**IMPLEMENTATION OF RSA ALGORITHM**

**I. Introduction**

RSA(Rivest-Shamir-Adleman) Algorithm is an asymmetric or public-key cryptography algorithm which means it works on two different keys: Public Key and Private Key. The Public Key is used for encryption and is known to everyone, while the Private Key is used for decryption and must be kept secret by the receiver. RSA Algorithm is named after Ron Rivest, Adi Shamir and Leonard Adleman, who published the algorithm in 1977.

**II. Code**

import math

def gcd(a, b):

    while b != 0:

        a, b = b, a % b

    return a

def mod\_inverse(e, phi):

    for d in range(1, phi):

        if (e \* d) % phi == 1:

            return d

    return None

def rsa():

    p = int(input("Enter a prime number (p): "))

    q = int(input("Enter another prime number (q): "))

    msg = int(input("Enter the plaintext message (as an integer): "))

    n = p \* q

    phi = (p - 1) \* (q - 1)

    print(f"n = {n}")

    print(f"phi = {phi}")

    e = 0

    for e in range(2, phi):

        if gcd(e, phi) == 1:

            break

    print(f"e = {e}")

    d = mod\_inverse(e, phi)

    if d is None:

        print("Error: Could not find modular inverse for d.")

        return

    print(f"d = {d}")

    print(f"Public key: ({e}, {n})")

    print(f"Private key: ({d}, {n})")

    C = (msg \*\* e) % n

    print(f"Original message: {msg}")

    print(f"Encrypted message: {C}")

    M = (C \*\* d) % n

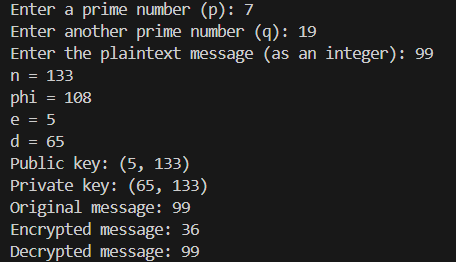
    print(f"Decrypted message: {M}")

rsa()

**III. Output**

**Example- 1 Example- 2**

**A screenshot of a computer

AI-generated content may be incorrect. **

**III. Explanation**

The algorithm for RSA is as follows:

1. Select 2 prime numbers, preferably large, p and q.
2. Calculate n = p\*q.
3. Calculate phi(n) = (p-1) \*(q-1)
4. Choose a value of e such as 1<e<phi(n) and gcd(phi(n), e) = 1.
5. Calculate d such that d = (e^-1) mod phi(n).

Here the public key is {e, n} and private key is {d, n.

If M is the plain text, then the cipher text C = (M^e) mod n.

Similarly, for decryption, the plain text M = (C^d) mod n.

As per Example -1 in the code,

Let p=5 and q=13, considering both to be prime numbers

* Now, n = p\*q = 5\*13 = 65
* phi(n) = (p-1) \*(q-1) = (5-1) \*(13-1) = 4\*12 = 48
* Value of e can be 5 since 1<5<48 and gcd (48, 5) = 1.
* Calculating d = 5^-1 mod 48 = 29.
* Therefore, public key = {5, 65} and private key = {29, 65.

Suppose our message is M=29.

**Encryption:** C = (M^e) mod n = 29^5 mod 65 = 9

**Decryption:** M = (C^d) mod n = 9^5 mod 65 = 29