Solving Congruences

Section 4.4

Section Summary

- Linear Congruences
- The Chinese Remainder Theorem
- Computer Arithmetic with Large Integers (*not currently included in slides, see text*)
- Fermat's Little Theorem
- Pseudoprimes
- Primitive Roots and Discrete Logarithms

Linear Congruences

Definition: 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*.

- The solutions to a linear congruence $ax \equiv b \pmod{m}$ are all integers x that satisfy the congruence.

Definition: An integer $\bar{a}$ such that $\bar{a}a \equiv 1 \pmod{m}$ is said to be an *inverse* of a modulo m.
Example: 5 is an inverse of 3 modulo 7 since $5 \cdot 3 = 15 \equiv 1 \pmod{7}$

- One method of solving linear congruences makes use of an inverse $\bar{a}$, if it exists. Although we can not divide both sides of the congruence by a, we can multiply by $\bar{a}$ to solve for x.

Inverse of a modulo m

- The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime. Two integers a and b are relatively prime when $\gcd(a,b) = 1$.

Theorem 1: If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m. (This means that there is a unique positive integer $\bar{a}$ less than m that is an inverse of a modulo m and every other inverse of a modulo m is congruent to $\bar{a}$ modulo m.)

Proof: Since $\gcd(a,m) = 1$, by Theorem 6 of Section 4.3, there are integers s and t such that $sa + tm = 1$.

- Hence, $sa + tm \equiv 1 \pmod{m}$.
- Since $tm \equiv 0 \pmod{m}$, it follows that $sa \equiv 1 \pmod{m}$
- Consequently, s is an inverse of a modulo m.
- The uniqueness of the inverse is Exercise 7. ◀

Finding Inverses

- The Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

  Example: Find an inverse of 3 modulo 7.

  Solution: Because gcd(3,7) = 1, by Theorem 1, an inverse of 3 modulo 7 exists.

  - Using the Euclidian algorithm: $7 = 2 \cdot 3 + 1$.
  - From this equation, we get $-2 \cdot 3 + 1 \cdot 7 = 1$, and see that -2 and 1 are Bézout coefficients of 3 and 7.
  - Hence, -2 is an inverse of 3 modulo 7.
  - Also every integer congruent to -2 modulo 7 is an inverse of 3 modulo 7, i.e., 5, -9, 12, etc.

Finding Inverses

Example: Find an inverse of 101 modulo 4620.

Solution: First use the Euclidian algorithm to show that gcd(101,4620) = 1.

$4620 = 45 \cdot 101 + 75$
$101 = 1 \cdot 75 + 26$
$75 = 2 \cdot 26 + 23$
$26 = 1 \cdot 23 + 3$
$23 = 7 \cdot 3 + 2$
$3 = 1 \cdot 2 + 1$
$2 = 2 \cdot 1$

Since the last nonzero remainder is 1, gcd(101,4620) = 1

Working Backwards:

$1 = 3 - 1 \cdot 2$
$1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
$1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$
$1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$
$1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$
$= 26 \cdot 101 - 35 \cdot 75$
$1 = 26 \cdot 101 - 35 \cdot (4620 - 45 \cdot 101)$
$= -35 \cdot 4620 + 1601 \cdot 101$

Bézout coefficients : -35 and 1601

1601 is an inverse of 101 modulo 4620

Using Inverses to Solve Congruences

- We can solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

  Example: What are the solutions of the congruence $3x \equiv 4 \pmod{7}$.

  Solution: We found that -2 is an inverse of 3 modulo 7 (two slides back). We multiply both sides of the congruence by -2 giving

  $$-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}.$$

  Because $-6 \equiv 1 \pmod{7}$ and $-8 \equiv 6 \pmod{7}$, it follows that if x is a solution, then $x \equiv -8 \equiv 6 \pmod{7}$

  We need to determine if every x with $x \equiv 6 \pmod{7}$ is a solution. Assume that $x \equiv 6 \pmod{7}$. By Theorem 5 of Section 4.1, it follows that $3x \equiv 3 \cdot 6 = 18 \equiv 4 \pmod{7}$ which shows that all such x satisfy the congruence.

