正確, 則(r, s)為m之合法簽署文，否則為非法簽署。

ElGamal數位簽署 example

1. 參數: p=11, g=2, Y=4, m=3,

2. 簽章: k=3, $r=2^3=8$, $s=3^{-1}(3-2*8)=7*(-13)=49=9$

3. (r, s)=(8, 9)

Class challenge: 以 Elgamal 簽章法

$$p = 11, g = 2, y = 5, m = 3, k = 3$$

求 簽署文(r,s)

➔

Elgamal 安全性分析及討論

1. DLP

2. 若第三者A欲偽造一合法簽署文，其任選r(或s)，欲求出s(或r)滿足(2)式，則面臨解離散對數問題。

3. 第三者已獲得一明文m之簽署及(r, s)，欲由(1)式求出x。則因(1)式中有二個未知數x及 k，他無從求出x。

4. 若 B 利用相同 k 簽署兩次，即m_1、m_2之簽署文為 (r, s_1) 及 (r, s_2)，則第三者可利用(1)式解聯立方程式

$$m_1 = xr + ks_1 \bmod p-1,$$

$$m_2 = xr + ks_2 \bmod p-1.$$

因有二方程式及二變數 (x及k)，則 x 可被求出。因此，在本系統中 k 不可重覆使用。

5. 第三者可以偽造出對m為合法的簽署文(r, s)，不過 m 並無法事先固定。

此偽造方法如下：

5.1) 第三者任選兩亂數 u 及 w，滿足$1 < u, w < p$-1且 $\gcd(w, p-1) = 1$。

5.2) 計算 $r = g^u y^{-w} \bmod p$ ， (3a)

$s = r w^{-1} \bmod p-1$ ， (3b)

及 $m = us \bmod p-1$。 (3c)

由(3a) – (3c)可得 $y^r r^s = y^r(g^u y^{-w})^s = y^r g^{us} y^{-ws} = y^r g^{us} y^{-r} = g^{us} = g^m \bmod p$。

因此(r, s)確為m之合法簽署文。

➔在此偽造過程中，由於 A 對m並無控制能力，故ElGamal 簽署仍為安全。

➔ 因為所有簽署系統均有此類似攻擊法存在，利用一單向赫序函數h與數位簽署合併使用，即可防止此攻擊法。

Elgamal in 實務

- Elgamal → Schnorr → DSA (DSS)
- (Digital Signature Algorithm，DSA)

- ElGamal Encryption Playground
  http://www.debjitbiswas.com/elgamal-encryption-playground/

- ElGamal Cryptosystem Interactive Demo
  https://www.cs.uri.edu/cryptography/publickeyelgamaldemo.htm

