中山大學
SUN YAT-SEN UNIVERSITY

Module I. Fundamentals of Information Security

Chapter 2

Cryptographic Techniques

Web Security: Theory & Applications

School of Data & Computer Science, Sun Yat-sen University

Outline

- **2.1 Cryptology Introduction**
  - Introduction
  - History
  - Concepts & Items
- **2.2 Symmetric Key Cryptographic Algorithms**
  - Introduction
  - Types & Modes
  - Data Encryption Standard (DES)
  - Advanced Encryption Standard (AES)

Outline

- **2.3 Mathematical Foundations of Public-Key Cryptography**
  - Prime factorizations of integers
  - The *Euclidean* Algorithm
  - *Bézout*‘s Theorem
  - Linear Congruence
  - The Extended_*Euclidean* Algorithm
  - The Chinese Remainder Theorem
  - *Euler*’s φ function
  - *Euler*’s Theorem
  - *Fermat*’s Little Theorem

Outline

- **2.4 Asymmetric Key Cryptographic Algorithms**
  - Introduction
  - The RSA Algorithm
  - Digital Signatures
- **2.5 Hashing Algorithms**
  - Introduction
  - Message-Digest Algorithm (MD5)
- **2.6 Typical Applications**
  - MD5 and Passwords
  - AES and WiFi Protected Access
  - RSA and e-Business

2.3 Mathematical Foundations

2.3.1 Prime factorizations of integers

- *Fundamental Theorem of Arithmetic* (算术基本定理)
  - Every positive integer greater than 1 can be written uniquely as a prime or as the product of two or more primes when the prime factors are written in order of non-decreasing size. (*Euclid*)
- *Greatest Common Divisor* (最大公因数)
  - Let a and b be integers, not both zero. The largest integer d such that $d|a$ and $d|b$ is called the *greatest common divisor* (GCD) of a and b, often denoted as gcd(a, b)

2.3 Mathematical Foundations

2.3.2 The *Euclidean* Algorithm (欧几里德辗转相除法)

—《几何原本.第VII卷》(公元前约300年)

- *Lemma* 0.
  - Let $a = bq + r$, where a, b, q, and r are integers. Then
    $$\gcd(a, b) = \gcd(b, r)$$
  - *Proof*.
    - ✧ Suppose d divides both a and b. Recall that if $d|a$ and $d|b$, then $d|a\text{-}bk$ for any integer k. It follows that d also divides $a\text{-}bq = r$. Hence, any common division of a and b is also a common division of b and r.
    - ✧ Suppose that d' divides both b and r, then d' also divides $bq+r = a$. Hence, any common divisor of b and r is also common divisor of a and b.
    - ✧ Consequently, $\gcd(a, b) = \gcd(b, r)$.
    - ✧ Note: $a = bq + r$, $0 \le r < b$, aka $r = a \bmod b$ if the quotient q ignored. r is the (*least positive*) remainder of the division.

2.3 Mathematical Foundations

2.3.2 The *Euclidean* Algorithm

- *Remark.*
  - ✧ Suppose a and b are positive integers, $a \geq b$. Let $r_0 = a$ and $r_1 = b$, we successively apply the division algorithm and the gcd is the last nonzero remainder

$$
\begin{aligned}
r_0 &= r_1 q_1 + r_2, && 0 < r_2 < r_1 \\
r_1 &= r_2 q_2 + r_3, && 0 < r_3 < r_2 \\
&\ \ \vdots \\
r_{n-2} &= r_{n-1} q_{n-1} + r_n, && 0 < r_n < r_{n-1} \\
r_{n-1} &= r_n q_n
\end{aligned}
$$

$$
\begin{aligned}
\gcd(a, b) &= \gcd(r_0, r_1) \\
&= \gcd(r_1, r_2) \\
&= \ldots \\
&= \gcd(r_{n-2}, r_{n-1}) \\
&= \gcd(r_{n-1}, r_n) \\
&= \gcd(r_n, 0) = r_n.
\end{aligned}
$$

2.3 Mathematical Foundations

2.3.2 The *Euclidean* Algorithm

– *Remark.*

✧ Suppose a and b are positive integers, $a \geq b$. Let $r_0=a$ and $r_1=b$, we successively apply the division algorithm and the gcd is the last nonzero remainder

$$
\begin{aligned}
r_0 &= r_1 q_1 + r_2, && 0<r_2<r_1 \\
r_1 &= r_2 q_2 + r_3, && 0<r_3<r_2 \\
&\ \ \dots \\
r_{n-2} &= r_{n-1} q_{n-1} + r_n, && 0<r_n<r_{n-1} \\
r_{n-1} &= r_n q_n + 0. \\
\gcd(a, b) &= \gcd(r_0, r_1) \\
&= \gcd(r_1, r_2) \\
&= \dots \\
&= \gcd(r_{n-2}, r_{n-1}) \\
&= \gcd(r_{n-1}, r_n) \\
&= \gcd(r_n, 0) = r_n.
\end{aligned}
$$

The last nonzero remainder

2.3 Mathematical Foundations

2.3.2 The *Euclidean* Algorithm

– *Example*.

- ✧ Find the GCD of 662 and 414
- ✧ Compute as

$$662 = 414\cdot 1+248$$
$$414 = 248\cdot 1+166$$
$$248 = 166\cdot 1+82$$
$$166 = 82\cdot 2 + 2$$
$$82 = 2\cdot 41$$

2.3 Mathematical Foundations

2.3.2 The *Euclidean* Algorithm

– *Example*.

