Lecture 9
Mathematics of Asymmetric-Key Encryption

Jason Lin

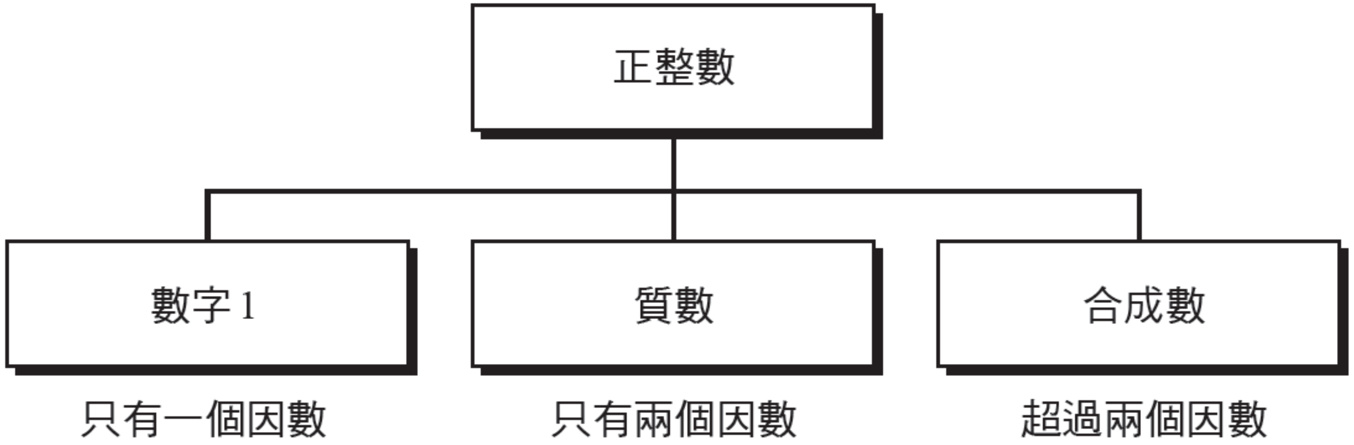
學習目標

- 介紹質數及其在密碼學上的應用
- 討論一些質數測試演算法，並探討它們的效能
- 討論因數分解的演算法及其在密碼學上的應用
- 說明中國餘數定理及其應用
- 介紹二次同餘
- 介紹模數下的指數和對數

9.1 質數

- 本節討論主題
  - 定義
  - 質數的個數
  - 質數的檢查
  - 尤拉 phi 函數
  - 費馬小定理
  - 尤拉定理

圖 9.1 正整數的三個群組



注意

一個質數只能被自己和 1 所整除。

範例 9.1

- 最小的質數為何？
- 解法：最小的質數是 2，因為 2 只能被 2（它自己）和 1 所整除。

範例 9.2

- 列出所有小於 10 的質數。
- 解法：小於 10 的質數有四個：2、3、5 和 7。有趣的是，我們注意到在 1 到 10 這個範圍中，質數所佔的百分比為 40%，但這個百分比會隨著範圍擴大而下降。

9.1.2 質數的個數

- 質數的個數

注意

我們有無限多個質數。

- $\pi(n)$: 小於或等於某個實數 n 的質數個數

$$[n/(\ln n)] < \pi(n) < [n/(\ln n - \boxed{1})]$$

Gaussian 跟 Legendre 的猜想漸近函數

Legendre常數

範例 9.3

- 以下是一個明顯的例子，假設所有質數所形成的集合如下：$\{2,3,5,7,11,13,17\}$。令 $P = 2 \times 3 \times 5 \times 7 \times 11 \times 13 \times 17 = 510510$，而 $P + 1 = 510511$。
  我們發現 $510511 = 19 \times 97 \times 277$；這些質數沒有任何一個是在原來的集合之中。因此，大於 17 的質數必定存在。

範例 9.4

- 求出小於 1,000,000 的質數個數。
- 解法：由近似值的關係式可知範圍為 72,383 到 78,543，真正的質數個數為 78,498。

表 9.1 埃拉托斯特尼（Eratosthenes）篩選法

- 如果 n 不太大，一個簡單計算 $\pi(n)$ 的方法

	2	3	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	~~25~~	~~26~~	~~27~~	~~28~~	29	~~30~~
31	~~32~~	~~33~~	~~34~~	~~35~~	~~36~~	37	~~38~~	~~39~~	~~40~~
41	~~42~~	43	~~44~~	~~45~~	~~46~~	47	~~48~~	~~49~~	~~50~~
~~51~~	~~52~~	53	~~54~~	~~55~~	~~56~~	~~57~~	~~58~~	59	~~60~~
61	~~62~~	~~63~~	~~64~~	~~65~~	~~66~~	67	~~68~~	~~69~~	~~70~~
71	~~72~~	73	~~74~~	~~75~~	~~76~~	~~77~~	~~78~~	79	~~80~~
~~81~~	~~82~~	83	~~84~~	~~85~~	~~86~~	~~87~~	~~88~~	89	~~90~~
~~91~~	~~92~~	~~93~~	~~94~~	~~95~~	~~96~~	97	~~98~~	~~99~~	~~100~~

9.1.3 質數的檢查

- 給定一個數值 n，如何判定 n 是否為質數？

  答案就是我們必須逐一測試所有小於 $\sqrt{n}$ 質數是否可以整除 n。
  我們知道這不是一個有效率的方法，但卻是個好的開始

範例 9.5

- 97 是否為質數？

解法：因為 $\lfloor\sqrt{97}\rfloor = 9$，所以小於 $\sqrt{97}$ 的質數為 2、3、5 和 7。我們必須確認 97 是否會被這些數的其中之一整除。結果發現這些數都不能整除 97，因此 97 是質數

