@vika

CRYPTOGRAPHY & NETWORK SECURITY

Contents

MODULE 1

CRYPTOLOGY

Definition: The science of securing and analyzing information to ensure confidentiality, integrity, and authentication.

Components of Cryptology

1. **Cryptography:**
   - Purpose: Create secure communication systems.
   - Focus: Design encryption and decryption techniques.
   - Methods: Symmetric encryption, asymmetric encryption, hashing, digital signatures.
2. **Cryptanalysis:**
   - Purpose: Break or analyze cryptographic systems.
   - Focus: Identify weaknesses in algorithms and implementations.
   - Techniques: Brute-force attacks, mathematical analysis, side-channel attacks, exploiting flaws.

Goals of Cryptology

- Balance secure cryptographic methods with understanding vulnerabilities through cryptanalysis.

In Cryptography

- **Protecting Information:** Through encryption and decryption.
- **Verifying Identities:** Using digital signatures and certificates.
- **Ensuring Data Integrity:** Detecting unauthorized modifications.

In Cryptanalysis

- Testing Security: Ensuring algorithms can withstand attacks.
- Improving Standards: Strengthening future cryptographic designs.

Applications of Cryptology

- **Cybersecurity**: Protects sensitive data in various sectors.
- **E-commerce**: Secures online transactions with encryption protocols (SSL/TLS).
- **Blockchain**: Uses cryptographic hashing and digital signatures for security.
- **Military Communication**: Protects sensitive communications in defense.
- **Post-Quantum Cryptography**: Develops algorithms resistant to quantum computing.

CRYPTOGRAPHY

Definition: Techniques for securing communication and data against adversaries.

Core Objectives (C-I-A-N):

- **Confidentiality**: Accessible only to authorized users.
- **Integrity**: Data remains unaltered during transmission/storage.
- **Authentication**: Verifies identities of communication parties.
- **Non-repudiation**: Sender cannot deny sending a message.

Key Concepts

- **Encryption**: Converts plaintext into ciphertext using an algorithm and key.
- **Decryption**: Converts ciphertext back to plaintext using a key.
- **Keys**: Used in algorithms; can be symmetric or asymmetric.

Types of Cryptography

- **Symmetric Cryptography:**
  1. Same key for encryption and decryption.
  2. Examples: AES, DES.
  3. Pros: Fast; Cons: Secure key sharing required.
- **Asymmetric Cryptography:**
  1. Uses a public key for encryption and a private key for decryption.
  2. Examples: RSA, ECC.
  3. Pros: Solves key sharing problem; Cons: Computationally intensive.
- **Hash Functions:**
  1. Converts data into a fixed-length hash.
  2. Examples: SHA-256, MD5.
  3. Purpose: Ensures data integrity, not encryption.

Difference Between Public Key and Secret Key:

Aspect	**Public Key (Asymmetric Encryption)**	**Secret Key (Symmetric Encryption)**
Definition	A key that is shared publicly for encryption or verifying digital signatures.	A key that is kept private and shared between communicating parties.
Key Pair	Used in combination with a private key.	Same key is used for both encryption and decryption.
Security	Security relies on keeping the private key secret while the public key can be freely shared.	Security relies on keeping the secret key confidential among all participants.
Speed	Slower due to complex mathematical operations.	Faster as it uses simpler algorithms like block ciphers.
Usage	- Encryption in systems like RSA. - Digital signatures. - Key exchange protocols (e.g., Diffie-Hellman).	- Encrypting bulk data (e.g., AES). - Securing communication in real-time systems.
Key Sharing	Public key is shared openly; private key is kept secret by its owner.	Requires secure key exchange between parties beforehand.
Algorithm Examples	RSA, ECC (Elliptic Curve Cryptography), DSA.	AES, DES, 3DES, Blowfish.
Use Cases	- Secure key exchanges. - Digital certificates. - SSL/TLS encryption.	- Securing stored data. - Real-time communication (e.g., video calls).

Scalability	Highly scalable since public keys can be distributed to many users.	Limited scalability as a unique key is needed for each pair of users.
Quantum Resistance	More vulnerable to quantum attacks.	Generally more resistant (with larger key sizes).