  The solutions are the integers x such that $x \equiv 6 \pmod{7}$, namely, 6,13,20 ... and $-1, -8, -15, \ldots$

The Chinese Remainder Theorem

- In the first century, the Chinese mathematician Sun-Tsu asked: (孙子算经）

  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，三三数之剩二，五五数之剩三，七七数之剩二。问物几何？
- This puzzle can be translated into the solution of the system of congruences:

  $x \equiv 2 \pmod{3}$,

  $x \equiv 3 \pmod{5}$,

  $x \equiv 2 \pmod{7}$?
- We'll see how the theorem that is known as the *Chinese Remainder Theorem* （中国剩余定理、秦九韶定理） can be used to solve Sun-Tsu's problem.

秦九韶（1208年－1268年）

- 这个定理（《数书九章》大衍求一术，1247年）是中国古代数学史上最完美和最值得骄傲的结果，它出现在中外每一本基础数论的教科书中，西方人称之为中国剩余定理。它不仅在抽象代数理论中有相应的推广和广泛的应用，也被应用到密码学、数值分析、快速傅里叶变换理论等诸多方面。
- 秦九韶字道古，在杭州浙江大学附近的西溪路上，曾有一座桥叫道古桥，就是为了纪念这位13世纪的数学家。此桥由秦九韶亲自设计并筹款，后由元代数学家朱世杰倡议命名。直到21世纪初才被拆除，十分可惜，不过在2012年春天，在浙大蔡天新教授的建议、努力和有关部门的支持下，将离原址不到百米的一座新建石桥命名为道古桥，并请数学家王元院士(1930-2021)题写了桥名。

The Chinese Remainder Theorem

Theorem 2: (*The Chinese Remainder Theorem*) Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers greater than one and $a_1, a_2, \ldots, a_n$ arbitrary integers. Then the system

$$x \equiv a_1 \pmod{m_1}$$
$$x \equiv a_2 \pmod{m_2}$$
$$\vdots$$
$$x \equiv a_n \pmod{m_n}$$

has a unique solution modulo $m = m_1 m_2 \cdots m_n$.
(That is, there is a solution x with $0 \leq x < m$ and all other solutions are congruent modulo m to this solution.)

- **Proof**: We'll show that a solution exists by describing a way to construct the solution. Showing that the solution is unique modulo m is Exercise 30.

continued →

The Chinese Remainder Theorem

To construct a solution first let $M_k = m/m_k$ for $k = 1,2,\ldots,n$ and $m = m_1 m_2 \cdots m_n$.

Since $\gcd(m_k, M_k) = 1$, by Theorem 1, there is an integer y_k, an inverse of M_k modulo m_k, such that
$M_k y_k \equiv 1 \pmod{m_k}$.
Form the sum
$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_n M_n y_n .$$

Note that because $M_j \equiv 0 \pmod{m_k}$ whenever $j \neq k$, all terms except the kth term in this sum are congruent to 0 modulo m_k.
Because $M_k y_k \equiv 1 \pmod{m_k}$, we see that $x \equiv a_k M_k y_k \equiv a_k \pmod{m_k}$, for $k = 1,2,\ldots,n$.
Hence, x is a simultaneous solution to the n congruences.

$x \equiv a_1 \pmod{m_1}$
$x \equiv a_2 \pmod{m_2}$
.
.
.
$x \equiv a_n \pmod{m_n}$

◀

The Chinese Remainder Theorem

Example: Consider the 3 congruences from Sun-Tsu's problem:
$x \equiv 2 \pmod 3$, $x \equiv 3 \pmod 5$, $x \equiv 2 \pmod 7$.