✧ Find the GCD of 414 and 662

✧ Compute as

$$662 = 414 \cdot 1 + 248$$

$$414 = 248 \cdot 1 + 166$$

$$248 = 166 \cdot 1 + 82$$

$$166 = 82 \cdot 2 + 2$$

$$82 = 2 \cdot 41$$

✧ So, gcd(414,662) = 2

2.3.2 The *Euclidean* Algorithm

- *The Euclidean Algorithm*

```
procedure gcd(a, b: positive integers)
begin
        x := a;
        y :=b;
   while (y≠0)
   begin
       r :=x mod y;
       x :=y;
       y :=r;
    end; {gcd(a, b)=x}
end;
```

✧ The time complexity (for **mod** operation) is $O(\log b)$ (where $a \geq b$)

2.3 Mathematical Foundations

2.3.2 The *Euclidean* Algorithm

- *The Euclidean Algorithm*
  - Another form of *Euclidean* Algorithm

```
function Euclid(a, b: positive integers): positive integer
begin
        if b=0 then return (a)
           else  return (Euclid(b, a mod b));
end;
```

✧ Think about it.

2.3 Mathematical Foundations

2.3.3 *Bézout*'s Theorem (1779, 贝祖定理)

- *Theorem* 1.
  - If a and b are *positive* integers, then there exits integers s and t such that gcd(a, b) = $sa + tb$.
  - *Remark*.
    - ✧ a and b are positive. s and t can be any integers.
    - ✧ The equation gcd$(a, b) = sa + tb$ is called *Bézout's identity* (贝祖恒等式). The integers s and t are called *Bézout coefficients* of a and b (贝祖系数).
    - ✧ Proof omitted.
  - *Example.*
    - ✧ gcd(252, 198) = 18.
    - ✧ By working backward through the divisions of *The Euclidean Algorithm*, we get $s = 4$, $t = -5$ such that

$$18 = 4 \cdot 252 + (-5) \cdot 198$$

    - ✧ Ref. to Section 2.3.5: *The Extended_Euclidean Algorithm*

2.3 Mathematical Foundations

2.3.3 *Bézout*'s Theorem (1779, 贝祖定理)

- *Lemma* 1.
  - If a, b and c are positive integers such that gcd(a, b) = 1 and $a|bc$ then $a|c$.
  - *Proof.*
    - ✧ By *Theorem*.1, there exits integers s and t such that $sa + tb = 1$, or

      $sac + tbc = c$.
    - ✧ Since $a|sac$ and $a|tbc$.
    - ✧ Therefore $a|c$.

2.3 Mathematical Foundations

2.3.3 *Bézout*'s Theorem (1779, 贝祖定理)

- *Lemma* 1.
  - If a, b and c are positive integers such that gcd(a, b) = 1 and $a|bc$ then $a|c$.
- *Lemma* 2.
  - If p is a prime and $p|a_1a_2 \ldots a_n$, where each a_i is an integer, then $p|a_i$ for some i.
  - *Proof.*
    - ✧ If gcd(p, a_1) = 1, by *Lemma*.1 it should be $p|a_2 \ldots a_n$.
      and if gcd(p, a_2) = 1, by *Lemma*.1 it should be $p|a_3 \ldots a_n$.
      … …
      until an i ($i \leq n$) found such that gcd(p, a_i) $\neq$ 1.
    - ✧ In this case $p|a_i$ for p is a prime.
    - ✧ The existence of such an i is assured or gcd(p, $a_1a_2 \ldots a_n$) = 1, a contradiction.

2.3 Mathematical Foundations

2.3.3 *Bézout*'s Theorem (1779, 贝祖定理)

- *Lemma* 1.
  - If a, b and c are positive integers such that gcd(a, b) = 1 and $a|bc$ then $a|c$.
- *Lemma* 2.
  - If p is a prime and $p|a_1a_2 \ldots a_n$, where each a_i is an integer, then $p|a_i$ for some i.
- *Lemma* 3.
  - The uniqueness of the prime factorization of a positive integer.
  - *Proof.*

2.3 Mathematical Foundations

2.3.3 *Bézout*'s Theorem (1779, 贝祖定理)

- *Lemma* 1.
  - If a, b and c are positive integers such that gcd(a, b) = 1 and $a|bc$ then $a|c$.
- *Theorem* 2.
  - Let m be a positive integer, and a, b and c be integers, $c \neq 0$. If $ac \equiv bc \pmod{m}$ and gcd(c, m) = 1, then $a \equiv b \pmod{m}$.
  - *Proof.*
    - ✧ $ac \equiv bc \pmod{m}$ means $m|(ac - bc)$, or

      $m|(a - b)c$.
    - ✧ Now gcd(c, m) = 1. By *Lemma*.1, we have $m|(a - b)$, or

      $a \equiv b \pmod{m}$.

2.3 Mathematical Foundations

2.3.4 Linear Congruence (线性同余式/线性同余方程/模线性方程)

- *Definition* 1.
  - A congruence of the form

    $$ax \equiv b \pmod{m}.$$

    where m is a positive integer, a and b are integers, and x is a variable, is called a linear congruence.
  - *Remark.*
    - ✧ How to find all integers x that satisfy the congruence

      $$ax \equiv b \pmod{m}?$$

- *Definition* 2. (模 m 逆元)
  - If there is an integer y such that the linear congruence

    $$ya \equiv 1 \pmod{m}.$$

    holds, then y is said to be an *inverse* of a modulo m.
    - ✧ $ya \equiv 1 \pmod{m}$ means $(ya - 1) = km$ for some integer k.

