CRYPTOGRAPHY AND NETWORK SECURITY

LAB 9 : ELGAMMAL ENCRYPTION AND DECRYPTION

NAME : OM SUBRATO DEY

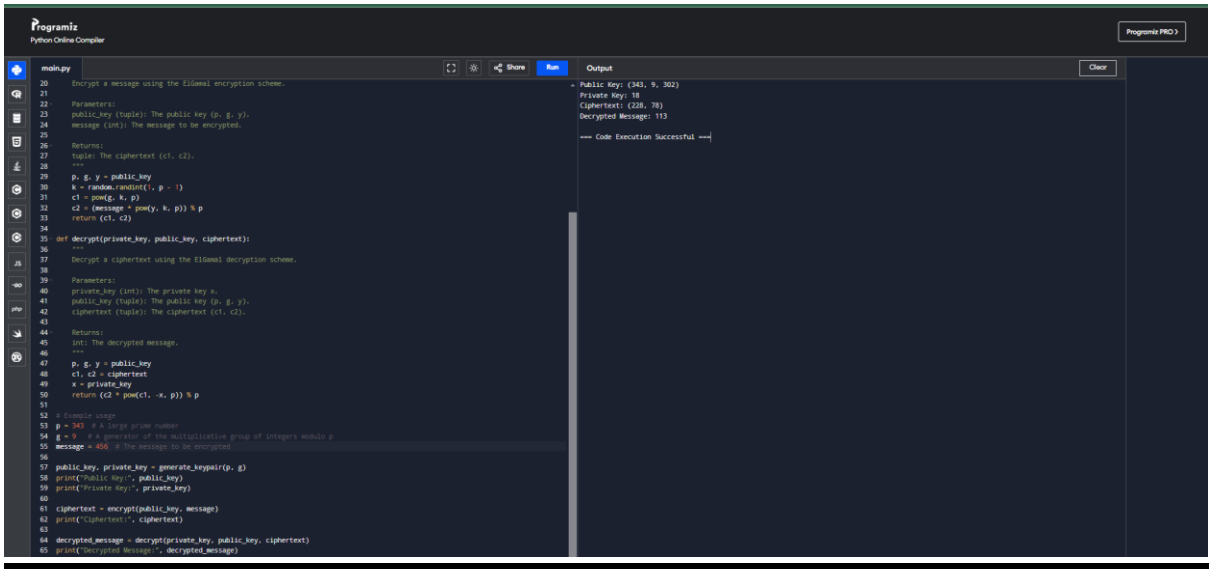
REGISTER NO.: 21BAI1876

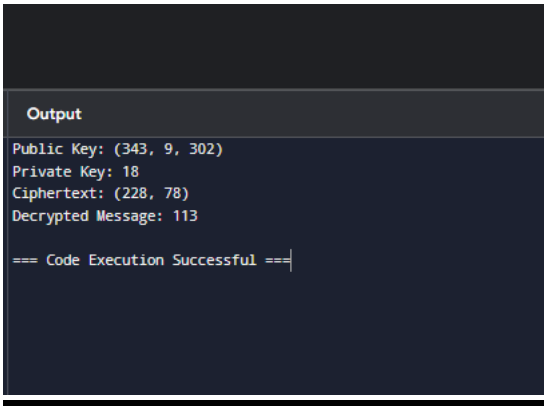
CODE:

```
import random

def generate_keypair(p, g):
    """
    Generate a public and private key pair.

    Parameters:
    p (int): A large prime number.
    g (int): A generator of the multiplicative group of integers modulo p.

    Returns:
    tuple: A tuple containing the public key (p, g, y) and the private key x.
    """
    x = random.randint(1, p - 1)
    y = pow(g, x, p)
    return ((p, g, y), x)

def encrypt(public_key, message):
    """
    Encrypt a message using the ElGamal encryption scheme.

    Parameters:
    public_key (tuple): The public key (p, g, y).
    message (int): The message to be encrypted.

    Returns:
    tuple: The ciphertext (c1, c2).
    """
    p, g, y = public_key
    k = random.randint(1, p - 1)
    c1 = pow(g, k, p)
    c2 = (message * pow(y, k, p)) % p
```

```
    return (c1, c2)

def decrypt(private_key, public_key, ciphertext):
    """
    Decrypt a ciphertext using the ElGamal decryption scheme.

    Parameters:
    private_key (int): The private key x.
    public_key (tuple): The public key (p, g, y).
    ciphertext (tuple): The ciphertext (c1, c2).

    Returns:
    int: The decrypted message.
    """
    p, g, y = public_key
    c1, c2 = ciphertext
    x = private_key
    return (c2 * pow(c1, -x, p)) % p

# Example usage
p = 343  # A large prime number
g = 9   # A generator of the multiplicative group of integers modulo p
message = 456  # The message to be encrypted

public_key, private_key = generate_keypair(p, g)
print("Public Key:", public_key)
print("Private Key:", private_key)

ciphertext = encrypt(public_key, message)
print("Ciphertext:", ciphertext)

decrypted_message = decrypt(private_key, public_key, ciphertext)
print("Decrypted Message:", decrypted_message)
```

OUTPUT:


Programiz
Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
Public Key: (343, 9, 302)
Private Key: 18
Ciphertext: (228, 78)
Decrypted Message: 113
=== Code Execution Successful ===


Output
Public Key: (343, 9, 302)
Private Key: 18
Ciphertext: (228, 78)
Decrypted Message: 113
=== Code Execution Successful ===