Department of Computer Science & Engineering Details of Lesson Plan

S.No.	Particulars	Details
1.	Course Name	Fundamentals of Cryptography
2.	Course Code	COM-403
3.	Academic Year	2024-2025
4.	Semester	4
5.	Number of Lesson Plans	45
6.	Faculty Assigned	Dr. Mir Aadil

Faculty Signature

Lesson Plan No. 1.1	**Course Name: Fundamentals of Cryptography** **Topic: Integer Arithmetic (Mathematical Foundations).**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Perform basic operations with integers (addition, subtraction, multiplication). b. Understand properties such as associativity, commutativity, and distributivity. c. Apply these properties in arithmetic expressions.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Integer Arithmetic examples
Teaching Development	**1. Introduction (5 minutes)** • Begin by creating a relatable context for integer operations. Pose a question like, “If you have a debt of 10 and only 3 in your account, where do you stand?” Use stories involving temperatures, finances, or sports scores. **2. Development (30 minutes)** • Transition into a discussion about the limitations of whole numbers and the introduction of negative numbers. Review set of whole numbers and the necessity for negative numbers. • Encourage student participation through group-based exploration of the number line and interactive tools. Define integers and notation, Explore operations using a number line, Prove properties with examples, Real-world applications • Next, guide students through discovering the properties of integer operations (commutative, associative, distributive) using visual models and peer discussion. Include moments where students can explain concepts in their own words or to a partner **3. Exercise (5 minutes)** Invite students to reflect by asking: “What surprised you about negative numbers today?” or “Where might you use this knowledge outside math class?” Encourage peer sharing and affirm diverse responses.
Closure	• Conclude with a summary linking integer arithmetic to real-world applications, such as banking, coding and encryption. Explain how today’s lesson connects to algebraic thinking.
Evaluation	Ask students to create and solve their own word problem involving integer operations.

Lesson Plan No. 1.2	**Course Name: Fundamentals of Cryptography** **Topic: Set of Integers & Binary Operations.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of a mathematical set. b. Identify the set of integers as a closed system under binary operations. c. Classify operations as binary and explore their properties.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Venn diagrams and Operation examples.
Teaching Development	**1. Introduction (5 minutes)** • Begin with a simple, tangible example: "If a sandwich and a drink cost Rs. 100, how many such combos can you make with Rs. 500?" Lead into the idea of sets and operations. Use a hands-on activity like sorting numbers with cards to illustrate set membership. Encourage students to define sets in their own words and find real-life analogies (e.g., a set of even numbers as members of a sports team). **2. Development (30 minutes)** • Define integers and notation, Introduce Z (integers) • Explore operations using number line • Prove properties with examples, define binary operations, Explore properties: closure, associativity, identity, inverse • Introduce binary operations through storytelling (e.g., "Imagine two machines that combine ingredients differently, what rules do they follow?"). Foster a collaborative learning environment where students classify operations based on observed patterns. Use Venn diagrams and interactive digital tools for exploration. **3. Exercise (5 minutes)** • Quick problem set on integer operations
Closure	• Importance of integer arithmetic • Guide a reflective wrap-up: "If you had to explain a binary operation to your sibling, how would you do it?" Encourage students to draw or act out an example. Summarize the key takeaways and point forward to how these operations will show up in algebra and computer science.
Evaluation	• Ask students to complete a chart matching operations with properties (e.g., associative, identity). • 5 Question Mini Quiz

Lesson Plan No. 1.3	**Course Name: Fundamentals of Cryptography** **Topic: Integer Arithmetic (Integer Division & Divisibility).**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Understand the division algorithm. b. Define divisibility and apply rules of divisibility. c. Perform integer division and calculate quotient and remainder.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with examples. c. Interactive calculator
Teaching Development	**1. Introduction (5 minutes)** • Ask: "What happens when we subtract 10 from 3?" • Show real-life examples involving debt and temperature. • Review set of whole numbers and the necessity for negative numbers. • Begin by recalling real-life experiences like dividing candies among friends or fitting items into containers. Ask: "Can we always divide things evenly?" **2. Development (30 minutes)** • Define integers and notation, and Explore operations using number line • Prove properties with examples, • Division algorithm, a divides b, Basic divisibility rules • Introduce the concept of divisibility with relatable examples (e.g., grouping chairs or books). Use physical objects or visuals to illustrate the division algorithm. In small groups, students test divisibility rules on fun datasets like their birthdates or classroom numbers. Encourage peer discussion to clarify misunderstandings. **3. Exercise (5 minutes)** • Invite students to summarize what divisibility means to them. Use questions like: "How does the remainder help us in understanding a division problem?"
Closure	• Wrap up by connecting this topic to factors and multiples, which will be useful in understanding modular arithmetic. • Assign a creative task like: "List 5 real-world situations where something is or isn't divisible."
Evaluation	• Use a short reflection: "What makes a number divisible by 3 or 5?" • Pair students to quiz each other using divisibility rules.

Lesson Plan No. 1.4	Course Name: Fundamentals of Cryptography Topic: Linear Diophantine Equations.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Define Diophantine equations. b. Solve linear Diophantine equations in two variables. c. Understand conditions for solvability.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with LDE c. Graph plotting tool
Teaching Development	**1. Introduction (5 minutes)** • Begin with a puzzle or riddle: “If I have only Rs. 2 and Rs. 5 coins, how do I make Rs. 17?” Coin combination problem (Rs. 5 & Rs. 2 = Rs. 17). • Use this to introduce the idea of equations with multiple solutions. Encourage students to act out or sketch possible coin combinations to visualize the concept. **2. Development (30 minutes)** • Define and solve Diophantine equations, GCD criteria, Graphical & algebraic methods. • Walk through solving linear Diophantine equations step-by-step with real-life context tickets, prices, budgeting. Use graphical methods and draw number lines or grids to show how combinations form a pattern. • Promote collaborative problem-solving by assigning each group an equation and asking them to solve and explain it creatively (e.g., as a story or visual). **3. Exercise (5 minutes)** • Ask reflective questions like: “How did it feel to find more than one solution to a problem?”
Closure	• Importance of integer arithmetic • Solution strategy summary • Connect the mathematical strategy to real-world problem solving, such as coding combinations or optimizing purchases.
Evaluation	• Reflective question: Reflect on solving approach. • 5 Question Mini Quiz

