Number Theory and Cryptography

Chapter 4

With Question/Answer Animations

Because learning changes everything.™

Chapter Motivation

Number theory is the part of mathematics devoted to the study of the integers and their properties.

Key ideas in number theory include divisibility and the primality of integers.

Representations of integers, including binary and hexadecimal representations, are part of number theory.

Number theory has long been studied because of the beauty of its ideas, its accessibility, and its wealth of open questions.

We'll use many ideas developed in Chapter 1 about proof methods and proof strategy in our exploration of number theory.

Mathematicians have long considered number theory to be pure mathematics, but it has important applications to computer science and cryptography studied in Sections 4.5 and 4.6.

Chapter Summary

Divisibility and Modular Arithmetic

Integer Representations and Algorithms

Primes and Greatest Common Divisors

Solving Congruences

Applications of Congruences

Cryptography

Divisibility and Modular Arithmetic

Section 4.1

Section Summary [1]

Division

Division Algorithm

Modular Arithmetic

Division

Definition: If a and b are integers with $a \neq 0$, then a *divides* b if there exists an integer c such that $b = ac$.

- When a divides b we say that a is a *factor* or *divisor* of b and that b is a multiple of a.
- The notation $a \mid b$ denotes that a divides b.
- If $a \mid b$, then b/a is an integer.
- If a does not divide b, we write $a \nmid b$.

Example: Determine whether $3 \mid 7$ and whether $3 \mid 12$.

Properties of Divisibility

Theorem 1: Let a, b, and c be integers, where $a \neq 0$.

i. If $a \mid b$ and $a \mid c$, then $a \mid (b + c)$;

ii. If $a \mid b$, then $a \mid bc$ for all integers c;

iii. If $a \mid b$ and $b \mid c$, then $a \mid c$.

Proof: (i) Suppose $a \mid b$ and $a \mid c$, then it follows that there are integers s and t with $b = as$ and $c = at$. Hence,

$b + c = as + at = a(s + t)$. Hence, $a \mid (b + c)$

(Exercises 3 and 4 ask for proofs of parts (ii) and (iii).)

Corollary: Let a, b, and c be integers, where $a \neq 0$, such that $a \mid b$ and $a \mid c$, then $a \mid mb + nc$ whenever m and n are integers.

Can you show how it follows easily from (ii) and (i) of Theorem 1?

Division Algorithm

When an integer is divided by a positive integer, there is a quotient and a remainder. This is traditionally called the "Division Algorithm," but is really a theorem.

Division Algorithm: If *a* is an integer and *d* a positive integer, then there are unique integers *q* and *r*, with $0 \le r < d$, such that $a = dq + r$ (*proved in Section* 5.2).

- *d* is called the *divisor*.
- *a* is called the *dividend*.
- *q* is called the *quotient*.
- *r* is called the *remainder*.

Definitions of Functions **div** and **mod**

$q = a \textbf{ div } d$

$r = a \textbf{ mod } d$

Examples:

- What are the quotient and remainder when 101 is divided by 11?
- **Solution**: The quotient when 101 is divided by 11 is 9 = 101 **div** 11, and the remainder is 2 = 101 **mod** 11.
- What are the quotient and remainder when −11 is divided by 3?
- **Solution**: The quotient when −11 is divided by 3 is −4 = −11 **div** 3, and the remainder is 1 = −11 **mod** 3.

Congruence Relation

Definition: If a and b are integers and m is a positive integer, then a is *congruent* to b *modulo* m if m divides $a - b$.

- The notation $a \equiv b \pmod{m}$ says that a is congruent to b modulo m.
- We say that $a \equiv b \pmod{m}$ is a *congruence* and that m is its *modulus.*
- Two integers are congruent mod m if and only if they have the same remainder when divided by m.
- If a is not congruent to b modulo m, we write $a \not\equiv b \pmod{m}$

Example: Determine whether 17 is congruent to 5 modulo 6 and whether 24 and 14 are congruent modulo 6.

Solution:

- $17 \equiv 5 \pmod{6}$ because 6 divides $17 - 5 = 12$.
- $24 \not\equiv 14 \pmod{6}$ since $24 - 14 = 10$ is not divisible by 6.

More on Congruences

Theorem 4: Let m be a positive integer. The integers *a* and *b* are congruent modulo *m* if and only if there is an integer *k* such that $a = b + km$.

Proof:

- If $a \equiv b \pmod{m}$, then (by the definition of congruence) $m \mid a - b$. Hence, there is an integer *k* such that $a - b = km$ and equivalently $a = b + km$.
- Conversely, if there is an integer *k* such that $a = b + km$, then $km = a - b$. Hence, $m \mid a - b$ and $a \equiv b \pmod{m}$.

The Relationship between (mod m) and **mod** m Notations

The use of "mod" in $a \equiv b \pmod{m}$ and a **mod** $m = b$ are different.

- $a \equiv b \pmod{m}$ is a relation on the set of integers.
- In a **mod** $m = b$, the notation **mod** denotes a function.

The relationship between these notations is made clear in this theorem.

Theorem 3: Let a and b be integers, and let m be a positive integer. Then $a \equiv b \pmod{m}$ if and only if a **mod** $m = b$ **mod** m. (*Proof in the exercises*)

Congruences of Sums and Products*

Theorem 5: Let m be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$

Proof:

- Because $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, by Theorem 4 there are integers s and t with $b = a + sm$ and $d = c + tm$.
- Therefore,
  - $b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$ and
  - $b\,d = (a + sm)\,(c + tm) = ac + m(at + cs + stm)$.
- Hence, $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

Example: Because $7 \equiv 2 \pmod{5}$ and $11 \equiv 1 \pmod{5}$, it follows from Theorem 5 that

$$18 = 7 + 11 \equiv 2 + 1 = 3 \pmod{5}$$

$$77 = 7 \cdot 11 \equiv 2 \cdot 1 = 2 \pmod{5}$$

Algebraic Manipulation of Congruences*

Multiplying both sides of a valid congruence by an integer preserves validity.

If $a \equiv b \pmod{m}$ holds then $c \cdot a \equiv c \cdot b \pmod{m}$, where c is any integer, holds by Theorem 5 with $d = c$.

Adding an integer to both sides of a valid congruence preserves validity.

If $a \equiv b \pmod{m}$ holds then $c + a \equiv c + b \pmod{m}$, where c is any integer, holds by Theorem 5 with $d = c$.

Dividing a congruence by an integer does not always produce a valid congruence.

Example: The congruence $14 \equiv 8 \pmod{6}$ holds. But dividing both sides by 2 does not produce a valid congruence since $14/2 = 7$ and $8/2 = 4$, but $7 \not\equiv 4 \pmod{6}$.

See Section 4.3 for conditions when division is ok.