Applications of Cryptography

1. **Secure Communication:** Used in emails, messaging apps, and online transactions to ensure data confidentiality and integrity.
2. **Authentication:** Password storage and verification utilize cryptographic hashing.
3. **Digital Signatures:** Verify the authenticity and integrity of digital documents.
4. **Blockchain:** Cryptographic techniques secure, immutable, and decentralized ledgers.
5. **SSL/TLS Protocols:** Secure websites and online services by encrypting data exchanged over the internet.

Key Size Implications

1. **Number of Possible Keys:**
   - An n-bit key can represent 2^n combinations.
   - **Examples:**
     - For n = 128: $2^{128} \approx 3.4 \times 10^{38}$ possible keys.
     - For n = 256: $2^{256} \approx 1.2 \times 10^{77}$ possible keys.
   - Large key sizes make brute-force attacks computationally infeasible.
2. **Key Space and Security:**
   - **Key Space:** Total set of possible keys, which is 2^n.
   - Larger key spaces increase difficulty for attackers to guess the correct key.
   - **Modern systems typically use:**
     - 128-bit keys for symmetric encryption (e.g., AES-128).
     - 2048-bit keys or higher for asymmetric encryption (e.g., RSA).
3. **Brute-Force Attack Complexity:**
   - Time for a brute-force attack depends on key size, computational power, and technology.
   - On average, a brute-force attack on an n-bit key requires testing 2^(n-1) keys.
4. **Implications of Key Size:**
   - **Symmetric Encryption:**
     - Common key sizes: 128, 192, and 256 bits.
     - 128-bit keys are secure against classical computing brute-force attacks.
     - 256-bit keys provide extra security against quantum attacks.
   - **Asymmetric Encryption:**
     - Requires larger key sizes due to algorithmic structure.
     - A 2048-bit RSA key offers security roughly equivalent to a 112-bit symmetric key.

Summary Table

Key Size (n)	**Possible Keys (2^n)**	**Approximate Security**
64 bits	**1.8×10^{19}**	**Weak for modern systems**
128 bits	**3.4×10^{38}**	**Strong for symmetric encryption**
256 bits	**1.2×10^{77}**	**Extra security and quantum-resistant**
2048 bits (RSA)	**Very large**	**Common for asymmetric encryption**

RSA ALGORITHM (Rivest-Shamir-Adleman)

1. Key Generation:

1. Compute n: $\mathbf{n=p\times q}$ (p and q are two distinct large prime numbers.)
2. Compute Euler's totient function $\phi(n)$: $\boldsymbol{\phi(n)=(p-1)\times(q-1)}$
3. Choose a public exponent e: $\boldsymbol{1<e<\phi(n)}$, where $\mathbf{gcd}\ \boldsymbol{(e,\phi(n))=1}$.
4. Compute the private key d: $\boldsymbol{d\times e\equiv1(\mathrm{mod}\ \phi(n))}$

   (d is the modular multiplicative inverse of e modulo $\phi(n)$.)

2. Encryption: Given the message M (where $0\le M<n$), find ciphertext: $\mathbf{C=M^e\ mod\ n}$

3. Decryption: To decrypt the ciphertext C, compute the original message M: $\mathbf{M=C^d\ mod\ n}$

Step 1: Key Generation

1. **Choose two large prime numbers p and q:** Let p=61 and q=53.
2. **Compute n:** $n = p\times q = 61\times53 = 3233$
3. **Calculate Euler's totient function $\boldsymbol{\phi(n)}$:** $\phi(n) = (p-1)\times(q-1) = (61-1)\times(53-1) = 60\times52 = 3120$
4. **Choose a public exponent e:**
   e must be coprime with $\phi(n)$ and $1<e<\phi(n)$.
   Let e=17 (a commonly used small prime).
5. **Calculate the private key d:**
   d is the modular multiplicative inverse of e modulo $\phi(n)$, i.e., $d\times e \equiv 1(\mathrm{mod}\ \phi(n))$

   Using the **Extended Euclidean Algorithm**, d=2753.

   - Public key: (e,n)=(17,3233)
   - Private key: (d,n)=(2753,3233)

Step 2: Encryption

Let the message M=65.

