Course Name:
"CRYPTOGRAPHY"
Course Code: B031009T

C.L. Jain College, Firozabad
(Affili. to Dr. B.R. Ambedkar University, Agra)

Polynomial and modular arithmetic

Polynomial arithmetic plays a crucial role in various cryptographic algorithms, particularly in error-correcting codes and cryptographic hash functions.
Error-Correcting Codes: Polynomial arithmetic is used in the encoding and decoding processes of error-correcting codes, such as Reed-Solomon codes and BCH codes. These codes are utilized in data storage systems, communication systems, and digital media to detect and correct errors that may occur during transmission or storage.
Cryptographic Hash Functions: Polynomial arithmetic is also employed in the implementation of cryptographic hash functions. These hash functions take an input message and produce a fixed-size output, called a hash value or hash digest. Polynomial arithmetic operations, such as addition and multiplication modulo a prime number, are often used in the internal compression functions of cryptographic hash algorithms like SHA-1 and SHA-256.

Modular arithmetic is a fundamental concept in cryptography and is used extensively in various cryptographic algorithms and protocols.
RSA Cryptosystem: In the RSA cryptosystem, modular arithmetic plays a central role in both key generation and encryption/decryption operations. The security of RSA relies on the difficulty of factoring large semiprime numbers, which involves modular arithmetic operations such as modular exponentiation.
Diffie-Hellman Key Exchange: Modular arithmetic is utilized in the Diffie-Hellman key exchange protocol, which allows two parties to agree upon a shared secret key over an insecure channel. The security of Diffie-Hellman relies on the discrete logarithm problem, which involves modular exponentiation in a finite field.
Elliptic Curve Cryptography (ECC): Modular arithmetic is a fundamental operation in ECC, a cryptographic approach based on the algebraic structure of elliptic curves over finite fields. Modular arithmetic operations are used in point addition, scalar multiplication, and other operations within the ECC cryptographic algorithms.

Introduction to finite field of the form GF(p) and GF(2n)

Finite fields, also known as Galois fields, are algebraic structures used extensively in various areas of mathematics, computer science, and cryptography. They provide a finite set of elements along with operations that mimic those of the familiar arithmetic operations on integers.

Finite Field of the Form GF(p):

A finite field of the form $GF(p)$, where p is a prime number, is denoted as $GF(p)$ or Z_p . Elements of $GF(p)$ are integers modulo p, ranging from 0 to $p-1$. The arithmetic operations in $GF(p)$ include addition, subtraction, multiplication, and division (multiplicative inverse) modulo p. Addition and multiplication in $GF(p)$ follow the usual rules of arithmetic, but the result is reduced modulo p to remain within the range of 0 to $p-1$

For example, if $p = 7$, the elements of $GF(7)$ are $\{0, 1, 2, 3, 4, 5, 6\}$. Addition and multiplication in $GF(7)$ are performed modulo 7:

$$3 + 5 \equiv 1 (mod 7)$$

$$3 \times 5 \equiv 1 (mod 7)$$

Finite Field of the Form GF(2^n) :

A finite field of the form $GF(2^n)$ is denoted as $GF(2^n)$. Elements of $GF(2^n)$ are binary polynomials of degree less than n with coefficients in the field of two elements $GF(2)$, also known as the binary field). The arithmetic operations in $GF(2^n)$ are performed modulo an irreducible binary polynomial of degree n. Addition in $GF(2^n)$ is performed by XORing the coefficients of the polynomials, and multiplication is performed using polynomial multiplication modulo the irreducible polynomial.

For example, if $n = 3$, the elements of $GF(2^3)$ are binary polynomials of degree less than 3 with coefficients in $GF(2)$, such as $\{0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$. Addition and multiplication in $GF(2^3)$ are performed modulo an irreducible polynomial, such as $x^3 + x + 1$.

Fermat theorem

Fermat's Theorem, also known as Fermat's Little Theorem, is a fundamental result in number theory named after the French mathematician Pierre de Fermat. It provides a method for determining whether a given integer is prime and is often used in modular arithmetic and cryptography.

Fermat's Theorem states:

If p is a prime number and a is any integer not divisible by p, then

$$a^{p-1} \equiv 1(mod p)$$

In other words, if you raise any integer a (not divisible by prime p) to the power of $p-1$ and compute the result modulo p, the remainder will be 1.

