|  |  |
| --- | --- |
| **Ex. No. 6** | **IMPLEMENT THE SIGNATURE SCHEME – DIGITAL SIGNATURE STANDARD** |
| **Date of Exercise** | **.2024** |

**Aim**

To write a Python program to implement the signature scheme named digital signature standard (Euclidean Algorithm).

**Description**

Standardization: DSS is a U.S. federal standard for digital signatures, defined in FIPS 186, which specifies the Digital Signature Algorithm (DSA) as its main method for creating digital signatures.

Hash and Signing: It combines a cryptographic hash function (like SHA-1, SHA-256) and DSA, RSA, or ECDSA to generate a signature based on the message's hash, ensuring message integrity and authenticity.

Public Key Infrastructure (PKI): DSS is based on asymmetric encryption, utilizing a public-private key pair for verification and signing, with the public key shared openly while the private key remains confidential.

**ALGORITHM**

STEP 1: Alice and Bob are investigating a forgery case of x and y.

STEP 2: X had a document signed by him but says he did not sign that document digitally.

STEP 3: Alice reads the two prime numbers p and a.

STEP 4: He chooses a random co-prime alpha, beta, and the x’s original signature x.

STEP 5: With these values, he applies it to the elliptic curve cryptographic equation to obtain y.

STEP 6: Comparing this ‘y’ with the actual y’s document, Alice concludes that y is a forgery.

**Program**

print("URK21CS1128")

from math import gcd

p = int(input("Enter p value: "))

h = int(input("Enter H value: "))

Hmac = int(input("Enter Hmac value: "))

k, q, s, w, r = 0, 0, 0, 0, 0

for Q in range(p//2, 0, -1):

if(p-1) % Q == 0:

for j in range(2, Q//2+1):

if Q % j == 0:

break

else:

q = Q

if q != 0:

break

g = (h \*\* ((p-1)//q)) % p

x = 1

print("x: ",x)

for k in range(2,q):

if gcd(k,q) == 1:

r = (pow(g,k,p))%q

s = (pow(k,-1,q) \* (Hmac + (r\*x))) % q

if gcd(s,q) == 1:

print(f"k:{k}")

print(f"r:{r}")

print(f"s:{s}")

w = pow(s,-1,q)

print(f"w:{w}")

u1 = (Hmac \* w) % q

u2 = (r\*w)% q

print(f"u1:{u1}")

print(f"u2: {u2}")

y = pow(g,x,p)

print(f"y:{y}")

v = (pow(g,u1,p) \* pow(y,u2,p))% p % q

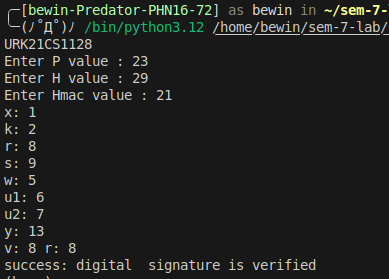
print(f"v: {v},r:{r}")

if v == r:

print("\n Success digital signature is verified")

break

**Output Screenshot**



**Result**

The program has executed successfully and the output is displayed in the console.