Computing the **mod** *m* Function of Products and Sums*

We use the following corollary to Theorem 5 to compute the remainder of the product or sum of two integers when divided by *m* from the remainders when each is divided by *m*.

Corollary: Let *m* be a positive integer and let *a* and *b* be integers. Then

$(a + b)$ $(\mathbf{mod}\ m) = ((a\ \mathbf{mod}\ m) + (b\ \mathbf{mod}\ m))\ \mathbf{mod}\ m$

and

$ab\ \mathbf{mod}\ m = ((a\ \mathbf{mod}\ m)\ (b\ \mathbf{mod}\ m))\ \mathbf{mod}\ m.$

(*proof in text*)

Arithmetic Modulo m [1]*

Definitions: Let $\mathbf{Z}_m$ be the set of nonnegative integers less than m: $\{0,1, \ldots., m-1\}$

- The operation $+_m$ is defined as $a +_m b = (a + b)$ **mod** m. This is *addition modulo m*.
- The operation $\cdot_m$ is defined as $a \cdot_m b = (a \cdot b)$ **mod** m. This is *multiplication modulo m*.
- Using these operations is said to be doing *arithmetic modulo m*.

Example: Find $7 +_{11} 9$ and $7 \cdot_{11} 9$.

Solution: Using the definitions above:

- $7 +_{11} 9 = (7 + 9)$ **mod** $11 = 16$ **mod** $11 = 5$
- $7 \cdot_{11} 9 = (7 \cdot 9)$ **mod** $11 = 63$ **mod** $11 = 8$

Arithmetic Modulo m 2*

The operations $+_m$ and $\cdot_m$ satisfy many of the same properties as ordinary addition and multiplication.

- *Closure*: If a and b belong to $\mathbf{Z}_m$, then $a +_m b$ and $a \cdot_m b$ belong to $\mathbf{Z}_m$.
- *Associativity*: If a, b, and c belong to $\mathbf{Z}_m$, then $(a +_m b) +_m c = a +_m (b +_m c)$ and $(a \cdot_m b) \cdot_m c = a \cdot_m (b \cdot_m c)$.
- *Commutativity*: If a and b belong to $\mathbf{Z}_m$, then $a +_m b = b +_m a$ and $a \cdot_m b = b \cdot_m a$.
- *Identity elements*: The elements 0 and 1 are identity elements for addition and multiplication modulo m, respectively.
  - If a belongs to $\mathbf{Z}_m$, then $a +_m 0 = a$ and $a \cdot_m 1 = a$.

Arithmetic Modulo m [3]*

- *Additive inverses*: If $a \neq 0$ belongs to $\mathbf{Z}_m$, then $m - a$ is the additive inverse of a modulo m and 0 is its own additive inverse.
  - $a +_m (m - a) = 0$ and $0 +_m 0 = 0$
- *Distributivity*: If a, b, and c belong to $\mathbf{Z}_m$, then
  - $a \cdot_m (b +_m c) = (a \cdot_m b) +_m (a \cdot_m c)$ and $(a +_m b) \cdot_m c = (a \cdot_m c) +_m (b \cdot_m c)$.

Exercises 42-44 ask for proofs of these properties.

Multiplicatative inverses have not been included since they do not always exist. For example, there is no multiplicative inverse of 2 modulo 6.

(*optional*) Using the terminology of abstract algebra, $\mathbf{Z}_m$ with $+_m$ is a commutative group and $\mathbf{Z}_m$ with $+_m$ and $\cdot_m$ is a commutative ring.

Integer Representations and Algorithms

Section 4.2

Section Summary 2

Integer Representations

- Base *b* Expansions
- Binary Expansions
- Octal Expansions
- Hexadecimal Expansions

Base Conversion Algorithm

Algorithms for Integer Operations

Representations of Integers

In the modern world, we use *decimal,* or *base* 10, *notation* to represent integers. For example when we write 965, we mean $9 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0$.

We can represent numbers using any base b, where b is a positive integer greater than 1.

The bases $b = 2$ (*binary*), $b = 8$ (*octal*) , and $b = 16$ (*hexadecimal*) are important for computing and communications

The ancient Mayans used base 20 and the ancient Babylonians used base 60.

Base b Representations

We can use positive integer b greater than 1 as a base, because of this theorem:

Theorem 1: Let b be a positive integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form:

$$n = a_k b^k + a_{k-1} b^{k-1} + \ldots. + a_1 b + a_0$$

where k is a nonnegative integer, $a_0, a_1, \ldots. a_k$ are nonnegative integers less than b, and $a_k \neq 0$. The a_j, $j = 0, \ldots, k$ are called the base-b digits of the representation.
(We will prove this using mathematical induction in Section 5.1.)

The representation of n given in Theorem 1 is called the *base b expansion of n* and is denoted by $(a_k a_{k-1} \ldots. a_1 a_0)_b$.

We usually omit the subscript 10 for base 10 expansions.

Binary Expansions

Most computers represent integers and do arithmetic with binary (base 2) expansions of integers. In these expansions, the only digits used are 0 and 1.

Example: What is the decimal expansion of the integer that has $(1\ 0101\ 1111)_2$ as its binary expansion?

Solution:

$(1\ 0101\ 1111)_2 = 1\cdot2^8 + 0\cdot2^7 + 1\cdot2^6 + 0\cdot2^5 + 1\cdot2^4 + 1\cdot2^3 + 1\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 = 351.$

Example: What is the decimal expansion of the integer that has $(11011)_2$ as its binary expansion?

Solution: $(11011)_2 = 1\cdot2^4 + 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 = 27.$

Octal Expansions

The octal expansion (base 8) uses the digits $\{0,1,2,3,4,5,6,7\}$.

Example: What is the decimal expansion of the number with octal expansion $(7016)_8$?

Solution: $7\cdot 8^3 + 0\cdot 8^2 + 1\cdot 8^1 + 6\cdot 8^0 = 3598$

Example: What is the decimal expansion of the number with octal expansion $(111)_8$?

Solution: $1\cdot 8^2 + 1\cdot 8^1 + 1\cdot 8^0 = 64 + 8 + 1 = 73$

Hexadecimal Expansions

The hexadecimal expansion needs 16 digits, but our decimal system provides only 10. So letters are used for the additional symbols. The hexadecimal system uses the digits {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}. The letters A through F represent the decimal numbers 10 through 15.

Example: What is the decimal expansion of the number with hexadecimal expansion $(2AE0B)_{16}$?

Solution:

$2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16^1 + 11 \cdot 16^0 = 175627$

Example: What is the decimal expansion of the number with hexadecimal expansion $(E5)_{16}$?