Lesson Plan No. 1.5	Course Name: Fundamentals of Cryptography Topic: Modulo Operator & Modular Arithmetic.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Understand modulo operation. b. Perform arithmetic operations using mod. c. Explore equivalence classes.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with: Modular clock visual Coding examples
Teaching Development	**1. Introduction (5 minutes)** • Start with a fun, concrete example like a clock: "If it's 9 o'clock now, what time will it be in 7 hours?" Use the clock to illustrate how numbers "wrap around" — the essence of mod. **2. Development (30 minutes)** • Let students physically act out counting around a circular object to understand mod. Then introduce the formal definition with relatable examples (e.g., days of the week, repeating patterns). What does 'mod 10' mean? Use clock analogy. • Define mod, Examples and Perform operations • Have students solve small mod problems using color-coded number circles. Encourage peer teaching: "Explain to your partner how mod works in your example." **3. Exercise (5 minutes)** • Quick problem set on integer operations
Closure	• Ask: "What's something you already do in life that's like 'mod' without realizing it?" • Summarize key concepts using a visual mod circle or digital display. • Link modular arithmetic to applications like cryptography and computer science.
Evaluation	• Reflective question: "Think of 3 situations in daily life where modular arithmetic applies and explain each briefly." • Modular Arithmetic Exercises

Lesson Plan No. 1.6	**Course Name: Fundamentals of Cryptography** **Topic: Set of Residues (Zn) & Congruence.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Perform basic operations with integers (addition, subtraction, multiplication). b. Understand properties such as associativity, commutativity, and distributivity. c. Apply these properties in arithmetic expressions.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Color-coded residue tables c. Hands-on congruence cards
Teaching Development	**1. Introduction (5 minutes)** • Begin with a hands-on question: "Which numbers give the same remainder when divided by 4?" • Use colored sticky notes or cards to sort numbers into residue classes. **2. Development (30 minutes)** • Introduce the set Zn using visuals and group discussions. • Make learning active by allowing students to build a "residue table" together. • Emphasize congruence through games: "Match pairs of numbers that are congruent mod 3." **3. Exercise (5 minutes)** •Quick class quiz using examples and non-examples of congruent numbers.
Closure	• Ask: "What does it mean when we say two numbers are congruent?" • Link this to broader uses like hashing and error detection in data systems. • End with a creative analogy (e.g., locker numbers, game levels).
Evaluation	• Exercise for congruent numbers in Z5 and Z6.

Lesson Plan No. 1.7	Course Name: Fundamentals of Cryptography Topic: Operations in Zn with Inverses.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Perform addition and multiplication in Zn. b. Understand concept of inverses in modular systems. c. Identify when inverses exist.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Inverse finder table c. Inverse group activity
Teaching Development	**1. Introduction (5 minutes)** • Ask a thinking question: "Can we always undo multiplication?". **2. Development (30 minutes)** • Lead students through operations in Zn using color-coded tables and "inverse finder" charts. Use real-world analogies like reversing a combination lock. • In small groups, let students explore the Extended Euclidean Algorithm with guided steps. Highlight "when and why" inverses exist in Zn. **3. Exercise (5 minutes)** •List all invertible elements in Z8 and justify your answers.
Closure	• Ask: "What surprised you about inverses in modular arithmetic?" • Discuss why understanding inverses is essential in cryptography and digital security. • Have students summarize key steps of finding an inverse to a partner.
Evaluation	• Evaluate the exercise given.

Lesson Plan No. 1.8	**Course Name: Fundamentals of Cryptography** **Topic: Addition & Multiplication Tables in Zn**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Perform basic operations with integers (addition, subtraction, multiplication). b. Understand properties such as associativity, commutativity, and distributivity. c. Apply these properties in arithmetic expressions.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with blank tables and Color-coded charts
Teaching Development	**1. Introduction (5 minutes)** • Open with a question: “Why do we need special tables for modular arithmetic?”. **2. Development (30 minutes)** • Use blank charts and guide students to build Zn tables from scratch (e.g., Z5, Z6). Encourage group work to identify patterns, identities, and inverses. • Discuss structure and symmetry. **3. Exercise (5 minutes)** •Construct the addition and multiplication tables for Z9 and describe what you observe
Closure	• Ask: “How is a Zn table similar to or different from the tables we use in regular arithmetic?” • Relate it to system design in computing and secure communications.
Evaluation	• How does Z5 compare to Z6? What’s different about inverses?

Lesson Plan No. 2.1	Course Name: Fundamentals of Cryptography Topic: Introduction to Encryption & Secret-Key Encryption.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic purpose of encryption. b. Explain the concept of secret-key (symmetric) encryption. c. Identify real-world applications of secret-key encryption.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Visual diagrams and Caesar cipher demo tool
Teaching Development	**1. Introduction (5 minutes)** • Begin with an engaging prompt: “Have you ever written a secret message only your friend could read?” • Introduce the concept of secrecy and encryption using a simple Caesar cipher puzzle. **2. Development (30 minutes)** • Encourage students to solve and create their own. • Guide them to discover the role of secret keys and explore the pros/cons of symmetric encryption using real-life examples like digital lockers or Bluetooth pairing. • Use a visual story or analogy, such as passing a note with a shared lock, to reinforce the concept. **3. Exercise (5 minutes)** • Quick problem set on integer operations
Closure	• Ask students to reflect: “How might secret-key encryption be used in everyday apps?” • Allow quick sharing and group summary of the key concept. Reinforce the takeaway: “If both people share the key, it’s symmetric encryption.”
Evaluation	• Use any alphabet-based cipher to encrypt your name and share the key logic with a peer. • Reflection Question: “Why is secrecy not enough without key protection?”

