**Lab Objectives**

1. Understand the working principles of RSA and Diffie-Hellman algorithms.
2. Perform manual computations for RSA and Diffie-Hellman key exchange.
3. Implement RSA encryption and decryption in Python.
4. Simulate the Diffie-Hellman key exchange process.
5. Analyze the security aspects of these algorithms.

**Activity 1: Understanding RSA Step-by-Step**

**Objective:** Manually perform RSA key generation, encryption, and decryption with small prime numbers. Use examples from the lecture slides in last week to understand the process.

**Task 1: Key Generation (Manual Calculation)**

1. Choose two small prime numbers, **p** and **q**.

P = 7

Q = 11

1. Compute **n = p × q**.

N =7 x 11

N = 77

1. Compute **ϕ(n) = (p-1) × (q-1)**.

**ϕ(n) = 6 x 10**

**ϕ(n) = 60**

1. Choose a small public exponent **e**.

E = 7

1. Compute the private key **d = e⁻¹ mod ϕ(n)** (find the modular inverse).

D = inverse(7 mod 60)

D = 43

1. Write down the public and private keys.

E = 7 D = 43

**Task 2: Encrypt a Message (Manual Calculation)**

1. Convert a small message (e.g., **M = 5**) into numerical form.

M = 6

1. Compute **C = M^e mod n**.

C = 6^7 mod 77

C = 41

**Task 3: Decrypt the Message (Manual Calculation)**

1. Compute **M = C^d mod n** to retrieve the original message.

41^43 mod 77 = 6

M = 6

**Activity 2: Understanding Diffie-Hellman Key Exchange**

**Objective:** Simulate the Diffie-Hellman key exchange by working in pairs and manually computing the values.

In this task, you will work in groups of two. **Do not sit near your partner; instead, communicate using Teams messages (or another suitable online channel).**

1. **Pairing and Communication:** Select a partner of your choice. **Do not sit near your partner.** Coordinate through Teams messages for communication.
2. **Selecting Public Values:** Decide on two public numbers **p** (prime number) and **g** via Teams messages. Ideally, g should be a primitive root modulo p for better security but for convenience, you can use any number.
3. **Choosing Private Values:** Each student privately selects a secret number (**a** for Student 1, **b** for Student 2). Keep this secret and do not share it even with your partner.
4. **Computing Public Keys:** Each student computes:

**P = 7**

**G = 9**

* + Student 1: **A = g^a mod p**

**9^49 mod 9 = 0**

**A = 2**

Student 2: **B = g^b mod p**

B = 1

1. **Exchanging Public Keys:** Share the computed values **A** and **B** with your partner via Teams messages. Note: Do not share private keys at any step.
2. **Computing the Shared Secret:** Each student computes the shared secret using their private key and the received public key:
   * Student 1: **s = B^a mod p**

**1^49 mod 7 = 1**

* + Student 2: **s = A^b mod p**

1. **Verification:** Both students should arrive at the same shared secret **s**. If the values differ, check your calculations, and repeat the procedure.

**LAB LOGBOOK REQUIREMENT:** Document in your lab logbook.

1. Your partner's name,
2. Values: p, g, s, and your private key (Do not write your partner’s private key)

**Optional Challenge (work in groups)**

* Work with your partner to simulate a **hybrid encryption** system where:
  + Diffie-Hellman is used to exchange a symmetric encryption key.
  + AES is used for encrypting and decrypting messages (you can use Python library for performing encryption and decryption).
* This mimics **real-world encryption systems**.
* Share any secret message using the D with your partner using encryption.
* Your partner will decrypt it using same key (which you shared using Diffie-Hellman)