- Let $m = 3 \cdot 5 \cdot 7 = 105$, $M_1 = m/3 = 35$, $M_2 = m/5 = 21$, $M_3 = m/7 = 15$.
- We see that
  - 2 is an inverse of $M_1 = 35$ modulo 3 since $35 \cdot 2 \equiv 2 \cdot 2 \equiv 1 \pmod 3$
  - 1 is an inverse of $M_2 = 21$ modulo 5 since $21 \equiv 1 \pmod 5$
  - 1 is an inverse of $M_3 = 15$ modulo 7 since $15 \equiv 1 \pmod 7$
- Hence,
$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3$$
$$= 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 = 233 \equiv 23 \pmod{105}$$
- We have shown that 23 is the smallest positive integer that is a simultaneous solution. Check it!

Back Substitution

- We can also solve systems of linear congruences with pairwise relatively prime moduli by rewriting a congruences as an equality using Theorem 4 in Section 4.1, substituting the value for the variable into another congruence, and continuing the process until we have worked through all the congruences. This method is known as *back substitution*.

Example: Use the method of back substitution to find all integers x such that $x \equiv 1 \pmod 5$, $x \equiv 2 \pmod 6$, and $x \equiv 3 \pmod 7$.

Solution: By Theorem 4 in Section 4.1, the first congruence can be rewritten as $x = 5t + 1$

Back Substitution

$x = 5t + 1$, where t is an integer.

- Substituting into the second congruence yields $5t + 1 \equiv 2 \pmod 6$.
- $5t + 1 + 4 \equiv 2 + 4 \pmod 6$. $5(t+1) \equiv 0 \pmod 6$. So that $t \equiv 5 \pmod 6$.
- Using Theorem 4 again gives $t = 6u + 5$ where u is an integer.
- Substituting this back into $x = 5t + 1$, gives $x = 5(6u + 5) + 1 = 30u + 26$.
- Inserting this into the third equation gives $30u + 26 \equiv 3 \pmod 7$.
- $30u + 26 + 4 \equiv 3 + 4 \pmod 7$. $30(u+1) \equiv 0 \pmod 7$. So that $u \equiv 6 \pmod 7$.
- By Theorem 4, $u = 7v + 6$, where v is an integer.
- Substituting this expression for u into $x = 30u + 26$, tells us that $x = 30(7v + 6) + 26 = 210v + 206$.

Translating this back into a congruence we find the solution $x \equiv 206 \pmod{210}$.

Fermat's Little Theorem

Pierre de Fermat
(1601-1665)

Theorem 3: (*Fermat's Little Theorem*) If p is prime and a is an integer not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$
Furthermore, for every integer a we have $a^p \equiv a \pmod{p}$
(*proof outlined in Exercise 19*)

Fermat's little theorem is useful in computing the remainders modulo p of large powers of integers.
Example: Find 7^{222} **mod** 11.
By Fermat's little theorem, we know that $7^{10} \equiv 1 \pmod{11}$, and so $(7^{10})^k \equiv 1 \pmod{11}$, for every positive integer k.
Therefore,

$$7^{222} = 7^{22 \cdot 10 + 2} = (7^{10})^{22} 7^2 \equiv (1)^{22} \cdot 49 \equiv 5 \pmod{11}.$$

Hence, 7^{222} **mod** $11 = 5$.

注：可以推广到欧拉函数上去。对于正整数n，欧拉函数φ(n)表示：小于n的正整数中，与n互质的数的个数。a,n互质。
$a^{\varphi(n)} \equiv 1 \pmod{n}$

Pseudoprimes(伪质数)

- By Fermat's little theorem $n > 2$ is prime, where
  $2^{n-1} \equiv 1 \pmod{n}$.
- But if this congruence holds, n may not be prime. Composite integers n such that $2^{n-1} \equiv 1 \pmod{n}$ are called *pseudoprimes* to the base 2.
  Example: The integer 341 is a pseudoprime to the base 2.
  $341 = 11 \cdot 31$
  $2^{340} \equiv 1 \pmod{341}$ (*see in Exercise* 37)
- We can replace 2 by any integer $b \geq 2$.
  Definition: Let b be a positive integer. If n is a composite integer, and $b^{n-1} \equiv 1 \pmod{n}$, then n is called a *pseudoprime to the base b*.

