RSA公開金鑰密碼機制

- 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及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

1. 選兩個大質數 p 和 q (至少100位數)，令 $N = p \cdot q$
2. 再計算$\varnothing(N)=(p-1)(q-1)$，並選一個與$\varnothing(N)$互質數 e
   $\varnothing(N)$為Euler's Totient函數，其意為與N互質之個數
3. (e,N) 即為公開金鑰

   加密法為 $C = M^e \bmod N$

4. 選一個數 d，滿足 $e \cdot d \bmod \varnothing(N) = 1$
5. d 即為解密金鑰(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

1. 接收方選 $p=3$ ， $q=11$；此時 $N = p \bullet q = 33$
2. 找出 1 個與 $(p-1) \text{ x } (q-1) = (3-1)(11-1) = 2 \text{ x } 10 = 20$ 互質數 $e=3$
3. $(e, N) = (3,33)$ 即為接收方的公開金鑰
4. 接收方選一個數 $d=7$ 當作解密金鑰，
   滿足 $e \bullet d \equiv 1 \bmod 20$ $(7 \text{ x } 3 \equiv 1 \bmod 20)$

令明文 $M = 19$

加密：$C = M^e \bmod N = 19^3 \bmod 33 = 28$

解密：$M = C^d \bmod N = 28^7 \bmod 33 = 19$

相關的數學理論

費瑪(Fermat)定理:

若p為質數且(a,p)互質，則 $a^{p-1} \bmod p = 1$

Fermat's Little Theorem

If p is a prime, and a is not a multiple of p, then Fermat's little theorem says

$$a^{p-1} \bmod p = 1$$

Ex. $2^6 \bmod 7 = 1$

尤拉定理

Euler定理:

若a，n互質，則 $a^{Ø(n)}$ mod n = 1

Euler's Theorem

If gcd(a,n)=1, then

$$a^{\phi(n)} \bmod n = 1$$

where ϕ() called Euler phi function.

尤拉函數:

$Ø(P) = P\text{-}1$ 若P為質數

$Ø(N) = Ø(PQ)$

$= Ø(P)Ø(Q)$

$= (P\text{-}1)(Q\text{-}1)$

It is the number of positive integers less than n that are relatively prime to n.

If n is a prime, $\phi(n)=n\text{-}1$.

If $n=pq$, where p and q are prime, then $\phi(n)=(p\text{-}1)(q\text{-}1)$

模數係下的乘法反元素問題

一般的乘法反元素問題

- 找到一數 x 使得 $(a \times x) = 1$
- x的解為 $x = a^{-1}$

模數係的乘法反元素問題

- 找到一數 x 使得 $1 = (a \times x) \bmod N$
- x的解為 $x = a^{-1} \bmod N$

模數係下的乘法反元素問題(續)

Modular Inverse Problem

- 若a 與N互質，則$x = a^{-1} \bmod n$ 有唯一解
- 若a 與N不互質，則$x = a^{-1} \bmod n$ 無解

  例如： 5模14的乘法反元素為3，但2模14的乘法反元素就不存在。

- 這也就是為何在RSA密碼系統中，使用者所選擇的公開金鑰e必須與$\varnothing(N)$互質的原因。
- 一般來說有兩種方法可以用來解乘法反元素的問題

  - 利用尤拉定理

  - 利用擴展歐幾理德演算法

如何找到模數係下的乘法反元素

方法一：利用尤拉定理 (Euler's Theorem)

$x= a^{-1} \mod n \rightarrow a \times x \mod n =1 \rightarrow$

$\boldsymbol{x=a^{\phi(n)-1} \mod n}$

理由是依據尤拉定理：If $gcd(a,n)=1$, then $a^{\phi(n)} \mod n=1$

例如： 5模 7的乘法反元素如何求得?