範例 9.6

- 301 是否為質數？

  解法：因為 $\lfloor\sqrt{301}\rfloor = 17$，所以我們需要檢查 2、3、5、7、11、13 和 17。數值 2、3 和 5 無法整除 301，但是 7 可以，因此 301 不是質數

尤拉 phi 函數

- 尤拉 phi 函數（Euler's phi function，$\phi(n)$），又稱為尤拉 totient 函數（Euler's totient function），主要是用來找尋小於等於正整數 n 與 n 互質的正整數個數，在密碼學中扮演非常重要的角色
  - $\phi(1) = 1$
  - $\phi(p) = p - 1$，若 p 是一個質數
  - $\phi(m \times n) = \phi(m) \times \phi(n)$，若正整數 m 和 n 互質
  - $\phi(p^e) = p^e - p^{e-1}$，若 p 是質數且 e 為正整數

尤拉 phi 函數 (續)

- 我們可以合併以上四個規則計算出 $\phi(n)$ 的值。舉例來說，如果 n 可以分解成

$$n = p_1^{e_1} \times p_2^{e_2} \times \cdots \times p_k^{e_k}$$

則可以使用規則 3 和規則 4

注意

求出 $\phi(n)$ 的難度相當於對 n 進行因數分解的難度。

尤拉函數規則三的證明

- 【證明】若正整數 m 與 n 互質，則 $\phi(mn) = \phi(m)\phi(n)$
  - 假設一正整數 $x < mn$，且 $x = k_1 m + p = k_2 n + q$
  - 又因為 $(k_1 m + p, m) = (p, m)$，$(k_2 n + q, n) = (q, n)$，所以若 p、q 分別與 m、n 互質，則 x 一定與 mn 互質，反之亦然
  - 透過中國餘數定理如下：

    方程組 $\begin{cases} x \equiv p \pmod{m} \\ x \equiv q \pmod{n} \end{cases}$ 的解可以寫成 $x = pnn^{-1} + qmm^{-1} \pmod{mn}$
  - 故數對 (p, q) 要滿足 $0 < p \le m$，$0 < q \le n$，$(p, m) = 1$，$(q, n) = 1$；與小於 mn 且與 mn 互質的正整數 x 一一對應
  - 由尤拉函數的定義與乘法原理，前一種數對 (p, q) 的個數為 $\phi(m)\phi(n)$；而後一種數 x 的個數為 $\phi(mn)$，故得證

尤拉函數規則四的證明

- 【證明】若 p 是質數且 e 為正整數，則 $\phi(p^e) = p^e - p^{e-1}$
  - 對於所有 $x \le p^e$ 使得 $\gcd(x, p^e) \neq 1$ 的皆為 p 之倍數，也就是 $x = \{1p, 2p, 3p, \ldots, p^{e-1}p\}$，總共 p^{e-1} 個數字與 p^e 不互質，故得證

範例 9.7

- $\phi(13)$ 的值為何？
- 解法：由於 13 是質數，$\phi(13) = (13 - 1) = 12$

範例 9.10

- 我們可以說 $\phi(49) = \phi(7) \times \phi(7) = 6 \times 6 = 36$ 嗎？

  解法：不可。

  因為規則 3 只有在 m 和 n 互質時才可以使用。
  在此，$49 = 7^2$，我們可以使用規則 4：$\phi(49) = 7^2 - 7^1 = 42$

範例 9.11

- 在 $\mathbf{Z}_{14*}$ 中有多少個元素？
- 解法：答案是 $\phi(14) = \phi(7) \times \phi(2) = 6 \times 1 = 6$
- 這些成員是 1、3、5、9、11 和 13

注意

有趣的觀點：若 $n > 2$，則 $\phi(n)$ 的值是偶數。

9.1.5 費馬小定理

- 令質數 p 與底數 a 互質
- 第一種版本

$$\boxed{a^{p-1} \equiv 1 \pmod p}$$

- 第二種版本

$$\boxed{a^{p} \equiv a \pmod p}$$

Pierre de Fermat (1601 – 1665)

費馬小定理的證明

- 因為 $\gcd(a,p)=1$，考慮 $1\times a, 2\times a, 3\times a, \ldots, (p-1)\times a$ 共 $(p-1)$ 個數
- 將這 $(p-1)$ 個數分別除以 p ，餘數分別為 $r_1, r_2, r_3, \ldots, r_{p-1}$，則整數集合 $\{r_1, r_2, r_3, \ldots, r_{p-1}\}$ 為正整數集合 $\{1, 2, 3, \ldots, (p-1)\}$ 的重新排列，即每一個餘數為從 1 到 $(p-1)$ 恰好各出現一次
- 這是因為對於任兩個相異的 ka 而言 $(k=1,2,3,\ldots,(p-1))$，其差不是 p 的倍數，所以不會有相同餘數出現，且又因為任一個 ka 不為 p 的倍數，所以餘數不為 0，因此

$$1\cdot 2\cdot 3\cdots(p-1)\equiv(1\cdot a)\cdot(2\cdot a)\cdots[(p-1)\cdot a]\pmod p$$

