Ex. No. 6	Implement Signature Scheme – Digital Signature Standard
Date of Exercise	05.10.2023

Aim

To implement digital signature standards and to prove its cryptography.

Description:

DSS is a set of rules and guidelines that specify how digital signatures should be generated and verified for the purpose of ensuring data integrity and authentication in electronic communications. Digital signatures are cryptographic techniques used to verify the authenticity and integrity of digital documents or messages.

DSA (Digital Signature Algorithm): The DSA is the core algorithm specified in DSS for generating and verifying digital signatures. DSA is based on the mathematical properties of modular exponentiation and the discrete logarithm problem. It is designed to provide data integrity and authentication.

Program:

```
import random


p = int(input("P: "))

q = int(input("Q: "))

x = int(input("Private_Key: "))

M = int(input("Hashed message: "))
```

```
h = random.randint(2, p-2)

g = int(h ** ((p-1)/q))


print(h, g)


y = (g ** x) % p

k = random.randint(1, q-1)


print("Public_Key: ", y)

print("Random Integer: ", k)


r = ((g**k) % p) % q


for i in range(1, q):

    if k*i % q == 1:

        s = (i*(M + x*r)) % q

        break


dig_sign = [r, s]
```

```
print("Digital Signature: ", dig_sign)


w = 0


r, s = dig_sign[0], dig_sign[1]


for i in range(2, 20):
    if (i*s) % q == 1:
        w = i
        break


u1 = (M * w) % q
u2 = (r * w) % q
v = (((g ** u1) * (y ** u2)) % p) % q


print("W: ", w)
print("u1: ", u1)
print("u2: ", u2)
print("V: ", v)
```

```
if v == r:
    print("Signature is Verified")
else:
    print("Signature is not Verified")
```

Output Screenshot:

```
P: 7
Q: 3
Private_Key: 2
Hashed message: 3
2 4
Public_Key:  2
Random Integer:  1
Digital Signature:  [1, 2]
W:  2
u1:  0
u2:  2
V:  1
Signature is Verified
```

Result

Thus, implementation of transposition cipher using Cryptography has been executed successfully.