Solution: $14 \cdot 16^1 + 5 \cdot 16^0 = 224 + 5 = 229$

Base Conversion[1]

To construct the base b expansion of an integer n:

- Divide n by b to obtain a quotient and remainder.
  $n = bq_0 + a_0 \quad 0 \le a_0 \le b$
- The remainder, a_0, is the rightmost digit in the base b expansion of n. Next, divide q_0 by b.
  $q_0 = bq_1 + a_1 \quad 0 \le a_1 \le b$
- The remainder, a_1, is the second digit from the right in the base b expansion of n.
- Continue by successively dividing the quotients by b, obtaining the additional base b digits as the remainder. The process terminates when the quotient is 0.

Algorithm for Base *b* Expansions*

procedure *base b expansion* (n, b: positive integers with $b > 1$)
  $q := n$
  $k := 0$
  while ($q \neq 0$)
    $a_k := q$ **mod** b
    $q := q$ **div** b
    $k := k + 1$
  return $(a_{k-1}, \ldots, a_1, a_0)$ $\{(a_{k-1} \ldots a_1 a_0)_b$ is base b expansion of $n\}$

q represents the quotient obtained by successive divisions by b, starting with $q = n$.

The digits in the base b expansion are the remainders of the division given by q **mod** b.

The algorithm terminates when $q = 0$ is reached.

Base Conversion [2]

Example: Find the octal expansion of $(12345)_{10}$

Solution: Successively dividing by 8 gives:

- $12345 = 8 \cdot 1543 + 1$
- $1543 = 8 \cdot 192 + 7$
- $192 = 8 \cdot 24 + 0$
- $24 = 8 \cdot 3 + 0$
- $3 = 8 \cdot 0 + 3$

The remainders are the digits from right to left yielding $(30071)_8$.

Base Conversion 3

Example: Find the binary expansion of $(1693)_{10}$

Solution: Successively dividing by 2 gives:

dividend	quotient	remainder
1693	846	1
846	423	0
423	211	1
211	105	1
105	52	1
52	26	0
26	13	0
13	6	1
6	3	0
3	1	1
1	0	1

remainders in reverse order = 11010011101_2

Comparison of Hexadecimal, Octal, and Binary Representations

TABLE 1 Hexadecimal, Octal, and Binary Representation of the Integers 0 through 15.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Octal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Binary	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

Initial 0s are not shown

Each octal digit corresponds to a block of 3 binary digits.

Each hexadecimal digit corresponds to a block of 4 binary digits.

So, conversion between binary, octal, and hexadecimal is easy.

Conversion Between Binary, Octal, and Hexadecimal Expansions

Example: Find the octal and hexadecimal expansions of $(11\ 1110\ 1011\ 1100)_2$.

Solution:

- To convert to octal, we group the digits into blocks of three $(011\ 111\ 010\ 111\ 100)_2$, adding initial 0s as needed. The blocks from left to right correspond to the digits 3,7,2,7, and 4. Hence, the solution is $(37274)_8$.
- To convert to hexadecimal, we group the digits into blocks of four $(0011\ 1110\ 1011\ 1100)_2$, adding initial 0s as needed. The blocks from left to right correspond to the digits 3,E,B, and C. Hence, the solution is $(3EBC)_{16}$.

Binary Addition of Integers

Algorithms for performing operations with integers using their binary expansions are important as computer chips work with binary numbers. Each digit is called a *bit*.

procedure *add* (*a, b*: positive integers)
{the binary expansions of *a* and *b* are $(a_{n-1},a_{n-2},\ldots,a_0)_2$ and $(b_{n-1},b_{n-2},\ldots,b_0)_2$, respectively}
$c := 0$
for $j := 0$ to $n - 1$
$d := \lfloor (a_j + b_j + c)/2 \rfloor$
$s_j := a_j + b_j + c - 2d$
$c := d$
$s_n := c$
return $(s_0,s_1,\ldots, s_n)$ {the binary expansion of the sum is $(s_n,s_{n-1},\ldots,s_0)_2$}

The number of additions of bits used by the algorithm to add two *n*-bit integers is $O(n)$.

Binary Multiplication of Integers

Algorithm for computing the product of two *n* bit integers.

procedure multiply (a, b: positive integers)
{the binary expansions of a and b are $(a_{n-1}, a_{n-2}, \ldots, a_0)_2$ and
$(b_{n-1}, b_{n-2}, \ldots, b_0)_2$, respectively}
for $j := 0$ to $n - 1$
  if $b_j = 1$ **then** $c_j = a$ shifted j places
  else $c_j := 0$
{$c_o, c_1, \ldots, c_{n-1}$ are the partial products}
$p := 0$
for $j := 0$ to $n - 1$
  $p := p + c_j$
return p {p is the value of ab}

The number of additions of bits used by the algorithm to multiply two n-bit integers is $O(n^2)$.

Binary Modular Exponentiation

In cryptography, it is important to be able to find b^n mod m efficiently, where b, n, and m are large integers.

Use the binary expansion of n, $n = (a_{k-1},\ldots,a_1,a_0)_2$, to compute b^n .

Note that:

$$b^n = b^{a_{k-1} \cdot 2^{k-1} + \cdots + a_1 \cdot 2 + a_0} = b^{a_{k-1} \cdot 2^{k-1}} \cdots b^{a_1 \cdot 2} \cdot b^{a_0}$$

Therefore, to compute b^n, we need only compute the values of b, b^2, $(b^2)^2 = b^4$, $(b^4)^2 = b^8$, …, and then multiply the terms in this list, where $a_j = 1$.

Example: Compute 3^{11} using this method.

Solution: Note that $11 = (1011)_2$ so that $3^{11} = 3^8\, 3^2\, 3^1 = ((3^2)^2)^2\, 3^2\, 3^1 = (9^2)^2 \cdot 9 \cdot 3 = (81)^2 \cdot 9 \cdot 3 = 6561 \cdot 9 \cdot 3 = 117{,}147$.

Binary Modular Exponentiation Algorithm

Algorithm for computing the product of two *n* bit integers.

procedure *modular exponentiation* (b: integer, $n = (a_{k-1}a_{k-2}\dots a_1a_0)_2$, m: positive integers)

$x := 1$

$power := b$ **mod** m

for $i := 0$ to $k - 1$

  if $a_i = 1$ **then** $x := (x \cdot power)$ **mod** m

  $power := (power \cdot power)$ **mod** m

return x {x equals b^n **mod** m }

$O((\log m)^2 \log n)$ bit operations are used to find b^n **mod** m.

Primes and Greatest Common Divisors

Section 4.3

Section Summary 3

Prime Numbers and their Properties

Conjectures and Open Problems About Primes

Greatest Common Divisors and Least Common Multiples

The Euclidian Algorithm

gcds as Linear Combinations

Primes

Definition: A positive integer p greater than 1 is called *prime* if the only positive factors of p are 1 and p. A positive integer that is greater than 1 and is not prime is called *composite*.