Lesson Plan No. 2.2	**Course Name: Fundamentals of Cryptography** **Topic: Public-Key Encryption.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Explain the concept of asymmetric encryption. b. Understand how public and private keys work together. c. Identify advantages of public-key encryption.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with RSA animation c. Padlock analogy video
Teaching Development	**1. Introduction (5 minutes)** • Start with a puzzle: "Can someone send a secure message without having met you?" • Use the padlock analogy — everyone has a public lock (key), but only the owner has the private key. **2. Development (30 minutes)** • Demonstrate RSA key pairs visually or through a simulation. • Encourage group exploration: Assign each pair a public/private key and have them simulate encrypting a message. • Highlight the real-world use (e.g., HTTPS in browsers, secure emails). **3. Exercise (5 minutes)** • Write down three benefits of public-key encryption with one practical example for each.
Closure	• Ask: "When might public-key encryption be better than shared secret keys?" • Link the idea to trust and authentication online. Allow students to brainstorm places they use it unknowingly.
Evaluation	• Scenario-based quiz: "Would you use symmetric or public-key in this case?"

Lesson Plan No. 2.3	Course Name: Fundamentals of Cryptography Topic: Block Ciphers	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Define block cipher encryption. b. Understand how plain text is divided and encrypted in blocks. c. Explore basic block cipher modes of operation.
Teaching Aids (if any)	d. Chalkboard/Whiteboard e. Slides AES block diagram. ECB vs CBC visual aid.
Teaching Development	**1. Introduction (5 minutes)** • Start by asking: "What if your message is too long to send securely all at once?" • Explain block ciphers with a paper-cutting analogy. The analogy can be breaking a message into blocks, like slicing a loaf of bread. **2. Development (30 minutes)** • Define block cipher • Block size • Modes: ECB, CBC, CFB • Use a DES and an AES diagram and show how different modes (ECB, CBC) handle blocks differently. • Use a visual race analogy (e.g., relay vs solo runners) to explain chaining in CBC. **3. Exercise (5 minutes)** • Write a few lines comparing ECB and CBC with an example.
Closure	• Ask: "Which mode do you think is more secure and why?" • Compare student reasoning and link to real-world threats like pattern exposure. • Homework: Compare ECB vs CBC for security
Evaluation	• Ask difference between block size and key size

Lesson Plan No. 2.4	Course Name: Fundamentals of Cryptography Topic: Stream Ciphers.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Explain the concept of stream cipher. b. Compare block and stream ciphers. c. Identify use-cases for stream ciphers.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Key stream animation, RC4 demo
Teaching Development	**1. Introduction (5 minutes)** • Begin with: "What if we want to encrypt something letter by letter, or even bit by bit?" Why do we even need such a thing. **2. Development (30 minutes)** • Introduce stream ciphers using an analogy like a faucet dripping data one bit at a time. Client Sever, Music and Video Stream • Use XOR truth tables and visual animations to show how a keystream works. • Compare with block ciphers to highlight flexibility. Let students try encrypting a short sentence using a given key stream and XOR operation. **3. Exercise (5 minutes)** • Write one benefit and one drawback of using a stream cipher over a block cipher.
Closure	• Ask: "In what kind of systems would stream ciphers be useful?" • Guide students to reflect on live communication tools, radio signals, or video streaming.
Evaluation	• Binary XOR quiz with student-generated key streams.

Lesson Plan No. 2.5	Course Name: Fundamentals of Cryptography Topic: Hybrid Encryption.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Define hybrid encryption and why it's used. b. Understand how symmetric and asymmetric encryption are combined. c. Explore practical use-cases like SSL/TLS.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with SSL/TLS example, Digital envelope diagram, etc.
Teaching Development	**1. Introduction (5 minutes)** • Pose a problem: “Symmetric is fast, public-key is secure — can we use both?” • A small debate to ignite the interest. **2. Development (30 minutes)** • Introduce hybrid encryption with real-world analogies (e.g., sending a gift with a lock and key). Show how symmetric encryption handles speed, and public-key ensures secure key exchange. • Walk through HTTPS or SSL/TLS handshake using visual diagrams. Students can role-play the encryption process: one plays a sender, another a receiver, and others observe and explain each step. • Discuss real applications like secure messaging apps and bank transactions. **3. Exercise (5 minutes)** • Quick problem set on integer operations
Closure	• Ask students: “Why is hybrid encryption common in modern systems?” • Invite them to draw a diagram of the encryption flow or explain it in pairs. Link the idea to daily internet use.
Evaluation	• Diagram labeling exercise for phases. • Exit prompt: “What’s the biggest strength of hybrid encryption?”

Lesson Plan No. 2.6	Course Name: Fundamentals of Cryptography Topic: Message Authentication Codes (MACs).	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Define message authentication and its importance. b. Explain the use of MACs for data integrity. c. Demonstrate generation and verification of a MAC.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with HMAC visualization c. Python demo of hash + key
Teaching Development	**1. Introduction (5 minutes)** • Begin by asking: “How do you know if a message was changed?” • Data Integrity and verification for tampering. **2. Development (30 minutes)** • Introduce the idea of authenticity and integrity using a sealed envelope analogy. Demonstrate how MACs work using HMAC + SHA-256 with a live demo or animation. • Types of MAC. • Let students simulate MAC creation and verification using Paper and pen representing message and key. Reinforce understanding with examples of real-world systems that use MACs (e.g., financial transactions, software updates). **3. Exercise (5 minutes)** •Ask students: “How is MAC different from encryption?”
Closure	• Allow small group discussion and summary of the MAC process. • Link it to trust in digital communications.
Evaluation	• Reflective question: Find 2 situations in life where integrity matters and describe how MACs could be useful.

Lesson Plan No. 2.7	**Course Name: Fundamentals of Cryptography** **Topic: Non-repudiation in Cryptography.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Define non-repudiation and its significance. b. Explain how digital signatures support non-repudiation. c. Identify legal applications of non-repudiation.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with circuit diagrams
Teaching Development	**1. Introduction (5 minutes)** • What is a signature and what is its significance. • Ask: "Can someone deny sending a signed email?". **2. Development (30 minutes)** • Introduce digital signatures as a combination of encryption and hashing. • Break down the process visually and with analogies (e.g., fingerprints). • Explain how digital signatures are created. • Explain Public Key Infrastructure (PKI). **3. Exercise (5 minutes)** •Write about a scenario where non-repudiation prevents serious issues.
Closure	• Prompt reflection: "When have you needed proof that someone said or did something?" • Link the concept to trust in systems like banking, emails, or agreements. Summarize using a real-world case example.
Evaluation	• Quick quiz on key terms (signing, verifying, denial). • Exit ticket: "What's one reason digital signatures are essential?"