$5^{6-1} \mod 7 = 5^5 \mod 7 =3$ where $\phi(n)=6$

Note: 若n為質數，則$\phi(n)$ 可以很輕易求得；但若n為一很大的非質數，則要求的$\phi(n)$相當於解質因數分解的困難度

如何找到模數係下的乘法反元素(續)

方法 2：擴展歐幾理德演算法 (Extended Euclidean Algorithm)

歐幾理德演算法 (Euclidean Algorithm)常被用在解最大公因數的問題上

→ Find $gcd(a,n)$

Let $r_0=n$, $r_1=a$, we get

$r_0=r_1g_1+r_2$, $r_1=r_2g_2+r_3$, ... , $r_{j-2}=r_{j-1}g_{j-1}+r_j$, ...,

$r_{m-4}=r_{m-3}g_{m-3}+r_{m-2}$, $r_{m-3}=r_{m-2}g_{m-2}+r_{m-1}$,

$r_{m-2}=r_{m-1}g_{m-1}+r_m$, $r_{m-1}=r_mg_m$

如何找到模數係下的乘法反元素(續)

根據擴展歐幾理德演算法，若a與n兩數互質，一定可以找到兩個整數s與t，使得 $gcd(a, n) = sa + tn = 1$.

其作法如下：

If $gcd(a,n)=1$, we get $sa + tn = 1$.

We can find s and t by using

$r_m = gcd(a,n) = r_{m-2} - r_{m-1} g_{m-1}$

because $r_{m-1} = r_{m-3} - r_{m-2} g_{m-2}$

so $gcd(a,n) = r_{m-2} - (r_{m-3} - r_{m-2} g_{m-2}) g_{m-1}$

$= (1+g_{m-1} g_{m-2}) r_{m-2} - g_{m-1} r_{m-3}$ and so on.

$sa+tn = 1 \rightarrow sa + tn \bmod n=1 \rightarrow$

$sa \bmod n=1 \rightarrow s = a^{-1} \bmod n$

RSA密碼系統的正確性

$C = E(M) = M^e \ mod\ n$　　　$M = D(C) = C^d \ mod\ n$

$C^d=(M^e)^d=M^{ed} \ mod\ n$ since $ed = 1 \ mod\ (p-1)(q-1)$

so $M^{ed} = M^{a(p-1)(q-1)+1} = MM^{a(p-1)(q-1)} = MM^{a\phi(n)} \ mod\ n$

根據尤拉定理 (Euler's Theorem)，得到

$M \times 1 = M$

因式分解的問題 Factoring Problem

Factoring a number means finding its prime number

Ex: $10 = 2 \times 5$; $60 = 2 \times 2 \times 3 \times 5$

但是解一個大數的值因數分解問題是很困難的。

請試著分解: 3337

RSA 的安全性便是植基於解因式分解的難題上

RSA 演算法

指數運算：計算 $x = A^B \ mod\ N$?

```
a1:= A;     b1:=B;      x:=1;
while b1 ≠ 0 do
begin
    while b1 mod 2 = 0 do
    begin
        b1 := b1 div 2;
        a1 := (a1 × a1) mod N;
    end
    b1 := b1 - 1;
    x := (x × a1) mod N
end
```

計算 $x = A^{17} \ mod\ N$

$$= A^{10001} \ mod\ N$$

$$= A \bullet (((A^2)^2)^2)^2$$

計算 $x = A^{13} \ mod\ N$

$$= A^{1101} \ mod\ N$$

$$= A \bullet ((A^2)^2) \bullet ((A^2)^2)^2$$

計算 $x = A^{7} \ mod\ N$

$$= A^{111} \ mod\ N$$

$$= A \bullet (A^2) \bullet (A^2)^2$$

Public-Key Cryptosystems之特性

1. $D(d, E(e, M)) = M$，可還原性
2. d 和 e很容易求得
3. 若公開(e, n)，別人很難從(e, n)求得d，即只有自己知道如何解密(以e加密)
4. $E(e, D(d, M)) = M$

Public-key Cryptosystems一定要能忍受 Chosen-Plaintext Attack

Public-Key Cryptosystems之特性(續)

- 滿足1~3項稱之為trap-door one-way function
  - “one-way”因易加密而不易解密
  - “trap-door”若知一些特別資訊即可解密
- 滿足1~4項稱之為trap-door one-way permutation
- 1~3項為public-key cryptosystems之要求
- 若同時滿足第4項要求，則該保密法可用來製作數位簽章。

