Cryptography and Network Security

UNIT II – Complete Notes

B.Tech Electronics and Computer Engineering | AKTU

Topics Covered

Modular Arithmetic • Groups • Fields • GF(p) • Extended Euclidean Algorithm • AES • Fermat’s Theorem • Euler’s Theorem • Primality Testing • Chinese Remainder Theorem • Discrete Logarithm • Public-Key Cryptography • RSA • RSA Security

Simple language • Exam-oriented explanations • Important formulas • Examples • Revision tables

Contents

1. Introduction to Unit II

Unit II introduces the mathematical concepts used in modern cryptography. These concepts are important for understanding AES, public-key cryptography, RSA and discrete-logarithm based systems.

The unit can be understood in three parts:

- Mathematical foundation: modular arithmetic, prime numbers, groups, fields, finite fields, GCD and modular inverse.
- Symmetric encryption: Advanced Encryption Standard (AES).
- Public-key cryptography: Fermat and Euler theorems, CRT, discrete logarithms, public-key principles and RSA.

Note: Learn the formulas together with their meaning. In numerical questions, write the formula first, substitute the values, and then show the modular calculation.

2. Modular Arithmetic

2.1 Meaning

Modular arithmetic is arithmetic based on remainders. If a number a is divided by n and the remainder is r, then a mod n = r.

$$17 \bmod 5 = 2$$

The notation a ≡ b (mod n) means that a and b have the same remainder when divided by n. Equivalently, n divides a − b.

$$17 \equiv 5 \pmod{12}$$

2.2 Basic Operations

Operation	General Form	Example
Addition	(a + b) mod n	15 + 13 = 28; 28 mod 7 = 0
Subtraction	(a − b) mod n	15 − 8 = 7; 7 mod 7 = 0
Multiplication	(a × b) mod n	7 × 8 = 56; 56 mod 9 = 2
Exponentiation	a^k mod n	3^4 = 81; 81 mod 5 = 1

2.3 Modular Multiplicative Inverse

The inverse of a modulo n is a number x such that ax ≡ 1 (mod n). The inverse exists when gcd(a,n) = 1.

$$ax \equiv 1 \pmod{n}$$

Example: For the inverse of 3 modulo 7, 3 × 5 = 15 and 15 mod 7 = 1. Therefore $3^{-1} \equiv 5 \pmod{7}$.

Note: Modular inverses are important in RSA and in many number-theoretic cryptographic calculations.

3. Prime and Relatively Prime Numbers

3.1 Prime Number

A prime number is an integer greater than 1 having exactly two positive divisors: 1 and the number itself.

Examples: 2, 3, 5, 7, 11, 13, 17 and 19.

3.2 Composite Number

A composite number is greater than 1 and has more than two positive divisors. For example, 12 = 2 × 2 × 3, so 12 is composite.

3.3 Relatively Prime Numbers

Two numbers are relatively prime, or coprime, when their greatest common divisor is 1.

gcd(a,b) = 1

Example: 8 and 15 are relatively prime because their only common positive divisor is 1.

3.4 Importance in Cryptography

- Prime numbers are used to construct cryptographic keys.
- Coprime numbers are needed when calculating modular inverses.
- RSA requires the public exponent e to satisfy gcd(e, φ(n)) = 1.

4. Groups and Abelian Groups

4.1 Group

A group is a set together with an operation that satisfies four basic properties: closure, associativity, identity and inverse.

Property	Meaning
Closure	Combining any two elements of the set gives another element of the set.
Associativity	(a * b) * c = a * (b * c).
Identity	There is an element e such that a * e = e * a = a.
Inverse	Every element has another element that gives the identity when combined with it.

4.2 Example

The integers under addition, (Z, +), form a group. The identity is 0 and the inverse of a is −a.

4.3 Abelian Group

A group is called an Abelian group when the operation is also commutative.

a * b = b * a

Example: Integer addition is commutative because 3 + 5 = 5 + 3.

Note: Groups provide mathematical structures used in several areas of modern cryptography.

5. Fields

A field is an algebraic structure in which addition, subtraction, multiplication and division follow the required field rules. Every non-zero element has a multiplicative inverse.

5.1 Important Idea

- Addition and multiplication are the main operations.
- There is an additive identity, 0.
- There is a multiplicative identity, 1.
- Every non-zero element has a multiplicative inverse.
- The operations satisfy the required associative, commutative and distributive properties.

