ST. FRANCIS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
SECURITY LAB

Experiment – 6: Implementation and Simulation of Diffie-Hellman Key Exchange Algorithm

Aim: Write a program to implement Diffie Hellman (DH) Key Exchange algorithm.

Objective: After performing the experiment, the students will be able to understand the Diffie-Hellman Key exchange algorithm.

Lab objective mapped: L502.2: Students should be able to analyse and implement public key algorithms.

Prerequisite: Basic knowledge of asymmetric key cryptography.

Requirements: PYTHON

Pre-Experiment Theory:

Diffie Hellman (DH) key exchange is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher. The efficiency of this algorithm is based on computation of discrete logarithm problem for large prime values.

Algorithm:
Alice and Bob are two users who wish to establish secure communication using the Diffie Hellman algorithm. We can assume that Alice and Bob know nothing about each other but are in communication.

Steps:

Global Public Elements

q *prime number*

α $\alpha \leq q$

User A key Generation

select private X_A $X_A \leq q$

calculate public Y_A $Y_A = \alpha^{X_A} \bmod q$

User B Key Generation

select private X_B $X_B \leq q$

calculate public Y_B $Y_B = \alpha^{X_B} \bmod q$

Calculation of Secret key by User A

$$K = (Y_B)^{X_A} \bmod q$$

Calculation of secret key by user B

$$K = (Y_A)^{X_B} \bmod q$$

As shown above,

1. Alice and Bob agree on two large positive integers, q and α, where q is a prime number and α is a primitive root mod q.
2. Alice randomly chooses another large positive integer, X_A, which is smaller than q. X_A will serve as Alice's private key.
3. Bob similarly chooses his own private key, X_B.
4. Alice computes her public key, Y_A, using the formula $Y_A = \alpha^{X_A} mod\ q$
5. Bob similarly computes his public key, Y_B, using the formula $Y_B = \alpha^{X_B} mod\ q$
6. Alice and Bob exchange public keys over the insecure channel.
7. Alice computes the shared secret key, K, using the formula $K = Y_B^{X_A} mod\ q$
8. Bob computes the same shared secret key, K, using the formula $K = Y_A^{X_B} mod\ q$
9. Alice and Bob communicate using the symmetric algorithm of their choice and the shared secret key, K, which was never transmitted over the insecure channel.

Procedure:

1. Access the Cryptography virtual lab link http://cse29-iiith.vlabs.ac.in/
2. Goto 'List of Experiments' – 'Diffie-Hellman Key Establishment' - 'Simulation' tab.
3. Understand the simulation process of Diffie Hellman Key exchange. Follow the key exchange process by generating prime and generator numbers. Take screenshot.
4. Write a program in Python for Diffie Hellman Key exchange.
    a. For Key generation, ask user to enter the value of prime number q & α and public keys of A and B, X_A & X_B. (Note that values of q, α, X_A & X_B cannot be random, they should satisfy criteria as per DH algorithm)
    b. Program should calculate the private keys for both Alice and Bob as per DH algorithm.
    c. Provide a set of public (e, n) and private key (d, n) of Alice and Bob as the output to the user.
5. Test the output of program for following exercise:
    a. Alice and Bob decide to use Diffie Hellman key exchange with $q = 23,\ \propto = 7,\ X_A = 3$ & $X_B = 6$. Find their public and private keys and the shared key.
    b. For $q = 7,\ \propto = 3,\ X_A = 2$ & $X_B = 5$ Validate public keys (Ans: 2 &5), private keys (Ans: 2 & 5) and the shared key (Ans: K=4).
    c. For $q = 17,\ \propto = 5,\ X_A = 4$ & $X_B = 6$ Validate public keys (Ans: 13 & 2), private keys (Ans: 4 & 6) and the shared key (Ans: K=16).

Output:

1. Attach the complete code performing key generation of both the parties.
2. Attach the program output for key generation (display public key & private key for Both A & B) for the inputs given in all three exercises above.
3. Attach the Screenshot taken for key generation using Cryptography virtual lab simulation.

Post Experimental Exercise-

1. Solve all three exercises mentioned in the procedure on the journal sheet. [Theoretical result and attached code's output should match].
2. Discuss five practical applications of Diffie Hellman key exchange algorithm.