2.3 Mathematical Foundations

2.3.4 Linear Congruence (线性同余式/线性同余方程/模线性方程)

- *Theorem* 3.
  - If a and m are relatively prime integers and m>1, then an inverse of a modulo m exits. Furthermore, this inverse is unique modulo m. (模 m 逆元存在定理)
  - *Remark.*
    - ✧ If the condition gcd(a, m) = 1, m>1 holds
    - ✧ Then there is a unique positive integer, less than m, denoted by a^{-1}, that is an inverse of a modulo m, and any other inverse of a modulo m is congruence to a^{-1} modulo m.

2.3 Mathematical Foundations

2.3.4 Linear Congruence (线性同余式/线性同余方程/模线性方程)

- *Example.*
  - Find an inverse of 3 modulo 7.
- *Solution.*
  - gcd(3, 7) = 1. Then by *Theorem.*3, the inverse of 3 modulo 7 exits.
  - We use the *Euclidean Algorithm* (or *Extended_Euclidean Algorithm*) to find gcd(3, 7). It ends at $7 = 2\cdot3+1$.
  - As the example following *Theorem.*1, by working backward through the divisions of the *Euclidean Algorithm*, we get $-2\cdot3+1\cdot7 = 1$.
    - Now gcd(a, m) = 1.
    - By *Theorem.*1, there exists integer s and t such that

      $$sa + tm = 1, \text{ or } sa - 1 = -tm.$$

    - By *Definition.*2, s is an inverse of a modulo m.
  - So -2 is an inverse of 3 modulo 7.
  - Every integers congruent to -2 modulo 7 is an inverse of 3 modulo 7, such as 5, -9, 12, and so on.

2.3 Mathematical Foundations

2.3.4 Linear Congruence (线性同余式/线性同余方程/模线性方程)

- *Example.*
  - ✧ Find the solutions of $3x \equiv 4 \pmod 7$.
- *Solution.*
  - ✧ We already know -2 is an inverse of 3 modulo 7. Multiplying both sides of the congruence by -2 show that

    $-2 \cdot 3x \equiv -2 \cdot 4 \pmod 7$, or

    $-2 \cdot 3x \equiv -8 \pmod 7$
  - ✧ We know $-2 \cdot 3 \equiv 1 \pmod 7$
  - ✧ So $-2 \cdot 3x \pmod 7 \equiv [-2 \cdot 3 \pmod 7] \cdot [x \pmod 7] \pmod 7$, or

    $-2 \cdot 3x \pmod 7 \equiv x \pmod 7$
  - ✧ Therefore

    $x \equiv -8 \pmod 7 \equiv 6 \pmod 7$
  - ✧ The solution are all the x such that $x \equiv 6 \pmod 7$. That is, 6, 13, 20, . . ., and -1, -8, -15, . . .

2.3.4 Linear Congruence (线性同余式/线性同余方程/模线性方程)

– *Remark.*

✧ How to working backward through the divisions of the *Euclidean* Algorithm? By the Algorithm we have the sequence of

$$
\begin{aligned}
a_0 &= b_0 q_1 + b_1 \\
b_0 &= b_1 q_2 + b_2 \\
b_1 &= b_2 q_3 + b_3 \\
&\;\;\ldots \\
b_{k-1} &= b_k q_{k+1} + b_{k+1} \\
b_k &= b_{k+1} q_{k+2} + \gcd(a_0, b_0)
\end{aligned}
$$

$\gcd(a_0, b_0)$ is the last non-zero remainder

✧ Then $\gcd(a_0, b_0) = f(b_k, b_{k+1}, q_{k+2}) = f^{(1)}(b_{k-1}, b_k, q_{k+1}, q_{k+2})$

$= f^{(2)}(b_{k-2}, b_{k-1}, q_k, q_{k+1}, q_{k+2}) = \ldots$

$= f^{(k+1)}(a_0, b_0, q_1, q_2, q_3, \ldots, q_{k+2})$

2.3 Mathematical Foundations

2.3.4 Linear Congruence (线性同余式/线性同余方程/模线性方程)

- *Example.*
  - ✧ To find gcd(287, 91), by *Euclidean Algorithm* we have the sequence of

    $287 = 91 \cdot 3 + 14$

    $91 = 14 \cdot 6 + 7$

    $14 = 7 \cdot 2 + 0$

  - ✧ Or let

    $a_0 = 287,\ b_0 = 91,$

    $a_0 = b_0 q_1 + b_1$

    $b_0 = b_1 q_2 + b_2$

    $b_1 = b_2 q_3$, here $\gcd(a_0, b_0) = b_2$ (= 7)

  - ✧ Then

    $$\begin{aligned}\gcd(a_0, b_0) = b_2 = b_0 - b_1 q_2 = b_0 - (a_0 - b_0 q_1) q_2 \\ = -a_0 q_2 + b_0 (1 + q_1 q_2) \\ = -6 a_0 + 19 b_0\end{aligned}$$

2.3 Mathematical Foundations

2.3.4 Linear Congruence (线性同余式/线性同余方程/模线性方程)

– *Example.*
  - ✧ Find an inverse of 101 modulo 4620.