1. Convert M to an integer (if it's not already an integer). In this case, M=65.
2. Encrypt M using the public key (e,n):

$C=M^e \bmod n \Rightarrow C=65^{17} \mathrm{mod} 3233$

Computing $65^{17}\mathrm{mod}3233$ (using fast modular exponentiation): C=2790

Ciphertext: C=2790

Step 3: Decryption

To retrieve the original message M, use the private key (d,n):

$M=C^d \bmod n \Rightarrow M=2790^{2753}\bmod 3233$

Using fast modular exponentiation: M=65

Decrypted message: M=65

EULER'S TOTIENT FUNCTION ($\Phi(N)$):

Counts the number of positive integers less than n that are relatively prime (coprime) to n. Two numbers are coprime if their greatest common divisor (GCD) is 1.

Key Formulas:

1. If $n=p$ (where p is a prime number): $\boldsymbol{\phi(p)=p-1}$
2. If $n=p^k$: $\boldsymbol{\phi(p^k)=p^k-p^{k-1}=p^k\times(1-1/p)}$
3. If n has a prime factorization $n=p_1^{k1}\times p_2^{k2}\times\ldots\times p_m^{km}$: $\boldsymbol{\phi(n)=n\times(1-1/p_1)\times(1-1/p_2)\times\ldots\times(1-1/p_m)}$

EUCLIDEAN ALGORITHM:

The Euclidean Algorithm is a method to compute the **Greatest Common Divisor (GCD)** of two integers efficiently. It works on the principle that the GCD of two numbers does not change if the larger number is replaced by its remainder when divided by the smaller number.

Key Formula:

For two integers a and b ($a>b$): **$\mathrm{GCD}(a,b) = \mathrm{GCD}(b, a \bmod b)$**

This process is repeated until the remainder becomes 0. The non-zero remainder from the last step is the GCD.

Extended Euclidean Algorithm:

The extended Euclidean algorithm finds coefficients x and y such that: **$ax + by = \mathrm{GCD}(a, b)$**

This is useful in computing the **modular multiplicative inverse**.

Q) Compute GCD of 48 and 18

1. **Divide a by b and find the remainder:**
   $48\div18=2$ (quotient), $48 \bmod 18=12$

   GCD(48,18) = GCD(18,12)
2. **Repeat with the smaller number $b=12$:**
   $18\div12=1$ (quotient), $18 \bmod 12=6$
   GCD(18,12) = GCD(12,6)
3. **Repeat again with $b=6$:**
   $12\div6=2$ (quotient), $12 \bmod 6=0$

   GCD(12,6) = 6
4. **Stop when the remainder is 0:**
   The GCD is the last non-zero remainder, which is **6**.

Q) Extended Euclidean Algorithm: 48,18

Steps to find x and y such that $48x+18y=6$

1. From the last step of the Euclidean algorithm:
   $12 = 48 - 18\cdot2$
   $6 = 18 - 12\cdot1$
2. Substitute $12 = 48 - 18\cdot2$ into $6 = 18 - 12\cdot1$:
   $6 = 18 - (48 - 18\cdot2)$
   $6 = 18 - 48 + 36$
   $6 = -48 + 54$
3. Solution: $x=-1$, $y=3$
4. Verification: $48(-1) + 18(3) = -48 + 54 = 6$

Finding the Modular Inverse Using the Extended Euclidean Algorithm

The modular inverse of $a \bmod m$ is a number x such that: $\mathbf{a \cdot x \equiv 1 (mod\ m)}$

This means x satisfies the equation: $\mathbf{a \cdot x + m \cdot y = 1}$

The inverse exists only if $\gcd(a,m)=1$

Q) Find the modular inverse of 17 mod 43

We need to find x such that: $17 \cdot x \equiv 1 (mod 43)$

Step 1: Apply the Extended Euclidean Algorithm

1. Start with m=43, a=17, and find the remainders:

   $43 = 17 \cdot 2 + 9$

   $17 = 9 \cdot 1 + 8$

   $9 = 8 \cdot 1 + 1$

   $8 = 1 \cdot 8 + 0$

   The GCD is 1, so 17 and 43 are coprime, and an inverse exists.

Step 2: Back-Substitution to Find the Coefficients

We now backtrack to express 1 as a linear combination of 43 and 17.