Example: The integer 7 is prime because its only positive factors are 1 and 7, but 9 is composite because it is divisible by 3.

The Fundamental Theorem of Arithmetic

Theorem: Every positive integer greater than 1 can be written uniquely as a prime or as the product of two or more primes where the prime factors are written in order of nondecreasing size.

Examples:

- $100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$
- $641 = 641$
- $999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$
- $1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$

The Sieve of Eratosthenes 1

Eratosthenes

(276-194 B.C.)

The *Sieve of Eratosthenes* can be used to find all primes not exceeding a specified positive integer. For example, begin with the list of integers between 1 and 100.s

a. Delete all the integers, other than 2, divisible by 2.

b. Delete all the integers, other than 3, divisible by 3.

c. Next, delete all the integers, other than 5, divisible by 5.

d. Next, delete all the integers, other than 7, divisible by 7.

e. Since all the remaining integers are not divisible by any of the previous integers, other than 1, the primes are:

{2,3,5,7,11,15,1719,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89, 97}

The Sieve of Eratosthenes[2]

TABLE 1 The Sieve of Eratosthenes.

Integers divisible by 2 other than 2 receive an underline.

1	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

Integers divisible by 3 other than 3 receive an underline.

1	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

Integers divisible by 5 other than 5 receive an underline.

1	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

Integers divisible by 7 other than 7 receive an underline; integers in color are prime.

1	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

If an integer n is a composite integer, then it has a prime divisor less than or equal to $\sqrt{n}$.

To see this, note that if $n = ab$, then $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$.

Trial division, a very inefficient method of determining if a number n is prime, is to try every integer $i \leq \sqrt{n}$ and see if n is divisible by i.

Infinitude of Primes

Euclid

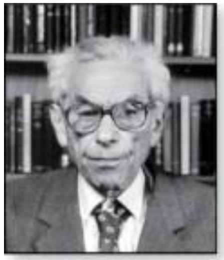
(325 B.C.E. – 265 B.C.E.)

Theorem: There are infinitely many primes. (Euclid)

Proof: Assume finitely many primes: $p_1, p_2, \ldots.., p_n$

- Let $q = p_1 p_2 \cdots p_n + 1$
- Either q is prime or by the fundamental theorem of arithmetic, it is a product of primes.
  - But none of the primes p_j divides q since if $p_j \mid q$, then p_j divides $q - p_1 p_2 \cdots p_n = 1$.
  - Hence, there is a prime not on the list $p_1, p_2, \ldots.., p_n$. It is either q, or if q is composite, it is a prime factor of q. This contradicts the assumption that $p_1, p_2, \ldots.., p_n$ are all the primes.
- Consequently, there are infinitely many primes.

This proof was given by Euclid in *The Elements*. The proof is considered to be one of the most beautiful in all mathematics. It is the first proof in *The Book,* inspired by the famous mathematician Paul Erdős' imagined collection of perfect proofs maintained by God.



Paul Erdős

(1913-1996)

Representing Functions

Marin Mersenne

(1588-1648)

Definition: Prime numbers of the form $2^p - 1$, where p is prime, are called *Mersenne primes*.

- $2^2 - 1 = 3$, $2^3 - 1 = 7$, $2^5 - 1 = 37$, and $2^7 - 1 = 127$ are Mersenne primes.
- $2^{11} - 1 = 2047$ is not a Mersenne prime since $2047 = 23 \cdot 89$.
- There is an efficient test for determining if $2^p - 1$ is prime.
- The largest known prime numbers are Mersenne primes.
- As of mid 2011, 47 Mersenne primes were known, the largest is $2^{43,112,609} - 1$, which has nearly 13 million decimal digits.
- The *Great Internet Mersenne Prime Search* (*GIMPS*) is a distributed computing project to search for new Mersenne Primes.

http://www.mersenne.org/

Distribution of Primes*

Mathematicians have been interested in the distribution of prime numbers among the positive integers. In the nineteenth century, the *prime number theorem* was proved which gives an asymptotic estimate for the number of primes not exceeding x.

Prime Number Theorem: The ratio of the number of primes not exceeding x and $x/\ln x$ approaches 1 as x grows without bound. ($\ln x$ is the natural logarithm of x)

- The theorem tells us that the number of primes not exceeding x, can be approximated by $x/\ln x$.
- The odds that a randomly selected positive integer less than n is prime are approximately $(n/\ln n)/n = 1/\ln n$.

Primes and Arithmetic Progressions (*optional*)*

Euclid's proof that there are infinitely many primes can be easily adapted to show that there are infinitely many primes in the following $4k + 3$, k = 1,2,… (See Exercise 55)

In the 19th century G. Lejuenne Dirichlet showed that every arithmetic progression $ka + b$, $k = 1,2, \ldots$, where a and b have no common factor greater than 1 contains infinitely many primes. (The proof is beyond the scope of the text.)

Are there long arithmetic progressions made up entirely of primes?

- 5,11, 17, 23, 29 is an arithmetic progression of five primes.
- 199, 409, 619, 829, 1039,1249,1459,1669,1879,2089 is an arithmetic progression of ten primes.

In the 1930s, Paul Erdős conjectured that for every positive integer n greater than 1, there is an arithmetic progression of length n made up entirely of primes. This was proven in 2006, by Ben Green and Terrence Tau.

Terence Tao

(Born 1975)

Generating Primes

The problem of generating large primes is of both theoretical and practical interest.

We will see (in Section 4.6) that finding large primes with hundreds of digits is important in cryptography.

So far, no useful closed formula that always produces primes has been found. There is no simple function $f(n)$ such that $f(n)$ is prime for all positive integers n.

But $f(n) = n^2 - n + 41$ is prime for all integers 1,2,…, 40. Because of this, we might conjecture that $f(n)$ is prime for all positive integers n. But $f(41) = 41^2$ is not prime.

More generally, there is no polynomial with integer coefficients such that $f(n)$ is prime for all positive integers n. (See supplementary Exercise 23.)

Fortunately, we can generate large integers which are almost certainly primes. See Chapter 7.

Conjectures about Primes

Even though primes have been studied extensively for centuries, many conjectures about them are unresolved, including:

Goldbach's Conjecture: Every even integer n, $n > 2$, is the sum of two primes. It has been verified by computer for all positive even integers up to $1.6 \cdot 10^{18}$. The conjecture is believed to be true by most mathematicians.

There are infinitely many primes of the form $n^2 + 1$, where n is a positive integer. But it has been shown that there are infinitely many primes of the form $n^2 + 1$, where n is a positive integer or the product of at most two primes.

The Twin Prime Conjecture: The twin prime conjecture is that there are infinitely many pairs of twin primes. Twin primes are pairs of primes that differ by 2. Examples are 3 and 5, 5 and 7, 11 and 13, etc. The current world's record for twin primes (as of mid 2011) consists of numbers $65{,}516{,}468{,}355 \cdot 2^{333{,}333} \pm 1$, which have 100,355 decimal digits.