Lesson Plan No. 2.8	Course Name: Fundamentals of Cryptography Topic: Digital Certificates.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Define digital certificates and Certificate Authorities (CAs). b. Understand certificate structure (X.509). c. Explain how browsers validate HTTPS using certificates.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with circuit diagrams
Teaching Development	**1. Introduction (5 minutes)** • Why do we trust a website with our card details? Who provides surety and authentication. **2. Development (30 minutes)** • Introduce digital certificates using the browser lock icon as an anchor. Explain Certificate Authorities (CAs) and the idea of a chain of trust using a trust pyramid analogy. • Use screenshots from real websites to dissect certificates (issuer, expiry, subject). • Guide students in checking a real certificate and interpreting its components. **3. Exercise (5 minutes)** • Students can investigate and share certificate info from their favorite sites.
Closure	• Ask: “What surprised you about how trust works online?” • Discuss expiry dates, revocation, and how browsers react to invalid certificates. Summarize using a trust chain diagram.
Evaluation	• Class discussion: “What if a CA is compromised?” and • “You see a certificate warning—what do you do?”

Lesson Plan No. 3.1	**Course Name: Fundamentals of Cryptography** **Topic: The Shift Cipher (Caesar Cipher).**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of Caesar shift cipher. b. Encrypt and decrypt messages using shift cipher. c. Recognize vulnerabilities in simple shift ciphers.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Caesar cipher wheel and Encryption chart
Teaching Development	**1. Introduction (5 minutes)** • Start with a storytelling hook: “Did you know Julius Caesar had a secret way to send messages?” **2. Development (30 minutes)** • Introduce the Caesar cipher with a hands-on puzzle. Use cipher wheels or strips and let students encode/decode simple words. • Guide them to understand the math behind the cipher: E(x) = (x + k) mod 26. Let them work in pairs to encrypt and decrypt using different keys and analyze patterns. • Use class discussion to highlight simplicity and weaknesses **3. Exercise (5 minutes)** Ask: “Why might Caesar’s cipher fail today?” Cryptanalysis of Key space.
Closure	Connect to how encryption has evolved. Let students summarize the pros and cons of the cipher
Evaluation	• Encrypt a short sentence using a Caesar shift of 5 and bring it to class for a peer to decrypt.

Lesson Plan No. 3.2	**Course Name: Fundamentals of Cryptography** **Topic: Substitution Cipher.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Define monoalphabetic substitution cipher. b. Encrypt messages using random key mappings. c. Understand frequency analysis vulnerability.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Substitution chart, Frequency Table of English letters
Teaching Development	**1. Introduction (5 minutes)** • Begin with a question: “What if we shuffled the whole alphabet instead of just shifting it?” • Will it save Caesar message. **2. Development (30 minutes)** • Introduce monoalphabetic substitution using mapping tables. • Mathematics behind it. • Discuss how letter frequency reveals patterns. Use frequency charts and famous quotes to demonstrate attacks on substitution ciphers. • Let students work in small groups to break a short, encoded message using frequency analysis. **3. Exercise (5 minutes)** • Do a frequency analysis of a paragraph from your favorite book.
Closure	• “What clues helped you break the substitution cipher?” • Link the activity to language patterns and cryptanalysis. Let students reflect on the difficulty of keeping a substitution cipher secret.
Evaluation	• Pair quiz: Break each other’s cipher. • Reflection: “What made some letters easier to decode?”

Lesson Plan No. 3.3	Course Name: Fundamentals of Cryptography Topic: Affine Cipher	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Explain the mathematics behind the affine cipher. b. Perform encryption and decryption using affine formulas. c. Understand invertibility and modular arithmetic.
Teaching Aids (if any)	c. Chalkboard/Whiteboard d. Slides with Modulo arithmetic chart, Affine calculator
Teaching Development	**1. Introduction (5 minutes)** • Begin with a scenario: “What if we combine multiplication and addition in a cipher?” • Discuss what purpose will it serve? **2. Development (30 minutes)** • Introduce the formula: E(x) = (ax + b) mod 26. Demonstrate with small values of a and b. • Let students explore which values of a are valid (must be co-prime with 26). Use visual tools or interactive calculators to experiment with different keys. **Exercise (5 minutes)** • Group work: Encrypt and decrypt words using chosen (a, b) pairs.**3.** • Quick problem set on integer operations
Closure	• What happens if ‘a’ and 26 share a factor?” • Summarize how the mathematical structure affects cipher strength and decryption. Link to modular arithmetic foundations.
Evaluation	• Group challenge: Identify valid keys. • Oral quiz: Explain each part of the affine formula.

Lesson Plan No. 3.4	**Course Name: Fundamentals of Cryptography** **Topic: Hill Cipher.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Define Hill cipher using linear algebra. b. Encrypt using matrix multiplication mod 26. c. Identify security challenges and key matrix rules.
Teaching Aids (if any)	a. 2x2 matrix visual aid b. Slides with Modular inverse calculator
Teaching Development	**1. Introduction (5 minutes)** • Ask: “Can we use matrices to encrypt?” • If yes, what can be the confusion and diffusion benefits. **2. Development (30 minutes)** • Introduce the Hill cipher using 2×2 matrices. Walk students through multiplying a key matrix with a vector (plaintext letters). Provide practice blocks with pre-selected keys. • Use visuals and animations to explain matrix multiplication mod 26. • In groups, have students encrypt a word, then decode it using the inverse matrix. Use colored grids and real-time examples to help them visualize the operation. **3. Exercise (5 minutes)** • Quick problem set on integer operations
Closure	• Prompt reflection: “What challenges did you face with matrix encryption?” • Connect to areas where matrix math is used in modern encryption. Highlight the importance of invertibility. • Homework: “Find the modular inverse of a 2×2 matrix using your class notes or a calculator.”1
Evaluation	• Encrypt a word and verify each step. • Short quiz on identifying valid Hill matrices.