Pseudoprimes

- Given a positive integer n, such that $2^{n-1} \equiv 1 \pmod{n}$:
  - If n does not satisfy the congruence, it is composite.
  - If n does satisfy the congruence, it is either prime or a pseudoprime to the base 2.
- Doing similar tests with additional bases b, provides more evidence as to whether n is prime.
- Among the positive integers not exceeding a positive real number x, compared to primes, there are relatively few pseudoprimes to the base b.
  - For example, among the positive integers less than 10^{10} there are 455,052,512 primes, but only 14,884 pseudoprimes to the base 2.

Carmichael Numbers (*optional*)

Robert Carmichael
(1879-1967)

- There are composite integers n that pass all tests with bases b such that $\gcd(b,n) = 1$.
  Definition: A composite integer n that satisfies the congruence $b^{n-1} \equiv 1 \pmod{n}$ for all positive integers b with $\gcd(b,n) = 1$ is called a *Carmichael* number卡米切尔数.
  Example: The integer 561 is a Carmichael number. To see this:
  - 561 is composite, since $561 = 3 \cdot 11 \cdot 17$.
  - If $\gcd(b, 561) = 1$, then $\gcd(b, 3) = 1$, then $\gcd(b, 11) = \gcd(b, 17) = 1$.
  - Using Fermat's Little Theorem: $b^2 \equiv 1 \pmod{3}$, $b^{10} \equiv 1 \pmod{11}$, $b^{16} \equiv 1 \pmod{17}$.
  - Then
    $b^{560} = (b^2)^{280} \equiv 1 \pmod{3}$,
    $b^{560} = (b^{10})^{56} \equiv 1 \pmod{11}$,
    $b^{560} = (b^{16})^{35} \equiv 1 \pmod{17}$.
  - It follows (*see Exercise* 29) that $b^{560} \equiv 1 \pmod{561}$ for all positive integers b with $\gcd(b,561) = 1$. Hence, 561 is a Carmichael number.
- Even though there are infinitely many Carmichael numbers, there are other tests (described in the exercises) that form the basis for efficient probabilistic primality testing. (*see Chapter* 7)

Primitive Roots （原根）

Definition: A primitive root modulo a prime p is an integer r in $\mathbf{Z}_p$ such that every nonzero element of $\mathbf{Z}_p$ is a power of r.

Example: Since every element of $\mathbf{Z}_{11}$ is a power of 2, 2 is a primitive root of 11.

Powers of 2 modulo 11: $2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 5, 2^5 = 10, 2^6 = 9, 2^7 = 7, 2^8 = 3, 2^9 = 6, 2^{10} = 1$.

Example: Since not all elements of $\mathbf{Z}_{11}$ are powers of 3, 3 is not a primitive root of 11.

Powers of 3 modulo 11: $3^1 = 3, 3^2 = 9, 3^3 = 5, 3^4 = 4, 3^5 = 1$, and the pattern repeats for higher powers.

Important Fact: There is a primitive root modulo p for every prime number p.

。

定义可推广：g在模m下的**乘法阶**恰好等于欧拉函数 φ(m)，则称g为模m的一个**原根**。

模m原根存在的条件是：$m = 1, 2, 4, p^\alpha, 2p^\alpha$,其中 p 为奇素数，$\alpha \geq 1$

Discrete Logarithms

Suppose p is prime and r is a primitive root modulo p. If a is an integer between 1 and $p-1$, that is an element of $\mathbf{Z}_p$, there is a unique exponent e such that $r^e = a$ in $\mathbf{Z}_p$, that is, $r^e \bmod p = a$.

Definition: Suppose that p is prime, r is a primitive root modulo p, and a is an integer between 1 and $p-1$, inclusive. If $r^e \bmod p = a$ and $1 \leq e \leq p-1$, we say that e is the *discrete logarithm* of a modulo p to the base r and we write $\log_r a = e$ (where the prime p is understood).