Conclusion:

The Diffie-Hellman key exchange algorithm is a fundamental cryptographic technique that enables two parties to securely establish a shared secret key over an insecure communication channel. This shared key can then be used for various encryption and authentication purposes.

References: Virtual Cryptography Lab link: http://cse29-iiith.vlabs.ac.in/

CODE:

```
def diffie_hellman_key_exchange(q, alpha, XA, XB):
    # Calculate the public keys
    YA = pow(alpha, XA, q)  # Alice's public key
    YB = pow(alpha, XB, q)  # Bob's public key

    # Calculate the shared secret keys
    shared_secret_Alice = pow(YB, XA, q)  # Alice calculates the shared key
    shared_secret_Bob = pow(YA, XB, q)    # Bob calculates the shared key

    return YA, YB, shared_secret_Alice, shared_secret_Bob

def main():
    print("Diffie-Hellman Key Exchange")

    # Input prime number q and base α
    q = int(input("Enter the prime number q: "))
    alpha = int(input("Enter the base α: "))

    # Input private keys XA and XB
    XA = int(input("Enter the private key of Alice XA: "))
    XB = int(input("Enter the private key of Bob XB: "))

    # Calculate the public keys and shared secret keys
    YA, YB, shared_secret_Alice, shared_secret_Bob = diffie_hellman_key_exchange(q, alpha, XA, XB)
    # Output the public keys and shared secret keys
```

```
    print("\nPublic Keys:")
    print(f"Alice's public key YA: {YA}")
    print(f"Bob's public key YB: {YB}")
    print("\nShared Secret Keys:")
    print(f"Alice's shared secret key: {shared_secret_Alice}")
    print(f"Bob's shared secret key: {shared_secret_Bob}")

    # Check if the shared secret keys match
    if shared_secret_Alice == shared_secret_Bob:
        print("The shared secret keys match. Key exchange
successful!")
    else:
        print("The shared secret keys do not match. Key exchange
failed.")

if __name__ == "__main__":
    main()
```

Output:

1. Alice and Bob decide to use Diffie Hellman key exchange with $q = 23, \propto = 7, X_A = 3 \,\&\, X_B = 6$. Find their public and private keys and the shared key.

```
Diffie-Hellman Key Exchange
Enter the prime number q: 23
Enter the base α: 7
Enter the private key of Alice XA: 3
Enter the private key of Bob XB: 6

Public Keys:
Alice's public key YA: 21
Bob's public key YB: 4

Shared Secret Keys:
Alice's shared secret key: 18
Bob's shared secret key: 18
The shared secret keys match. Key exchange successful!
```

2. For $q = 7, \propto = 3, X_A = 2 \& X_B = 5$ Validate public keys (Ans: 2 &5), private keys (Ans: 2 & 5) and the shared key (Ans: K=4).

```
Diffie-Hellman Key Exchange
Enter the prime number q: 7
Enter the base α: 3
Enter the private key of Alice XA: 2
Enter the private key of Bob XB: 5

Public Keys:
Alice's public key YA: 2
Bob's public key YB: 5

Shared Secret Keys:
Alice's shared secret key: 4
Bob's shared secret key: 4
The shared secret keys match. Key exchange successful!
```

3. For $q = 17, \propto = 5, X_A = 4 \& X_B = 6$ Validate public keys (Ans: 13 & 2), private keys (Ans: 4 & 6) and the shared key (Ans: K=16)

```
Diffie-Hellman Key Exchange
Enter the prime number q: 17
Enter the base α: 5
Enter the private key of Alice XA: 4
Enter the private key of Bob XB: 6

Public Keys:
Alice's public key YA: 13
Bob's public key YB: 2

Shared Secret Keys:
Alice's shared secret key: 16
Bob's shared secret key: 16
The shared secret keys match. Key exchange successful!
```

Public Information:

Prime Number:

7237 Generate Prime

Generator G:

26 Another Generator

Alice

Key: 1837 Generate A

4278 Calculate Ga

Send Ga to B

Received: 3888

Calculate Gab 1485

Bob

Key: 651 Generate B

3888 Calculate Gb

Send Gb to A

Received: 4278

Calculate Gba 1485