5.2 Examples

The rational numbers Q, real numbers R and complex numbers C are examples of fields.

> Note: Finite fields are especially important in cryptography because they provide a finite set in which arithmetic can be performed predictably.

6. Finite Fields and GF(p)

6.1 Finite Field

A finite field is a field containing a finite number of elements. It is commonly written as GF(q), where q is the number of elements.

6.2 GF(p)

When p is prime, GF(p) contains p elements: 0, 1, 2, ..., p − 1. All arithmetic is performed modulo p.

GF(5) = {0, 1, 2, 3, 4}

Example: In GF(5), 3 + 4 = 7 and 7 mod 5 = 2. Therefore 3 + 4 = 2 in GF(5). Similarly, 3 × 4 = 12 and 12 mod 5 = 2.

6.3 Why GF(p) Matters

- It gives a finite mathematical environment for cryptographic calculations.
- It is used as a foundation for several cryptographic algorithms and constructions.
- Finite-field arithmetic is also important in the study of AES and other modern cryptographic systems.

7. Extended Euclidean Algorithm

The Extended Euclidean Algorithm (EEA) finds the greatest common divisor of two integers and also finds integers x and y satisfying Bézout's identity.

$$ax + by = gcd(a,b)$$

7.1 Why EEA is Important

- It finds GCD.
- It can find a modular multiplicative inverse.
- Modular inverses are required in RSA key generation.

7.2 Worked Example

Find the inverse of 7 modulo 26.

First use Euclidean division:

$$26 = 7 \times 3 + 5$$

$$7 = 5 \times 1 + 2$$

$$5 = 2 \times 2 + 1$$

$$2 = 1 \times 2 + 0$$

Therefore gcd(26,7) = 1. Work backwards:

$$1 = 5 - 2 \times 2$$

$$1 = 3 \times 5 - 2 \times 7$$

$$1 = 3 \times 26 - 11 \times 7$$

Therefore $7(-11) \equiv 1 \pmod{26}$. Since $-11 \equiv 15 \pmod{26}$, the inverse is:

$$7^{-1} \equiv 15 \pmod{26}$$

8. Advanced Encryption Standard (AES)

8.1 Introduction

AES stands for Advanced Encryption Standard. It is a symmetric-key block cipher used to protect digital information. AES uses the same secret key family for encryption and decryption.

8.2 AES Sizes

Version	Key Size	Block Size	Rounds
AES-128	128 bits	128 bits	10
AES-192	192 bits	128 bits	12
AES-256	256 bits	128 bits	14

Note: The AES block size is always 128 bits. The key size changes the number of rounds.

8.3 AES State

A 128-bit data block contains 16 bytes. AES arranges these bytes as a 4 × 4 matrix called the state.

128 bits = 16 bytes = 4 × 4 byte state

8.4 Main AES Operations

- SubBytes: substitutes every byte using the AES S-box.
- ShiftRows: cyclically shifts the rows of the state.
- MixColumns: transforms each column using finite-field arithmetic.
- AddRoundKey: XORs the state with the round key.

8. AES Encryption – Detailed Process

8.5 Initial Transformation

Before the regular rounds, AES performs AddRoundKey. The plaintext state is XORed with the first round key.

State = Plaintext XOR RoundKey

8.6 Rounds 1 to 9 in AES-128

Each of the first nine rounds performs four operations in this order:

SubBytes → ShiftRows → MixColumns → AddRoundKey

8.7 Final Round

The final round performs SubBytes, ShiftRows and AddRoundKey. MixColumns is not performed in the final round.

Final round: SubBytes → ShiftRows → AddRoundKey

8.8 SubBytes

Every byte is replaced by another byte using the AES S-box. This introduces non-linearity and makes simple mathematical relationships harder to exploit.

8.9 ShiftRows

Row 0 is not shifted. Row 1 is shifted left by one byte, row 2 by two bytes, and row 3 by three bytes.

8.10 MixColumns

Each column is transformed using matrix multiplication over GF(2^8). This provides diffusion by spreading the influence of bytes across a column.

8.11 AddRoundKey

The current state is XORed with the round key. The round keys are generated from the original key by the AES key-expansion process.

8. AES Decryption

AES decryption reverses the encryption process and uses inverse transformations.

Encryption Operation	Inverse Operation
SubBytes	InvSubBytes
ShiftRows	InvShiftRows
MixColumns	InvMixColumns
AddRoundKey	AddRoundKey

8.12 Decryption Steps

- Use the round keys in the appropriate reverse order.
- Apply inverse ShiftRows.
- Apply inverse SubBytes.
- Apply AddRoundKey.
- Apply inverse MixColumns where required.

