U18CO018
Shubham Shekhaliya
CNS
Assignment – 9

Write a program to implement the Digital Signature Standard (DSS) algorithm.

Code:-

```
from Crypto.Util.number import *
from random import *
from hashlib import sha1

# Hash of message in SHA1
def hash_function(message):
    hashed=sha1(message.encode("UTF-8")).hexdigest()
    return hashed

# Modular Multiplicative Inverse
def mod_inverse(a, m) :
    a=a%m;
    for x in range(1,m) :
        if((a*x)%m==1) :
            return(x)
    return(1)

# Global parameters are q,p, and g
def parameter_generation():

    # primes of 8 bits in length in binary
    # q is prime divisor
    q=getPrime(5)
    # p is prime modulus
    p=getPrime(10)

    # Always p should be greater than q
    # because p-1 must be a multiple of q

    # to make sure that p not equal to q while generating randomly
    # and q is prime divisor of p-1
    while((p-1)%q!=0):
        p=getPrime(10)
        q=getPrime(5)
```

```
    print("Prime divisor (q): ",q)
    print("Prime modulus (p): ",p)

    flag=True
    while(flag):
        h=int(input("Enter integer between 1 and p-1(h): "))
        # h must be in between 1 and p-1
        if(1<h<(p-1)):
            g=1
            while(g==1):
                g=pow(h,int((p-1)/q))%p
            flag=False
        else:
            print("Wrong entry")
    print("Value of g is : ",g)

    # returning them as they are public globally
    return(p,q,g)

def per_user_key(p,q,g):

    # User private key:
    x=randint(1,q-1)
    print("Randomly chosen x(Private key) is: ",x)

    # User public key:
    y=pow(g,x)%p
    print("Randomly chosen y(Public key) is: ",y)

    # returning private and public components
    return(x,y)

def signature(name,p,q,g,x):
    with open(name) as file:
        text=file.read()
        hash_component = hash_function(text)
        print("Hash of document sent is: ",hash_component)
    r=0
    s=0
    while(s==0 or r==0):
        k=randint(1,q-1)
        r=((pow(g,k))%p)%q
        i=mod_inverse(k,q)
```

```
        # converting hexa decimal to binary
        hashed=int(hash_component,16)
        s=(i*(hashed+(x*r)))%q

    # returning the signature components
    return(r,s,k)

def verification(name,p,q,g,r,s,y):
    with open(name) as file:
        text=file.read()
        hash_component = hash_function(text)
        print("Hash of document received is: ",hash_component)

    # computing w
    w=mod_inverse(s,q)
    print("Value of w is : ",w)

    hashed=int(hash_component,16)

    # computing u1, u2 and v
    u1=(hashed*w)%q
    u2=(r*w)%q
    v=((pow(g,u1)*pow(y,u2))%p)%q

    print("Value of u1 is: ",u1)
    print("Value of u2 is: ",u2)
    print("Value of v is : ",v)

    if(v==r):
        print("The signature is valid!")
    else:
        print("The signature is invalid!")

global_var=parameter_generation()
keys=per_user_key(global_var[0],global_var[1],global_var[2])

# Sender's side (signing the document):
print()
file_name=input("Enter the name of document to sign: ")
components=signature(file_name,global_var[0],global_var[1],global_var[2],keys[0])

print("r(Component of signature) is: ",components[0])
print("k(Randomly chosen number) is: ",components[2])
print("s(Component of signature) is: ",components[1])
```

```
# Receiver's side (verifying the sign):
print()
file_name=input("Enter the name of document to verify: ")
verification(file_name,global_var[0],global_var[1],global_var[2],components[0],co
mponents[1],keys[1])
```

Output:-

```
PS D:\Course-Work\7th SEM\CNS\Assignment-9> python -u "d:\Course-Work\7th SEM\CNS\Assignment-9\CNS9.py"
Prime divisor (q):  17
Prime modulus (p):  647
Enter integer between 1 and p-1(h): 100
Value of g is :  53
Randomly chosen x(Private key) is:  8
Randomly chosen y(Public key) is:  218

Enter the name of document to sign: CNS9.txt
Hash of document sent is:  0318d138e57dfdca894d3074c112a1caa1c36bd9
r(Component of signature) is:  4
k(Randomly chosen number) is:  16
s(Component of signature) is:  6

Enter the name of document to verify: CNS9.txt
Hash of document received is:  0318d138e57dfdca894d3074c112a1caa1c36bd9
Value of w is :  3
Value of u1 is:  5
Value of u2 is:  12
Value of v is :  4
The signature is valid!
PS D:\Course-Work\7th SEM\CNS\Assignment-9>
```