Greatest Common Divisor 1

Definition: Let *a* and *b* be integers, not both zero. The largest integer *d* such that *d* | *a* and also *d* | *b* is called the greatest common divisor of *a* and *b*. The greatest common divisor of *a* and *b* is denoted by gcd(*a,b*).

One can find greatest common divisors of small numbers by inspection.

Example: What is the greatest common divisor of 24 and 36?

Solution: gcd(24, 36) = 12

Example: What is the greatest common divisor of 17 and 22?

Solution: gcd(17,22) = 1

Greatest Common Divisor 2

Definition: The integers a and b are *relatively prime* if their greatest common divisor is 1.

Example: 17 and 22

Definition: The integers a_1, a_2, …, a_n are *pairwise relatively prime* if $\gcd(a_i, a_j) = 1$ whenever $1 \leq i<j \leq n$.

Example: Determine whether the integers 10, 17 and 21 are pairwise relatively prime.

Solution: Because gcd(10,17) = 1, gcd(10,21) = 1, and gcd(17,21) = 1, 10, 17, and 21 are pairwise relatively prime.

Example: Determine whether the integers 10, 19, and 24 are pairwise relatively prime.

Solution: Because gcd(10,24) = 2, 10, 19, and 24 are not pairwise relatively prime.

Finding the Greatest Common Divisor Using Prime Factorizations

Suppose the prime factorizations of *a* and *b* are:

$$a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}, \qquad b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n},$$

where each exponent is a nonnegative integer, and where all primes occurring in either prime factorization are included in both. Then:

$$\gcd(a,b) = p_1^{\min(a_1,b_1)} p_2^{\min(a_2,b_2)} \ldots p_n^{\min(a_n,bn)},$$

This formula is valid since the integer on the right (of the equals sign) divides both *a* and *b*. No larger integer can divide both *a* and *b*.

Example: $120 = 2^3 \cdot 3 \cdot 5 \quad 500 = 2^2 \cdot 5^3$

$\gcd(120,500) = 2^{\min(3,2)} \cdot 3^{\min(1,0)} \cdot 5^{\min(1,3)} = 2^2 \cdot 3^0 \cdot 5^1 = 20$

Finding the gcd of two positive integers using their prime factorizations is not efficient because there is no efficient algorithm for finding the prime factorization of a positive integer.

Least Common Multiple

Definition: The least common multiple of the positive integers *a* and *b* is the smallest positive integer that is divisible by both *a* and *b*. It is denoted by lcm(*a*,*b*).

The least common multiple can also be computed from the prime factorizations.

$$\operatorname{lcm}(a,b) = p_1^{\max(a_1,b_1)} p_2^{\max(a_2,b_2)} \cdots p_n^{\max(a_n,bn)},$$

This number is divided by both *a* and *b* and no smaller number is divided by *a* and *b*.

Example: $\operatorname{lcm}(2^3 3^5 7^2, 2^4 3^3) = 2^{\max(3,4)} 3^{\max(5,3)} 7^{\max(2,0)} = 2^4 3^5 7^2$

The greatest common divisor and the least common multiple of two integers are related by:

Theorem 5: Let a and b be positive integers. Then

$ab = \gcd(a,b) \cdot \operatorname{lcm}(a,b)$

(*proof is Exercise* 31)

Euclidean Algorithm 1

The Euclidian algorithm is an efficient method for computing the greatest common divisor of two integers. It is based on the idea that gcd(*a*,*b*) is equal to gcd(*a*,*c*) when $a > b$ and *c* is the remainder when a is divided by *b*.

Euclid
(325 B.C.E. – 265 B.C.E.)

Example: Find gcd(91, 287):

- $287 = 91 \cdot 3 + 14$ — Divide 287 by 91
- $91 = 14 \cdot 6 + 7$ — Divide 91 by 14
- $14 = 7 \cdot 2 + 0$ — Divide 14 by 7

Stopping condition

gcd(287, 91) = gcd(91, 14) = gcd(14, 7) = 7

Euclidean Algorithm 2

The Euclidean algorithm expressed in pseudocode is:

procedure *gcd* (*a, b*: positive integers)
  $x := a$
  $y := b$
  while $y \neq 0$
    $r := x$ **mod** y
    $x := y$
    $y := r$
  return x {gcd(a,b) is x}

In Section 5.3, we'll see that the time complexity of the algorithm is O (log b), where a > b.

Correctness of Euclidean Algorithm [1] *

Lemma 1: Let $a = bq + r$, where a, b, q, and r are integers. Then $\gcd(a,b) = \gcd(b,r)$.

Proof:

- Suppose that d divides both a and b. Then d also divides $a - bq = r$ (by Theorem 1 of Section 4.1). Hence, any common divisor of a and b must also be any common divisor of b and r.
- Suppose that d divides both b and r. Then d also divides $bq + r = a$. Hence, any common divisor of a and b must also be a common divisor of b and r.
- Therefore, $\gcd(a,b) = \gcd(b,r)$.

Correctness of Euclidean Algorithm [2] *

Suppose that a and b are positive integers with $a \geq b$.

Let $r_0 = a$ and $r_1 = b$.

Successive applications of the division algorithm yields:

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

Eventually, a remainder of zero occurs in the sequence of terms: $a = r_0 > r_1 > r_2 > \cdots \geq 0$. The sequence can't contain more than a terms.

By Lemma 1

$\gcd(a,b) = \gcd(r_0,r_1) = \cdots = \gcd(r_{n-1},r_n) = \gcd(r_n, 0) = r_n$.

Hence the greatest common divisor is the last nonzero remainder in the sequence of divisions.

gcd's as Linear Combinations

Étienne Bézout

(1730-1783)

Bézout's Theorem: If a and b are positive integers, then there exist integers s and t such that $\gcd(a,b) = sa + tb$.

(*proof in exercises of Section* 5.2)

Definition: If a and b are positive integers, then integers s and t such that $\gcd(a,b) = sa + tb$ are called *Bézout coefficients* of a and b. The equation $\gcd(a,b) = sa + tb$ is called *Bézout's identity.*

By Bézout's Theorem, the gcd of integers a and b can be expressed in the form $sa + tb$ where s and t are integers. This is a *linear combination* with integer coefficients of a and b.

- $\gcd(6,14) = (-2)\cdot 6 + 1\cdot 14$

gcd's as Linear Combinations

Example: Express gcd(252,198) = 18 as a linear combination of 252 and 198.

