SEM - VII - 2022-23

CNS Lab

B3 - 2019BTECS00094 - Sweety Shrawan Gupta

Assignment 11

Diffie-Hellman Key Exchange Algorithm

Theory:-

Diffie–Hellman key exchange is a method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as conceived by Ralph Merkle and named after Whitfield Diffie and Martin Hellman.

Code:

```
# Enter the approved prime number and the primitive root g.
Prime_no = int(input("Enter Prime No. q: "))
g = int(input("Enter Primitive root (a<q) : "))
# Enter  private key chosen by A and B
PkXa = int(input("Enter Private key of A (xa<q) : "))
PkXb = int(input("Enter Private key of B (xb<q) : "))
# Calculate public key of A and B
ya = g**PkXa % Prime_no
yb = g**PkXb % Prime_no
# Calculate shared session key K
ka = yb**PkXa % Prime_no
kb = ya**PkXb % Prime_no
print("A's Public Key Ya =",ya)
print("B's Public Key Yb =",yb)
print("Shared session key k =",ka)
```

Output:

```
In [1]: runfile('D:/CNS Lab/Diffie_hellman.py', wdir='D:/CNS Lab')

Enter Prime No. q: 23

Enter Primitive root (a<q) : 9

Enter Private key of A (xa<q) : 4

Enter Private key of B (xb<q) : 3
A's Public Key Ya = 6
B's Public Key Yb = 16
Shared session key k = 9

In [2]:
```

Diffie Hellman key exchange - live interaction of 2 programs (socket programming)

Code:

Server:

```
import java.net.*;
import java.io.*;

public class Server {
    public static void main(String[] args) throws IOException {
        try {
            int port = 8088;

            // Server Key
```

```
            int b = 3;

            // Client p, g, and key
            double clientP, clientG, clientA, B, Bdash;
            String Bstr;

            // Established the Connection
            ServerSocket serverSocket = new ServerSocket(port);
            System.out.println("Waiting for client on port " +
serverSocket.getLocalPort() + "...");
            Socket server = serverSocket.accept();
            System.out.println("Just connected to " +
server.getRemoteSocketAddress());

            // Server's Private Key
            System.out.println("From Server : Private Key = " + b);

            // Accepts the data from client
            DataInputStream in = new
DataInputStream(server.getInputStream());

            clientP = Integer.parseInt(in.readUTF()); // to accept p
            System.out.println("From Client : P = " + clientP);

            clientG = Integer.parseInt(in.readUTF()); // to accept g
            System.out.println("From Client : G = " + clientG);

            clientA = Double.parseDouble(in.readUTF()); // to accept A
            System.out.println("From Client : Public Key = " + clientA);

            B = ((Math.pow(clientG, b)) % clientP); // calculation of B
            Bstr = Double.toString(B);

            // Sends data to client
            // Value of B
            OutputStream outToclient = server.getOutputStream();
            DataOutputStream out = new DataOutputStream(outToclient);

            out.writeUTF(Bstr); // Sending B
```

```
            Bdash = ((Math.pow(clientA, b)) % clientP); // calculation of
Bdash

            System.out.println("Secret Key to perform Symmetric Encryption
= "
                    + Bdash);
            server.close();
        }

        catch (SocketTimeoutException s) {
            System.out.println("Socket timed out!");
        } catch (IOException e) {
        }
    }
}
```

```
D:\CNS Lab>java Server
Waiting for client on port 8088...
```

```
Just connected to /127.0.0.1:62912
From Server : Private Key = 3
From Client : P = 23.0
From Client : G = 9.0
From Client : Public Key = 6.0
Secret Key to perform Symmetric Encryption = 9.0
```

Client:

```
import java.net.*;
import java.io.*;

public class Client {
    public static void main(String[] args) {
        try {
```

```
            String pstr, gstr, Astr;
            String serverName = "localhost";
            int port = 8088;

            // Declare p, g, and Key of client
            int p = 23;
            int g = 9;
            int a = 4;
            double Adash, serverB;

            // Established the connection
            System.out.println("Connecting to " + serverName
                    + " on port " + port);
            Socket client = new Socket(serverName, port);
            System.out.println("Just connected to "
                    + client.getRemoteSocketAddress());

            // Sends the data to client
            OutputStream outToServer = client.getOutputStream();
            DataOutputStream out = new DataOutputStream(outToServer);

            pstr = Integer.toString(p);
            out.writeUTF(pstr); // Sending p

            gstr = Integer.toString(g);
            out.writeUTF(gstr); // Sending g

            double A = ((Math.pow(g, a)) % p); // calculation of A
            Astr = Double.toString(A);
            out.writeUTF(Astr); // Sending A

            // Client's Private Key
            System.out.println("From Client : Private Key = " + a);

            // Accepts the data
            DataInputStream in = new
DataInputStream(client.getInputStream());

            serverB = Double.parseDouble(in.readUTF());
            System.out.println("From Server : Public Key = " + serverB);
```

```
            Adash = ((Math.pow(serverB, a)) % p); // calculation of Adash

            System.out.println("Secret Key to perform Symmetric Encryption
= "
                    + Adash);
            client.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
D:\CNS Lab>java Client
Connecting to localhost on port 8088
Just connected to localhost/127.0.0.1:8088
From Client : Private Key = 4
From Server : Public Key = 16.0
Secret Key to perform Symmetric Encryption = 9.0
```