– *Solution.*
  - ✧ To find gcd(101, 4620), by *Euclidean Algorithm* we have the sequence of

$$
\begin{aligned}
4620 &= 45\cdot 101 + 75 \\
101 &= 1\cdot 75 + 26 \\
75 &= 2\cdot 26 + 23 \\
26 &= 1\cdot 23 + 3 \\
23 &= 7\cdot 3 + 2 \\
3 &= 1\cdot 2 + 1 \\
2 &= 2\cdot 1.
\end{aligned}
$$

remainders

gcd(101, 4620)

quotients

2.3 Mathematical Foundations

2.3.4 Linear Congruence (线性同余式/线性同余方程/模线性方程)

– *Example.*

✧ Find an inverse of 101 modulo 4620.

– *Solution.*

✧ Now gcd(101, 4620) = 1. We can find the *Bézout coefficients* for 101 and 4620 by working backwards through these steps, expressing gcd(101, 4620) = 1 in terms of each successive pair of remainders.

$$
\begin{aligned}
1 &= 3 - 1\cdot 2 = 3 - 1\cdot(23 - 7\cdot 3) = -1\cdot 23 + 8\cdot 3 \\
&= -1\cdot 23 + 8\cdot(26 - 1\cdot 23) = 8\cdot 26 - 9\cdot 23 \\
&= 8\cdot 26 - 9\cdot(75 - 2\cdot 26) = -9\cdot 75 + 26\cdot 26 \\
&= -9\cdot 75 + 26\cdot(101 - 1\cdot 75) = 26\cdot 101 - 35\cdot 75 \\
&= 26\cdot 101 - 35\cdot(4620 - 45\cdot 101) \\
&= -35\cdot 4620 + 1601\cdot 101.
\end{aligned}
$$

✧ Now -35 and 1601 are *Bézout coefficients* of 4620 and 101, and 1601 is an inverse of 101 modulo 4620.

2.3 Mathematical Foundations

2.3.5 The Extended_*Euclidean* Algorithm (扩展欧几里德算法)

– *Remark.*

- ✧ Let $ax + by = \gcd(a, b)$, $a \geq b > 0$. (*Theorem.*1)
- ✧ How to find x, y, and $\gcd(a, b)$? (*Diophantus* equation)
- ✧ Let $a' = b$, $b' = a \bmod b$. By *Bézout's Theorem* we have

  $\gcd(a', b') = a'x' + b'y'$ or

  $\gcd(b, a \bmod b) = bx' + (a \bmod b)y'$.
- ✧ By *Lemma* 0, we know that

  $\gcd(a, b) = \gcd(b, a \bmod b) = \gcd(a', b')$.
- ✧ Then $\gcd(a, b) = \gcd(a', b')$

$$
\begin{aligned}
&= a'x' + b'y' \\
&= bx' + (a \bmod b)y' \\
&= bx' + (a - (a \text{ div } b)b)y' \\
&= ay' + b(x' - (a \text{ div } b)y')
\end{aligned}
$$

So $x = y'$, and $y = x' - (a \text{ div } b)y'$ is a solution to the equation

$ax + by = \gcd(a', b') = \gcd(a, b)$.

2.3 Mathematical Foundations

2.3.5 The Extended_*Euclidean* Algorithm (扩展欧几里德算法)

– *Remark.*

- Let $a'' = b'$, $b'' = a' \bmod b'$ we also have
  $$\gcd(a'', b'') = a'y'' + b'(x'' - (a' \text{ div } b')y'').$$
  So $x' = y''$, and $y' = x'' - (a' \text{ div } b')y''$ is a solution to the equation
  $$a'x' + b'y' = \gcd(a'', b'') = \gcd(a', b') = \gcd(a, b).$$
- Let $a^{(3)} = b''$, $b^{(3)} = a'' \bmod b''$ we also have
  $$\gcd(a^{(3)}, b^{(3)}) = a''y^{(3)} + b''(x^{(3)} - (a'' \text{ div } b'')y^{(3)}).$$
  So $x'' = y^{(3)}$, and $y'' = x^{(3)} - (a'' \text{ div } b'')y^{(3)}$.

  . . . . . .
- Let $a^{(k+1)} = b^{(k)}$, $b^{(k+1)} = a^{(k)} \bmod b^{(k)}$ we have
  $$\gcd(a^{(k+1)}, b^{(k+1)}) = a^{(k)}y^{(k+1)} + b^{(k)}(x^{(k+1)} - (a^{(k)} \text{ div } b^{(k)})y^{(k+1)}).$$
  So $x^{(k)} = y^{(k+1)}$, and $y^{(k)} = x^{(k+1)} - (a^{(k)} \text{ div } b^{(k)})y^{(k+1)}$.
- Continue this process until $b^{(k+1)} = a^{(k)} \bmod b^{(k)} = 0$ obtained.
- Then $\gcd(a, b) = \gcd(a', b') = \gcd(a'', b'') = \ldots = \gcd(a^{(k+1)}, b^{(k+1)})$
  $= \gcd(a^{(k+1)}, 0)$
  $= a^{(k+1)}$ $(=b^{(k)})$.

