**Project 5 - RSA Public Key Algorithm**

**1. Source code**

// RSA Algorithim.cpp : This file contains the 'main' function. Program execution begins and ends there.

//

#include <iostream>

#include <conio.h>

using namespace std;

bool isPrime(int n)

{

for (int i = 2; i <= n / 2; ++i) if (n % i == 0) return false;

return true;

}

int gcd(int a, int b) {

if (b == 0)

return a;

return gcd(b, a % b);

}

int main()

{

int p, q;

cout << "Input p: "; cin >> p;

while ((p > 10000 || p < 5001) || !isPrime(p))

{

cout << p << " must be a prime number in range [5000, 10000], please type again\n";

cout << "Input p: "; cin >> p;

}

cout << "Input p successfully" << endl;

cout << "Input q: "; cin >> q;

while ((q > 10000 || q < 5001) || !isPrime(q))

{

cout << q << " must be a prime number in range [5000, 10000], please type again\n";

cout << "Input q: "; cin >> q;

}

cout << "Input q successfully" << endl;

cout << "Verified: p and q are prime numbers in range[5000,10000]" << endl << endl;

int n = q \* p;

int phi = (q - 1) \* (p - 1);

cout << "Modulo n: " << n << endl;

cout << "Totien value O: " << phi << endl;

int e = 2;

while (e < phi)

{

if (gcd(e, phi) == 1 && gcd(e, n) == 1) break;

++e;

}

int k = 1;

int d = e;

while (d == e)

{

while ((k \* phi + 1) % e != 0) ++k;

d = (k \* phi + 1) / e;

++k;

}

printf("Public key is: <%d, %d>\n", e, n);

printf("Private key is: <%d, %d>\n\n", d, n);

if ((d \* e) % phi == 1) printf("%d and %d are multiplicative inverse based for O: %d", d, e, phi);

else printf("%d and %d are not multiplicative inverse based for O", d, e);

\_getch();

}

**2. A screen shot showing the validation and p, q**

Text

Description automatically generated

**3. A screen shot showing n, O, a private key, and public key.**

Text

Description automatically generated

**4. A screen shot showing the verification that d and e are multiplicative inverse for O**