1. From $9 = 8 \cdot 1 + 1$ => $1 = 9 - 8$
2. Substitute $8 = 17 - 9 \cdot 1$:

   $1 = 9 - (17 - 9 \cdot 1)$ => $1 = 9 \cdot 2 - 17$
3. Substitute $9=43-17 \cdot 2$:

   $1 = (43 - 17 \cdot 2) \cdot 2 - 17$ => $1 = 43 \cdot 2 - 17 \cdot 5$

Step 3: Modular Inverse

From the equation $1 = 43 \cdot 2 - 17 \cdot 5$ => $-5 \cdot 17 \equiv 1 (mod 43)$

Thus, x = −5. Since modular inverses are always positive, convert −5 to a positive equivalent modulo 43: X = 43 − 5 = 38

Final Answer: The modular inverse of 17 mod 43 is 38.

Verification: $17 \cdot 38 \bmod 43 = 646 \bmod 43 = 1$

GENERATOR

An integer g is a generator modulo n if: $\mathbf{g^k \bmod n\ (0 \le k < \phi(n))}$ produces all integers in the reduced residue system modulo n, where $\phi(n)$ is Euler's totient function.

Steps to Find a Generator

1. Compute $\phi(n)$. (The number of integers less than n that are coprime to n.)
2. Determine the prime factors of $\phi(n)$.
3. For each candidate g (from 2 to n−1), check if: $\mathbf{g^{\phi(n)/p} \equiv 1\ (mod n)}$ for all prime factors p of $\phi(n)$

If g satisfies this condition, it is a generator.

Q) Find a Generator Modulo 7

Step 1: Compute $\phi(7)$

Since 7 is prime: $\phi(7) = 7-1 = 6$

Step 2: Prime Factors of $\phi(7)$

$\phi(7)=6$ and its prime factors are 2 and 3.

Step 3: Check Candidates

Candidates for g are 2,3,4,5,6 (integers less than 7).

Candidate g=2:

1. Compute $2^{6/2} \bmod 7 = 2^3 \bmod 7 = 8 \bmod 7 = 1$: g=2 fails because $2^{6/2} \bmod 7 \equiv 1$.

Candidate g=3:

1. Compute $3^{6/2} \bmod 7 = 3^3 \bmod 7 = 27 \bmod 7 = 6$ => $3^{6/2} \bmod 7$!≡ 1
2. Compute $3^{6/3} \bmod 7 = 3^2 \bmod 7 = 9 \bmod 7 = 2$ => $3^{6/3} \bmod 7$!≡ 1

Thus, g=3 is a generator.

Step 4: Verify g=3

Compute all powers of g=3 $\bmod 7$:

$3^1 \bmod 7 = 3$, $3^2 \bmod 7 = 2$, $3^3 \bmod 7 = 6$, $3^4 \bmod 7 = 4$, $3^5 \bmod 7 = 5$, $3^6 \bmod 7 = 1$

These powers produce all integers coprime to 7 modulo 7: 1,2,3,4,5,6

Answer: The generator modulo 7 is g=3.

FERMAT'S LITTLE THEOREM (Primality Testing)

Fermat's Little Theorem is a fundamental result in number theory that provides a property of integers modulo a prime number. It is widely used in cryptography, particularly in RSA encryption.

Theorem Statement

If p is a prime number and a is an integer such that **gcd(a,p)=1**, then: $\mathbf{a^{p-1} \equiv 1 \pmod{p}}$

Alternatively, for any integer a: $\mathbf{a^p \equiv a \pmod{p}}$

Applications of Fermat's Little Theorem

1. **Checking Primality**: Fermat's theorem is the basis for Fermat's primality test.
2. **Finding Modular Inverses**: The theorem is used to find the modular inverse of a $\bmod p$. If p is prime, the modular inverse of a $\bmod p$ is: $a^{p-2} \bmod p$

Q1) Verification of Fermat's Little Theorem

Let p=7 (prime) and a=3:

$3^{7-1} \equiv 36 \pmod{7}$

=> $3^6 \bmod 7$

=> $729 \bmod 7 = 1$

Thus: $3^{7-1} \equiv 1 \pmod{7}$

Q2) Finding a Modular Inverse

Find the modular inverse of 3 $\bmod 7$:
Using Fermat's theorem, the modular inverse is:

$3^{7-2} \equiv 35 \pmod{7}$

=> $3^5 \bmod 7$

=> $243 \bmod 7 = 5$

Thus, the modular inverse of 3 $\bmod 7$ is 5.

EULER'S THEOREM (Primality Testing)

Euler's theorem is a generalization of Fermat's Little Theorem, applicable to any modulus n, not just primes. It is a fundamental result in number theory and cryptography.