2.3 Mathematical Foundations

2.3.5 The Extended_*Euclidean* Algorithm (扩展欧几里德算法)

- *Remark.*
  - ✧ Since we have

    $$\gcd(a, b) = a^{(k+1)}.$$

  - ✧ Then The equation

    $$a^{(k+1)}x^{(k+1)} + b^{(k+1)}y^{(k+1)} = \gcd(a, b).$$

    has a solution $x^{(k+1)} = 1$, $y^{(k+1)} = 0$.
    - in fact, $y^{(k+1)}$ can take any positive integer because $b^{(k+1)}=0$.

2.3 Mathematical Foundations

2.3.5 The Extended_*Euclidean* Algorithm (扩展欧几里德算法)

- *Remark.*
  - Since we have

    $$\gcd(a, b) = a^{(k+1)}.$$

  - Then The equation

    $$a^{(k+1)}x^{(k+1)} + b^{(k+1)}y^{(k+1)} = \gcd(a, b).$$

    has a solution $x^{(k+1)} = 1$, $y^{(k+1)} = 0$.
    - in fact, $y^{(k+1)}$ can take any positive integer because $b^{(k+1)}=0$.
  - If we have put every $a^{(i)}$ and $b^{(i)}$ in the process on record, by working backward,

    $$x^{(k)} = y^{(k+1)}, \text{ and } y^{(k)} = x^{(k+1)} - (a^{(k)} \text{ div } b^{(k)})y^{(k+1)}.$$

    we can finally find x and y.

2.3 Mathematical Foundations

2.3.5 The Extended_*Euclidean* Algorithm (扩展欧几里德算法)

- *The Extended_Euclidean Algorithm.*

```
ADT triple {
    x, y, d: longint;
} ee;
triple function Extended_Euclid (a, b: positive integers)
begin
    if b=0 then return(1, 0, a);
    ee :=Extended_Euclid (b, a mod b);
    x :=ee.y;
    y :=ee.x - (a div b)*ee.y;
    return (x, y, ee.d);
end;
```

2.3.5 The Extended_*Euclidean* Algorithm (扩展欧几里德算法)

- *Example.*
  - ✧ Find the GCD of 662 and 414
- *Solution.*
  - ✧ Construct a forward procedure

    $a^{(k+1)} = b^{(k)}$,

    $b^{(k+1)} = a^{(k)} \bmod b^{(k)}$.

    until k=5, $b^{(5)} = 0$.
  - ✧ We get: gcd(a, b) = $a^{(5)} = b^{(4)}$.

k	*a*	*b*
0	662	414
1	414	248
2	248	166
3	166	82
4	82	2
5	2	0

2.3.5 The Extended_*Euclidean* Algorithm (扩展欧几里德算法)

- *Example.*
  - ✧ Find the GCD of 662 and 414
- *Solution.*
  - ✧ Construct a forward procedure

    $a^{(k+1)} = b^{(k)}$,

    $b^{(k+1)} = a^{(k)} \bmod b^{(k)}$.

    until k=5, $b^{(5)} = 0$.
  - ✧ We get: gcd(a, b) = $a^{(5)} = b^{(4)}$.
  - ✧ Take $x^{(5)}$=1, $y^{(5)}$=0.

k	*a*	*b*	*x*	*y*
0	662	414		
1	414	248		
2	248	166		
3	166	82		
4	82	2		
5	2	0	1	0

2.3 Mathematical Foundations

2.3.5 The Extended_*Euclidean* Algorithm (扩展欧几里德算法)

- *Example.*
  - ✧ Find the GCD of 662 and 414
- *Solution.*
  - ✧ Construct a forward procedure

    $a^{(k+1)} = b^{(k)}$,

    $b^{(k+1)} = a^{(k)} \bmod b^{(k)}$.

    until k=5, $b^{(5)} = 0$.
  - ✧ We get: $\gcd(a, b) = a^{(5)} = b^{(4)}$.
  - ✧ Take $x^{(5)}=1$, $y^{(5)}=0$.
  - ✧ Construct a backward process

    $x^{(k)} = y^{(k+1)}$,

    $y^{(k)} = x^{(k+1)} - (a^{(k)} \operatorname{div} b^{(k)})y^{(k+1)}$.
  - ✧ Now the *Diophantus* equation

    $$662x + 414y = \gcd(662, 414)$$

    has a solution of $x = -5$, $y = 8$.

k	*a*	*b*	*x*	*y*
0	662	414	-5	8
1	414	248	3	-5
2	248	166	-2	3
3	166	82	1	-2
4	82	2	0	1
5	2	0	1	0

2.3 Mathematical Foundations

2.3.6 The Chinese Remainder Theorem (中国剩余定理)

- History of the *CRT.*
  - In 4^{ST} century, the Chinese mathematician Sun-Tsu ask: There are certain things whose number is unknown. When divided by 3 , the remainder is 2; when divided by 5, the remainder is 3; when divided by 7, the remainder is 2. What will be the number of things? (有物不知其数, 三分之余二, 五分之余三, 七分之余二, 此物几何? -《孙子算经》魏晋南北朝)
    - ✧ The notion of congruences was first introduced and used by *Gauss* in his *Disquisitiones Arithmeticae* (算术探究) of 1801. This puzzle can be: What are the solutions of the systems of congruences

$$\begin{cases} x \equiv 2 \pmod 3 \\ x \equiv 3 \pmod 5 \\ x \equiv 2 \pmod 7 \end{cases}$$