Fermat's Theorem is a crucial result in number theory with applications in various fields, including cryptography, where it is used in primality testing and the development of cryptographic algorithms such as RSA.

Example:

Let's consider the prime number $p = 7$ and an integer $a = 3$ (not divisible by 7).
According to Fermat's Theorem:

$$3^{7-1} \equiv 3^6 \equiv 1(mod 7)$$

To verify this result, let's calculate 3^6
and find its remainder when divided by 7:

$$3^6 = 729$$

$$729 \div 7 = 104 remainder 1$$

Since the remainder is 1 when 3^6 is divided by 7, Fermat's Theorem holds true for this example.

Euler's theorem

Euler's Theorem states:

If a and n are coprime positive integers (i.e., they have no common factors other than 1), then:

$$a^{\phi(n)} \equiv 1(mod n)$$

$\phi(n)$ denotes Euler's totient function, which represents the number of positive integers less than or equal to n that are coprime to n.

Euler's Theorem is a generalization of Fermat's Little Theorem and has numerous applications in number theory, cryptography, and computer science. It forms the basis for various cryptographic algorithms, including RSA (Rivest-Shamir-Adleman), which is widely used for secure communication and digital signatures.

Chinese remainder theorem

The Chinese Remainder Theorem (CRT) is a fundamental theorem in number theory with wide applications in mathematics and computer science, including cryptography, error-correcting codes, and computer algorithms. The theorem is named after the Chinese mathematician Sunzi (Sun Tzu), who described it in the 3rd century AD.
The Chinese Remainder Theorem states that if we have a system of simultaneous congruences (i.e., equations involving remainders when divided by certain integers), then there exists a unique solution modulo the product of the moduli of the congruences, under certain conditions.

More formally, let $n_1, n_2, ..., n_k$ be pairwise coprime positive integers (i.e., they have no common factors other than 1), and let $a_1, a_2, ..., a_k$ be arbitrary integers. Then, the system of congruences:

$$x \equiv a_1 (mod n_1)$$

$$x \equiv a_2 (mod n_2)$$

$$...$$

$$x \equiv a_3 (mod n_3)$$

has a unique solution modulo $N = n_1 \cdot n_2 \cdot ... \cdot n_k$, where N is the product of the moduli n_i. Moreover, if $N_i = N/n_i$ and if b_i is the modular multiplicative inverse of N_i modulo n_i, then the solution to the system of congruences is given by:

$$x \equiv \sum_{i=1}^{k} a_i b_i N_i (mod N)$$

The Chinese Remainder Theorem is useful in various applications, including solving systems of linear congruences, simplifying computations in modular arithmetic, and breaking down large computations into smaller, more manageable parts. In cryptography, the theorem is used in various algorithms, such as RSA and Shamir's Secret Sharing Scheme, for efficient key generation, encryption, and decryption.
Overall, the Chinese Remainder Theorem provides a powerful tool for solving congruence problems and has widespread applications in mathematics and computer science.

Discrete logarithm

The discrete logarithm problem is a fundamental problem in mathematics and computer science, particularly in the field of cryptography. It is the inverse operation of exponentiation in modular arithmetic and can be stated as follows:

Given a prime number p, an integer base g, and an integer residue y, find the integer exponent x such that $g^x \equiv ymod(p)$

In other words, the discrete logarithm problem asks to determine the exponent x when the base g, the prime modulus p, and the residue y are known. This problem is considered difficult to solve efficiently for large prime numbers and is the foundation of several cryptographic algorithms, including Diffie-Hellman key exchange and the ElGamal encryption scheme. The difficulty of solving the discrete logarithm problem efficiently is based on the absence of known algorithms that can solve it in polynomial time. Unlike the related problem of integer factorization, which can be efficiently solved by some algorithms such as the General Number Field Sieve (GNFS), no such efficient algorithm is known for the discrete logarithm problem in general. Various cryptographic protocols and systems rely on the assumed difficulty of the discrete logarithm problem for their security. For example:

Diffie-Hellman Key Exchange:

In this protocol, two parties can agree on a shared secret key over an insecure channel without needing to exchange the key directly. The security of this protocol is based on the difficulty of solving the discrete logarithm problem in the finite field used for the computations.

ElGamal Encryption Scheme:

This encryption scheme is based on the difficulty of solving the discrete logarithm problem in the multiplicative group of a finite field. The security of ElGamal encryption relies on the assumption that an attacker cannot efficiently compute the discrete logarithm of a given value.

Thank You