即 $1\cdot 2\cdot 3\cdots(p-1)\equiv 1\cdot 2\cdot 3\cdots(p-1)\,a^{p-1}\pmod p$

$a^{p-1}\equiv 1\pmod p$

範例 9.12

- 計算 $6^{10} \bmod 11$。
- 解法：我們可得 $6^{10} \bmod 11 = 1$

  這是在 $p = 11$ 時，使用費馬小定理的第一種版本計算出來的

範例 9.13

- 計算 $3^{12} \bmod 11$。
- 解法：此處指數（12）和模數（11）是不同的。藉由替換法，這個式子可以使用費馬小定理來求解

$$3^{12} \bmod 11 = (3^{11} \times 3) \bmod 11 = (3^{11} \bmod 11)\ (3 \bmod 11) = (3 \times 3) \bmod 11 = 9$$

乘法反元素

$a^{-1} \bmod p = a^{p-2} \bmod p$

已知 a 在 $\bmod p$ 的乘法反元素 a^{-1}，滿足 $a \times a^{-1} \equiv 1 \pmod p$；
透過費馬小定理的第一種版本，得知 $a^{p-1} \equiv 1 \pmod p$。
讓 $a^{p-1} \bmod p \equiv a \times a^{p-2} \bmod p \equiv a \times a^{-1} \bmod p = 1$，得證。

範例 9.14

- 當模數是質數時，我們可以不使用歐幾里德延伸演算法來求出乘法反元素的解：
  - $8^{-1} \bmod 17 = 8^{17-2} \bmod 17 = 8^{15} \bmod 17 = 15 \bmod 17$
  - $5^{-1} \bmod 23 = 5^{23-2} \bmod 23 = 5^{21} \bmod 23 = 14 \bmod 23$
  - $60^{-1} \bmod 101 = 60^{101-2} \bmod 101 = 60^{99} \bmod 101 = 32 \bmod 101$
  - $22^{-1} \bmod 211 = 22^{211-2} \bmod 211 = 22^{209} \bmod 211 = 48 \bmod 211$

9.1.6 尤拉定理

- 令正整數 n 與底數 a 互質
- 第一種版本

$$\boxed{a^{\phi(n)} \equiv 1 \pmod{n}}$$

- 第二種版本

$$\boxed{a^{k\times\phi(n)+1} \equiv a \pmod{n}}$$

Leonhard Euler (1707 – 1783)

尤拉定理的證明

- 因為 $\gcd(a,n)=1$，假設 $S=\{x_1,x_2,\dots,x_{\phi(n)}\}$ 為小於 n 與 n 互質之正整數，考慮 $x_1\times a, x_2\times a, x_3\times a, \dots, x_{\phi(n)}\times a$ 共 $\phi(n)$ 個數
- 將他們分別除以 n ，餘數分別為 $r_1, r_2, r_3, \dots, r_{\phi(n)}$，則集合 $\{r_1, r_2, r_3, \dots, r_{\phi(n)}\}$ 為集合 $\{x_1,x_2,\dots,x_{\phi(n)}\}$ 的重新排列，即每一種餘數從 x_1 到 $x_{\phi(n)}$ 恰好各出現一次
- 這是因為對於任兩個相異的ka而言($k=x_1,x_2,x_3,\dots,x_{\phi(n)}$)，其差不是 n 的倍數，所以不會有相同餘數出現，且又因為任一個 ka 不為 n 的倍數（a 跟 k 皆與 n 互質），所以餘數不為 0，因此

$$x_1\cdot x_2\cdot x_3\cdots x_{\phi(n)}\equiv(x_1\cdot a)\cdot(x_2\cdot a)\cdot(x_3\cdot a)\cdots(x_{\phi(n)}\cdot a)\pmod n$$

即 $x_1\cdot x_2\cdot x_3\cdots x_{\phi(n)}\equiv x_1\cdot x_2\cdot x_3\cdots x_{\phi(n)}\cdot a^{\phi(n)}\pmod n$

$a^{\phi(n)}\equiv 1\pmod n$

Ex. 尤拉

- $6^{24} \bmod 35 =$
  - Hint: $\phi(35) = 24$

- $20^{62} \bmod 77 =$
  - Hint: $\phi(77) = 60$

乘法反元素

- 當模數是合成數時，我們也可以使用尤拉定理求出乘法反元素

$$\boxed{a^{-1} \bmod n \equiv a^{\phi(n)-1} \bmod n}$$

莫仙尼（Mersenne）質數

- 一個可以表示成 $M_p = 2^p - 1$ 的數值被稱為莫仙尼數，其可能是質數，也可能不是

$$\boxed{M_p = 2^p - 1}$$

$M_2 = 2^2 - 1 = 3$
$M_3 = 2^3 - 1 = 7$
$M_5 = 2^5 - 1 = 31$
$M_7 = 2^7 - 1 = 127$
$M_{11} = 2^{11} - 1 = 2047$ **不是質數（$2047 = 23 \times 89$）**
$M_{13} = 2^{13} - 1 = 8191$
$M_{17} = 2^{17} - 1 = 131071$

費馬（Fermat）質數

- 一個可以表示成 $F_n = 2^{2^n} + 1$ 的數值被稱為費馬數，其有可能是質數，也可能不是

$$\boxed{F_n = 2^{2^n} + 1}$$

$F_1 = 5$
$F_2 = 17$
$F_3 = 257$
$F_4 = 65537$
$F_5 = 4294967297 = 641 \times 6700417$ 不是質數

9.2 質數測試法

- 迄今為止，人們尚未找到易於計算且能夠產生所有質數的公式，但對於質數所應該具備的性質已有了大量的研究
- 找出一個能正確且有效率地測試非常大的整數並判定其為質數或合成數的演算法，一直都是數論和密碼學的挑戰。然而，近年來所發展的一些方法，看起來十分有希望可以完成這項挑戰