The inverse operations finally recover the original plaintext block.

8.13 AES Key Expansion

AES expands the original secret key into multiple round keys. AES-128 requires 11 round keys: one for the initial AddRoundKey and one for each of the 10 rounds.

9. Fermat’s Little Theorem

Fermat’s Little Theorem is an important result in modular arithmetic. If p is prime and a is not divisible by p, then:

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

9.1 Example

Take a = 2 and p = 7. Since 7 is prime and gcd(2,7) = 1:

$$2^6 \equiv 1 \pmod{7}$$

Because 2^6 = 64 and 64 mod 7 = 1.

9.2 Modular Inverse Connection

For a prime modulus p and a not divisible by p:

$$a^{p-2} \equiv a^{-1} \pmod{p}$$

9.3 Uses

- Modular arithmetic.
- Finding inverses under a prime modulus.
- Number-theoretic reasoning in cryptography.
- Understanding public-key cryptographic mathematics.

10. Euler’s Theorem and Euler’s Totient Function

10.1 Euler’s Theorem

If gcd(a,n) = 1, then Euler’s theorem states:

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

10.2 Euler’s Totient Function

φ(n) counts the positive integers up to n that are relatively prime to n.

Example: For n = 8, the numbers 1, 3, 5 and 7 are relatively prime to 8. Therefore φ(8) = 4.

10.3 Useful Totient Formulas

Condition	Formula
p is prime	$\varphi(p) = p - 1$
p and q are distinct primes	$\varphi(pq) = (p - 1)(q - 1)$

10.4 Importance in RSA

For RSA, when n = pq for two distinct primes, $\varphi(n) = (p - 1)(q - 1)$. This value is used while generating the private exponent.

11. Primality Testing

Primality testing determines whether an integer is prime or composite. It is important in public-key cryptography because RSA requires large prime numbers.

11.1 Basic Trial-Division Test

For a number n, it is enough to test possible divisors up to √n. If no integer from 2 through √n divides n, the number is prime.

Example: For 29, √29 is approximately 5.38. Test 2, 3 and 5. None divides 29, so 29 is prime.

11.2 Probabilistic Tests

For very large numbers, testing every divisor is inefficient. Cryptographic software uses probabilistic tests.

Test	Basic Idea
Fermat test	Uses modular exponentiation to identify likely primes; some composite numbers can pass it.
Miller-Rabin	Returns composite or probably prime; repeating the test with different bases reduces the error probability.

Note: In cryptography, a result such as 'probably prime' means the probability of an incorrect classification is made extremely small by appropriate testing.

12. Chinese Remainder Theorem (CRT)

The Chinese Remainder Theorem gives a method for solving a set of simultaneous congruences when the moduli are pairwise relatively prime.

$$x \equiv a_1 \pmod{n_1},\ x \equiv a_2 \pmod{n_2},\ \ldots$$

If the moduli are pairwise coprime, there is a unique solution modulo $N = n_1 n_2 \ldots nk$.

12.1 Simple Example

Solve: $x \equiv 2 \pmod 3$ and $x \equiv 3 \pmod 5$.

The number 8 satisfies both conditions:

$$8 \bmod 3 = 2$$

$$8 \bmod 5 = 3$$

Therefore:

$$x \equiv 8 \pmod{15}$$

12.2 CRT in RSA

RSA private-key calculations can be performed separately modulo p and q, and the two results can then be combined using CRT. This is known as RSA-CRT and can make private-key operations significantly faster.

13. Discrete Logarithmic Problem

The Discrete Logarithm Problem (DLP) asks us to find an exponent x from an equation of the form:

g^x ≡ y (mod p)

Here g and y are known, p is a modulus, and x is the unknown exponent.

13.1 Simple Example

Example: Suppose 2^x ≡ 8 (mod 11). Since 2^3 = 8, x = 3.

13.2 Why It Matters

For small numbers, the discrete logarithm can be found by trial. For appropriately selected large parameters, the problem can become computationally difficult.

- Diffie-Hellman key exchange is based on related discrete-logarithm assumptions.
- ElGamal encryption uses discrete-logarithm based mathematics.
- ElGamal digital signatures also use related mathematics.
- Elliptic-curve cryptography uses related discrete-logarithm problems on elliptic curves.

