**Experiment 6**

**Date of Performance :**  **Date of Submission:**

**SAP Id: 60004190057** **Name : Junaid Altaf Girkar**

**Div:** **A** **Batch : A4**

**Aim of Experiment**

Design and Implement Diffie Hellman Key Exchange Algorithm

**Theory:**

The Diffie-Hellman algorithm is being used to establish a shared secret that can be used for secret communications while exchanging data over a public network using the elliptic curve to generate points and get the secret key using the parameters.

For the sake of simplicity and practical implementation of the algorithm, we will consider only 4 variables, one prime P and G (a primitive root of P) and two private values a and b.

P and G are both publicly available numbers. Users (say Alice and Bob) pick private values a and b and they generate a key and exchange it publicly. The opposite person receives the key and that generates a secret key, after which they have the same secret key to encrypt.

**Implementation Example:**

| **Step 1: Alice and Bob get public numbers P = 23, G = 9  Step 2: Alice selected a private key a = 4 and  Bob selected a private key b = 3  Step 3: Alice and Bob compute public values  Alice: x =(9^4 mod 23) = (6561 mod 23) = 6  Bob: y = (9^3 mod 23) = (729 mod 23) = 16  Step 4: Alice and Bob exchange public numbers  Step 5: Alice receives public key y =16 and  Bob receives public key x = 6  Step 6: Alice and Bob compute symmetric keys  Alice: ka = y^a mod p = 65536 mod 23 = 9  Bob: kb = x^b mod p = 216 mod 23 = 9  Step 7: 9 is the shared secret.** |
| --- |

**CODE:**

| from random import randint  # Both the persons will be agreed upon the # public keys Q and P # A prime number P is taken P = 23  # A primitive root for P, Q is taken Q = 9   print('The Value of P is :%d'%(P)) print('The Value of Q is :%d'%(Q))  # Alice will choose the private key a a = 4 print('The Private Key a for Alice is :%d'%(a))  # gets the generated key x = int(pow(Q,a,P))  # Bob will choose the private key b b = 3 print('The Private Key b for Bob is :%d'%(b))  # gets the generated key y = int(pow(Q,b,P))   # Secret key for Alice ka = int(pow(y,a,P))  # Secret key for Bob kb = int(pow(x,b,P))  print('Secret key for the Alice is : %d'%(ka)) print('Secret Key for the Bob is : %d'%(kb)) |
| --- |

**OUTPUT:**

| The Value of P is :23 The Value of Q is :9 The Private Key a for Alice is :4 The Private Key b for Bob is :3 Secret key for the Alice is : 9 Secret Key for the Bob is : 9 The Value of P is :23 The Value of Q is :9 The Private Key a for Alice is :4 The Private Key b for Bob is :3 Secret key for the Alice is : 9 Secret Key for the Bob is : 9 |
| --- |

**CONCLUSION**

Even while using ciphers for encryption, it is crucial that the key for encryption and decryption is secure and yet available to the sender and receiver. Diffie Hellman key exchange algorithm is an algorithm in which even the sender and receiver are unaware of others private key. We learnt about this algorithm and implemented it in Python