9.2 質數測試法 (續)

- 本節討論主題
  - 確定式的演算法
    - 整除性測試法
    - AKS 測試法
  - 機率式的演算法
    - 費馬測試法
    - 平方根測試法
    - Miller-Rabin 測試法

整除性測試法

注意

整除性測試法的位元運算複雜度是以指數成長的。

```
Divisibility_Test (n)                         //測試n是否為質數
{
  r ← 2
  while (r < √n )
  {
   if (r | n) return  "a composite"
   r ← r + 1
  }
  return  "a prime"
}
```

範例 9.18

- 假設 n 有 200 個位元。對 n 執行整除性測試法演算法需要多少次位元運算？
- 解法：這個演算法的位元運算複雜度為 $2^{(\log_2 n)/2}$，表示此演算法需要 2^{100} 次位元運算。在一部 1 秒可以執行 2^{30} 次（約 10 億次）位元運算的電腦上，此演算法需要 2^{70} 秒來完成這個測試，這樣無法在合理的時間內算完。

AKS 測試法

- AKS（Agrawal–Kayal–Saxena）質數測試是一個決定型質數測試演算法，主要是基於以下定理：
  正整數 $n\ (\geq 2)$ 是質數若且唯若 $(x+1)^n-(x^n+1) \equiv 0 \pmod{n}$
- 這個定理是費馬小定理的一般化，可以簡單的使用二項式係數的特徵來證明出此定理。
  對任何 $0<k<n$，$\binom{n}{k}=\frac{[n(n-1)(n-2)\cdots(n-k+1)]}{(n-k)!} \equiv 0 \pmod{n}$，若且唯若 n 是質數。

範例 9.19

- 假設 n 有 200 個位元。對 n 執行 AKS 演算法需要多少次位元運算？
- 解法：此演算法的位元運算複雜度為 $O((\log_2 \log_2 n)^{12})$，表示這個演算法只需執行 $(\log_2 200)^{12} = 39{,}547{,}615{,}483$ 次位元運算。在一部 1 秒可以執行 10 億次位元運算的電腦上，這個演算法僅需 40 秒

費馬測試法

$$\boxed{\text{若 } n \text{ 是質數，則 } a^{n-1} \equiv 1 \bmod n\text{。}}$$

若 n 是質數，$a^{n-1} \bmod n \equiv 1$ 一定會成立。
若 n 是合成數，$a^{n-1} \bmod n \equiv 1$ 有可能會成立。

範例 9.20

- 561 能否通過費馬測試法？
- 解法：我們使用 2 為底數：

$$2^{561-1} = 1 \bmod 561$$

這個整數可以通過費馬測試法，

但它不是質數，因為 $561 = 33 \times 17$。

平方根測試法

若 n 是質數，只有 $\sqrt{1} \bmod n = \pm 1$。
若 n 是合成數，除了 $\sqrt{1} \bmod n = \pm 1$，可能還有其他解。

範例 9.21

- 當 n 為 7（質數）時，1 在模 n 下的平方根為何？
- 解法：其平方根只有 1 和 -1。
  我們可以發現：

$$
\begin{array}{ll}
1^2 = 1 \bmod 7 & (-1)^2 = 1 \bmod 7 \\
2^2 = 4 \bmod 7 & (-2)^2 = 4 \bmod 7 \\
3^2 = 2 \bmod 7 & (-3)^2 = 2 \bmod 7
\end{array}
$$

範例 9.23

- 當 n 為 17（質數）時，1 在模 n 下的平方根為何？
- 解法：其平方根只有兩個解：1 和 -1。

$$
\begin{array}{ll}
1^2 = 1 \bmod 17 & (-1)^2 = 1 \bmod 17 \\
2^2 = 4 \bmod 17 & (-2)^2 = 4 \bmod 17 \\
3^2 = 9 \bmod 17 & (-3)^2 = 9 \bmod 17 \\
4^2 = 16 \bmod 17 & (-4)^2 = 16 \bmod 17 \\
5^2 = 8 \bmod 17 & (-5)^2 = 8 \bmod 17 \\
6^2 = 2 \bmod 17 & (-6)^2 = 2 \bmod 17 \\
7^2 = 15 \bmod 17 & (-7)^2 = 15 \bmod 17 \\
8^2 = 13 \bmod 17 & (-8)^2 = 13 \bmod 17
\end{array}
$$

注意，我們並不需要測試比 8 大的整數，因為 $9 = -8 \bmod 17$，以此類推。

範例 9.24

- 當 n 為 8（合成數）時，1 在模 n 下的平方根為何？

$$
\begin{array}{ll}
1^2 = 1 \bmod 8 & (-1)^2 = 1 \bmod 8 \\
 & \cdots \\
3^2 = 1 \bmod 8 & (-3)^2 = 1 \bmod 8 \\
 & \cdots \\
5^2 = 1 \bmod 8 & (-5)^2 = 1 \bmod 8
\end{array}
$$

- 當 n 為 22（合成數）時，1 在模 n 下的平方根為何？

$$
1^2 = 1 \bmod 22 \qquad (-1)^2 = 1 \bmod 22
$$

Miller-Rabin 測試法

- 令輸入 n 為正奇數，且 $n-1=m\times 2^k$，其中 $k\geq 1$，m 為奇數
- 若 n 為質數 p，則 $a^2 \bmod p=1$ 若且為若 $a \bmod p=\pm 1$
- 若 a 為整數且 $1<a<p-1$，則下列情況會有一個成立
  - $a^m \bmod p=\pm 1$
  - 在 $1<j<k-1$ 中的某些數值會滿足 $(a^m)^{2^j}\equiv -1 \pmod p$

