# CSS Experiment 4

Deep Kothari
TE COMP B
Roll no 23

**Aim:**

To Analyze and implement Diffie-Hellman Key exchange algorithm

**Theory:**

**Diffie-Hellman Key Exchange algorithm:**

Diffie-Hellman algorithm is one of the most important algorithms used for establishing a shared secret. At the time of exchanging data over a public network, we can use the shared secret for secret communication. We use an elliptic curve for generating points and getting a secret key using the parameters.

- We will take four variables, i.e., **P (prime), G (the primitive root of P),** and **a and b (private values)**.
- The variables **P** and **G** both are publicly available. The sender selects a private value, either a or b, for generating a key to exchange publicly. The receiver receives the key, and that generates a secret key, after which the sender and receiver both have the same secret key to encrypt.

Let's understand the process step by step for user1 (sender) and user2 (receiver):

| **Steps** | **User1** | **User2** |
|---|---|---|
| 1. | **P, G** => available public keys. | **P, G** => available public keys. |
| 2. | **a** is selected as a private key. | **b** is selected as a private key. |
| 3. | Eq. to generate key: $x=G^a \bmod P$ | Eq. to generate key: $y=G^b \bmod P$ |
| 4. | After exchanging keys, user1 receives key y. | After exchanging keys, user2 receives key x. |
| 5. | User1 generates a secret key by using the received key y: $k_a=y^a \bmod P$ | User2 generates a secret key by using the received key x: $k_b=x^b \bmod P$ |

Algebraically, 5th step can be shown as follows:

$$k_a=k_b$$

It means that both the users have the symmetric secret key to encrypt.

**Example:**

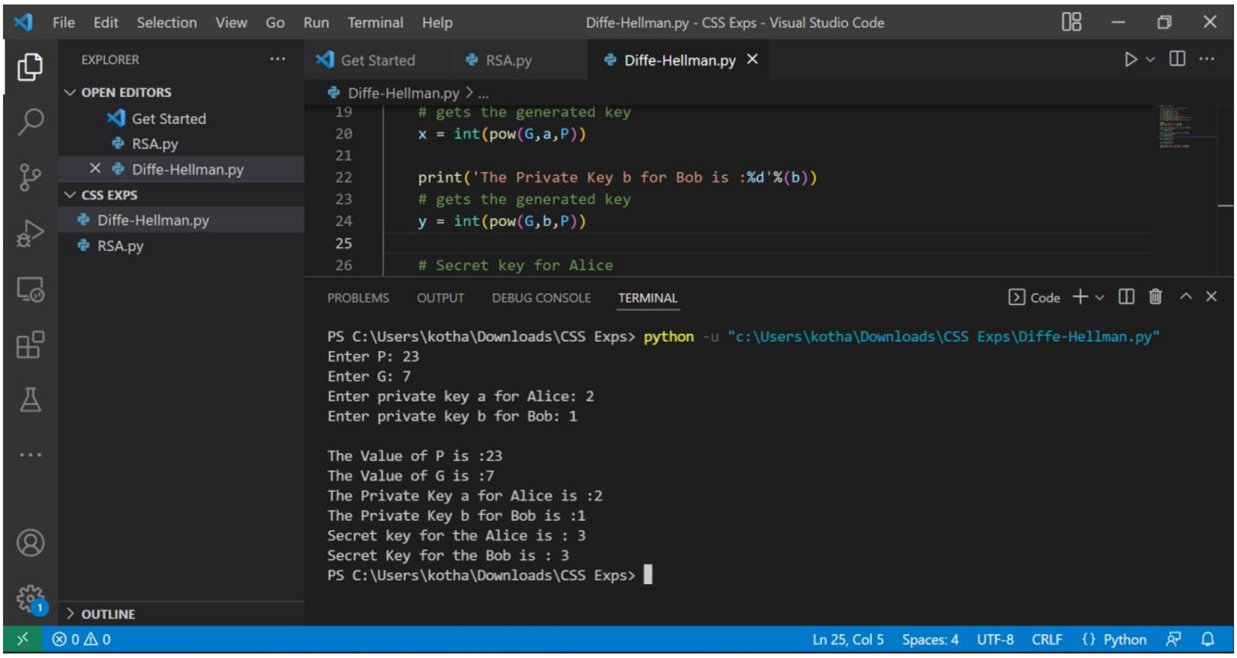
1. User1 and User2 get public keys P = 33 and G = 8.
2. User1 selects a as a private key, i.e., 3, and User2 selects b as a private key, i.e., 2.
3. User1 calculate the public value:
   **x= ($8^3$ mod 33) =512mod33=17**
4. User2 calculate the public value:
   **y= ($8^2$ mod33) =64mod33= 31**
5. User1 and User2 exchange public keys, i.e., 17 and 31.
6. User1 receives public key y = 31 and User2 receives public key x = 17.
7. User1 and User2 calculate symmetric keys:
   **User1: $k_a$=$y_a$ modP = $31_3$ mod 33 = 29791 mod 33 = 25**
   **User2: $k_b$=$x_b$ modP = $17_2$ mod 33 = 289 mod 33 = 25**
8. 25 is the shared secret.

**Program for Diffie-Hellman Algorithm:**

```
if __name__ == '__main__':

    # Both the persons will be agreed upon the
    # public keys G and P
    # A prime number P is taken
    P = int(input("Enter P: "))
    # A primitive root G is taken
    G = int(input("Enter G: "))
    # Alice will choose the private key a
    a = int(input("Enter private key a for Alice: "))
    # Bob will choose the private key b
    b = int(input("Enter private key b for Bob: "))

    print()
    print('The Value of P is :%d'%(P))
    print('The Value of G is :%d'%(G))

    print('The Private Key a for Alice is :%d'%(a))
    # gets the generated key
    x = int(pow(G,a,P))

    print('The Private Key b for Bob is :%d'%(b))
    # gets the generated key
    y = int(pow(G,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 for Diffie-Hellman Algorithm:**




**Conclusion:**

Thus, we have seen how Diffie-Hellman Algorithm works by four variables to generate same secret key for both users. We have successfully designed and implemented a code for Diffie-Hellman Algorithm in python programming language and understood the concept of this asymmetric algorithm.