2.3 Mathematical Foundations

2.3.6 The Chinese Remainder Theorem (中国剩余定理)

- History of the *CRT.*
  - “大衍求一术”(《数书九章》，秦九韶，南宋，1247)
    - ✧ 求解一次同余式

2.3 Mathematical Foundations

2.3.6 The Chinese Remainder Theorem (中国剩余定理)

- *Theorem* 4.
  - Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers and $a_1, a_2, \ldots, a_n$ arbitrary integers. Then the system

$$
\begin{aligned}
x &\equiv a_1 \pmod{m_1}\\
x &\equiv a_2 \pmod{m_2}\\
&\ldots\\
x &\equiv a_n \pmod{m_n}
\end{aligned}
$$

    has a unique solution modulo m, $m = m_1 m_2 \ldots m_n$.
  - That is, there is a solution x with $0 \leq x \leq m$ to the system, and all other solutions to the system are congruent modulo m to this solution.

2.3 Mathematical Foundations

2.3.6 The Chinese Remainder Theorem (中国剩余定理)

- *Theorem 4.*
  - *Proof.*
    - ✧ Let $M_k = m/m_k$ for k=1, 2, . . ., n. That is, M_k is the product of the moduli except for m_k, and M_s mod m_k = 0 when $s \neq k$.
    - ✧ We know gcd(M_k, m_k) = 1 for k=1, 2, . . ., n because $m_1, m_2, \ldots, m_n$ are pairwaise relatively prime integers. From *Theorem.*3, there is an integer y_k, an inverse of M_k modulo m_k, such that
      $$M_k y_k \equiv 1 \pmod{m_k},\ k = 1, 2, \ldots, n.$$
    - ✧ Now form the sum
      $$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \ldots + a_n M_n y_n.$$
    - ✧ Then we know x is a simultaneous solution by showing
      $$x \bmod m_k = a_k M_k y_k \bmod m_k = a_k \bmod m_k,\ k = 1, 2, \ldots, n.$$
      or $$x \equiv a_k \pmod{m_k},\ k = 1, 2, \ldots, n.$$

2.3.6 The Chinese Remainder Theorem (中国剩余定理)

- *Example.*
  - ✧ Find the solutions of the systems of congruences

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

- *Solution.*
  - ✧ Let $m = 3 \cdot 5 \cdot 7 = 105$, then

$M_1 = 5 \cdot 7 = 35$, $y_1 = 2$ (an inverse of M_1 modulo 3)

$M_2 = 3 \cdot 7 = 21$, $y_2 = 1$ (an inverse of M_2 modulo 5)

$M_3 = 3 \cdot 5 = 15$, $y_3 = 1$ (an inverse of M_3 modulo 7)

  - ✧ A solution is

$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 = 233 \equiv 23 \pmod{105}$$

2.3 Mathematical Foundations

2.3.6 The Chinese Remainder Theorem (中国剩余定理)

– *Remark.*

- Let $m_1, m_2, \ldots, m_n$ ($m_i \geq 2$, i=1, 2, . . ., n) be pairwaise relatively prime integers and $m = m_1 m_2 \ldots m_n$. By *The Chinese Remainder Theorem*, any integer x with $0 \leq x \leq m$ can be uniquely represented by the n-tuple

$$(a_1, a_2, \ldots, a_n),\ a_i = x \bmod m_i,\ i=1, 2, \ldots, n.$$

- Keeping ($m_1, m_2, \ldots, m_n$) in secret, it is very difficult to decrypt x from ($a_1, a_2, \ldots, a_n$).
  - As we know, $x = a_1 M_1 y_1 + a_2 M_2 y_2 + \ldots + a_n M_n y_n$.
- As in Sun-Tsu's example, (m_1, m_2, m_3)=(3, 5, 7) is the secret key. The number x=23 is represented by (a_1, a_2, a_3)=(2, 3, 2):

$$a_1 = x \bmod m_1 = 23 \bmod 3 = 2$$
$$a_2 = x \bmod m_2 = 23 \bmod 5 = 3$$
$$a_3 = x \bmod m_3 = 23 \bmod 7 = 2$$

2.3 Mathematical Foundations

2.3.7 *Euler's* φ function (*Euler*'s Totient function, 欧拉 φ 函数)

- *Definition.*
  - For an integer m, consider the ring $Z_m = \{0, . . . , m-1\}$. *Euler*'s φ function $\varphi(m)$ is the number of integers in Z_m which are coprime to m.
    - ✧ Denote the collection of all the integers coprime to m in Z_m as Z_m', $\varphi(m) = |Z_m'|$. (Z_m', reduced residue system of m, 既约剩余系)
    - ✧ $\varphi(m)$ is the number of positive integers less than and prime to m.
  - *Example.*
    - ✧ $\varphi(8) = 4$.
      - ◌ 1, 3, 5, 7 are coprime to 8. $Z_8'=\{1, 3, 5, 7\}$
    - ✧ Convention: $\varphi(1) = 1$.