Theorem Statement

If a and n are coprime integers (gcd(a,n)=1), then: $\mathbf{a^{\phi(n)} \equiv 1 \ (mod n)}$

where ϕ(n) is **Euler's totient function**.

Relation to Fermat's Little Theorem

- If n=p, where p is prime, then $\phi(p) = p-1$.
  Euler's theorem reduces to Fermat's Little Theorem: $a^{p-1} \equiv 1 \ (mod p)$

Example: Apply Euler's Theorem

Q) Compute $7^{40} \bmod 15$

Step 1: Check if 7 and 15 are coprime: Since gcd(7,15) = 1, Euler's theorem applies.

Step 2: Calculate ϕ(15)

The prime factorization of 15 is 15=3×5. Using the formula for Euler's totient function:

$\phi(15) = 15\cdot(1 - 1/3)\cdot(1 - 1/5) = 15\cdot23\cdot45 = 8$

Step 3: Apply Euler's Theorem

By the theorem:

$7^{\phi(15)} \equiv 7^8 \equiv 1 \ (mod15)$

Step 4: Simplify $7^{40} \bmod 15$

Since $7^{40} = (7^8)^5$, and $7^8 \equiv 1 \ (mod15)$:

$7^{40} \equiv 1^5 \equiv 1 \ (mod15)$

Answer: $7^{40} \bmod 15 = 1$

CHINESE REMAINDER THEOREM (CRT)

The Chinese Remainder Theorem provides a way to solve a system of simultaneous congruences with pairwise coprime moduli. It is particularly useful in modular arithmetic and number theory.

Theorem Statement

Let n1,n2,…,nk be **pairwise coprime** integers **(gcd(ni,nj)=1 for i≠j)**. For any integers a1,a2,…,ak , the system of congruences: **x≡a1(modn1)**; **x≡a2(modn2); x≡ak(modnk)** has a unique solution **modulo N=n1⋅n2⋅⋯⋅nk.**

Steps to Solve Using CRT

1. Compute **N=n1⋅n2⋅⋯⋅nk**.
2. For each modulus ni, calculate **Ni=N/ni**.
3. Find the modular inverse of Ni mod ni, denoted as **yi** , such that: **Ni⋅yi ≡ 1 (mod ni)**
4. Compute the solution x as: **x = ∑(i=1,k) ai⋅Ni⋅yi (modN)**

Q) Solve the system of congruences:

x≡2(mod3); x≡3(mod4); x≡1(mod5)

Step 1: a1=2; a2=3; a3=1; n1=3; n2=4; n3=5

Compute N: N = 3·4·5 = 60

Step 2: Compute Ni:

N1 = N/3 = 20

N2 = N/4 = 15

N3 = N/5 = 12

Step 3: Find Modular Inverses

1. N1 = 20 mod 3: Solve 20·y1 ≡ 1 (mod3)
   20 ≡ 2 (mod3) => Solve 2·y1 ≡ 1 (mod3)
   y1 = 2 (since 2·2 = 4 ≡ 1 (mod3))

2. N2 = 15 mod 4: Solve 15·y2 ≡ 1 (mod4)
   15 ≡ 3 (mod4) => Solve 3·y2 ≡ 1 (mod4)
   y2 = 3 (since 3·3 = 9 ≡ 1 (mod4))

3. N3 = 12 mod 5: Solve 12·y3 ≡ 1 (mod5)
   12 ≡ 2 (mod5) => Solve 2·y3 ≡ 1 (mod5)
   y3 = 3 (since 2·3 = 6 ≡ 1 (mod5))

Step 4: Compute x

X = (a1·N1·y1) + (a2·N2·y2) + (a3·N3·y3) (modN)

X = (2·20·2) + (3·15·3) + (1·12·3) (mod60)

X = (80) + (135) + (36) (mod60)

X = 251 (mod60)

X = 251 mod 60 = 11

Answer: X ≡ 11 (mod60)

DISCRETE LOGARITHM PROBLEM (DLP)

The **Discrete Logarithm Problem (DLP)** is a mathematical challenge that underpins the security of many cryptographic systems. It involves finding the exponent x in the equation: $\mathbf{g^x \equiv y \pmod p}$

Here:

- g is a **generator** (a primitive root modulo p).
- p is a prime number.
- y is a known result (modular exponentiation result).
- x is the **discrete logarithm** of y to the base g modulo p.