Example 1: We write $\log_2 3 = 8$ since the discrete logarithm of 3 modulo 11 to the base 2 is 8 as $2^8 = 3$ modulo 11.

Example 2: We write $\log_2 5 = 4$ since the discrete logarithm of 5 modulo 11 to the base 2 is 4 as $2^4 = 5$ modulo 11.

There is no known polynomial time algorithm for computing the discrete logarithm of a modulo p to the base r (when given the prime p, a root r modulo p, and a positive integer $a \in \mathbf{Z}_p$). The problem plays a role in cryptography as will be discussed in Section 4.6.

不对称性：
知道p r e 求 a 易
知道p r a 求 e 难

Exercise

- Sec 4.4 9, 21

Applications of Congruences

Section 4.5

Section Summary

- Hashing Functions
- Pseudorandom Numbers
- Check Digits
- Public Key Cryptography

Hashing Functions

Definition: A *hashing function h* assigns memory location $h(k)$ to the record that has k as its key.

- A common hashing function is $h(k) = k$ **mod** m, where m is the number of memory locations.
- Because this hashing function is onto, all memory locations are possible.

Example: Let $h(k) = k$ **mod** 111. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

h(064212848) = 064212848 **mod** 111 = 14
h(037149212) = 037149212 **mod** 111 = 65
h(107405723) = 107405723 **mod** 111 = 14, but since location 14 is already occupied, the record is assigned to the next available position, which is 15.

- The hashing function is not one-to-one as there are many more possible keys than memory locations. When more than one record is assigned to the same location, we say a *collision* occurs. Here a collision has been resolved by assigning the record to the first free location.
- For collision resolution, we can use a *linear probing function*:
  $h(k,i) = (h(k) + i)$ **mod** m, where i runs from 0 to $m - 1$.
- There are many other methods of handling with collisions. You may cover these in a later CS course.

Pseudorandom Numbers

- Randomly chosen numbers are needed for many purposes, including computer simulations.
- *Pseudorandom numbers* are not truly random since they are generated by systematic methods.
- The *linear congruential method* is one commonly used procedure for generating pseudorandom numbers.
- Four integers are needed: the *modulus m*, the *multiplier a*, the *increment c*, and *seed* x_0, with $2 \leq a < m$, $0 \leq c < m$, $0 \leq x_0 < m$.
- We generate a sequence of pseudorandom numbers $\{x_n\}$, with $0 \leq x_n < m$ for all n, by successively using the recursively defined function

$$x_{n+1} = (ax_n + c) \bmod m.$$

(*an example of a recursive definition, discussed in Section* 5.3*)*

- If psudorandom numbers between 0 and 1 are needed, then the generated numbers are divided by the modulus, x_n / m.

Pseudorandom Numbers

- **Example**: Find the sequence of pseudorandom numbers generated by the linear congruential method with modulus $m = 9$, multiplier $a = 7$, increment $c = 4$, and seed $x_0 = 3$.
- **Solution**: Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \bmod 9$, with $x_0 = 3$.

$x_1 = 7x_0 + 4 \bmod 9 = 7 \cdot 3 + 4 \bmod 9 = 25 \bmod 9 = 7,$
$x_2 = 7x_1 + 4 \bmod 9 = 7 \cdot 7 + 4 \bmod 9 = 53 \bmod 9 = 8,$
$x_3 = 7x_2 + 4 \bmod 9 = 7 \cdot 8 + 4 \bmod 9 = 60 \bmod 9 = 6,$
$x_4 = 7x_3 + 4 \bmod 9 = 7 \cdot 6 + 4 \bmod 9 = 46 \bmod 9 = 1,$
$x_5 = 7x_4 + 4 \bmod 9 = 7 \cdot 1 + 4 \bmod 9 = 11 \bmod 9 = 2,$
$x_6 = 7x_5 + 4 \bmod 9 = 7 \cdot 2 + 4 \bmod 9 = 18 \bmod 9 = 0,$
$x_7 = 7x_6 + 4 \bmod 9 = 7 \cdot 0 + 4 \bmod 9 = 4 \bmod 9 = 4,$
$x_8 = 7x_7 + 4 \bmod 9 = 7 \cdot 4 + 4 \bmod 9 = 32 \bmod 9 = 5,$
$x_9 = 7x_8 + 4 \bmod 9 = 7 \cdot 5 + 4 \bmod 9 = 39 \bmod 9 = 3.$

