LABORATORY

CEL62: Cryptography and System Security
Winter 2021

Experiment 2:	**Diffie Hellman Key Exchange**

NAME: SAKSHI PATIL
BATCH: C
UID: 2018130039

Experiment 2: Diffie Hellman Key Exchange

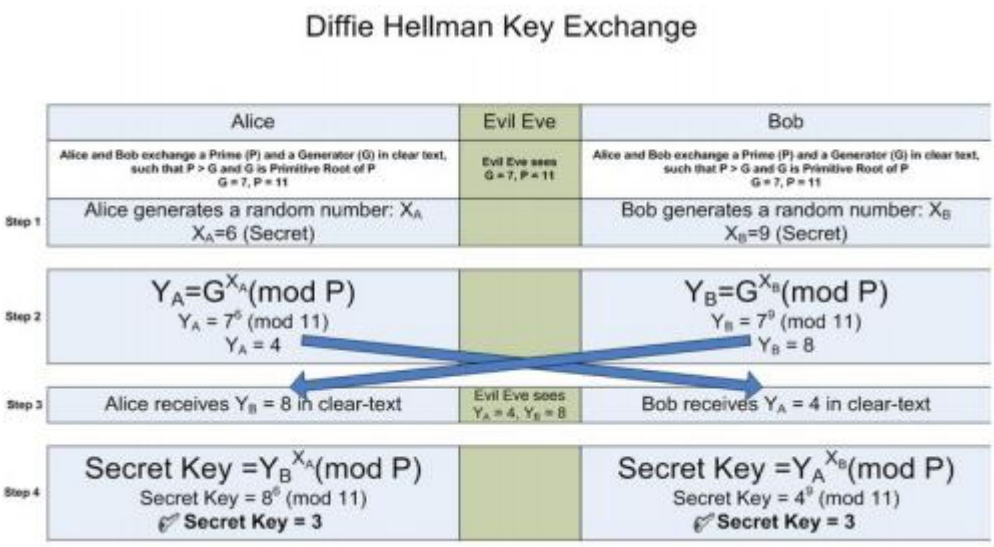
OBJECTIVE

Implement Diffie Hellman key exchange algorithm in Scilab/C/Python/R.

INTROUCTION TO DIFFIE HELLMAN KEY EXCHANGE ALGORITHM

Diffie –Hellman Key exchange algorithm:

The Diffie–Hellman key exchange method 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. Although Diffie–Hellman key agreement itself is an anonymous (non-authenticated) key-agreement protocol, it provides the basis for a variety of authenticated protocols, and is used to provide perfect forward secrecy in Transport Layer Security's ephemeral modes (referred to as EDH or DHE depending on the cipher suite).



SUBMISSION

You need to submit a detailed lab report to describe what you have done and what you have observed as per the suggested output format for all method ; you also need to provide explanation to the observations that are interesting or surprising. In your report, you need to answer all the questions as per the suggested formant listed above.

Code:

```
import sympy , secrets

def diffieHellmanExchange(g, n, x, y):
    """
    x => Alice's secret
    y => Bob's secret
    X => Alice computes from x and sends to Bob
    Y => Bob computes from y and sends to Alice
    """
    X = (g**x)%n
    Y = (g**y)%n
    return Y, X

if __name__ == '__main__':
    g = sympy.randprime(0, 10)
    print("The Prime Number g:",g)

    n = sympy.randprime(g, 1e6)
    print("The Prime Number n:",n)

    x,y = secrets.randbelow(20), secrets.randbelow(20)

    print("The secret key x(Alice's key):",x)
    print("The secret key y(Bob's key):",y)

    k1,k2 = diffieHellmanExchange(g,n,x,y)
    print("K1(Key recieved by Alice from Bob):",k1)
    print("K2(Key recieved by Bob from Alice):",k2)

    s1,s2 = (k1**x)%n,(k2**y)%n
    print("Secret Keys:",s1,s2)
```

Output:

Diffie Hellman Key Exchange

```
C:\Users\Swara>python diffie.py
The Prime Number g: 7
The Prime Number n: 721561
The secret key x(Alice's key): 19
The secret key y(Bob's key): 7
K1(Key recieved by Alice from Bob): 101982
K2(Key recieved by Bob from Alice): 653999
Secret Keys: 362629 362629
```

Observations:

- This is an interesting algorithm as it securely allows the two parties to verify each other over an insecure channel.
- It would be nearly impossible to crack the key for this algorithm if the value of p is large enough.

Conclusion:

- I learnt the procedure for encrypting messages using various cryptographical algorithms.
- I observed the various shortcomings a cryptographical algorithm can have and read about ways to overcome those shortcomings.