The DLP is **computationally hard** to solve when **p is large**, making it ideal for cryptographic applications like **Diffie-Hellman key exchange** and **ElGamal encryption**.

Steps to Solve the DLP

1. Given g, p, and y, **find** x such that: $\mathbf{g^x \equiv y \pmod p}$
2. Test **successive powers** of g modulo p until $\mathbf{g^x \equiv y \pmod p}$.

Q) Solve for x in: $3^x \equiv 4 \pmod 7$

Step-by-Step Solution:

1. **Given Data**: g=3; p=7; y=4
2. **Check Successive Powers of g Modulo p**: Compute $\mathbf{g^x \equiv y \pmod p}$ for x=1,2,3,…:
   - $3^1 \bmod 7 = 3$

  - 3^2 mod7 = 9 mod7 = 2
  - 3^3 mod7 = 27 mod7 = 6
  - 3^4 mod7 = 81 mod7 = 4

3. **Find x**: $3^4 \equiv 4$ (mod7); Therefore, x=4.

GROUPS

A set G with a binary operation * is called a **group** if the following conditions are satisfied:

Property	Description
Closure	a∗b ∈G for all a,b ∈G
Associativity	(a∗b)∗c = a∗(b∗c)
Identity	a∗e = e∗a = a for all a ∈G
Inverses	$a*a^{-1} = a^{-1}*a = e$

Types of Groups

1. **Abelian Group**:
   - A group is **Abelian** (or **commutative**) if a∗b = b∗a for all a,b ∈G.
   - Example: The set of integers $\mathbb{Z}$ under addition.
2. **Non-Abelian Group**:
   - A group is **non-Abelian** if a∗b ≠ b∗a for some a,b ∈G.
   - Example: The group of 2×2 invertible matrices under matrix multiplication.

Examples of Groups

1. Integers Under Addition ($\mathbb{Z}$,+)
2. Nonzero Real Numbers Under Multiplication ($\mathbb{R}^*$,·) => Set: $\mathbb{R}^*=\mathbb{R}\setminus\{0\}$
3. Symmetric Group S_3 => Set: All permutations of {1,2,3} (6 elements).

FIELD

A field ($\mathbb{F}$,+,·) is a set F with two binary operations (addition and multiplication) such that:

Property	Description
Closure	For any a,b ∈F, a+b ∈F and a·b ∈F
Associativity	(a+b)+c = a+(b+c), (a·b)·c = a·(b·c) for all a,b,c ∈F
Commutativity	a+b = b+a, a·b =b·a for all a,b ∈F
Identity Elements	Additive Identity: There exists 0∈F such that a+0 = a Multiplicative Identity: There exists 1∈F such that a·1 = a
Additive Inverses	For each a∈F, there exists −a∈F such that a + (−a) = 0
Multiplicative Inverses	For each a∈F, a≠0, there exists a^{-1}∈F such that $a \cdot a^{-1}$ = 1 a
Distributivity	a·(b+c) = (a·b) + (a·c) for all a,b,c ∈F

Additional Notes on Fields

1. **Finite Fields:** Fields with a finite number of elements are called finite fields or Galois fields (F_q, where $q=p^n$ for some prime p and n≥1).

2. **Infinite Fields:** Examples include R (real numbers), Q (rational numbers), and C (complex numbers).
3. **Characteristic:** The smallest positive integer p such that $p \cdot 1=0$ in the field. If no such p exists, the field has characteristic 0.

Examples of Fields: (Q,+,.), (R,+,.), (C,+,.)

Q1) Finite Field F_5

Let F= {0,1,2,3,4} with addition and multiplication modulo 5:

- Addition: $2+3 = 5 \bmod 5 = 0$
- Multiplication: $3 \cdot 4 = 12 \bmod 5 = 2$
- Additive inverse: $-2 = 3$ −because $2+3 \equiv 0 \bmod 5$
- Multiplicative inverse: $3^{-1} = 2$ because $3 \cdot 2 \equiv 1 \bmod 5$.

Q2) Real Numbers (R)

- Additive identity: 0.
- Multiplicative identity: 1.
- Additive inverse: $-a$ for $a \in R$
- Multiplicative inverse: $a^{-1} = 1/a$, $a \neq 0$