Lesson Plan No. 3.5	**Course Name: Fundamentals of Cryptography** **Topic: Permutation Cipher (Transposition).**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Understand transposition (permutation) ciphers. b. Apply key-based rearrangement of characters. c. Explore columnar transposition.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Grid chart, Encryption demo
Teaching Development	**1. Introduction (5 minutes)** • Start with an activity: "Let's rearrange a sentence so it looks like nonsense but keeps all the same letters!" **2. Development (30 minutes)** • Explain transposition ciphers using a grid system. Walk students through a step-by-step encryption of a word using a given permutation key (e.g., 4312). • Have students work in pairs: one encrypts using a grid, the other tries to decrypt without the key. Discuss patterns and difficulties. • Use a puzzle or word scramble to make the concept interactive and tactile. **3. Exercise (5 minutes** •Decrypt a message encrypted with a given permutation key.
Closure	• Ask: "What made this cipher different from substitution?" • Encourage students to compare strengths and weaknesses. Summarize that transposition rearranges symbols but doesn't change them.
Evaluation	• Group activity: Encrypt and decrypt short phrases using transposition. • Exit prompt: "What's one strength and one weakness of a permutation cipher?"

Lesson Plan No. 3.6	**Course Name: Fundamentals of Cryptography** **Topic: Stream Cipher (Reinforcement).**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. a. Review concept of stream cipher. b. b. Encrypt using XOR and pseudo-random key streams. c. c. Discuss synchronization and key generation.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with XOR truth table and Bitwise encryption tools
Teaching Development	**1. Introduction (5 minutes)** • Begin with: “If you send a message letter by letter, how do you keep it safe?” • Review stream ciphers using real-world examples like live video streaming or walkie-talkies. **2. Development (30 minutes)** • Explain the role of XOR with key streams. • Guide students to encrypt a binary string using XOR and a pseudo-random key stream. Provide visual XOR tables and hands-on examples. • Group task: Encrypt and decrypt using different key streams and compare results. **3. Exercise (5 minutes)** •Encrypt your name in binary using XOR with a simple key stream.
Closure	• Prompt reflection: “How is stream encryption like a secret radio channel?” • Reinforce how synchronization and randomness are essential. Let students share which systems they think use stream ciphers.
Evaluation	• XOR worksheet and decoding challenge. • Reflection: “What happens if sender and receiver lose sync?”

Lesson Plan No. 3.7	**Course Name: Fundamentals of Cryptography** **Topic: Cryptanalysis: Affine & Substitution Cipher.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Learn methods to break substitution and affine ciphers. b. Use frequency analysis and known plaintext. c. Recognize limits of classical cryptography.
Teaching Aids (if any)	c. Chalkboard/Whiteboard d. Slides with Frequency histogram, etc. e. Affine cipher cracker tool
Teaching Development	**1. Introduction (5 minutes)** • Begin with a challenge: “Can you break this message without a key?” using an example of frequency analysis. **2. Development (30 minutes)** • Introduce frequency analysis with a simple encrypted message. Use bar graphs or histograms to compare letter frequencies. • Let students work in teams to crack a substitution or affine cipher using logic and trial. Provide guided steps and frequency hints. • Draw parallels between cryptanalysis and solving a mystery—what clues help? Reinforce strategies like guessing common letters (E, T, A). **3. Exercise (5 minutes)** • Try decoding a cipher text using only frequency patterns and describe your steps.
Closure	• Ask: “What made cryptanalysis easier or harder today?” • Allow teams to share which strategies worked. Emphasize how patterns in language are both helpful and risky.
Evaluation	• Decoding challenge with support tools. • Group discussion: “Why are simple ciphers insecure?” • Short quiz: Match ciphertexts to likely decryption strategies.

Lesson Plan No. 3.8	Course Name: Fundamentals of Cryptography Topic: Cryptanalysis: Vigenère, Hill, One-Time Pad.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Analyze classical ciphers like Vigenère and Hill. b. Understand brute-force and Kasiski method. c. Realize the strength of the One-Time Pad.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Vigenère square, Hill cipher matrices, OTP visuals, etc.
Teaching Development	**1. Introduction (5 minutes)** • Open with a critical thinking question: “If substitution is weak, what if we rotate the key?” • Introduce the Vigenère cipher using a keyword and Vigenère square. **2. Development (30 minutes)** • Guide students to encrypt and decrypt short messages using repeated keys. • Explain how longer keys make cryptanalysis harder, and demonstrate the Kasiski method for breaking Vigenère. • Move to Hill and One-Time Pad, showing how complexity increases. Emphasize OTP’s perfect security—if used correctly. **3. Exercise (5 minutes)** • Break students into groups: one analyzes Vigenère, another tries Hill, and one explores why OTP is unbreakable (with discussion).
Closure	• Ask: “Which cipher felt strongest, and why?” • Summarize that security improves with randomness, key length, and structural complexity. Highlight trade-offs between usability and security. • Homework: “Research a historical use of the One-Time Pad and summarize it in 100 words.”
Evaluation	• Case study quiz: Match cipher type to its weakness. • Peer presentations on how they cracked each cipher.

Lesson Plan No. 4.1	**Course Name: Fundamentals of Cryptography** **Topic: Piling-up Lemma.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Understand the statistical foundation of the piling-up lemma. b. Apply the lemma in estimating the bias of linear approximations. c. Recognize its role in linear cryptanalysis..
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Visual of XOR and probability table c. Python code demo (Jupyter Notebook)
Teaching Development	**1. Introduction (5 minutes)** • Start with a guiding question: “How do small biases add up in cryptography?” • Use a simple XOR game to demonstrate the idea of combining bits and detecting patterns. **2. Development (30 minutes)** • Introduce the Piling-up Lemma step-by-step using a visual probability tree. • Explain each component (bias, XOR, independence) using color-coded examples. Pair students for a hands-on activity: compute the combined bias from individual XORs. • Encourage students to work through practical scenarios from cryptanalysis and reflect on why small patterns matter in big systems. **3. Exercise (5 minutes)** • Try calculating the bias of a 3-variable XOR expression and explain your steps.
Closure	• Ask: “What does the Piling-up Lemma teach us about cipher strength?” • Connect this idea to real-world implications—where seemingly secure systems can leak information due to bias.
Evaluation	• Oral Q&A on conceptual understanding of bias and lemma. • Small group task: Walk through a bias computation and explain logic.