The sequence generated is 3,7,8,6,1,2,0,4,5,3,7,8,6,1,2,0,4,5,3,...
It repeats after generating 9 terms.

- Commonly, computers use a linear congruential generator with increment $c = 0$. This is called a *pure multiplicative generator*. Such a generator with modulus $2^{31} - 1$ and multiplier $7^5 = 16{,}807$ generates $2^{31} - 2$ numbers before repeating.

Check Digits: UPCs

- A common method of detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function. If the final digit is not correct, then the string is assumed not to be correct.

Example: Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit is determined by the congruence:

$3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \pmod{10}$.

a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?
b. Is 041331021641 a valid UPC?

Solution:

a. $3\cdot7 + 9 + 3\cdot3 + 5 + 3\cdot7 + 3 + 3\cdot4 + 3 + 3\cdot1 + 0 + 3\cdot4 + x_{12} \equiv 0 \pmod{10}$
$21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$
$98 + x_{12} \equiv 0 \pmod{10}$
$x_{12} \equiv 0 \pmod{10}$ So, the check digit is 2.

b. $3\cdot0 + 4 + 3\cdot1 + 3 + 3\cdot3 + 1 + 3\cdot0 + 2 + 3\cdot1 + 6 + 3\cdot4 + 1 \equiv 0 \pmod{10}$
$0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 = 44 \equiv 4 \not\equiv \pmod{10}$
Hence, 041331021641 is not a valid UPC.

Check Digits:ISBNs

Books are identified by an *International Standard Book Number* (ISBN-10), a 10 digit code. The first 9 digits identify the language, the publisher, and the book. The tenth digit is a check digit, which is determined by the following congruence

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11}.$$

The validity of an ISBN-10 number can be evaluated with the equivalent $\sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$.

a. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
b. Is 084930149X a valid ISBN10?

Solution:

a. $X_{10} \equiv 1\cdot0 + 2\cdot0 + 3\cdot7 + 4\cdot2 + 5\cdot8 + 6\cdot8 + 7\cdot0 + 8\cdot0 + 9\cdot8 \pmod{11}$.
$X_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \pmod{11}$.
$X_{10} \equiv 189 \equiv 2 \pmod{11}$. Hence, $X_{10} = 2$.

b. $1\cdot0 + 2\cdot8 + 3\cdot4 + 4\cdot9 + 5\cdot3 + 6\cdot0 + 7\cdot1 + 8\cdot4 + 9\cdot9 + 10\cdot10 =$
$0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0 \pmod{11}$
Hence, 084930149X is not a valid ISBN-10.

X is used for the digit 10.

- A *single error* is an error in one digit of an identification number and a *transposition error* is the accidental interchanging of two digits. Both of these kinds of errors can be detected by the check digit for ISBN-10. (*see text for more details*)
- *The ISBN of our book is 978-7-111-38550-9*

Public Key Cryptography

- All classical ciphers, including shift and affine ciphers, are *private key cryptosystems*. Knowing the encryption key allows one to quickly determine the decryption key.
- All parties who wish to communicate using a private key cryptosystem must share the key and keep it a secret.
- In public key cryptosystems, first invented in the 1970s, knowing how to encrypt a message does not help one to decrypt the message. Therefore, everyone can have a publicly known encryption key. The only key that needs to be kept secret is the decryption key.

