Cryptography-CS362, Assignment 2
Topic: Diffie Hellman Key Agreement Protocol
Assignment Due Date: 14/4/2022

In this assignment, you have to demonstrate working of Diffie Hellman Key Agreement protocol. You have to use openssl library (a famous and widely used library that implements many of the cryptographic algorithms) to demonstrate. At the end, show that a same key is shared between two users i.e. User A and User B. You have to generate the following files and upload the files on the assignment upload link:
1) Global Parameter file
2) Public key Private key files for User A and User B
3) Shared key file for User A and User B

You can use below steps to write this assignment or any material available on Internet.

1. Check if openssl is installed in your linux system or not. If not, go to this link to install openssl in your system.

2. Generate a Diffie-Hellman global domain parameters and save it in a file DHparam.pem
   Use the command: openssl genpkey -genparam -algorithm DH -out DHparam.pem

3. Display the generated global public parameters, using the following commands. See the difference between both the commands.
   ```
   cat DHparam.pem
   openssl pkeyparam -in DHparam.pem -text
   ```

4. The global public parameters generated in above steps can now be used by User A and User B in the protocol to generate their own public and private key. Save the keys in files DHkeyA.pem and DHkeyB.pem for User A and B respectively. Use the following commands for this step.
   For User A : openssl genpkey -paramfile DHparam.pem -out DHkeyA.pem
   For User B : openssl genpkey -paramfile DHparam.pem -out DHkeyB.pem

5. Display the public and private key using following command
   ```
   openssl pkey -in DHkeyA.pem -text -noout
   openssl pkey -in DHkeyB.pem -text -noout
   ```

6. Extract the public keys of user A and user B into separate file viz., DHpubA.pem and DHpubB.pem

7. Let us consider, both the users have exchanged their public keys with each other. That means, user A has DHpubB.pem and user B has DHpubA.pem. Using this keys, generate a shared secret key (128 bit binary file) at both sides using following command.
   ```
   openssl pkeyutl -derive -inkey DHkeyA.pem -peerkey DHpubB.pem -out sharedkeyA.bin
   openssl pkeyutl -derive -inkey DHkeyB.pem -peerkey DHpubA.pem -out sharedkeyB.bin
   ```

8. Check if same key is generated at both sides.

Last updated: April 7, 2022: by Sankita J Patel for Cryptography-CS362 assignment