Solution: First use the Euclidean algorithm to show gcd(252,198) = 18

i. $252 = 1 \cdot 198 + 54$
ii. $198 = 3 \cdot 54 + 36$
iii. $54 = 1 \cdot 36 + 18$
iv. $36 = 2 \cdot 18$

- Now working backwards, from iii and i above
  - $18 = 54 - 1 \cdot 36$
  - $36 = 198 - 3 \cdot 54$
- Substituting the 2nd equation into the 1st yields:
  - $18 = 54 - 1 \cdot (198 - 3 \cdot 54) = 4 \cdot 54 - 1 \cdot 198$
- Substituting $54 = 252 - 1 \cdot 198$ (from i)) yields:
  - $18 = 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 = 4 \cdot 252 - 5 \cdot 198$

This method illustrated above is a two pass method. It first uses the Euclidian algorithm to find the gcd and then works backwards to express the gcd as a linear combination of the original two integers. A one pass method, called the *extended Euclidean algorithm*, is developed in the exercises.

Consequences of Bézout's Theorem*

Lemma 2: If a, b, and c are positive integers such that $\gcd(a, b) = 1$ and $a \mid bc$, then $a \mid c$.

Proof: Assume $\gcd(a, b) = 1$ and $a \mid bc$

- Since $\gcd(a, b) = 1$, by Bézout's Theorem there are integers s and t such that

$$sa + tb = 1.$$

- Multiplying both sides of the equation by c, yields $sac + tbc = c$.
- From Theorem 1 of Section 4.1:
  $a \mid tbc$ (part ii) and a divides $sac + tbc$ since $a \mid sac$ and $a \mid tbc$ (part i)
- We conclude $a \mid c$, since $sac + tbc = c$.

Lemma 3: If p is prime and $p \mid a_1 a_2 \cdots a_n$, then $p \mid a_i$ for some i.

(*proof uses mathematical induction; see Exercise* 64 *of Section* 5.1)

Lemma 3 is crucial in the proof of the uniqueness of prime factorizations.

Uniqueness of Prime Factorization *

We will prove that a prime factorization of a positive integer where the primes are in nondecreasing order is unique. (This part of the fundamental theorem of arithmetic. The other part, which asserts that every positive integer has a prime factorization into primes, will be proved in Section 5.2.)

Proof: (*by contradiction*) Suppose that the positive integer n can be written as a product of primes in two distinct ways:

$$n = p_1 p_2 \cdots p_s \text{ and } n = q_1 q_2 \cdots p_t.$$

- Remove all common primes from the factorizations to get
- By Lemma 3, it follows that divides, for some *k,* contradicting the assumption that and are distinct primes.
- Hence, there can be at most one factorization of n into primes in nondecreasing order.

Dividing Congruences by an Integer*

Dividing both sides of a valid congruence by an integer does not always produce a valid congruence (see Section 4.1).

But dividing by an integer relatively prime to the modulus does produce a valid congruence:

Theorem 7: Let m be a positive integer and let a, b, and c be integers. If $ac \equiv bc \pmod{m}$ and $\gcd(c,m) = 1$, then $a \equiv b \pmod{m}$.

Proof: Since $ac \equiv bc \pmod{m}$, $m \mid ac - bc = c(a - b)$ by Lemma 2 and the fact that $\gcd(c,m) = 1$, it follows that $m \mid a - b$. Hence, $a \equiv b \pmod{m}$.

Solving Congruences

Section 4.4

Section Summary 4

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 [1] *

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$_2$*

Example: Find an inverse of 101 modulo 4620.

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

$42620 = 45 \cdot 101 + 75$

$101 = 1 \cdot 75 + 26$

$75 = 2 \cdot 26 + 23$

$26 = 1 \cdot 23 + 3$

$23 = 7 \cdot 3 + 2$

$3 = 1 \cdot 2 + 1$

$2 = 2 \cdot 1$

Since the last nonzero remainder is 1, gcd(101,4260) = 1

Bézout coefficients : − 35 and 1601

Working Backwards:

$1 = 3 - 1 \cdot 2$

$1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$

$1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$

$1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$

$1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$

$= 26 \cdot 101 - 35 \cdot 75$

$1 = 26 \cdot 101 - 35 \cdot (42620 - 45 \cdot 101)$

$= -35 \cdot 42620 + 1601 \cdot 101$

1601 is an inverse of 101 modulo 42620

Using Inverses to Solve Congruences*

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

Example: What are the solutions of the congruence $3x \equiv 4(\bmod 7)$.

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

$$-2 \cdot 3x \equiv -2 \cdot 4 (\bmod 7).$$

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 $3x \equiv 3 \cdot 6 = 18 \equiv 4(\bmod 7)$ 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,…

The Chinese Remainder Theorem [1]*

In the first century, the Chinese mathematician Sun-Tsu asked:

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?

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* can be used to solve Sun-Tsu's problem.

The Chinese Remainder Theorem [2]*

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

$x \equiv a_1 \pmod{m_1}$

$x \equiv a_2 \pmod{m_2}$

.

.

.

$x \equiv a_n \pmod{m_n}$

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.

The Chinese Remainder Theorem [3]*

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.

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

The Chinese Remainder Theorem 4*

Example: Consider the 3 congruences from Sun-Tsu’s problem:

x ≡ 2 (mod 3), x ≡ 3 (mod 5), x ≡ 2 (mod 7).

- Let m = 3· 5 · 7 = 105, M1 = m/3 = 35, M3 = m/5 = 21, M3 = m/7 = 15.
- We see that
  - 2 is an inverse of M1 = 35 modulo 3 since 35 · 2 ≡ 2 · 2 ≡ 1 (mod 3)
  - 1 is an inverse of M2 = 21 modulo 5 since 21 ≡ 1 (mod 5)
  - 1 is an inverse of M3 = 15 modulo 7 since 15 ≡ 1 (mod 7)
- Hence,
  x = a1M1y1 + a2M2y2 + a3M3y3
  = 2 · 35 · 2 + 3 · 21 · 1 + 2 · 15 · 1 = 233 ≡ 23 (mod 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 $x = 5t + 1$, where t is an integer.

- Substituting into the second congruence yields $5t + 1 \equiv 2 \pmod{6}$.
- Solving this tells us 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 $30u + 26 \equiv 3 \pmod{7}$.
- Solving this congruence tells us 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 $x = 30(7v + 6) + 26 = 210u + 206$.

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

Fermat's Little Theorem

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*)

Pierre de Fermat

(1601-1665)

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** $11 = 5$.

Pseudoprimes[1]*

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 [2]*

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.

Example: The integer 561 is a Carmichael number. To see this:

- 561 is composite, since 561 = 3 · 11 · 13.
- 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

$$b^{560} = \left(b^2\right)^{280} \equiv 1 \pmod 3,$$

$$b^{560} = \left(b^{10}\right)^{56} \equiv 1 \pmod{11},$$

$$b^{560} = \left(b^{16}\right)^{35} \equiv 1 \pmod{17}.$$

- 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*

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

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.