用Alice公钥（公开）加密，只有Alice可用私钥（保密）解密。定向发密文。

Clifford Cocks
(Born 1950)

The RSA Cryptosystem

- A public key cryptosystem, now known as the RSA system was introduced in 1976 by three researchers at MIT.

Ronald Rivest
(Born 1948)

Adi Shamir
(Born 1952)

Leonard
Adelman
(Born 1945)

- It is now known that the method was discovered earlier by Clifford Cocks, working secretly for the UK government.
- The public encryption key is (n,e), where $n = pq$ (the modulus) is the product of two large (300 digits) primes p and q, and an exponent e that is relatively prime to $(p-1)(q-1)$. The two large primes can be quickly found using probabilistic primality tests, discussed earlier. But $n = pq$, with approximately 600 digits, cannot be factored in a reasonable length of time.

Factorization is believed to be a difficult problem, as opposed to finding large primes p and q, which can be done quickly. The most efficient factorization methods known (as of 2017) require billions of years to factor 600-digit integers.

RSA Encryption

- To encrypt a message using RSA using a key (n,e) （公钥）:
  i. Translate the plaintext message M into sequences of two digit integers representing the letters. Use 00 for A, 01 for B, etc.
  ii. Concatenate the two digit integers into strings of digits.
  iii. Divide this string into equally sized blocks of $2N$ digits where $2N$ is the largest even number 2525…25 with $2N$ digits that does not exceed n.
  iv. The plaintext message M is now a sequence of integers $m_1, m_2, \ldots, m_k$.
  v. Each block (an integer) is encrypted using the function $C = M^e \bmod n$.

Example: Encrypt the message STOP using the RSA cryptosystem with key(2537,13).
- $2537 = 43 \cdot 59$,
- $p = 43$ and $q = 59$ are primes and $\gcd(e,(p-1)(q-1)) = \gcd(13, 42 \cdot 58) = 1$.

Solution: Translate the letters in STOP to their numerical equivalents 18 19 14 15.
- Divide into blocks of four digits (because $2525 < 2537 < 252525$) to obtain 1819 1415.
- Encrypt each block using the mapping $C = M^{13} \bmod 2537$.
- Since $1819^{13} \bmod 2537 = 2081$ and $1415^{13} \bmod 2537 = 2182$, the encrypted message is 2081 2182.

RSA Decryption

- To decrypt a RSA ciphertext message, the decryption key d, an inverse of e modulo $(p-1)(q-1)$ is needed. The inverse exists since $\gcd(e,(p-1)(q-1)) = \gcd(13, 42 \cdot 58) = 1$.
- With the decryption key d（私钥）, we can decrypt each block with the computation $M = C^d \bmod n$. （$n=p \cdot q$） (*see text for full derivation*)
- RSA works as a public key system since the only known method of finding d is based on a factorization of n into primes. There is currently no known feasible method for factoring large numbers into primes.

Example: The message 0981 0461 is received. What is the decrypted message if it was encrypted using the RSA cipher from the previous example.

Solution: The message was encrypted with $n = 43 \cdot 59$ and exponent 13. An inverse of 13 modulo $42 \cdot 58 = 2436$ (*exercise 2 in Section* 4.4) is $d = 937$.
- To decrypt a block C, $M = C^{937} \bmod 2537$.
- Since $0981^{937} \bmod 2537 = 0704$ and $0461^{937} \bmod 2537 = 1115$, the decrypted message is 0704 1115. Translating back to English letters, the message is HELP.

- *不经过素因子分解，从n直接计算(p–1)(q –1)或d的值，被相信很难*
- $C = M^e \bmod n$，*只已知C、e、n反求M，被相信很难*

Cryptographic Protocols: Key Exchange

- *Cryptographic protocols* are exchanges of messages carried out by two or more parties to achieve a particular security goal.
- *Key exchange* is a protocol by which two parties can exchange a secret key over an insecure channel without having any past shared secret information. Here the *Diffe-Hellman key agreement protcol* is described by example.
  i. Suppose that Alice and Bob want to share a common key.
  ii. Alice and Bob agree to use a prime p and a primitive root a of p.
  iii. Alice chooses a secret integer k_1 and sends a^{k1} **mod** p to Bob.
  iv. Bob chooses a secret integer k_2 and sends a^{k2} **mod** p to Alice.
  v. Alice computes $(a^{k2})^{k1}$ **mod** p.
  vi. Bob computes $(a^{k1})^{k2}$ **mod** p.

