

## Experiment No 6

**Aim:** Implementation of Diffie Hellman Key exchange algorithm.

**Description:**

**Diffie-Hellman algorithm:**

The **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.

1. We will take four variables, i.e., **P (prime)**, **G (the primitive root of P)**, and **a and b (private values)**.
2. 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.

Step-by-Step explanation is as follows:

Alice	Bob
Public Keys available = P, G	Public Keys available = P, G
Private Key Selected = a	Private Key Selected = b
Key generated $\Rightarrow x = G^a \bmod P$	Key generated $\Rightarrow y = G^b \bmod P$
The exchange of generated keys takes place	
Key received = y	key received = x
Generated Secret Key $\Rightarrow k_a = y^a \bmod P$	Generated Secret Key $\Rightarrow k_b = x^b \bmod P$
Algebraically, it can be shown that- $k_a = k_b$	
Users now have a symmetric secret key to encrypt	

Roll No: 2103163

Batch: C32

Name: Om Shete

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:  $x = (9^4 \bmod 23) = (6561 \bmod 23) = 6$

Bob:  $y = (9^3 \bmod 23) = (729 \bmod 23) = 16$

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:  $ka = y^a \bmod p = 6^4 \bmod 23 = 9$

Bob:  $kb = x^b \bmod p = 6^3 \bmod 23 = 9$

Step 7: 9 is the shared secret.

**Code:**

```
#include <cmath>
#include <iostream>
using namespace std;

long long int power(long long int a, long long int b, long long int P) {
    if (b == 1)
        return a;

    else
        return (((long long int)pow(a, b)) % P);
}

int main() {
    long long int P, G, x, a, y, b, ka, kb;

    cout << "Enter the value of P: " << endl;
    cin >> P;
```

Roll No: 2103163

Batch: C32

Name: Om Shete

```
cout << "Enter the value of G: " << endl;
cin >> G;

cout << "Enter the private key a for Alice: " << endl;
cin >> a;

x = power(G, a, P);

cout << "Enter the private key a for Bob: " << endl;
cin >> b;

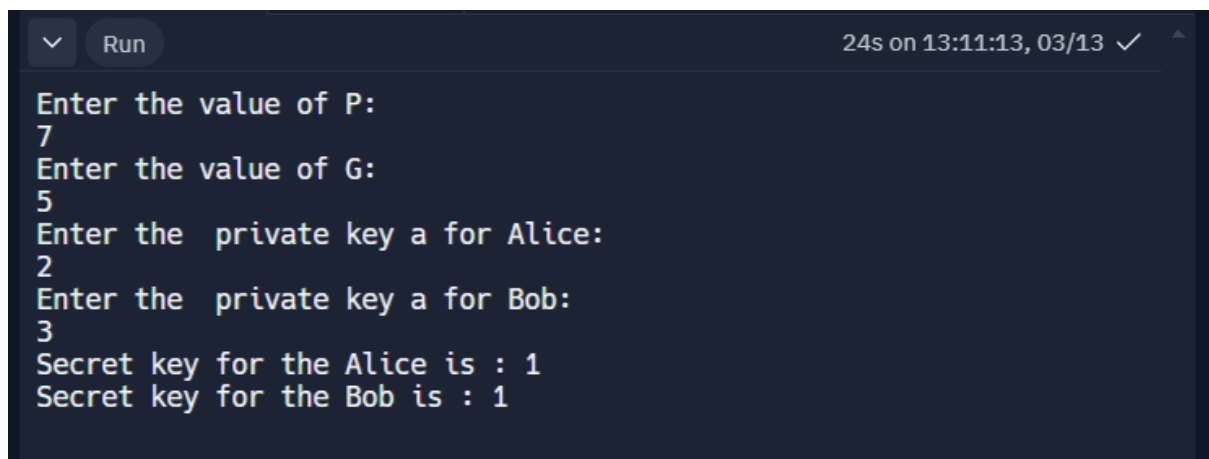
y = power(G, b, P);

ka = power(y, a, P);
kb = power(x, b, P);
cout << "Secret key for the Alice is : " << ka << endl;

cout << "Secret key for the Bob is : " << kb << endl;

return 0;
}
```

## Results:



The screenshot shows a terminal window with a dark background. At the top, there is a 'Run' button and a status bar indicating '24s on 13:11:13, 03/13' with a checkmark. The program's output is as follows:

```
Enter the value of P:
7
Enter the value of G:
5
Enter the private key a for Alice:
2
Enter the private key a for Bob:
3
Secret key for the Alice is : 1
Secret key for the Bob is : 1
```

Roll No: 2103163

Batch: C32

Name: Om Shete