Lesson Plan No. 4.2	Course Name: Fundamentals of Cryptography Topic: Linear Approximations of S-boxes	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Understand S-box structures b. Learn to derive linear approximations c. Evaluate approximation biases.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Sample S-box c. Excel sheet for bias calculations
Teaching Development	**1. Introduction (5 minutes)** • Start with a digital metaphor: “Can a complicated function still be predicted a little bit?” • Introduce the structure and role of S-boxes using simplified examples.. **2. Development (30 minutes)** • Use grids and input/output tables to demonstrate mappings. • Guide students to build a Linear Approximation Table (LAT) for a 4-bit S-box using guided steps. Encourage teamwork and pattern recognition. • Explain approximation bias with support visuals and statistical plots to show probability differences. **3. Exercise (5 minutes)** • Choose a new 4-bit S-box and build a LAT with approximation bias values.
Closure	• Ask: “How do linear approximations help us analyze complex ciphers?” • Let students summarize what bias means in this context and how it helps break down a block cipher.
Evaluation	• Group discussion on emerging patterns from LATs. • Quick quiz: Identify correct approximations and bias signs.

Lesson Plan No. 4.3	**Course Name: Fundamentals of Cryptography** **Topic: Data Encryption Standard (DES).**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Understand DES structure and Feistel network b. Trace round operations c. Explain key scheduling and weaknesses
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Diagram of DES, Rounds and Key Generation c. Online DES simulation tool
Teaching Development	**1. Introduction (5 minutes)** • Ask: “What makes an old cipher like DES still important today?” • Introduce DES structure with a storytelling angle: each round is like a stage in a spy mission. Break down Feistel structure using animations or drawings.. **2. Development (30 minutes)** • Guide students through one round of DES manually—Expansion, XOR, S-box, P-box. Use hands-on materials like puzzle cards or diagramming tools. • Have students collaborate in solving one round of DES using a simplified key and block, then present their steps to the class. **3. Exercise (5 minutes)** • Draw and label each part of one round of DES. Add a note on what each part does.
Closure	• Ask: “What’s one thing DES teaches us about building ciphers?” • Link to the need for complexity and key security. Point out DES limitations (key size, susceptibility to brute force).
Evaluation	• Worksheet on DES operations. • Reflection prompt: “What was the hardest part of tracing a round?”

Lesson Plan No. 4.4	Course Name: Fundamentals of Cryptography Topic: Advanced Encryption Standard (AES).	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Learn AES architecture and round structure b. - Understand SubBytes, ShiftRows, MixColumns, AddRoundKey c. - Compare with DES
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Animated AES architecture, round steps, SubBytes, ShiftRows, MixColumns, AddRoundKey etc. c. NIST AES animation
Teaching Development	**1. Introduction (5 minutes)** • Open with a comparison question: "How does AES fix the problems DES had?" **2. Development (30 minutes)** • Introduce AES visually using an animation or diagram of its 4 core operations: SubBytes, ShiftRows, MixColumns, AddRoundKey. • Walk through a single AES round with color-coded blocks. Use hands-on practice with small matrices or digital block editors. • Divide students into groups: each handles one transformation and then teaches it to the rest of the class. Highlight the substitution-permutation structure. **3. Exercise (5 minutes)** • Compare AES to DES in terms of structure, key length, and resistance to attack.
Closure	• "Which AES step do you think is the most powerful against attackers, and why?" • Summarize the modular and layered nature of AES. Link to security strength and speed.
Evaluation	• Reflective quiz on transformation purposes. • Peer assessment: Groups review each other's explanations of AES rounds.

Lesson Plan No. 4.5	**Course Name: Fundamentals of Cryptography** **Topic: Hash Functions and Data Integrity.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Define hash functions and properties (collision, pre-image) b. Understand integrity checks using hashes c. Analyze real-life applications
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Hash function demo (MD5, SHA1). c. Integrity check example using files
Teaching Development	**1. Introduction (5 minutes)** • Start with a real-world hook: “How can you be sure a file hasn’t been tampered with?” • Introduce hash functions using analogies (e.g., fingerprint or barcode). **2. Development (30 minutes)** • Show how a tiny change leads to a completely different hash. • Use tools like *sha256sum* to hash files and compare outputs. Let students modify a simple text file and observe the difference. • Demonstrate real-life examples like Git, digital receipts, and secure downloads. • Students work in small teams to create and compare hashes for documents, emphasizing immutability and uniqueness. **3. Exercise (5 minutes)** List three examples of where data integrity is important and how hashes might help.
Closure	• “Why is it hard to reverse a hash?” • Encourage sharing of how hashes are useful in everyday digital life (e.g., password storage). Summarize the key properties: deterministic, irreversible, and collision-resistant.
Evaluation	• Matching quiz: Terms vs properties (e.g., collision, pre-image resistance).

Lesson Plan No. 4.6	**Course Name: Fundamentals of Cryptography** **Topic: SHA-512.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Learn the internal structure of SHA-512 b. - Understand bit operations and message padding c. - Analyze differences with SHA-256
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with SHA-512 block diagram diagrams, round diagram and word generation Logic. c. Algorithm Walkthrough.
Teaching Development	**1. Introduction (5 minutes)** • Pose a guiding question: “How does SHA-512 process a large message securely?” **2. Development (30 minutes)** • Use a diagram to walk through initial constants, message padding, and 80-round computation. Break it down into understandable phases: pre-processing, block expansion, and round transformation. • Provide a code walkthrough or online tool for students to step through hashing in real time. • Encourage students to pair up and trace a simple hash transformation on a small input message. • Use structured handouts and worksheets for processing bit-level operations (XOR, AND, shifts). **3. Exercise (5 minutes)** • How SHA-512 handles large messages.
Closure	• Ask: “What makes SHA-512 stronger than older algorithms like SHA-1?” • Link the structure of SHA-512 to defense against collision attacks. Encourage students to compare SHA-256 and SHA-512.
Evaluation	• Worksheet on padding, word expansion, and rounds.

