SCT211-0002/2020
MWAURA DAVID NJUNG'E
ICS2411 : CRYPTOGRAPHY

NUMBER THEORY AND CRYPTOGRAPHY

Recap

- **Number Theory** is the part of mathematics devoted to the study of the set of integers and their properties.
- We will first introduce the notion of divisibility of integers, which we use to introduce modular, or clock, arithmetic.
- Integers can be represented with any positive integer b greater than 1 as a base.
- Here, we will look at base b representations of integers and give an algorithm for finding them.
- Additionally, we will look at prime numbers, how to solve linear congruences, pseudoprimes and other applications of the numbers' theory

4.1 Divisibility and Modular Arithmetic

Divisibility: An integer a divides b (written as a|ba | ba|b) if there exists an integer c such that b=ac

Basic Properties of Divisibility as given in Theorem 1:

i. If a|b and a|c, then a|(b+c).
ii. If a|b, then a|bc for any integer c.
iii. If a|b and b|cb , then a|c.

Proof for (i): . Suppose that a | b and a | c. Then, from the definition of divisibility, it follows that there are integers s and t with b = as and c = at. Hence : **b + c = as + at = a(s + t)**
Therefore, a divides b + c. This establishes part (i) of the theorem.

The Division Algorithm - Let a be an integer and d a positive integer. Then there are unique integers q and r, with 0 ≤ r < d, such that a = dq + r.

$$a = dq + r, \quad 0 \leq r < d$$

Modular Arithmetic

- In some situations, we care only about the remainder of an integer when it is divided by some specified positive integer.
- The notation a mod m is used to represent the remainder when an integer a is divided by the positive integer m
- **Congruences**: 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. We use the notation $a \equiv b \pmod{m}$ to indicate 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 (plural moduli). If a and b are not congruent modulo m, we write $a \not\equiv b \pmod{m}$.
- Although both notations $a \equiv b \pmod{m}$ and a mod m = b include "mod," they represent different concepts. The first represents a relation on the set of integers, whereas the second represents a function. However, the relation $a \equiv b \pmod{m}$ and the mod m function are closely related, as described in Theorem 3
- **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.
- Theorem 4 further provides a useful way to work with 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 $a \equiv b \pmod{m}$, by the definition of congruence (Definition 3), we know that $m \mid (a - b)$. This means that there is an integer k such that a − b = km, so that a = b + km. Conversely, if there is an integer k such that a = b + km, then km = a − b. Hence, m divides a − b, so that $a \equiv b \pmod{m}$.
- The sett of all integers congruent to an integer a modulo m is called the **congruence class.**

Modular Arithmetic

- Theorem 5 shows that additions and multiplications preserve congruences.
- **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:** We use a direct 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 $d = c + tm$. Hence, $b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$

and

$$bd = (a + sm)(c + tm) = ac + m(at + cs + stm)$$

Hence,

$$a + c \equiv b + d \pmod{m} \text{ and } ac \equiv bd \pmod{m}$$

Arithmetic Modulo m

- We can define arithmetic operations on Zm, the set of nonnegative integers less than m, that is, the set {0, 1, … , m − 1}.
- In particular, we define addition of these integers, denoted by:

  $+_m$ **by** $a +_m b = (a + b) \bmod m$,

  where the addition on the right-hand side of this equation is the ordinary addition of integers, and we define multiplication of these integers, denoted by $\cdot_m$ by

  $a \cdot_m b = (a \cdot b) \bmod m$,

  where the multiplication on the right-hand side of this equation is the ordinary multiplication of integers
- The operations $+_m$ and $\cdot_m$ are called addition and multiplication modulo m and when we use these operations, we are said to be doing **arithmetic modulo** m

4.2 Integer Representations and Algorithms

1. Number Bases and Representations

- Integers can be expressed using any integer greater than one as a base.
- Common bases in computing:
  - Decimal (Base 10) – used in daily life.
  - Binary (Base 2) – used in computer arithmetic.
  - Octal (Base 8) – compact representation of binary.
  - Hexadecimal (Base 16) – commonly used for memory addresses and color codes.
- Given a base b and an integer n, we will show how to construct the base b representation of this integer. We will also explain how to quickly convert between binary and octal and between binary and hexadecimal notations