Note: The security idea is that computing a forward exponentiation can be efficient while recovering the secret exponent from the public result is intended to be computationally difficult.

14. Principles of Public-Key Cryptosystems

Public-key cryptography, also called asymmetric cryptography, uses a pair of related keys: a public key and a private key.

Key	Purpose
Public key	Can be openly distributed; depending on the system it may be used for encryption or signature verification.
Private key	Must be kept secret; depending on the system it may be used for decryption or digital signing.

14.1 Basic Confidentiality Model

If Bob wants to receive a confidential message, he publishes his public key. Alice encrypts the message using Bob’s public key. Bob decrypts it using his private key.

$$C = E(K_{public}, P)$$

$$P = D(K_{private}, C)$$

14.2 Requirements

- Key-pair generation must be practical.
- Encryption using the public key must be practical.
- Legitimate decryption using the private key must be practical.
- An attacker should not be able to feasibly derive the private key from public information.
- The system should be secure under appropriate attack models.

15. RSA Algorithm

RSA is a public-key cryptographic algorithm named after Ron Rivest, Adi Shamir and Leonard Adleman. It uses modular arithmetic and large prime numbers.

15.1 RSA Key Generation

The standard educational sequence is:

- Choose two large prime numbers p and q.
- Calculate n = pq.
- Calculate φ(n) = (p − 1)(q − 1).
- Choose e such that 1 < e < φ(n) and gcd(e, φ(n)) = 1.
- Find d such that ed ≡ 1 (mod φ(n)).
- Public key = (e, n). Private key = (d, n).

15.2 Small Educational Example

Choose p = 11 and q = 13.

$$n = 11 \times 13 = 143$$

$$\varphi(n) = (11-1)(13-1) = 120$$

Choose e = 7 because gcd(7,120) = 1.

$$7d \equiv 1 \pmod{120}$$

One solution is d = 103 because 7 × 103 = 721 and 721 mod 120 = 1.

$$\text{Public Key} = (7,143)$$

$$\text{Private Key} = (103,143)$$

15. RSA Encryption and Decryption

15.3 Encryption

Represent the plaintext as an integer M in the valid RSA range. The ciphertext is calculated using the public key:

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

15.4 Decryption

The receiver uses the private key:

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

15.5 Worked Small Example

Using the educational key pair above, let M = 9.

$$C = 9^7 \bmod 143 = 48$$

The receiver computes:

$$M = 48^{103} \bmod 143 = 9$$

Note: This is a small classroom example. Real RSA uses much larger parameters and secure padding; raw textbook RSA should not be used for real applications.

15.6 RSA Process Summary

Stage	Operation
Key generation	Generate p, q, n, φ(n), e and d.
Encryption	Use public key (e,n): C = M^e mod n.
Transmission	Send ciphertext C.
Decryption	Use private key (d,n): M = C^d mod n.

16. Security of RSA

16.1 Factorization

The RSA modulus is n = pq. The public key contains n, but p and q are kept secret. An attacker who can efficiently factor n can derive φ(n) and potentially recover the private exponent.

16.2 Strong Key Generation

- Use sufficiently large, randomly generated prime numbers.
- Use a cryptographically secure random-number generator.
- Avoid weak or predictable primes.
- Follow secure parameter-generation practices.

16.3 Small Exponent and Padding Issues

The commonly used public exponent 65537 provides efficient exponentiation. However, RSA must be used with an appropriate padding scheme and secure protocol design. Raw textbook RSA is deterministic and does not provide the security expected from modern encryption.

16.4 RSA Padding

For encryption, RSA-OAEP is a standard padding approach. For signatures, RSA-PSS is a modern standardized approach.

16.5 Side-Channel Attacks

An implementation may leak information through execution time, power consumption, cache behavior or other physical effects. These are called side-channel attacks.

16.6 RSA-CRT

RSA-CRT performs private-key calculations modulo p and q separately and combines the results with the Chinese Remainder Theorem. This can make RSA private-key operations faster.

Note: RSA security depends on both sound mathematics and correct implementation. A mathematically strong algorithm can still be insecure if keys, randomness, padding or implementation are handled incorrectly.