注意

Miller-Rabin 測試法需要從步驟 0 執行到步驟 $k-1$。

演算法 9.2 Miller-Rabin 測試法的虛擬碼

```
Miller_Rabin_Test (n, a)                    //n 是要被測試的數，a 是底數
{
    Find m and k such that n − 1 = m × 2^k
    T ← a^m mod n
    if (T = ± 1)   return "a prime"
    for (i ← 1 to k − 1)                    //k – 1 為所需要執行的最大步驟數
    {
        T ← T^2 mod n
        if (T = +1) return "a composite"
        if (T = −1) return "a prime"
    }
    return "a composite"
}
```

Miller-Rabin 測試法的證明 (1/2)

- 【證明】令 $n-1=m\times 2^k$，其中 m 和 n 皆為奇數。
  若 $a^m \not\equiv 1 \pmod{n}$ … ① 和 $a^{2^i m} \not\equiv -1 \pmod{n}$ … ②
  對所有 $i=0,1,\dots,k-1$ 和 $n \nmid a$，則 n 為合成數
- 假設 p 為質數，
  $\boxed{p-1}=m\times 2^k$，且 m 是奇數
  $a^m, a^{2m}, a^{2^i m}, \dots, \underbrace{a^{\boxed{2^k m}}}_{\boxed{1}} \pmod{p}$，且 $p \nmid a \Rightarrow \gcd(p,a)=1$
- 令 $x^2 \equiv 1 \pmod{p}$，得到唯一兩種可能的解 $x=\pm 1 \pmod{p}$

Miller-Rabin 測試法的證明 (2/2)

- 因此，現在將 $a^{2^k m}$ 一直取平方根，總會得到 1 和 -1，以下有兩種之一的可能情況會發生
  1. 如果得到了 -1，存在某些 $a^{2^i m} \pmod p$ 不為 1，因為 -1 為 1 的一個平方根解，即②式不成立。
  2. 從未得到過 -1，代表所有的 $a^{2^i m}$ 皆為 1，也因此使得①式不成立。
- 總結上述的分析，必須要①②式皆成立才能確定 n 為合成數，故得證

範例 9.25

- 561 能否通過 Miller-Rabin 測試法？
- 解法：我們用 2 為底數，令 $561 - 1 = 35 \times 2^4$，這表示 $m = 35$、$k = 4$ 和 $a = 2$。

初始化：	$T = 2^{35} \bmod 561 = 263 \bmod 561$	
$k = 1$：	$T = 263^2 \bmod 561 = 166 \bmod 561$	
$k = 2$：	$T = 166^2 \bmod 561 = 67 \bmod 561$	
$k = 3$：	$T = 67^2 \bmod 561 = +1 \bmod 561$	→ **是合成數**

範例 9.26

- 已知 27 不是質數，讓我們用 Miller-Rabin 測試法來驗證看看。
- 解法：使用 2 為底數，令 $27 - 1 = 13 \times 2^1$，這表示 $m = 13$、$k = 1$ 和 $a = 2$。這種情形下，因為 $k - 1 = 0$，我們只執行初始化的步驟：$\mathrm{T} = 2^{13} \bmod 27 = 11 \bmod 27$。然而，因為此演算法不會進入迴圈， 所以它會回傳是合成數。

範例 9.28

- 數值 4033 是一個合成數（37×109）。它可以通過建議的質數測試法嗎？
- 解法
  - 我們先執行整除性測試法，測試了質數 2、3、5、7、11、17 和 23，都無法整除 4033。
  - 改以 2 為底數執行 Miller-Rabin 測試法，$4033 - 1 = 63 \times 2^6$，這表示 m 為 63 而 k 為 6。

範例 9.28 (續)

初始化：$T \equiv 2^{63} \pmod{4033} \equiv 3521 \pmod{4033}$

$k = 1$ $\quad T \equiv T^2 \equiv 3521^2 \pmod{4033} \equiv -1 \pmod{4033}$ $\quad \rightarrow$ **通過測試**

- 但我們並不滿足於此。我們繼續以 3 為底數進行測試。

初始化：$T \equiv 3^{63} \pmod{4033} \equiv 3551 \pmod{4033}$

$k = 1$ $\quad T \equiv T^2 \equiv 3551^2 \pmod{4033} \equiv 2443 \pmod{4033}$

$k = 2$ $\quad T \equiv T^2 \equiv 2443^2 \pmod{4033} \equiv 3442 \pmod{4033}$

$k = 3$ $\quad T \equiv T^2 \equiv 3442^2 \pmod{4033} \equiv 2443 \pmod{4033}$

$k = 4$ $\quad T \equiv T^2 \equiv 2443^2 \pmod{4033} \equiv 3442 \pmod{4033}$

$k = 5$ $\quad T \equiv T^2 \equiv 3442^2 \pmod{4033} \equiv 2443 \pmod{4033}$ $\quad \rightarrow$ **測試失敗（合成數）**

9.3 因數分解

- 因數分解（Factorization）的問題從過去一直被研究至今，而這樣的研究在未來似乎會持續下去。因數分解在公開金鑰密碼系統（參見第十章）的安全性扮演一個非常重要的角色。
- 本節討論主題
  - 算術的基本定理
  - 因數分解的方法
  - 費馬分解法