2. Representations of Integers

- We use decimal notations to express integers.
- In decimal notation, an integer n is written as a sum of the form

  $$a_k 10^k + a_{k-1} 10^{k-1} + \cdots + a_1 10 + a_0,$$

  where a_j is an integer with $0 \le a_j \le 9$ for $j = 0, 1, \ldots, k$.
- Ie; , 965 is used to denote $9 \cdot 102 + 6 \cdot 10 + 5$
- computers usually use binary notation (with 2 as the base) when carrying out arithmetic, and octal (base 8) or hexadecimal (base 16) notation when expressing characters, such as letters or digits.

Binary Expansions - Choosing 2 as the base gives binary expansions of integers. Each digit is either a 0 or 1

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

Solution: We have $(1\ 0101\ 1111)^2 = 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 351$

Octal and Hexadecimal Expansions - Base 8 expansions are called octal expansions and base 16 expansions are hexadecimal expansions.

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

Solution: Using the definition of a base b expansion with b = 8 tells us that

$(7016)_8 = 7 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8 + 6 = 3598.$

- Sixteen different digits are required for hexadecimal expansions. Usually, the hexadecimal digits used are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F, where the letters A through F represent the digits corresponding to the numbers 10 through 15 (in decimal notation)

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

Solution: : Using the definition of a base b expansion with b = 16 tells us that

$(2AE0B)_{16} = 2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16 + 11 = 175627$

Base Conversion – Uses an algorithm for constructing the base b expansion of an integer n by first dividing n by b to obtain a quotient and a remainder

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

Solution: First, divide 12345 by 8 to obtain $12345 = 8 \cdot 1543 + 1$. Successively dividing quotients by 8 gives successive remainders which are then arrange from the bottom, left to right. Hence:

the left of 12345 in base 8. Hence, $(12345)_{10} = (30071)_{8.}$

3. Conversions Between Common Bases

- Binary to Octal: Group binary digits into triplets (3 bits per octal digit).
- Binary to Hexadecimal: Group binary digits into quadruplets (4 bits per hex digit).
- Octal to Binary: Convert each octal digit to its 3-bit binary equivalent.
- Hexadecimal to Binary: Convert each hex digit to its 4-bit binary equivalent.

Algorithm for Base Conversion (Pseudocode)

```
procedure base_b_expansion(n, b: positive integers with b > 1)
    q := n
    k := 0
    while q ≠ 0
        ak := q mod b
        q := q div b
        k := k + 1
    return (ak-1, ..., a1, a0)
```

This is a **greedy algorithm**, as it picks the largest possible digit at each step.

4.3 Primes and Greatest Common Divisors

- A prime number is an integer greater than 1 that is divisible only by 1 and itself.
- Composite numbers are integers greater than 1 that are not prime.
- Every positive integer greater than 1 can be uniquely factored as a product of primes in non-decreasing order.
- **Trial Division**: A method for checking primality by dividing the number by all primes up to its square root.
- **Euclidean Algorithm**: This is an efficient method to compute the greatest common divisor (GCD) of two integers.
- **The Sieve of Eratosthenes-**A systematic technique to find all prime numbers up to a specified integer by eliminating multiples of each prime.
- **Infinitude of Primes-**It has been proven since ancient times that there are infinitely many primes.
- **The distribution of primes** among integers approximates x/ln x, where x is a large number and ln x is the natural logarithm of x.
- **Dirichlet's Theorem on Arithmetic Progressions-**There are infinitely many primes in any arithmetic progression of the form ak+b where a and b are integers with no common factors greater than 1.

