Seiyun University, Yemen
Department of Information Security

Subject Name: Special Topics in Information Security
Bachelor of Science , Fourth Level - First Semester, 2025

Instructor:
Prof. Ahmed Abuamer

Lab Assignment No. 01

RSA Encryption & Homomorphic Property

Submission Instructions:

1. Answer all questions with appropriate explanations and examples where required.
2. All coding must be done in **Google Colab**..
3. Save notebook as RSA_Homomorphic_Lab _<StudentName>_<EnrollNo>.ipynb.
4. The Answers to questions must be handwritten.
5. Export a single PDF that includes:
    a. Python code
    b. Output screenshots
    c. Handwritten answers (*scanned or photographed clearly*)
    d. Diagrams (if any)
6. Send the PDF to info@ahmedabuamer.com and include the following in your email body:
    a. Student's Name
    b. Enrollment Number
    c. Mobile Number
    d. GitHub Repository Link (*with your notebook uploaded*)
7. **Due Date: 15 November 2025**
8. Ensure your programs are:
    a. Well-formatted
    b. Properly tested
    c. Commented for clarity

Objectives:

- Understand how RSA public-key encryption works.
- Learn how to generate small RSA key pairs (public and private keys).
- Implement encryption and decryption functions in Python.
- Demonstrate and verify the multiplicative homomorphic property of RSA:

$$E(m_1) \times E(m_2) = E(m_1 \times m_2) \, (mod\ n)$$

- Analyze why this property is useful but also dangerous in certain contexts (e.g., chosen-ciphertext attacks).

Lab Task: Implement and Demonstrate RSA Encryption & its Homomorphic Property

Write a Python script that performs the following:

1. Generate small RSA key pair:
    a. Choose small primes p and q.
    b. Compute n = p * q and Euler's totient φ(n) = (p - 1)(q - 1).
    c. Choose public key e and compute private key d.
2. Implement:
    a. encrypt(m, e, n)
    b. decrypt(c, d, n)
3. Encrypt two plaintext messages m_1 and m_2.
4. Compute:
    a. $E(m_1)$ and $E(m_2)$
    b. Multiply ciphertexts: $E(m_1) \times E(m_2) \, mod\ n$
    c. Decrypt the result and verify that it equals $m_1 * m_2 \, mod\ n$
5. Log all steps, including intermediate values (p, q, n, φ, e, d, ciphertexts, results).

Steps:

1. Generate p, q, n, φ(n), and keys (e, d).
2. Implement encryption and decryption functions.
3. Select any two plaintexts (e.g., 5 and 9).
4. Encrypt both and print ciphertexts..
5. Multiply the ciphertexts modulo n
6. Decrypt the product and verify that it equals m1 × m2 mod n.
7. Log all calculations and verify correctness.

Sample Output Log:

```
p = 11, q = 13
n = 143, ϕ = 120
Public key (e, n) = (7, 143)
Private key (d, n) = (103, 143)


m1 = 5, m2 = 9
E(m1) = 47, E(m2) = 48
E(m1)*E(m2) mod n = 110
Decrypted result = 45
Expected (m1*m2 mod n) = 45


Homomorphic property verified: E(m1)E(m2) ≡ E(m1*m2) mod n
```

Expected Outcome:

After completing this lab, students should be able to:

- Successfully generate RSA key pairs..
- Demonstrate how encryption and decryption work mathematically.
- Verify the homomorphic multiplication property.
- Understand how mathematical relationships in RSA can be both useful and dangerous.
- Develop and document secure coding habits for cryptography.

Lab Questions: (Answer in Handwriting)

Q1. Define RSA algorithm and explain each step briefly.

Q2. What is the homomorphic property in cryptography?

Q3. Explain why RSA is multiplicatively homomorphic.

Q4. What are the advantages and risks of homomorphic encryption?

Q5. Why do we use the **mod n** operation in RSA?

Q6. Compute manually for a small case (p=3, q=11, e=7, m1=2, m2=5).

Q7. Why is key size important for security?

Q8. Explain what happens if e and φ(n) are not coprime.

Q9. Describe one real-world application of homomorphic encryption.

GOOD LUCK!