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.

Applications of Congruences

Section 4.5

Section Summary

Hashing Functions

Pseudorandom Numbers

Check Digits

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 $h(k) = k \textbf{ mod } m$, where m is the number of memory locations.
- Because this hashing function is onto, all memory locations are possible.

Example: Let $h(k) = k \textbf{ mod } 111$. 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*:

$h(k,i) = (h(k) + i) \textbf{ mod } m$, 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 [1] *

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 $2 \le a < m$, $0 \le c < m$, $0 \le x_0 < m$.

We generate a sequence of pseudorandom numbers $\{x_n\}$, with $0 \le x_n < m$ for all n, by successively using the recursively defined function $x_{n+1} = (ax_n + c) \ \mathbf{mod} \ m.$

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

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

Pseudorandom Numbers 2*

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

$$x_{n+1} = (7x_n + 4) \textbf{ mod } 9, \text{ with } x_0 = 3.$$

$$\begin{aligned}
x_1 &= 7x_0 + 4 \textbf{ mod } 9 = 7\cdot 3 + 4 \textbf{ mod } 9 = 25 \textbf{ mod } 9 = 7,\\
x_2 &= 7x_1 + 4 \textbf{ mod } 9 = 7\cdot 7 + 4 \textbf{ mod } 9 = 53 \textbf{ mod } 9 = 8,\\
x_3 &= 7x_2 + 4 \textbf{ mod } 9 = 7\cdot 8 + 4 \textbf{ mod } 9 = 60 \textbf{ mod } 9 = 6,\\
x_4 &= 7x_3 + 4 \textbf{ mod } 9 = 7\cdot 6 + 4 \textbf{ mod } 9 = 46 \textbf{ mod } 9 = 1,\\
x_5 &= 7x_4 + 4 \textbf{ mod } 9 = 7\cdot 1 + 4 \textbf{ mod } 9 = 11 \textbf{ mod } 9 = 2,\\
x_6 &= 7x_5 + 4 \textbf{ mod } 9 = 7\cdot 2 + 4 \textbf{ mod } 9 = 18 \textbf{ mod } 9 = 0,\\
x_7 &= 7x_6 + 4 \textbf{ mod } 9 = 7\cdot 0 + 4 \textbf{ mod } 9 = 4 \textbf{ mod } 9 = 4,\\
x_8 &= 7x_7 + 4 \textbf{ mod } 9 = 7\cdot 4 + 4 \textbf{ mod } 9 = 32 \textbf{ mod } 9 = 5,\\
x_9 &= 7x_8 + 4 \textbf{ mod } 9 = 7\cdot 5 + 4 \textbf{ mod } 9 = 39 \textbf{ mod } 9 = 3.
\end{aligned}$$

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. $3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$
$21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$
$98 + x_{12} \equiv 0 \pmod{10}$
$x_{12} \equiv 2 \pmod{10}$ So, the check digit is 2.

b. $3 \cdot 0 + 4 + 3 \cdot 1 + 3 + 3 \cdot 3 + 1 + 3 \cdot 0 + 2 + 3 \cdot 1 + 6 + 3 \cdot 4 + 1 \equiv 0 \pmod{10}$
$0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 = 44 \equiv 4 \not\equiv 0 \pmod{10}$
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 $\sum_{i=1}^{9} i x_i \equiv 0 \pmod{11}$.

a. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?

b. Is 084930149X a valid ISBN10?

X is used for the digit 10.

Solution:

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

b. $1\cdot 0 + 2\cdot 8 + 3\cdot 4 + 4\cdot 9 + 5\cdot 3 + 6\cdot 0 + 7\cdot 1 + 8\cdot 4 + 9\cdot 9 + 10\cdot 10 =$
$0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0 \pmod{11}$

Hence, 084930149X is not a valid ISBN-10.

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*)

Cryptography

Section 4.6

Section Summary [6]

Classical Cryptography

Cryptosystems

Public Key Cryptography

RSA Cryptosystem

Cryptographic Protocols

Primitive Roots and Discrete Logarithms

Caesar Cipher [1]

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A. This process of making a message secret is an example of *encryption*.

Here is how the encryption process works:

- Replace each letter by an integer from $\mathbf{Z}_{26}$, that is an integer from 0 to 25 representing one less than its position in the alphabet.
- The encryption function is $f(p) = (p + 3)$ **mod** 26. It replaces each integer p in the set $\{0,1,2,\ldots,25\}$ by $f(p)$ in the set $\{0,1,2,\ldots,25\}$.
- Replace each integer p by the letter with the position $p + 1$ in the alphabet.

Example: Encrypt the message "MEET YOU IN THE PARK" using the Caesar cipher.

Solution: 12 4 4 19   24 14 20   8 13   19 7 4   15 0 17 10.

Now replace each of these numbers p by $f(p) = (p + 3)$ **mod** 26.

15 7 7 22   1 17 23   11 16   22 10 7   18 3 20 13.

Translating the numbers back to letters produces the encrypted message

"PHHW BRX LQ WKH SDUN."

Caesar Cipher 2

To recover the original message, use $f^{-1}(p) = (p-3)$ **mod** 26. So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters. This process of recovering the original message from the encrypted message is called *decryption*.

The Caesar cipher is one of a family of ciphers called *shift ciphers.* Letters can be shifted by an integer k, with 3 being just one possibility. The encryption function is

$$f(p) = (p + k) \textbf{ mod } 26$$

and the decryption function is

$$f^{-1}(p) = (p-k) \textbf{ mod } 26$$

The integer k is called a *key*.

Shift Cipher [1]

Example 1: Encrypt the message "STOP GLOBAL WARMING" using the shift cipher with $k = 11$.

Solution: Replace each letter with the corresponding element of $\mathbf{Z}_{26}$.

18 19 14 15   6 11 14 1 0 11   22 0 17 12   8 13 6.

Apply the shift $f(p) = (p + 11)$ **mod** 26, yielding

3 4 25 0   17 22 25 12 11 22   7 11 2 23   19 24 17.

Translating the numbers back to letters produces the ciphertext

"DEZA RWZMLW HLCXTYR."

Shift Cipher 2

Example 2: Decrypt the message "LEWLYPLUJL PZ H NYLHA ALHJOLY" that was encrypted using the shift cipher with $k = 7$.

Solution: Replace each letter with the corresponding element of $\mathbf{Z}_{26}$.

11 4 22 11 24 15 11 20 9 1 15 25 7 13 24 11 7 0 0 11 7 9 14 11 24.

Shift each of the numbers by $-k = -7$ modulo 26, yielding

4 23 15 4 17 8 4 13 2 4 8 18 0 6 17 4 0 19 19 4 0 2 7 4 17.

