AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Department of Computer Science and Engineering

Program: Bachelor of Science in Computer Science and Engineering

Course Code: CSE 4174
Course Title: Cyber Security Lab
Academic Semester: Spring 2023

Assignment Topic: RSA (Rivest-Shamir-Adleman) Algorithm

Submitted on: 4/12/2023

Submitted by

Name: Parvez Ahammed
Student ID: 20200104129
Lab Section: C2

Question:
Devise a program using the RSA algorithm demonstrating the key set up and
encryption-decryption.

Answer:

```
#include <bits/stdc++.h>
using namespace std;
#define endl "\n"
#define ll long long int

int PUBLIC_KEY;
int PPRIVATE_KEY;
int n;
void viewInformation(int p, int q, int n, int
PHI, int e, int d)
{
    cout << "Prime 1 (p) = " << p << endl;
    cout << "Prime 2 (q) = " << q << endl;
    cout << "n (n) = " << n << endl;
    cout << "PHI (phi) = " << PHI << endl;
    cout << "Public Key (e) = " << e << endl;
    cout << "Private Key (d) = " << d << endl;
}
void init()
{
    int p, q;

    cin >> p >> q;

    n = p * q;
```

```
    int PHI = (p - 1) * (q - 1);
    int e = 2;
    while (true) {
        if (__gcd(e, PHI) == 1)
            break;
        e++;
    }
    PUBLIC_KEY = e;
    int d = 2;
    while (true) {
        if ((d * e) % PHI == 1)
            break;
        d++;
    }
    PPRIVATE_KEY = d;

    viewInformation(p, q, n, PHI, e, d);
}

ll modVal(int key, int message, int n)
{
    ll text = 1;
    while (key--) {
        text *= message;
        text %= n;
    }
    return text;
}

vector<int> encrypt(string message)
{
    vector<int> form;
    for (int i = 0; i < message.size(); i++)
```

```
        form.push_back(modVal(PUBLIC_KEY,
message[i], n));
    return form;
}

string decrypt(vector<int> encoded)
{
    string s;
    for (int i = 0; i < encoded.size(); i++)
        s.push_back(modVal(PPRIVATE_KEY,
encoded[i], n));
    return s;
}

int main()
{

    init();
    string message;
    cin.ignore();
    getline(cin, message);
    vector<int> coded = encrypt(message);
    cout << "Initial message:" << endl;
    cout << message;
    cout << endl;
    cout << endl;
    cout << "The encoded message (encrypted by
public key)" << endl;
    for (int i = 0; i < coded.size(); i++)
        cout << coded[i];
    cout << endl;
    cout << "The decoded message (decrypted by
private key)" << endl;
```

```
    cout << decrypt(coded) << endl;
    return 0;
}

/*
Sample Input :
587 599
Chilling with friends

Sample Output :
Prime 1 (p) = 73
Prime 2 (q) = 151
n (n) = 11023
PHI (phi) = 10800
Public Key (e) = 7
Private Key (d) = 1543
Initial message:
Chilling with friends

The encoded message (encrypted by public key)
10896648384041734173840314310665183615984032056483183456078478405194314377676 79
The decoded message (decrypted by private key)
Chilling with friends

*/
```

TC 1 Passed 106ms

Input: Copy

```
587 599
Chilling with friends
```

Expected Output: Copy

```
Prime 1 (p) = 587
Prime 2 (q) = 599
n (n) = 351613
PHI (phi) = 350428
Public Key (e) = 3
Private Key (d) = 233619
Initial message:
Chilling with friends

The encoded message (encrypted by public key)
300763700251027862048732048731027862761613788832768278707102786154444700253276863
697509210278632707527616129677411 4423
The decoded message (decrypted by private key)
Chilling with friends
```

Received Output: Copy

```
Prime 1 (p) = 587
Prime 2 (q) = 599
n (n) = 351613
PHI (phi) = 350428
Public Key (e) = 3
Private Key (d) = 233619
Initial message:
Chilling with friends

The encoded message (encrypted by public key)
300763700251027862048732048731027862761613788832768278707102786154444700253276863
697509210278632707527616129677411 4423
The decoded message (decrypted by private key)
Chilling with friends
```