GCD AND LCM

Greatest common Divisor

- The largest integer that divides both of two integers is called the greatest common divisor of these integers.
- Let a and b be integers, not both zero. The largest integer d such that d | a and d | b is called the greatest common divisor of a and b. The greatest common divisor of a and b is denoted by gcd(a, b).
- One way to find the greatest common divisor of two integers is to find all the positive common divisors of both integers and then take the largest divisor

Least Common Multiple

- It is the smallest positive integer that is divisible by both a and b, is denoted by lcm(a, b)
- The least common multiple exists because the set of integers divisible by both a and b is nonempty (because ab belongs to this set, for instance), and every nonempty set of positive integers has a least element (by the well-ordering property.
- The LCM of a and b is given by

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

  where max(x, y) denotes the maximum of the two numbers x and y.
- This formula is valid because a common multiple of a and b has at least max(a_i , b_i) factors of pi in its prime factorization, and the least common multiple has no other prime factors besides those in a and b.

4.4 Solving Congruences

1.Linear Congruences

- A linear congruence of the form: ax ≡ b (mod m) where m is a positive integer, a and b are integers, and x is a variable, is called a linear congruence
- We solve a linear congruence by finding all integers x that satisfy the congruence.
- One method that we will describe uses an integer a such that aa ≡ 1 (mod m), if such an integer exists. Such an integer a is said to be an inverse of a modulo m. Theorem 1 guarantees that an inverse of a modulo m exists whenever a and m are relatively prime

2.**Finding Modular Inverses**

- If gcd (a,m) =1, an inverse of a modulo m exists.
- The modular inverse of a (denoted a^{-1})satisfies:

$$a \cdot a^{-1} \equiv 1 \pmod{m}$$

The inverse can be found using the **Extended Euclidean Algorithm**.

3.**Solving Linear Congruences**

- Multiply both sides by a^{-1} to find x.
- The general solution takes the form: $x \equiv x_0 \pmod{m}$

Where x_0 is a particular solution

4.Fermat's Little Theorem

- If p is prime and p∤a, then:

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

Used in modular arithmetic and cryptography

4.5 Applications of Congruences

- Used in hashing functions for efficient memory allocation.
- Generate pseudorandom numbers for simulations and algorithms.
- Compute check digits to verify the correctness of identification numbers.

1.Hashing Functions

- Hashing maps large keys (e.g., Social Security numbers) to smaller memory locations.
- Common hashing function: h (k)=k mod .
- Collisions occur when multiple keys map to the same location and are resolved using techniques like linear probing.

2.Pseudorandom Number Generation

Uses the **linear congruential method:**

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

3.Check Digits for Error Detection

- Extra digits (parity check bits) are added to ensure data integrity.
- Parity check bits detect errors but cannot always correct them.
- Used in barcodes, ISBNs, and other identification systems.

4.6 Cryptography

Private Key Cryptography:

- Traditional ciphers like shift and affine ciphers use private keys, where encryption and decryption keys are the same or easily derived.
- Secure communication requires sharing the secret key, making key distribution a challenge.
- AES is a modern, highly secure private key cryptosystem.

Public Key Cryptography:

- Introduced in the 1970s to eliminate the need for key sharing.
- The encryption key is public, while only the recipient has the private decryption key.
- Breaking encryption without the private key requires immense computational effort.

RSA Cryptosystem:

- Invented in 1976 by Ronald Rivest, Adi Shamir, and Leonard Adleman, though Clifford Cocks had secretly discovered it earlier at GCHQ in 1973.
- Based on the difficulty of factoring large numbers (product of two large primes).
- Each user has a public key (n, e) and a private key (d).

RSA Encryption Process

RSA Encryption Process:

- Convert plaintext letters into numerical equivalents.
- Encrypt using the formula:

$$C = M^e \mod n$$

- Produces ciphertext blocks.

RSA Decryption Process:

- Requires the private key ddd, which is the modular inverse of e mod (p−1)(q−1).
- Decrypt using:

$$M = C^d \mod n$$

- Ensures only the intended recipient can recover the plaintext.

Security of RSA:

- Based on the difficulty of factoring large numbers.
- Practical RSA keys use 200+ digit primes, making brute-force attacks infeasible.
- Fermat's Little Theorem and the Chinese Remainder Theorem help ensure correct decryption.

THE END