9.3.1 算術的基本定理

- 最大公因數

$$a = p_1^{a_1} \times p_2^{a_2} \times \cdots \times p_k^{a_k} \qquad b = p_1^{b_1} \times p_2^{b_2} \times \cdots \times p_k^{b_k}$$

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} \times p_2^{\min(a_2, b_2)} \times \cdots \times p_k^{\min(a_k, b_k)}$$

- 最小公倍數

$$a = p_1^{a_1} \times p_2^{a_2} \times \cdots \times p_k^{a_k} \qquad b = p_1^{b_1} \times p_2^{b_2} \times \cdots \times p_k^{b_k}$$

$$\mathrm{lcm}(a, b) = p_1^{\max(a_1, b_1)} \times p_2^{\max(a_2, b_2)} \times \cdots \times p_k^{\max(a_k, b_k)}$$

$$\mathrm{lcm}(a, b) \times \gcd(a, b) = a \times b$$

演算法 9.3 試除因數分解法的虛擬碼

```
Trial_Division_Factorization (n)                //n 是要分解的數值
{
    a ← 2
    while (a ≤ √n )
    {
        while (n mod a = 0)
        {
            output a                              //一個接一個地輸出因數
            n = n / a
        }
        a ← a + 1
    }
    if (n > 1) output n                           //n 沒有其他的因數
}
```

範例 9.30

- 使用試除因數法來找出 1523357784 的因數。
- 解法：我們執行一個程式並得到以下的結果

$$1523357784 = 2^3 \times 3^2 \times 13 \times 37 \times 43987$$

9.3.3 費馬分解法

- 費馬因數分解法（Fermat Factorization Method，演算法 9.4）可以將一個數值 n 分解成兩個正整數 a 和 b（不一定為質數），使得 $n = a \times b$

$$n = x^2 - y^2 = a \times b \text{，其中 } a = (x + y) \text{ 和 } b = (x - y)$$

演算法 9.4 費馬因數分解法的虛擬碼

```
Feramat_Factorization (n)                         //n 是要分解的數值
{
    x ← √n                                        // 大於 √n 的最小整數
    while (x < n)
    {
    w ← x^2 − n
    if (w is perfect square)   y ← √w;  a ← x+y;  b ← x−y;   return a and b
    x ← x + 1
    }
}
```

9.4 中國餘數定理

- 中國餘數定理（Chinese Remainder Theorem, CRT）是用來求解模數兩兩相異且互質之單變數的同餘方程組。此方程組如下所述：

$$
\begin{aligned}
x &\equiv a_1 \pmod{m_1} \\
x &\equiv a_2 \pmod{m_2} \\
&\cdots \\
x &\equiv a_k \pmod{m_k}
\end{aligned}
$$

範例 9.35

- 以下範例為模數均相異的同餘方程組：

$$x \equiv 2 \pmod 3$$
$$x \equiv 3 \pmod 5$$
$$x \equiv 2 \pmod 7$$

我們將在下一節說明這個方程組的解法；目前，先注意到這個方程組的解為 $x = 23$
這個值可以滿足所有的方程式：$23 \equiv 2 \pmod 3$、$23 \equiv 3 \pmod 5$ 和 $23 \equiv 2 \pmod 7$

範例 9.35 (續)

- 解法：我們可以依照以下的步驟來解這個方程組：
  - 求出 $M = m_1 \times m_2 \times \cdots \times m_k$，這是所有方程式共同的模數
  - 求出 $M_1 = M/m_1, M_2 = M/m_2, \ldots, M_k = M/m_k$
  - 在相對應的模數 $(m_1, m_2, \ldots, m_k)$ 下，求出 $M_1, M_2, \ldots, M_k$ 的乘法反元素 $M_1^{-1}, M_2^{-1}, \ldots, M_k^{-1}$
  - 這些方程式共同的解為：

$$x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \cdots + a_k \times M_k \times M_k^{-1}) \bmod M$$

範例 9.36

- $x \equiv 2 \pmod 7$ and $x \equiv 3 \pmod 9$

  $x = 30$

- $x \equiv 4 \pmod 5$ and $x \equiv 10 \pmod{11}$

  $x = 54$

- $x \equiv 7 \pmod{13}$ and $x \equiv 11 \pmod{12}$

  $x = 59$

範例 9.38

- 假設我們需要計算 $z = x + y$，其中 $x = 123$ 和 $y = 334$，但系統只允許使用小於 100 的數值，這些數值可以用下列表示法來代替：

$$x \equiv 24 \pmod{99} \qquad y \equiv 37 \pmod{99}$$

$$x \equiv 25 \pmod{98} \qquad y \equiv 40 \pmod{98}$$

$$x \equiv 26 \pmod{97} \qquad y \equiv 43 \pmod{97}$$

範例 9.38 (續)

- 將每一組模數相同的 x 和 y 相加，我們得到

$$
\begin{aligned}
x + y &\equiv 61 \pmod{99} & \quad z &\equiv 61 \pmod{99} \\
x + y &\equiv 65 \pmod{98} & \quad z &\equiv 65 \pmod{98} \\
x + y &\equiv 69 \pmod{97} & \quad z &\equiv 69 \pmod{97}
\end{aligned}
$$

現在，我們可以用中國餘數定理來解這三個方程式，並求出 z，其中一個可以滿足這些方程式的解為 $z = 457$

9.5 二次同餘

- 在密碼學中，我們也必須探討二次同餘（Quadratic Congruence），亦即形式為 $a_2x^2 + a_1x + a_0 \equiv 0 \pmod{n}$ 的方程式
- 此處我們將探討的範圍限制在 $a_2 = 1$ 和 $a_1 = 0$ 的二次方程式，亦即形式為 $x^2 \equiv a \pmod{n}$ 的方程式