RSA數位簽章

張三使用自己的祕密金鑰對文件 ***M*** 做數位簽章***S***

張三

$$S = M^{d_{張}} \bmod N_{張}$$

一般使用時會先將文件*M*作HASH函數處理，使得*HASH(M)*比*N*小

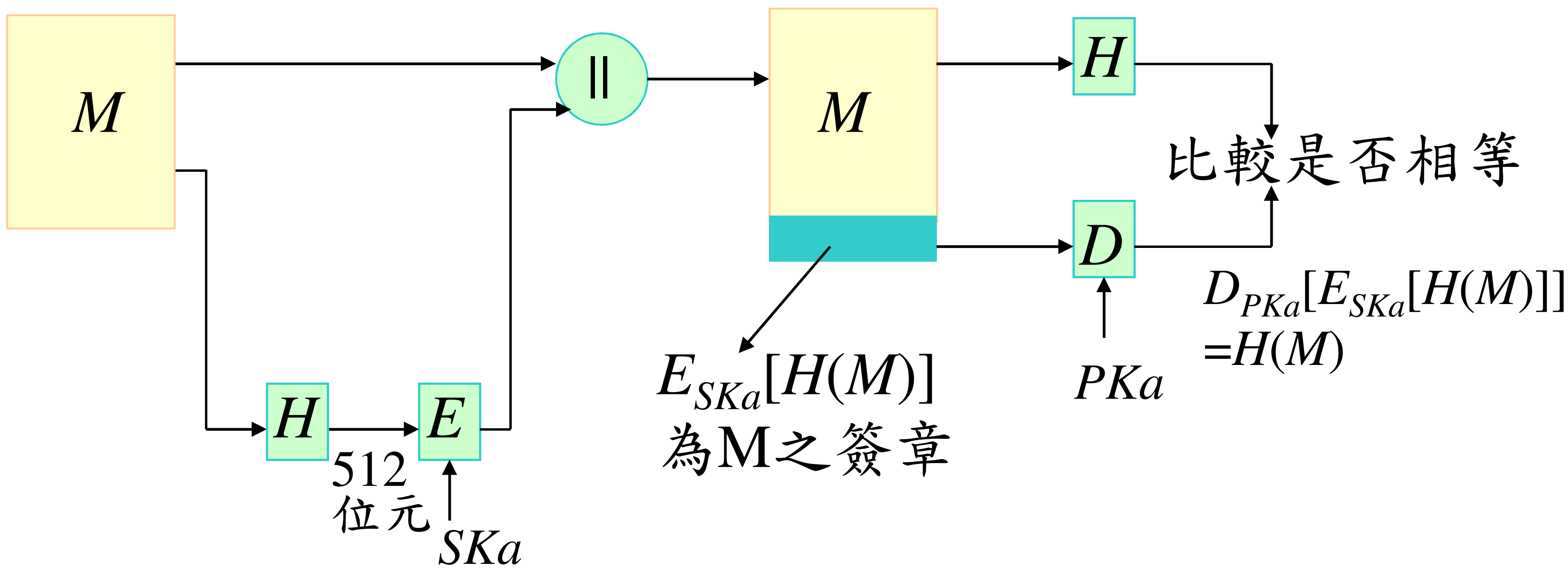
傳送*M*及*S*

李四

$$M =? S^{e_{張}} \bmod N_{張}$$

李四使用張三的公開金鑰確認數位簽章及文件

數位簽章法



RSA數位簽章+加密

張三對文件 M 做數位簽章後用李四的公用金鑰將簽章加密

張三

$$C = (M^{d_{張}} \bmod N_{張})^{e_{李}} \bmod N_{李}$$

傳送密文 C

李四

$$M = (C^{d_{李}} \bmod N_{李})^{e_{張}} \bmod N_{張}$$

李四使用私有金鑰解開密文 C 再用張三的公開金鑰確認數位簽章