  At the end of the protocol, Alice and Bob have their shared key
  $(a^{k2})^{k1}$ **mod** $p = (a^{k1})^{k2}$ **mod** p.
- To find the secret information from the public information would require the adversary to find k_1 and k_2 from a^{k1} **mod** p and a^{k2} **mod** p respectively. This is an instance of the discrete logarithm problem, considered to be computationally infeasible when p and a are sufficiently large.

Cryptographic Protocols: Digital Signatures

Adding a *digital signature* to a message is a way of ensuring the recipient that the message came from the purported sender.

- Suppose that Alice's RSA public key is (n,e) and her private key is d. Alice encrypts a plain text message x using $E_{(n,e)}(x) = x^e$ **mod** n. She decrypts a ciphertext message y using $D_{(n,e)}(y) = y^d$ **mod** n.
- Alice wants to send a message M so that everyone who receives the message knows that it came from her.
  1. She translates the message to numerical equivalents and splits into blocks, just as in RSA encryption.
  2. She then applies her decryption function $D_{(n,e)}$ to the blocks and sends the results to all intended recipients.
  3. The recipients apply Alice's encryption function and the result is the original plain text since $E_{(n,e)}(D_{(n,e)}(x)) = x$.

  Everyone who receives the message can then be certain that it came from Alice.

用Alice私钥加密，用Alice公钥解密。因为私钥只有Alice知道，所以可以用于认证来源。

Cryptographic Protocols: Digital Signatures

Example: Suppose Alice's RSA cryptosystem is the same as in the earlier example with key(2537,13), $2537 = 43 \cdot 59$, $p = 43$ and $q = 59$ are primes and $\gcd(e,(p-1)(q-1)) = \gcd(13, 42 \cdot 58) = 1$.

Her decryption key is d = 937.

She wants to send the message "MEET AT NOON" to her friends so that they can be certain that the message is from her.

Solution: Alice translates the message into blocks of digits 1204 0419 0019 1314 1413.

1. She then applies her decryption transformation $D_{(2537,13)}(x) = x^{937} \bmod 2537$ to each block.
2. She finds (using her laptop, programming skills, and knowledge of discrete mathematics) that $1204^{937} \bmod 2537 = 817$, $419^{937} \bmod 2537 = 555$, $19^{937} \bmod 2537 = 1310$, $1314^{937} \bmod 2537 = 2173$, and $1413^{937} \bmod 2537 = 1026$.
3. She sends 0817 0555 1310 2173 1026.

When one of her friends receive the message, they apply Alice's encryption transformation $E_{(2537,13)}$ to each block. They then obtain the original message which they translate back to English letters.

Combing Digital Signatures and Digital encryptions

- Alice wants to send a message M to Bob so that Bob receives the message knows that it came from her. Bob also want the message is only for him.
  1. She translates the message to numerical equivalents and splits into blocks, just as in RSA encryption.
  2. She first applies Bob's encryption function $E_{(n',e')}$ to the blocks
  3. She then applies her decryption function $D_{(n,e)}$ to the blocks and sends the results to all intended recipients.
  4. Bob apply Alice's encryption function and his own decryption function $D_{(n',e')}$ the result is : $(D_{(n',e')}(E_{(n,e)}(D_{(n,e)}(E_{(n',e')}(x))))) = x$.

  Bob receives the message can then be certain that it came from Alice and it is only for him.

  用Bob公钥加密，用Alice私钥加密。
  用Alice公钥认证，用Bob私钥解密。

Exercise

- 无