Experiment-7

Aim → Implementation of secure key exchange and source authentication process using both symmetric and asymmetric cryptography in computer networking.

Objective 1 → An overview on Diffie-Helman algorithm.

The Diffie-Hellman algorithm is a method for securely exchanging cryptographic keys over a public chanel. It allows two parties (e.g., Alice and Bob) to create a shared secret key without transmitting it directly.

Keey concepts → • Uses modular arithmetic and exponentiation.
• Based on the difficulty of solving discrete logarithms.

How it Works →

i) Both parties agree on a public prime number p & base g.

ii) Each picks a private secret.

iii) They compute public keys: • Alice: $A = g^a \bmod p$
• Bob: $B = g^b \bmod p$

iv) They exchange these public keys.

v) Both compute the same shared key: • Alice: $s = B^a \bmod p$
• Bob: $s = A^b \bmod p$

vi) This shared key is then used for secure communication.

Use Cases → • Secure messaging
• VPNs
• TLS (used in HTTPS)

Objective 2 → Execution of Diffie-Helman algorithm for key exchange between a source and destination host.

Code →

```
import random
p = 23
g = 5
a = random.randint(1, p-2)
A = pow(g, a, p)
b = random.randint(1, p-2)
B = pow(g, b, p)
shared_key_alice = pow(B, a, p)
shared_key_bob = pow(A, b, p)
print(f"Alice's Private Key: {a}")
```

Name: _____ Regd No: _____

```
print("Bob's Private Key: {b}")
print("Alice's Public Key (A): {A}")
print("Bob's Public Key (B): {B}")
print("Shared Key at Alice : {shared_key_alice}")
print("Shared Key at Bob : {shared_key_bob}")
print("Key exchange successful:", shared_key_alice == shared_key_bob)
```

Output → Alice's Private Key : 12
Bob's Private Key : 13
Alice's Public Key (A): 18
Bob's Public Key (B): 21
Shared Key at Alice : 2
Shared Key at Bob : 2
Key exchange successful : True

Objective 3 → Execution of public key crypto-system for authentication verification of source using digital signature in cryptography process.

Code →
```
from cryptography.hazmat.primitives.asymmetric import rsa, padding
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives import serialization
private_key = rsa.generate_private_key(public_exponent=65537, key_size=2048)
public_key = private_key.public_key()
message = b"This is a secure message from Alice."
signature = private_key.sign(
        message,
        padding.PSS(mgf=padding.MGF1(hashes.SHA256()), salt_length=padding.
            PSS.MAX_LENGTH),
        hashes.SHA256()
)
try:
    public_key.verify(
        signature,
        message,
        padding.PSS(mgf=padding.MGF1(hashes.SHA256()), salt_length=padding.
                                                    PSS.MAX_LENGTH),
```

Name: _____ Regd No: _____

```
        hashes.SHA256()
    )
    print("Authentication successful: Signature is valid.")
except Exception as e:
    print("Authentication failed: Signature is invalid")
```

Output → Authentication successful: Signature is valid.

Conclusion → The experiment demonstrated how to establish secure communication in computer networks using both ~~them~~ symmetric & asymmetric cryptographic techniques. It covered:

i) Diffie-Hellman Algorithm for securely exchanging a shared key between two parties over an insecure channel. This key can later be used for encrypted communication.

ii) Digital Signatures using a public-key cryptosystem to verify the authenticity of the source. This ensures that a recieved message truly originated from the claimed sender and was not tampered with in transit.

Exercises →

1) Given the following parameters for Diffie-Hellman algorithm used by A and B for key exchange.

- The shared prime $q = 157$ and the primitive root $p = 5$.

~~Sol~~ Calculate → a) The value of Y_A and Y_B transmitted by both A and B.

b) The value of secured key (K) shared by both A and B.

Ans → Given, shared prime $q = 157$

Primitive root $p = 5$

Assume private keys: $X_A = 15$ (for A), $X_B = 13$ (for B)

a) Compute Public keys →

For A:

$$Y_A = p^{X_A} \bmod q = 05^{15} \bmod 157$$

First, compute $5^{15} \bmod 157$ using modular exponentiation:

$5^1 = 5$

$5^2 = 25$

$5^4 = 625 \bmod 157 = 157 \times 3 + 154 = 154$

$5^8 = (5^4)^2 = 154^2 = 23716 \bmod 157 = 23716 \div 157 = 151$, remainder $139 \Rightarrow 139$

Now express $5^{15} = 5^8 . 5^4 . 5^2 . 5^1$

$5^{15} \bmod 157 = 139 . 154 . 25 . 5 \bmod 157$

Name: _____ Regd No: _____

Step by step → $139 \cdot 154 = 21406 \bmod 157 = 21406 \div 157 = 136$, remainder 38

$38 \cdot 25 = 950 \bmod 157 = 950 \div 157 = 6$, remainder 8

$8 \cdot 5 = 40$

$\boxed{Y_A = 40}$

For B →

$Y_B = 5^{13} \bmod 157$

Use: $5^{13} = 5^8 \cdot 5^4 \cdot 5^1 = 139 \cdot 159 \cdot 5$

From above: $139 \cdot 154 = 21406 \bmod 157 = 38$

$38 \cdot 5 = 190 \bmod 157 = 33$

$\boxed{Y_B = 33}$

b) Compute Shared Key K.

A computes:

$K = Y_B^{X_A} \bmod q = 33^{15} \bmod 157$

Use modular exponentiation:

First compute powers of 33 modulo 157:

- $33^2 = 1089 \bmod 157 = 157 \times 6 + 117 = 117$
- $33^4 = 117^2 = 13689 \bmod 157 = 13689 \div 157 = 87$, remainder 90
- $33^8 = 90^2 = 8100 \bmod 157 = 8100 \div 157 = 51$, remainder 93

Now:

$33^{15} = 33^8 \cdot 33^4 \cdot 33^2 \cdot 33 = 93 \cdot 90 \cdot 117 \cdot 33 \bmod 157$

Step by step:

- $93 \cdot 90 = 8370 \bmod 157 = 8370 \div 157 = 53$, remainder 89
- $89 \cdot 117 = 10413 \bmod 157 = 10413 \div 157 = 66$, remainder 21
- $21 \cdot 33 = 693 \bmod 157 = 693 \div 157 = 4$, remainder 65

Shared key $K = \boxed{65}$

2) Given a scenario, where B has received a document from A through internet. Explain, how B confirms that the document has been transmitted by A only (~~Only~~ not any adversary) using the concept of digital signature as an use of public key crypto system.

Ans→ Scenario Explanation Using Digital Signature →

Name: _____ Regd No: _____

a) A (sender) signs the document →

- A computes a hash (message digest) of the document using a hash function like SHA-256.
- A then encrypts the hash with their private key to create a digital signature.
- A sends both the document and the digital signature to B over the internet.

b) B (Reciever) verifies the signature:

- B recieves the document and the signature.
- B computes the hash of the recieved document using the same hash function.
- B decrypts the digital signature using A's public key, obtaining the hash that A originally sent.

Name: _____ Regd No: _____