Lesson Plan No. 4.7	Course Name: Fundamentals of Cryptography Topic: Message and Message Digest Encryption.	Course No.: COM-403

Objectives	At the end of the lesson the student shall be able to: a. Understand how messages are hashed before encryption b. Analyze combined use of hashing and encryption c. Learn about digital signatures
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with diagrams of Email encryption demo, Flowchart of sign-then-encrypt vs encrypt-then-sign
Teaching Development	**1. Introduction (5 minutes)** • Start with a scenario: "If a sender signs a message with a hash, is it still secure?" • Introduce the concept of signing the hash (digest) instead of the entire message. **2. Development (30 minutes)** • Use diagrams to explain sign-then-encrypt and encrypt-then-sign. • Demonstrate this process using GPG or visual flowcharts. Have students simulate both workflows using roleplay (sender, receiver, attacker). • Discuss digital signatures and how they provide integrity and non-repudiation. **3. Exercise (5 minutes)** •Find a real-world example where digital signing is used and explain why.
Closure	• Ask: "Which method makes more sense: sign-then-encrypt or encrypt-then-sign? Why?" • Allow students to defend their answer. Link the lesson to secure email, contracts, and communication platforms.
Evaluation	• Mini debate or reflection write-up: Pros and cons of both models. • Peer quiz: Match encryption scenarios with correct strategy.

Lesson Plan No. 4.8	**Course Name: Fundamentals of Cryptography** **Topic: Linear Cryptanalysis Case Study (DES).**	**Course No.: COM-403**

Objectives	At the end of the lesson the student shall be able to: a. Apply linear approximations to attack DES b. Identify useful linear trails c. Estimate success probabilities
Teaching Aids (if any)	c. Chalkboard/Whiteboard d. Real attack paper summary e. Slides with diagrams, Chart of bias calculations
Teaching Development	**1. Introduction (5 minutes)** • Ask: “How do attackers find patterns even in secure-looking ciphers?” • Review DES structure and walk through a linear trail over a few rounds. **2. Development (30 minutes)** • Use guided bias computation with the Piling-up Lemma to estimate success probabilities. • Provide examples of real research or case studies where linear cryptanalysis was used. Use a worksheet to track biases across operations. • Break the class into groups: each follows a potential linear trail and determines if it’s promising. Summarize findings as a class. **3. Exercise (5 minutes)** Try extending a 2-round linear trail on a simplified DES.
Closure	• Prompt discussion: “What are the limits of linear cryptanalysis?” • Emphasize the importance of structure, bias, and trail length. Discuss why modern ciphers are designed to resist this type of attack.
Evaluation	• Reflective question: “What makes a linear trail useful?”

Lesson Plan No. 4.9	Course Name: Fundamentals of Cryptography Topic: AES vs DES vs Hash Functions – Summary and Comparison.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Compare encryption techniques b. - Understand evolution in security design c. - Recognize future directions (quantum, post-quantum)
Teaching Aids (if any)	a. Comparison table b. - Timeline of crypto evolution
Teaching Development	**1. Introduction (5 minutes)** • Start with a timeline: “How has encryption evolved from DES to AES to SHA-512?” **2. Development (30 minutes)** • Display a comparison chart with features like block size, key size, structure, speed, and vulnerabilities. • Assign small groups to research and present one algorithm. Then, complete a matrix together as a class. Discuss trade-offs in security, performance, and adoption. • Introduce the idea of quantum computing and future threats to current systems. **3. Exercise (5 minutes)** Read a summary of NIST’s post-quantum cryptography initiative and summarize one proposed algorithm
Closure	• Ask: “If you had to design your own encryption system, what would you prioritize and why?” • Encourage reflection on strengths and weaknesses of each method. Link to emerging fields like post-quantum cryptography.
Evaluation	• Written quiz with fill-in-the-blank, true/false, and application questions. • Exit ticket: “Which algorithm do you trust the most and why?”

Lesson Plan No. 1.1	**Course Name: Fundamentals of Cryptography** **Topic: ElGamal Cryptosystem.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. - Understand the mathematical foundations of ElGamal. b. - Perform encryption and decryption using ElGamal. c. - Analyze the security assumptions behind ElGamal.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Visualization of discrete log c. Online ElGamal calculator
Teaching Development	**1. Introduction (5 minutes)** • Start with: “Can we build secure encryption without sharing a key beforehand?” **2. Development (30 minutes)** • Introduce ElGamal with a visual analogy of padlocks and open mailboxes. Explain the mathematics of discrete logarithms using simple, relatable numbers. • Walk through key generation, encryption, and decryption using small values. Use group-based examples to help students simulate message exchange. • Reinforce understanding with visual models showing public/private key relationships. **3. Exercise (5 minutes)** •Write a Python function to implement ElGamal encryption for small numbers.
Closure	• Explain: “What makes ElGamal useful for encryption in the real world?” • Connect to digital applications like secure file transfer and blockchain. Let students summarize key steps of ElGamal in their own words.
Evaluation	• Reflective quiz on encryption and decryption steps. • Group problem-solving: Encrypt/decrypt using given keys.

Lesson Plan No. 5.2	**Course Name: Fundamentals of Cryptography** **Topic: Shanks' Algorithm.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Understand Shanks' algorithm for solving discrete log problems. b. Learn baby-step giant-step technique. c. Analyze its complexity and application in cryptanalysis.
Teaching Aids (if any)	a. Step table and equation derivation b. Visual diagram of algorithm process
Teaching Development	**1. Introduction (5 minutes)** • Pose a puzzle: "How would you find 'x' in the equation $a^x \equiv b \bmod p$ • Introduce Shanks' baby-step giant-step algorithm visually using tables. **2. Development (30 minutes)** • Break down the process with a worked example and graph each step. • Let students manually solve small examples using step tables. Reinforce logic with analogies like distance jumping. • Assign team-based challenges where each group applies the algorithm to a different discrete log problem. **3. Exercise (5 minutes)** "Implement Shanks' algorithm in code and test it on a small prime."
Closure	• Ask: "What made solving the discrete log easier or harder using Shanks' method?" • Summarize its importance in both cryptanalysis and verifying security assumptions. •
Evaluation	• "Implement Shanks' algorithm in code and test it on a small prime."