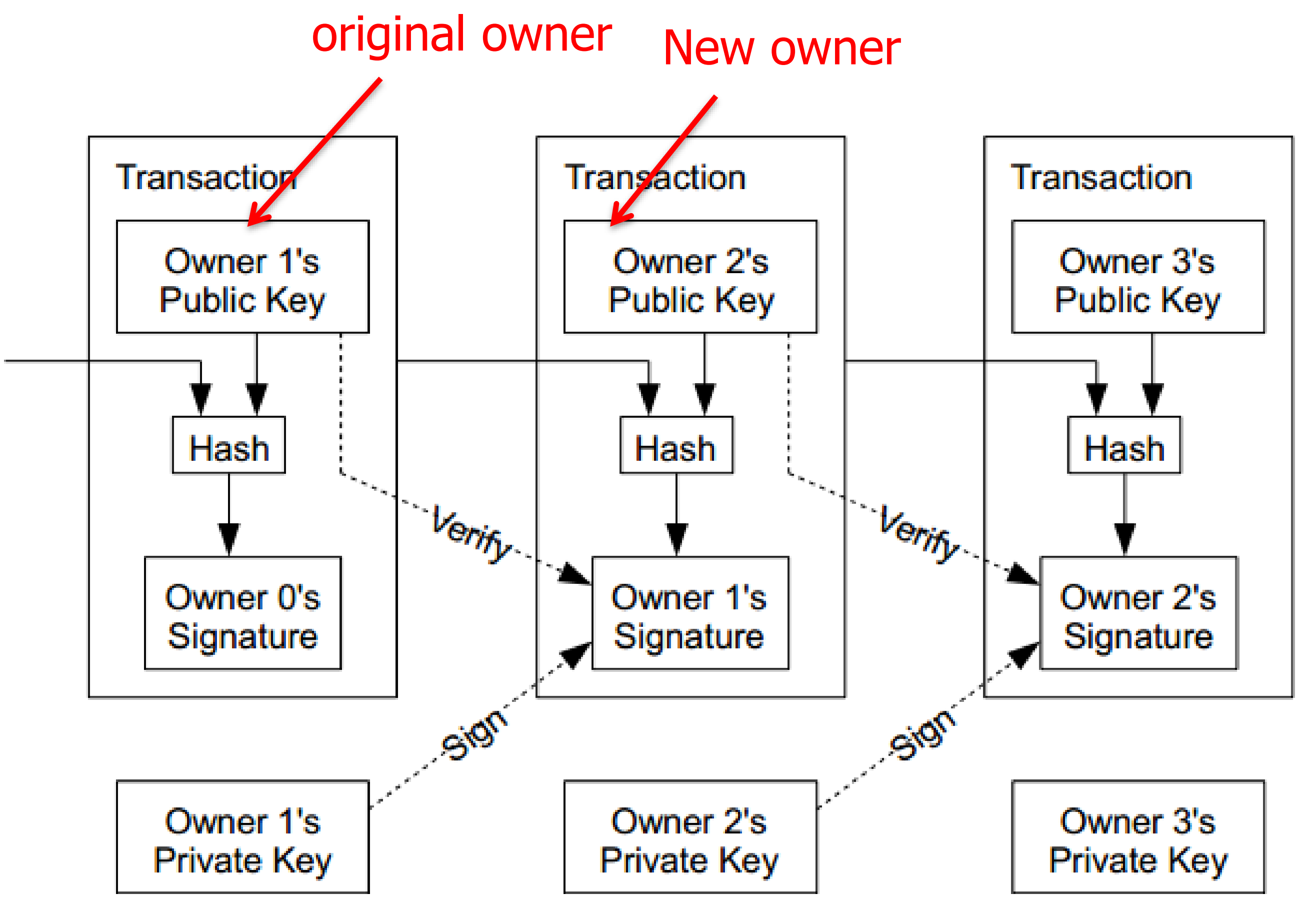
Lab

1. An Online RSA Public and Private Key Generator
http://travistidwell.com/blog/2013/09/06/an-online-rsa-public-and-private-key-generator/

1. RSA Public Key Encryption Demo - haneWIN.net
www.hanewin.net/encrypt/rsa/rsa-test.htm

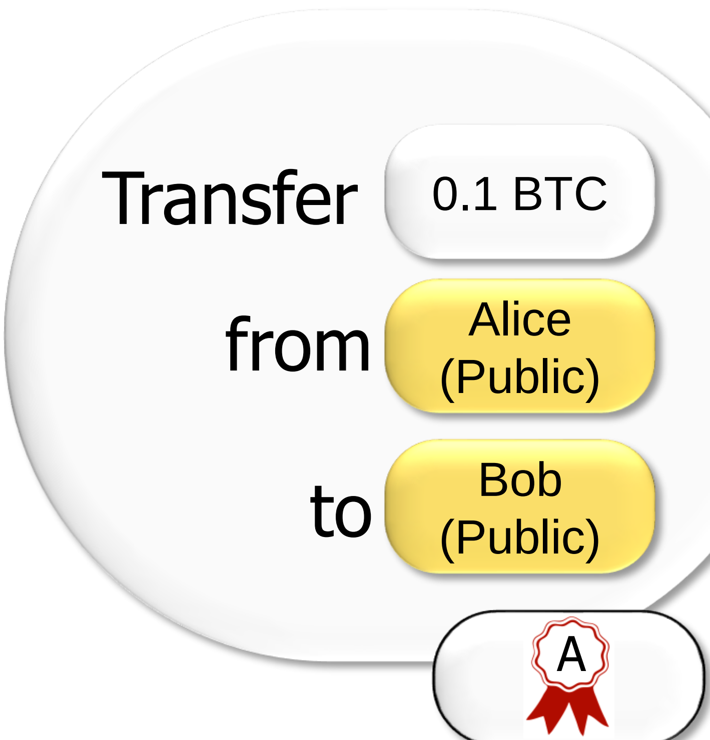
數位簽章套用在 Blockchain/BitCoin

BitCoin transfer: Sign(Previous transaction + New owner's public key)



Sending Bitcoins

To send money, we use transactions. These are messages like this:



In “short”, transactions like this: