Introduction to mathematical concepts

Cryptography summer course

DACLab

August 13, 2023

Department of Knowledge Engineering
VNUHCM - University of Science
{ndhy, nvqhuy}@hcmus.edu.vn

Table of contents

Introduction

Introduction to Cryptography

- Overview:
  - Cryptography is the practice and study of secure communication techniques.
  - It involves encryption, which converts plaintext into ciphertext, and decryption, the reverse process.
- Importance of Mathematics:
  - Mathematics forms the foundation of modern cryptography.
  - Complex mathematical problems are at the core of cryptographic algorithms.
- Secure Communication:
  - Cryptography enables secure communication over insecure channels.
  - Confidentiality, integrity, authentication, and non-repudiation are achieved through cryptographic protocols.

Introduction to Cryptography

- Data Protection:
  - Cryptography safeguards sensitive data from unauthorized access.
  - Applications include protecting passwords, financial information, and personal data.
- Future of Cryptography:
  - As technology evolves, cryptography must adapt to maintain security.
  - Quantum cryptography shows promise in providing unbreakable encryption.

Key concepts of modern cryptographic algorithms

- **Prime Numbers:** Prime numbers are fundamental for cryptography, as security depends on factoring large composites into primes.
- **Modular Arithmetic:** Modular arithmetic simplifies operations in encryption by handling remainders from integer divisions, preventing overflow.
- **One-Way Hash Functions:** One-way hash functions generate fixed-size outputs from any input, easy to compute but hard to reverse.
- **Randomness:** Crucial for strong cryptography, generates unpredictable keys and initialization vectors.

Prime Numbers and Factorization

Factorization

- Factorization is a fundamental mathematical concept with applications in various fields.
- Expressing a number or mathematical object as a product of smaller, simpler factors of the same kind:
  - Simple Factoring: Finding 2 numbers that multiply to form another number - like $32 = 4 \times 8$.
  - Prime Factorization: Breaking numbers down to their prime components - e.g., $81 = 3 \times 3 \times 3 \times 3$.
  - Greatest Common Factor (GCF): Simplifying expressions by factoring out the common factor - as in $2x + 10 = 2(x + 5)$.
  - Exercies: Factoring complex expressions like $x^2 - 14x - 32, 15x^2 - 26x + 11, or 150x^3 + 350x^2 + 180x + 420$.

Prime Numbers

- A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself: $\{2, 3, 5, 7, 11, 13, \ldots\}$
- Primality Testing
  - Primality testing involves determining whether a given number is prime or composite.
  - Efficient algorithms like Miller-Rabin and AKS have been developed for accurate and low-computational primality testing.
- Prime Generation
  - Prime number generation methods include the Sieve of Eratosthenes for efficient generation up to a limit and randomized approaches like Baillie-PSW for generating random probable primes.
  - These techniques are essential in cryptography for generating secure encryption keys and prime-based algorithms.

Modular Arithmetic & Modular Exponentiation

Modular Arithmetic

- In modular arithmetic, we pick a number n as the 'modulus', and our numbers range from 0 to $n-1$. Numbers wrap around after reaching n for sensible arithmetic.
- Example: With modulus 5, we work in $\{0, 1, 2, 3, 4\}$. We get $2+1=3$ and $2+2=4$. However, $2+3=5$ wraps to 0, and $2+4=6$ wraps to 1.
- Think of a clock: 1 o'clock to 12 o'clock, then back to 1 o'clock. This is like a modulus of 12, using $\{1, 2, ..., 12\}$. Yet, $\{0, 1, ..., 11\}$ is the same, as 0 and 12 wrap around similarly.
- Example: Assume the current time is $2:00$ p.m. Write this as $14:00$. Sixty five hours later, it would be $79:00$. Since $79 = 243 + 7$, it will be $7:00$ or 7 a.m
- As you can see, the modulo n arithmetic maps all integers into the set $\{0, 1, 2, 3, ..., n1\}$

Modular Arithmetic

Definition: Congruent modulo

Let $n \geq 2$ be a fixed integer. We say the two integers m_1 and m_2 are congruent modulo, denoted

$$m_1 \equiv m_2 \mod n$$

if and only if $n | (m_1 - m_2)$. The integer n is called the modulus of the congruence.

Theorem

Let $n \geq 2$ be a fixed integer. For any two integers m_1 and m_2

$$m_1 \equiv m_2 \mod n \Leftrightarrow m_1 \mod n = m_2 \mod n.$$

Corollary

Let $n \geq 2$ be a fixed integer. Then

$$a \equiv 0 \mod n \Leftrightarrow n | a.$$

Modular Arithmetic

Theorem

Let $n \geq 2$ be a fixed integer. If $a \equiv b \mod n$ and $c \equiv d \mod n$, then

$$a + c \equiv b + d \qquad \mod n$$

$$ac \equiv bd \qquad \mod n$$

Proof: Assume $a \equiv b \mod n$ and $c \equiv d \mod n$. Then $n|(a - b)$ and $n|(c - d)$. We can write $a - b = ns$, and$c - d = nt$ for some integers s and t. Consequently,

$$(a + c) - (b + d) = (a - b) + (c - d) = ns + nt = n(s + t).$$

where $s + t$ is an integer. This proves that $a + c \equiv b + d \mod n$. We also have

$$ac - bd = (b + ns)(d + nt) - bd = bnt + nsd + n^2st = n(bt + sd + nst).$$

where $bt + sd + nst$ is an integer. Thus, $n|(acbd)$, which means $ac \equiv bd \mod n$.

Modular Arithmetic

Properties of addition in modular arithmetic:

- If $a + b = c$, then $a \mod n + b \mod n \equiv c \mod n$
- If $a \equiv b \mod n$, then $a + k \equiv b + k \mod n$ for any integer k
- If $a \equiv b \mod n$ and $c \equiv d \mod n$, then $a + c \equiv b + d \mod n$
- If $a \equiv b \mod n$, then $-a \equiv -b \mod n$

Properties of modular multiplication:

- If $a \cdot b = c$, then $(a \mod n) \cdot (b \mod n) \equiv c \mod n$
- If $a \equiv b \mod n$, then $k \cdot a \equiv k \cdot b \mod n$ for any integer k
- If $a \equiv b \mod n$ and $c \equiv d \mod n$, then $a \cdot c \equiv b \cdot d \mod n$

Modular Exponentiation

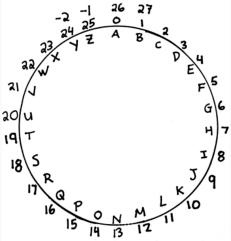
- **Modular exponentiation** is the process of repeatedly squaring and reducing a number modulo some integer, and then combining the results to find the required answer.

<u>Theorem</u>

If $a \equiv b \mod n$, then $a^k \equiv b^k \mod n$ for any positive integer k.

How do Modular Arithmetic and Caesar Ciphers relate?

- Since there are 26 letters in the English alphabet, let’s relate the letters a-z by numbers 0-25 as shown by the diagram below.



- Notice going from “a” to “D” was a shift of 3 letters over. Thus we can encrypt the word “pumpkin” by relating “p” with 15 on the wheel, adding 3 to get 18, and then we turn this back into a letter, which gives us “S”. Similarly “u” $\rightarrow$ 20 $\rightarrow$ 23 $\rightarrow$ X.

Example

Greatest Common Divisor and Euclidean Algorithm

Greatest Common Divisor and Euclidean Algorithm

Divisor

A divisor b of an integer a, denoted by $b \mid a$, is an integer satisfying $a = bq$ where q is an integer.

Example: All divisors of 12 *are* $\pm 1, \pm 2, \pm 3, \pm 4, \pm 6,$ *and* ± 12.

Common Divisor

An integer n is a commmon divisor of a and b iff $n \mid a$ and $n \mid b$.

Example: 3 *is a common divisor of* 12 *and* 18.

Greatest Common Divisor

An integer n is the greatest commmon divisor (GCD) of a and b iff there is no common divisor of a and b larger than n.

Then, n is denoted by $gcd(a, b)$.

Example: $gcd(12, 18) = 6$.

Greatest Common Divisor and Euclidean Algorithm

Theorem (Bézout's Lemma)

Let a and b be integers with the integer $d = gcd(a, b)$. Then, there exists integers x and y such that $ax + by = d$. Moreover, d is a divisor of $ax + by$ for all integers x and y.

Example: $gcd(12, 18) = 6 = 12 \times (-1) + 18 \times 1$.

Remarks

- The equation $ax + b = 0 \pmod{n}$ has a solution iff $gcd(a, n) \mid b$. Therefore, if $gcd(a, n) = 1$, the equation always has a solution.
- If a and n are coprime, i.e. $gcd(a, n) = 1$, there exists an integer b satisfying $ab = 1 \pmod{n}$. Then, b is called a modular multiplicative inverse of a and denoted by a^{-1}.
- With the prime number p, all integers a satisfying $p \nmid a$ are coprime with p. That means we can design modular operations like addition, subtraction, multiplication, division, and exponentiation.
  $\rightarrow$ The foundation of the cryptographic theory.

Greatest Common Divisor and Euclidean Algorithm

Theorem (Euclidean Algorithm)

Let a, b, and k be integers. Then, $gcd(a, b) = gcd(a + bk, b)$.

Euclidean Algorithm

Let a and b be positive integers. Find $gcd(a, b)$?

1. Loop until $b = 0$.
   1.1. $r \leftarrow a \mod b$.
   Note: r is an integer satisfying $0 \leq r < b$ and $b \mid (a - r)$.
   1.2. $(a, b) \leftarrow (b, r)$.
2. Return a.

Remark: By Euclidean Algorithm, we can find $gcd(a, b)$, but how can we find integers x and y satisfying $ax + by = gcd(a, b)$?

Greatest Common Divisor and Euclidean Algorithm

Theorem (Extended Euclidean Algorithm)

Let $r_1 = ax_1 + by_1$, $r_2 = ax_2 + by_2$, and $r = r_1 + kr_2$. Then, $r = ax + by$ where $x = x_1 + kx_2$ and $y = y_1 + ky_2$.

Extended Euclidean Algorithm

Let a and b be positive integers. Find d, x, and y such that $d = gcd(a, b) = ax + by$?

1. $(s_1, t_1, s_2, t_2) \leftarrow (1, 0, 0, 1)$.
2. Loop until $b = 0$.
   2.1. $r \leftarrow a \mod b$ and $q \leftarrow (a - r)/b$.
   Note: r is an integer satisfying $0 \leq r < b$ and $b \mid (a - r)$.
   2.2. $(a, b, s_1, t_1, s_2, t_2) \leftarrow (b, r, s_2, t_2, s_1 - qs_2, t_1 - qt_2)$.
3. Return (a, s_1, t_1).

Greatest Common Divisor and Euclidean Algorithm

Example (Extended Euclidean Algorithm)

Let $a = 12$ and $b = 18$. Find d, x, and y such that $d = gcd(x, y) = ax + by$?

The following table shows values of the variables after step 2.1 and before step 2.2:

a	b	s_1	t_1	s_2	t_2	q	r
12	18	1	0	0	1	0	12
18	12	0	1	1	0	1	6
12	6	1	0	-1	1	2	0

Therefore, $d = 6$, $x = -1$, and $y = 1$.

Remarks (Extended Euclidean Algorithm)

- The complexity of the algorithm is $O(\log(\min(a, b)))$. Thus, it is still considered efficient even with big integers.
- Bézout's Lemma shows the existence of modular multiplicative inverses, and Extended Euclidean Algorithm provides a method to find them. That is a crucial theoretical and practical basis to implement cryptosystems. This topic will be revisited in the RSA section.

Chinese Remainder Theorem

Chinese Remainder Theorem

<u>Chinese Remainder Theorem (CRT):</u> Let $n_1, \ldots, n_k$ be positive integers which are pairwise coprime with $N = n_1 \ldots n_k$, and $a_1, \ldots, a_k$ be integers. Then the following system of equations

$$\begin{cases} x = a_1 \pmod{n_1} \\ \vdots \\ x = a_k \pmod{n_k} \end{cases}$$

has a solution. Moverover, any two solutions x_1 and x_2 are congruent modulo N, i.e. $x_1 = x_2 \pmod{N}$.

Chinese Remainder Theorem

Example (CRT)

Consider the following system of equations:

$$\begin{cases} x = 7 \pmod{11} \\ x = 2 \pmod{13} \end{cases}$$

$$\iff \begin{cases} x = 7 - 4 \times 11 \pmod{11} \\ x = 2 - 3 \times 13 \pmod{13} \end{cases}$$

$$\iff \begin{cases} x = -37 \pmod{11} \\ x = -37 \pmod{13} \end{cases}$$

$$\iff x = -37 \pmod{11 \times 13} \iff x = -37 \pmod{143}$$

Therefore, all solutions $x = -37 \pmod{143}$.

Chinese Remainder Theorem

Remarks (CRT)

- A solution of the system of equations is $a_1p_1q_1 + \ldots a_kp_kq_k$ where $p_i = \frac{N}{n_i}$ and $q_i = p_i^{-1} \pmod{n_i}$. Thus, if all n_i are unchanged, then we can precalculate all p_k and q_k, so as soon as all a_i is determined we can immediately find the solution.
- The performance of the operations can be improved by two ways:
  - Because all n_i are usually much smaller than N, the calculations for each n_i are much simpler and efficient. After that, we can instantly get the final result by precalculated p_i and q_i of CRT.
  - The calculations for each n_i are independent from each other. Therefore, parallel implementation is possible to utilize the full potential of hardware processing.

Applications (CRT)

- Speed up the signing process of certificates and the decryption process of cryptosystems. For example, CRT is applied in the standard implementation of public-key cryptography based on RSA [1].
- Secret sharing: A secret is only recovered when at least any k people of a group join to decrypt together. For example, each person keep a piece of the secret which is a solution of a congruence equation, and to find the original secret, we need to solve a systems of equations. Therefore, CRT is essential to do that.

[1] https://www.rfc-editor.org/rfc/rfc8017.txt

Review & Exercises

Exercises