|  |  |
| --- | --- |
| **Walchand College of Engineering, Sangli**  *(Government Aided Autonomous Institute)* | |
| **AY 2025-26** | |
| **Course Information** | |
| **Programme** | B.Tech. (Computer Science and Engineering) |
| **Class, Semester** | Final Year B. Tech., Sem VII |
| **Course Code** | 6CS451 |
| **Course Name** | Cryptography and Network Security Lab |

**Experiment No. 08**

**Title –** Implement the Diffie-Hellman Key Exchange algorithm for a given problem.

**Objectives:**

To implement the **Diffie-Hellman Key Exchange Algorithm** to enable two parties to securely share a secret key over an insecure communication channel. This shared key can then be used for symmetric encryption or decryption in secure communication.

**Problem Statement:**

In secure communications, two parties (commonly referred to as Alice and Bob) need to agree on a secret key that can be used for encrypting and decrypting messages. However, they must do this over a public channel where an attacker (Eve) might be listening.

Implement the **Diffie-Hellman Key Exchange Algorithm**, which allows Alice and Bob to securely compute a shared secret key without directly transmitting it over the insecure channel. The algorithm should:

1. Accept a large prime number p and a primitive root modulo p, g.
2. Allow each party to select a private key (a for Alice, b for Bob).
3. Compute the corresponding public keys:
   * Alice computes A = g^a mod p
   * Bob computes B = g^b mod p
4. Exchange public keys between Alice and Bob.
5. Compute the shared secret key:
   * Alice computes K = B^a mod p
   * Bob computes K = A^b mod p
6. Validate that both computed secret keys are equal, i.e., K\_Alice == K\_Bob.

Additionally, demonstrate the correctness of the algorithm with an example and optionally simulate an attacker attempting to derive the secret key without access to the private keys.

**Equipment/Tools: Theory:**

**Procedure:**

**Steps:**

**Observations and Conclusion:**