9.5 二次同餘 (續)

- 本節討論主題
  - 模數為質數的二次同餘
  - 模數為合成數的二次同餘

範例 9.39

- 方程式 $x^2 \equiv 3 \pmod{11}$ 有兩個解，$x \equiv 5 \pmod{11}$ 和 $x \equiv -5 \pmod{11}$。但我們注意到 $-5 \equiv 6 \pmod{11}$，所以此方程式真正的解為 5 和 6，同時這兩個解是非同餘的

二次剩餘與非二次剩餘

- 在方程式 $x^2 \equiv a \pmod p$ 中，如果此方程式有兩個解，則 a 稱為二次剩餘（Quadratic Residue，QR）；如果此方程式無解，則 a 稱為非二次剩餘（Quadratic Nonresidue，NQR）

範例 9.41

- 在 $\mathbf{Z}_{11*}$ 中有十個元素，其中剛好有五個是二次剩餘，而另五個是非二次剩餘。換句話說，我們可以將 $\mathbf{Z}_{11*}$ 分成兩個不同的集合：QR 和 NQR，如下圖所示。

$$\boldsymbol{Z_{11^*}} = \{\mathbf{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}\}$$

集合 $\boldsymbol{QR} = \{\mathbf{1, 3, 4, 5, 9}\}$

每個元素都有平方根

集合 $\boldsymbol{NQR} = \{\mathbf{2, 6, 7, 8, 10}\}$

沒有任何元素有平方根

尤拉準則

- 若 $a^{(p-1)/2} \equiv 1 \pmod{p}$，則 a 在模 p 下是二次剩餘 QR
- 若 $a^{(p-1)/2} \equiv -1 \pmod{p}$ ，則 a 在模 p 下是非二次剩餘 NQR

尤拉準則的證明

- 由於 p 是一個奇質數，由費馬小定理知 $a^{p-1} \equiv 1 \pmod p$，但是 $p-1$ 是一個偶數，所以有$(a^{\frac{p-1}{2}} - 1) \cdot (a^{\frac{p-1}{2}} + 1) \equiv 0 \pmod p$
- p 是一個質數，所以 $a^{\frac{p-1}{2}} - 1$ 和 $a^{\frac{p-1}{2}} + 1$ 中必有一個是 p 的倍數，因此，$a^{\frac{p-1}{2}}$ 模 p 的餘數必然是 1 或 -1
- 已知若 a 是模 p 的二次剩餘，則存在 $x^2 \equiv a \pmod p$，p 跟 a、x 互質
- 根據費馬小定理得 $a^{\frac{p-1}{2}} \equiv x^{p-1} \equiv 1 \pmod p$
- 所以若 $a^{\frac{p-1}{2}} \equiv 1 \pmod p$，則 a 是模 p 的二次剩餘

範例 9.42

- 為了確認 14 或 16 在 $\mathbf{Z}_{23*}$ 是否為 QR，我們計算：

$14^{(23-1)/2} \bmod 23 \rightarrow 14^{11} \bmod 23 \rightarrow 22 \bmod 23 \rightarrow -1 \bmod 23$	**非二次剩餘**
$16^{(23-1)/2} \bmod 23 \rightarrow 16^{11} \bmod 23 \rightarrow 1 \bmod 23$	**二次剩餘**

解出模數為質數的二次方程式

- 特殊形式：$\boldsymbol{p = 4k + 3}$

$$x \equiv a^{(p+1)/4} \pmod{p} \qquad \text{和} \qquad x \equiv -a^{(p+1)/4} \pmod{p}$$

範例 9.43

- 求解下列各二次方程式：
  a. $x^2 \equiv 3 \pmod{23}$
  b. $x^2 \equiv 2 \pmod{11}$
  c. $x^2 \equiv 7 \pmod{19}$

範例 9.43 (續)

- 解法：
  a. 在第一個方程式，3 在 $\mathbf{Z}_{23}$ 中是 QR。
     解為 $x \equiv \pm 16 \pmod{23}$。
  b. 在第二個方程式，2 在 $\mathbf{Z}_{11}$ 中是 NQR。
     在 $\mathbf{Z}_{11}$ 中無解。
  c. 在第三個方程式，7 在 $\mathbf{Z}_{19}$ 中是 QR。
     解為 $x \equiv \pm 11 \pmod{19}$。

範例 9.44

- 求解下列各二次方程式：
  a. $x^2 \equiv 4 \pmod 7$
     $+2, -2$
  b. $x^2 \equiv 5 \pmod{11}$
     $+4, -4$
  c. $x^2 \equiv 7 \pmod{13}$
     No
  d. $x^2 \equiv 12 \pmod{17}$
     No

圖 9.5 模數為合成數之二次同餘方程式的解法

$$\begin{array}{l} x^2 \equiv a \bmod (n) \\ n = p_1 \times p_2 \times \ldots \times p_k \end{array} \longrightarrow \begin{array}{c} x^2 \equiv a_1 \pmod{p_1} \\ x^2 \equiv a_2 \pmod{p_2} \\ \cdots \\ x^2 \equiv a_k \pmod{p_k} \end{array} \longrightarrow \begin{array}{c} x_1 \equiv \pm\, b_1 \pmod{p_1} \\ x_2 \equiv \pm\, b_2 \pmod{p_2} \\ \cdots \\ x_k \equiv \pm\, b_k \pmod{p_k} \end{array}$$