2.3 Mathematical Foundations

2.3.7 *Euler's φ* function (*Euler*'s Totient function, 欧拉 φ 函数)

– *Example.*

- Let m = p^k, p is prime. Then $\varphi(m) = \varphi(p^k) = p^k - p^{k-1}$.
- *Proof.*
  - An integer n is coprime to m = p^k if and only if it contains no p as its factor. Integers in Z_m containing p as factor are

    1p, 2p, 3p, ..., $p^{(k-1)}$p，
  - Remove them from Z_m , m - p^{k-1} = p^k - p^{k-1} number of integers are left witch are coprime to m.
- *Example.*
  - $\varphi(8) = \varphi(2^3) = 2^3 - 2^2 = 4$.
  - When k=1, The equation becomes $\varphi(p) = p - 1$.
- The equation can be the form of

$$\phi(p^k) = p^k - p^{k-1} = p^k(1 - \frac{1}{p})$$

2.3 Mathematical Foundations

2.3.7 *Euler's* φ function (*Euler*'s Totient function, 欧拉 φ 函数)

- *Example.*
  - $\varphi(p) = p-1$ if p is prime. ($p \neq 1$ because 1 is not prime)
    - For p is coprime to any integer less then p. $Z_p' = \{1, 2, ..., p-1\}$
    - *Example.*
      - $\varphi(11) = 10$.
- *Example.*
  - Let m = pq, p and q are relatively prime. Then

    $$\varphi(m) = \varphi(pq) = \varphi(p)\varphi(q).$$
  - *Proof.*
    - Let $a \in Z_p'$, $b \in Z_q'$. Applying the *Chinese Remainder Theorem*, any $c \in Z_{pq}'$ can be uniquely represent as a ordered pair (a, b). The number of c, say $|Z_{pq}'|$, is $|Z_p'| \times |Z_p'|$.
  - *Example.*
    - $\varphi(56) = \varphi(8\times7) = \varphi(8)\times\varphi(7) = 4\times6 = 24$

2.3 Mathematical Foundations

2.3.7 *Euler's* φ function (*Euler*'s Totient function, 欧拉 φ 函数)

- *Example.*
  - Let $m = pq$, p and q are primes, $p \neq q$. Then

$$\varphi(m) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1).$$

- *Example.*
  - Let $m = p_1^{k_1} p_2^{k_2} \ldots p_r^{k_r}$ where p_i are primes and $k_i>0$ for $i=1..r$, $p_s \neq p_t$ for $1 \leq s<t \leq r$,. Then

$$\begin{aligned}\varphi(m) &= \varphi(p_1^{k_1})\varphi(p_2^{k_2})\ldots\varphi(p_r^{k_r}) \\ &= p_1^{k_1}[1-(1/ p_1)]\ p_2^{k_2}[1-(1/ p_2)]\ \ldots\ p_r^{k_r}[1-(1/ p_r)] \\ &= m\ [1-(1/ p_1)]\ [1-(1/ p_2)]\ \ldots\ [1-(1/ p_r)]\end{aligned}$$

  - *Example.*
    - $\varphi(1323) = \varphi(3^3 \times 7^2) = 1323 \times (1-1/3) \times (1-1/7) = 756$

2.3 Mathematical Foundations

2.3.8 *Euler's* Theorem (欧拉定理)

- *Theorem* 5.
  - Let a and m be integers such that gcd(a, m) = 1. Then

$$a^{\varphi(m)} \equiv 1 \pmod{m}.$$

    - ✧ $m > 0$.
  - *Remark.*
    - ✧ The existence of *inverse* of a modulo m
      - ◌ As defined in *Definition*.2, if there is an integer y such that $ya \equiv 1 \pmod{m}$, y is said to be an *inverse* of a modulo m.
      - ◌ Now $a^{\varphi(m)} = a \times a^{\varphi(m)-1} \equiv 1 \pmod{m}$. Thus $a^{\varphi(m)-1}$ is an inverse of a modulo m
    - ✧ *Euler's* Theorem is a generalization (arbitrary modulus) of *Fermat's* Little Theorem.

2.3 Mathematical Foundations

2.3.8 *Euler's* Theorem (欧拉定理)

- *Theorem* 5.
  - Let a and m be integers such that gcd(a, m) = 1. Then

$$a^{\varphi(m)} \equiv 1 \pmod{m}.$$

  - Proof.

(1) Let $Z_m' = \{x_1, x_2, ..., x_{\varphi(m)}\}$ be the reduced residue system of m, and let $S = \{ax_1 \bmod m, ax_2 \bmod m, ..., ax_{\varphi(m)} \bmod m\}$, then

$$Z_m' = S.$$

- Because a and x_i ($1 \le i \le \varphi(m)$) are all coprime to m, so ax_i are also coprime to m for $1 \le i \le \varphi(m)$. Therefore

$$ax_i \bmod m \in Z_m'.$$

- Now a is coprime to m. For any $x_i \neq x_j$, by Cancellation Law we get

$$ax_i \bmod m \neq ax_j \bmod m.$$

2.3 Mathematical Foundations

2.3.8 *Euler's* Theorem (欧拉定理)

- *Theorem* 5.
  - Let a and m be integers such that gcd(a, m) = 1. Then

$$a^{\varphi(m)} \equiv 1 \pmod{m}.$$

  - Proof.