17. Important Unit II Formulas

Topic	Formula
Congruence	a ≡ b (mod n)
Modular inverse	ax ≡ 1 (mod n)
Fermat	a^(p−1) ≡ 1 (mod p), when p is prime and gcd(a,p)=1
Euler	a^φ(n) ≡ 1 (mod n), when gcd(a,n)=1
Totient for prime	φ(p) = p − 1
Totient for two distinct primes	φ(pq) = (p−1)(q−1)
RSA modulus	n = pq
RSA public exponent	gcd(e,φ(n)) = 1
RSA private exponent	ed ≡ 1 (mod φ(n))
RSA encryption	C = M^e mod n
RSA decryption	M = C^d mod n
Discrete logarithm	g^x ≡ y (mod p)

18. Quick Comparison Tables

18.1 Prime vs Relatively Prime

Prime Number	Relatively Prime Numbers
Property of one number.	Relationship between two or more numbers.
Has exactly two positive divisors.	GCD of the numbers is 1.
Example: 7.	Example: 8 and 15.

18.2 Fermat vs Euler

Fermat's Little Theorem	Euler's Theorem
a^(p−1) ≡ 1 (mod p)	a^φ(n) ≡ 1 (mod n)
Uses a prime modulus p.	Works for general n when gcd(a,n)=1.
Special case.	More general result.

18.3 Symmetric vs Public-Key

Symmetric Cryptography	Public-Key Cryptography
Uses shared secret key material.	Uses public/private key pair.
Generally faster.	Generally more computationally expensive.
Key distribution is a major challenge.	Public key can be distributed openly.
Example: AES.	Example: RSA.

18.4 AES vs RSA

AES	RSA
Symmetric-key algorithm.	Public-key algorithm.
Block cipher.	Public-key cryptosystem.
128-bit block size.	Uses modulus n = pq.
128, 192 or 256-bit keys.	Security depends strongly on key generation and factorization difficulty.

19. Important Examination Questions

Short-Answer Questions

- 1. What is modular arithmetic?
- 2. Define a prime number.
- 3. What are relatively prime numbers?
- 4. What is a group?
- 5. What is an Abelian group?
- 6. Define a field.
- 7. What is GF(p)?
- 8. What is the Extended Euclidean Algorithm?
- 9. State Fermat’s Little Theorem.
- 10. State Euler’s theorem.
- 11. What is Euler’s totient function?
- 12. What is the Chinese Remainder Theorem?
- 13. Define the Discrete Logarithm Problem.
- 14. What is a public-key cryptosystem?
- 15. What is RSA?
- 16. What is the block size of AES?

Long-Answer / Numerical Questions

- 1. Explain modular arithmetic and modular multiplicative inverse with examples.
- 2. Explain groups and fields and state their important properties.
- 3. Explain finite field GF(p) with suitable examples.
- 4. Use the Extended Euclidean Algorithm to find a modular inverse.
- 5. State and explain Fermat’s Little Theorem with an example.
- 6. Explain Euler’s theorem and Euler’s totient function.
- 7. Solve a system of congruences using the Chinese Remainder Theorem.
- 8. Explain the Discrete Logarithm Problem and its role in cryptography.
- 9. Explain the structure and encryption process of AES.
- 10. Explain AES decryption and its inverse transformations.
- 11. Explain the complete RSA key-generation procedure.
- 12. Perform RSA encryption and decryption using a small numerical example.
- 13. Explain the security of RSA and the importance of secure key generation.
- 14. Explain RSA padding, side-channel concerns and RSA-CRT.
- 15. Compare symmetric and public-key cryptography.

Unit II – Final Revision

The mathematical foundation of Unit II is modular arithmetic, prime numbers, relatively prime numbers, groups, fields, finite fields, GCD and modular inverse.

AES is the major symmetric block cipher topic. Remember that AES always uses a 128-bit block and supports 128-, 192- and 256-bit keys.

Fermat's and Euler's theorems provide important modular relationships. Euler's totient function is especially important in RSA.

CRT provides an efficient method for solving simultaneous congruences and is also used to accelerate RSA private-key operations.

The discrete logarithm problem is the basis of several public-key cryptographic constructions.

RSA uses two large primes, $n = pq$, $\varphi(n) = (p-1)(q-1)$, a public exponent e and a private exponent d satisfying $ed \equiv 1 \pmod{\varphi(n)}$.

RSA: p,q → n,φ(n) → e → d → Public Key (e,n), Private Key (d,n)

For examinations, practise both conceptual answers and numerical problems involving modular arithmetic, EEA, CRT and RSA.

Note: These notes are written in simple language for learning and examination preparation. Real cryptographic implementations should use current standards, secure libraries and appropriate parameter sizes rather than classroom-sized examples.