Lesson Plan No. 5.3	**Course Name: Fundamentals of Cryptography** **Topic: Diffie-Hellman Problems.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Understand key exchange through Diffie-Hellman. b. Learn the DH key generation and shared secret computation. c. Identify potential vulnerabilities (MITM).
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with DH key exchange diagram c. Wireshark or simulation of exchange
Teaching Development	**1. Introduction (5 minutes)** • Ask: How can two people create a shared secret in public? How to communicate the key though insecure channels??? **2. Development (30 minutes)** • Introduce Diffie-Hellman key exchange with a color-mixing analogy or modular exponentiation visualization. • Walk through key generation and secret sharing. • Use diagrams and roleplay: students act as Alice and Bob to simulate the exchange. • Introduce potential attacks (e.g., MITM) to emphasize security needs. • **3. Exercise (5 minutes)** • Draw and explain the steps of DH key exchange and mention one vulnerability.
Closure	• Prompt discussion: “Why is it hard for attackers to reverse the key exchange?” • Link to VPNs, messaging apps, and TLS. Encourage a student summary of the full exchange flow.
Evaluation	• Key exchange quiz. • Scenario question: “How would MITM interfere with DH?”

Lesson Plan No. 5.4	**Course Name: Fundamentals of Cryptography** **Topic: RSA Algorithm.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Learn key generation process for RSA. b. Perform encryption and decryption operations. c. Understand importance of Euler's theorem in RSA.
Teaching Aids (if any)	a. RSA diagram and sample primes b. Python or online RSA demo
Teaching Development	**1. Introduction (5 minutes)** • Pose the question: "What if your encryption method depended on multiplying large primes?" **2. Development (30 minutes)** • Introduce RSA with small primes to demonstrate key generation. Use Euler's theorem and φ(n) in real calculations. • Let students encrypt and decrypt a 2-digit message using RSA steps in groups. • Use a card game or roleplay to represent public and private key operations. **3. Exercise (5 minutes)** • Write a Python script to generate RSA keys with small primes.
Closure	• Explain again: "What makes factoring so critical to RSA's security?" • Summarize the steps and relate RSA to secure email and file encryption.
Evaluation	• Quiz on each step of RSA: keygen, encryption, decryption. • Exit question: "Why must p and q be prime in RSA?"

Lesson Plan No. 5.5	Course Name: Fundamentals of Cryptography Topic: Signing and Encrypting.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. - Understand difference between signing and encryption. b. - Learn order of operations: sign-then-encrypt vs encrypt-then-sign. c. - Apply both for message authenticity and confidentiality.
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Diagrams of both approaches c. Digital signature demo
Teaching Development	**1. Introduction (5 minutes)** • Ask: “If you want privacy and authenticity, how do you achieve both?” • Introduce signing vs encryption with flowcharts. **2. Development (30 minutes)** • Explain the difference between sign-then-encrypt and encrypt-then-sign. • Use real-world scenarios like secure email or document verification. In pairs, students simulate both methods with visual aids or cards. • Discuss when each method is better and what threats it protects against. **3. Exercise (5 minutes)** •Write a short opinion piece: Sign-then-encrypt vs Encrypt-then-sign — which is better and why?
Closure	• Clarify : “Why does the order of signing and encrypting matter?” • Encourage students to summarize the benefits of each approach and reflect on their usefulness in digital security.
Evaluation	• Mini debate in class / Flowchart-based quiz. And/Or Peer-reviewed diagrams of message workflows.

Lesson Plan No. 5.6	Course Name: Fundamentals of Cryptography Topic: Multivariate Encryption Techniques.	Course No.: COM- 403

Objectives	At the end of the lesson the student shall be able to: a. Understand difference between signing and encryption. b. Learn order of operations: sign-then-encrypt vs encrypt-then-sign. c. Apply both for message authenticity and confidentiality.
Teaching Aids (if any)	a. Diagram of multivariate polynomials b. Paper on Rainbow or HFE
Teaching Development	**1. Introduction (5 minutes)** • Ask: “How can we build secure systems using polynomial equations?” **2. Development (30 minutes)** • Introduce multivariate cryptosystems with a visual of variable graphs and nonlinear functions. Use examples from Rainbow and HFE. • Explain public/private key creation and how trapdoors help in decryption. Students solve small systems of polynomial equations in groups. • Highlight relevance to post-quantum cryptography and emerging standards. **3. Exercise (5 minutes)** •Summarize how multivariate cryptography offers quantum resistance.
Closure	• “What makes multivariate systems hard to break with quantum computers?” • Reinforce concepts through analogies (e.g., untangling a knot of equations). Link to cryptography’s future.
Evaluation	• Concept check quiz: match terms like trapdoor, nonlinear system, multivariate. • Worksheet on solving basic systems.

Lesson Plan No. 5.7	**Course Name: Fundamentals of Cryptography** **Topic: Summary and Modern Applications.**	**Course No.: COM- 403**

Objectives	At the end of the lesson the student shall be able to: a. Compare various public-key techniques. b. Identify best-fit use cases for each algorithm. c. Understand developments in post-quantum cryptography
Teaching Aids (if any)	a. Chalkboard/Whiteboard b. Slides with Comparison table, NIST post-quantum initiative, etc.
Teaching Development	**1. Introduction (5 minutes)** • Start with: “Which encryption method fits best in a given scenario?” • Create a comparison chart of RSA, DH, ElGamal, multivariate systems. • **2. Development (30 minutes)** • Let groups fill out performance, security, speed, and quantum resistance. • Introduce blockchain, secure messaging, and cloud storage as modern application cases. Students match use cases to the most suitable algorithm. • Facilitate gallery walk or rotating presentations from each group. **3. Exercise (5 minutes)** •
Closure	• “Which algorithm would you recommend for future-proof security?” • Encourage reflection on trade-offs. Point to the role of open standards and innovation. Homework: “Write a report on one post-quantum algorithm and where it could be deployed.”
Evaluation	• Student presentations on chosen encryption techniques. • Quiz comparing algorithms’ strengths and limitations.