(2) Construct

$$\begin{aligned} & a^{\varphi(m)} x_1 x_2 \ldots x_{\varphi(m)} \pmod{m} \\ \equiv\ & (ax_1)(ax_2) \ldots (ax_{\varphi(m)}) \pmod{m} \\ \equiv\ & (ax_1 \bmod m)(ax_2 \bmod m) \ldots (ax_{\varphi(m)} \bmod m) \pmod{m} \\ \equiv\ & x_1 x_2 \ldots x_{\varphi(m)} \pmod{m}. \end{aligned}$$

- But x_i ($1 \le i \le \varphi(m)$) are coprime to m, and so is $x_1 x_2 \ldots x_{\varphi(m)}$.
- Therefore, by Cancellation Law,

$$a^{\varphi(m)} \equiv 1 \bmod m.$$

✧ Cancellation Law:

If gcd(c,p) = 1, then ac ≡ bc (mod p) ⇒ a ≡ b (mod p)

2.3 Mathematical Foundations

2.3.9 *Fermat's* Little Theorem (1640, 费马小定理)

- *Theorem 6.*
  - If p is a prime number and a is an integer not divisible by p, then

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

  - Further more, for every integer a, *Fermat's Little Theorem* is equivalent to

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

  - *Example.*
    - ✧ $a = 13$, $p = 7$, $a^p = 13^7 = 62748517$, $a^{p-1} = 13^6 = 4826809$

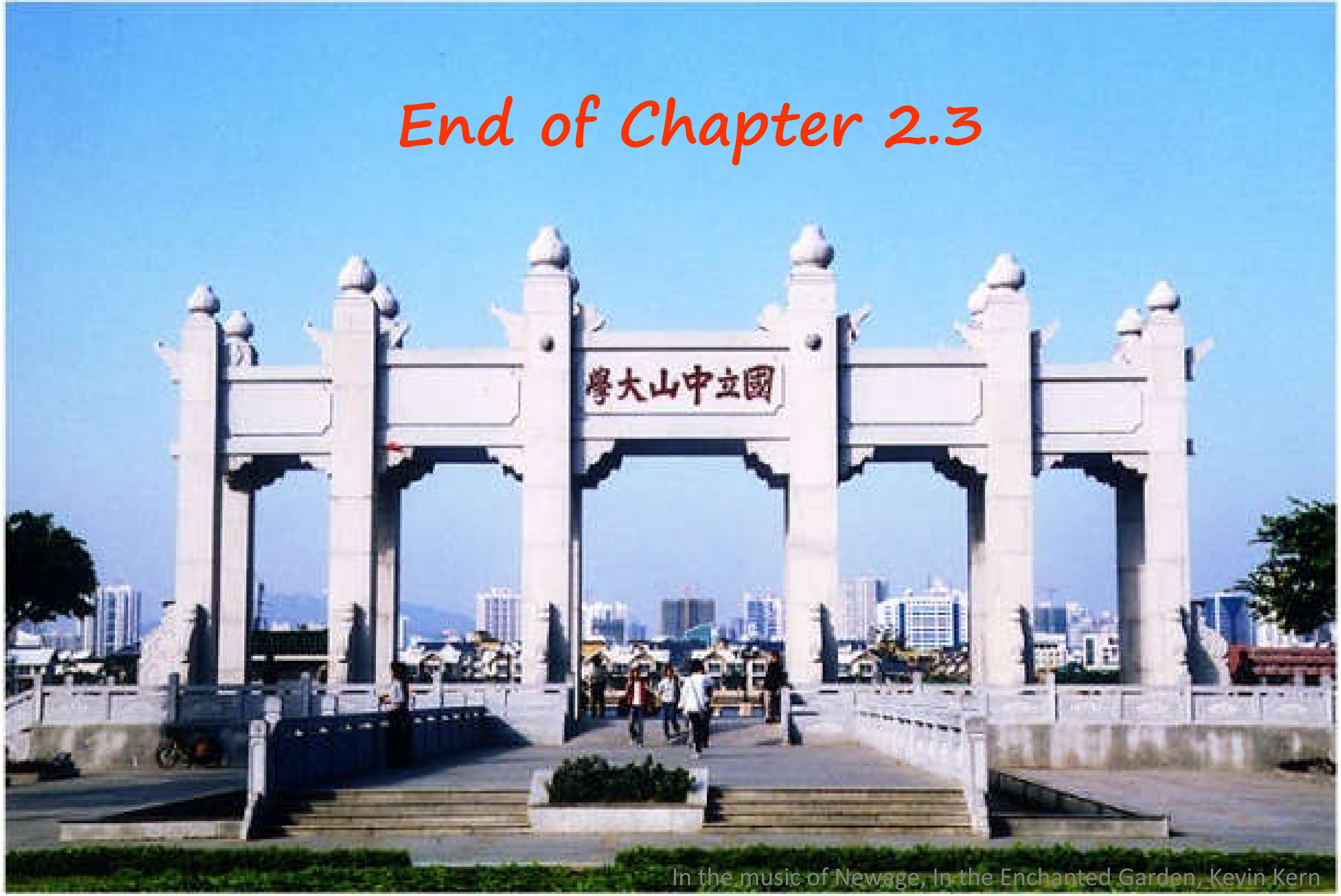
      $a^p - a = 62748517 - 13 = 62748504 = 8964072 \times 7$

      $a^{p-1} = 4826809 = 689544 \times 7 + 1 = qp + 1$
    - ✧ $a = 14$, $p = 7$, $a^p = 14^7 = 105413504$, $a^{p-1} = 14^6 = 7529536$

      $a^p - a = 105413504 - 14 = 105413490 = 15059070 \times 7$

      $a^{p-1} = 7529536 = 1075648 \times 7 = qp$, the *Theorem* failed.

End of Chapter 2.3



In the music of Newage, In the Enchanted Garden, Kevin Kern