Translating the numbers back to letters produces the decrypted message

"EXPERIENCE IS A GREAT TEACHER."

Affine Ciphers

Shift ciphers are a special case of *affine ciphers* which use functions of the form

$f(p) = (ap + b)$ **mod** 26,

where a and b are integers, chosen so that f is a bijection.

The function is a bijection if and only if gcd(a,26) = 1.

Example: What letter replaces the letter K when the function $f(p) = (7p + 3)$ **mod** 26 is used for encryption.

Solution: Since 10 represents K, $f(10) = (7 \cdot 10 + 3)$ **mod** 26 =21, which is then replaced by V.

To decrypt a message encrypted by a shift cipher, the congruence $c \equiv ap + b$ (mod 26) needs to be solved for p.

- Subtract b from both sides to obtain $c - b \equiv ap$ (mod 26).
- Multiply both sides by the inverse of a modulo 26, which exists since gcd(a,26) = 1.
- $\bar{a}(c - b) \equiv \bar{a}ap$ (mod 26), which simplifies to $\bar{a}(c - b) \equiv p$ (mod 26).
- $p \equiv \bar{a}(c - b)$ (mod 26) is used to determine p in $\mathbf{Z}_{26}$.s

Cryptanalysis of Affine Ciphers

The process of recovering plaintext from ciphertext without knowledge both of the encryption method and the key is known as *cryptanalysis* or *breaking codes*.

An important tool for cryptanalyzing ciphertext produced with a affine ciphers is the relative frequencies of letters. The nine most common letters in the English texts are E 13%, T 9%, A 8%, O 8%, I 7%, N 7%, S 7%, H 6%, and R 6%.

To analyze ciphertext:

- Find the frequency of the letters in the ciphertext.
- Hypothesize that the most frequent letter is produced by encrypting E.
- If the value of the shift from E to the most frequent letter is k, shift the ciphertext by $-k$ and see if it makes sense.
- If not, try T as a hypothesis and continue.

Example: We intercepted the message "ZNK KGXRE HOXJ MKZY ZNK CUXS" that we know was produced by a shift cipher. Let's try to cryptanalyze.

Solution: The most common letter in the ciphertext is K. So perhaps the letters were shifted by 6 since this would then map E to K. Shifting the entire message by −6 gives us "THE EARLY BIRD GETS THE WORM."

Block Ciphers 1

Ciphers that replace each letter of the alphabet by another letter are called *character* or *monoalphabetic* ciphers.

They are vulnerable to cryptanalysis based on letter frequency. *Block ciphers* avoid this problem, by replacing blocks of letters with other blocks of letters.

A simple type of block cipher is called the *transposition cipher*. The key is a permutation σ of the set $\{1,2,\ldots,m\}$, where m is an integer, that is a one-to-one function from $\{1,2,\ldots,m\}$ to itself.

To encrypt a message, split the letters into blocks of size *m,* adding additional letters to fill out the final block. We encrypt $p_1,p_2,\ldots,p_m$ as $c_1,c_2,\ldots,c_m = p_{\sigma(1)},p_{\sigma(2)},\ldots,p_{\sigma(m)}$.

To decrypt the $c_1,c_2,\ldots,c_m$ transpose the letters using the inverse permutation σ^{-1}.

Block Ciphers 2

Example: Using the transposition cipher based on the permutation σ of the set {1,2,3,4} with σ(1) = 3, σ(2) = 1, σ(3) = 4, σ(4) = 2,

a. Encrypt the plaintext PIRATE ATTACK

b. Decrypt the ciphertext message SWUE TRAEOEHS, which was encrypted using the same cipher.

Solution:

a. Split into four blocks PIRA TEAT TACK.
Apply the permutation σ giving IAPR ETTA AKTC.

b. σ^{-1}: $\sigma^{-1}(1) = 2$, $\sigma^{-1}(2) = 4$, $\sigma^{-1}(3) = 1$, $\sigma^{-1}(4) = 3$.

Apply the permutation σ^{-1} giving USEW ATER HOSE.

Split into words to obtain USE WATER HOSE.

Cryptosystems[1]

Definition: A *cryptosystem* is a five-tuple (P,C,K,E,D), where

- P is the set of plaintext strings,
- C is the set of ciphertext strings,
- K is the *keyspace* (set of all possible keys),
- E is the set of encryption functions, and
- D is the set of decryption functions.

The encryption function in E corresponding to the key *k* is denoted by E_k and the description function in D that decrypts cipher text encrypted using E_k is denoted by D_k. Therefore:

$D_k(E_k(p)) = p$, for all plaintext strings *p*.

Cryptosystems₂*

Example: Describe the family of shift ciphers as a cryptosystem.

Solution: Assume the messages are strings consisting of elements in $\mathbf{Z}_{26}$.

- P is the set of strings of elements in $\mathbf{Z}_{26}$,
- C is the set of strings of elements in $\mathbf{Z}_{26}$,
- K = $\mathbf{Z}_{26}$,
- E consists of functions of the form
  $E_k(p) = (p + k)$ **mod** 26 , and
- D is the same as E where $D_k(p) = (p - k)$ **mod** 26 .

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.

The RSA Cryptosystem

Clifford Cocks
(Born 1950)

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 (200 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 400 digits, cannot be factored in a reasonable length of time.

RSA Encryption

To encrypt a message using RSA using a key (n,e) :

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 $C = M^e$ **mod** 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 $C = M^{13}$ **mod** 2537.
- Since 1819^{13} mod 2537 = 2081 and 1415^{13} mod 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, we can decrypt each block with the computation $M = C^d$ **mod** $p\cdot q$. (*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, $M = C^{937}$ **mod** 2537.
- Since 0981^{937} **mod** 2537 = 0704 and 0461^{937} **mod** 2537 = 1115, the decrypted message is 0704 1115. Translating back to English letters, the message is HELP.

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 protocol* 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[1]*

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 $E_{(n,e)}(x) = x^d \textbf{ mod } n$. She decrypts a ciphertext message y using $D_{(n,e)}(y) = y^d \textbf{ mod } n$.

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.

Cryptographic Protocols: Digital Signatures [2]*

Example: Suppose Alice's RSA cryptosystem is the same as in the earlier example with key(2537,13), 2537 = 43· 59, *p* = 43 and *q* = 59 are primes and gcd(*e*,(*p*−1)(*q* −1)) = gcd(13, 42· 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 $D_{(2537,13)}(x) = x^{937}$ **mod** 2537 to each block.
2. She finds (using her laptop, programming skills, and knowledge of discrete mathematics) that 1204^{937} **mod** 2537 = 817, 419^{937} **mod** 2537 = 555 , 19^{937} **mod** 2537 = 1310, 1314^{937} **mod** 2537 = 2173, and 1413^{937} **mod** 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.