範例 9.45

- Assume that $x^2 \equiv 36 \pmod{77}$

  ① $x^2 \equiv 36 \pmod{7} \equiv 1 \pmod{7}$
  $x^2 \equiv 36 \pmod{11} \equiv 3 \pmod{11}$

  ② $x \equiv \pm 1 \pmod{7}$
  $x \equiv \pm 5 \pmod{11}$
  Case 1: $x \equiv +1 \pmod{7}$; $x \equiv +5 \pmod{11}$
  Case 2: $x \equiv +1 \pmod{7}$; $x \equiv -5 \pmod{11}$
  Case 3: $x \equiv -1 \pmod{7}$; $x \equiv +5 \pmod{11}$
  Case 4: $x \equiv -1 \pmod{7}$; $x \equiv -5 \pmod{11}$

複雜度

- 求解一個模數為合成數之二次同餘式的難度有多高？
  - 其主要難度在於模數的因數分解。 換句話說，求解一個模數為合成數之二次同餘式的難度，等同於對一個合成數進行因數分解。如之前所述，當 n 很大時，因數分解是不可行的。

注意

求解一個模數為合成數的二次同餘式的難度，等同於對該模數進行因數分解。

9.6 指數運算與對數運算

- 本節討論主題
  - 指數運算
  - 對數運算

指數運算：$y = a^x$ → **對數運算**：$x = \log_a y$

快速指數運算

- 我們可以使用平方暨乘演算法 (Square-and-Multiply Method) 來進行快速的指數運算。

$$y = a^{x_{n_b-1}\times 2^{n_b-1} + x_{n_b-2}\times 2^{n_b-2} + \cdots + x_1\times 2^1 + x_0\times 2^0}$$

其中 x_i 等於 0 或 1

$$\downarrow$$

$$y = \boxed{\boxed{a^{2^{n_b-1}}}\ \text{或}\ \boxed{1}} \times \boxed{\boxed{a^{2^{n_b-2}}}\ \text{或}\ \boxed{1}} \times\cdots\times \boxed{\boxed{a^{2}}\ \text{或}\ \boxed{1}} \times \boxed{\boxed{a}\ \text{或}\ \boxed{1}}$$

範例：

$$\boxed{y = a^9 = a^{1001_2} = a^8 \times 1 \times 1 \times a}$$

演算法 9.7 平方暨乘演算法的虛擬碼

```
Square_and_Multiply (a, x, n)
{
    y ← 1
    for (i ← 0 to n_b − 1)                      //n_b是x位元數
    {
        if (x_i = 1)    y ← a × y  mod n        //只有當位元等於1時才與該項相乘

        a ← a^2 mod n                           //最後一回合不用平方
    }
    return y
}
```

範例 9.46

- 下圖顯示利用演算法 9.7 來計算 $y = a^x$ 的流程（為了簡化，圖中省略模數）。在這個範例中，$x = 22 = (10110)_2$，其指數有五個位元

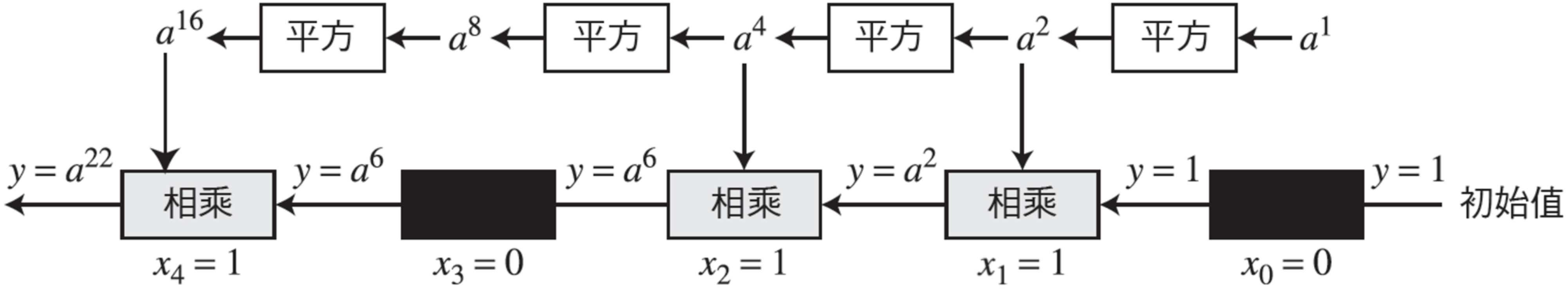


表 9.3 計算 $17^{22} \bmod 21$

i	x_i	乘法（初始值：$y = 1$）		平方（初始值：$a = 17$）
0	0		$\rightarrow$	$a = 17^2 \bmod 21 = 16$
1	1	$y = 1 \times 16 \bmod 21 = 16$	$\rightarrow$	$a = 16^2 \bmod 21 = 4$
2	1	$y = 16 \times 4 \bmod 21 = 1$	$\rightarrow$	$a = 4^2 \bmod 21 = 16$
3	0		$\rightarrow$	$a = 16^2 \bmod 21 = 4$
4	1	$y = 1 \times 4 \bmod 21 = 4$	$\rightarrow$	

計算 $21^{24} \bmod 8$

- The answer = 1

複雜度

注意

快速指數運算演算法的位元複雜度是以多項式成長。

演算法 9.8 模對數運算的暴力搜尋法

```
Modular_Logarithm (a, y, n)
{
    for (x = 1 to n −1)
    {
        if (y ≡ a^x mod n) return x
    }
    return failure
}
```