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: Fall 2023

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

Submitted on: **9/05/2024**

Submitted by

Name: **Nafisa Tasnim Neha**
Student ID: **20200204020**
Lab Section: **A2**

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

Code:

```
#include <bits/stdc++.h>
using namespace std; bool
is Prime (int num)
{
if (num <= 1) {
return false;
}
//int sqrtNum = static_cast<int>(sqrt(num)); int
sqrtNum=sqrt(num);
for (int i = 2; i <= sqrtNum; i++) { if
(num % i == 0) {
return false;
}
}
return true;
}
int calculateGCD(int a, int b)
{
while (b != 0)
{
int temp = b; b =
a % b;
a = temp;
```

```
}
return a;
}
int modInverse(int a, int m)
{
a = a % m;
for (int x = 1; x < m; x++) if
((a * x) % m == 1) return x;
return 1;
}
int modExp(int base, int exp, int mod)
{
int result = 1;
base = base % mod;  for (;
exp > 0; exp /= 2)
{
if (exp % 2 == 1)
result = (result * base) % mod; base
= (base * base) % mod;
}
return result;
}
int encrypt(int message, int e, int n)
{
return modExp(message, e, n);
}
```

```
int decrypt(int ciphertext, int d, int n)
{
return modExp(ciphertext, d, n);
}
int main()
{
int p, q;
// Key Setup
cout << "Enter the value of p: "; cin
>> p;
cout << "Enter the value of q: "; cin
>> q;
if ((!isPrime(p)) || (!isPrime(q)))
{
cout << "Enter prime numbers!!!"<<endl; cout
<< "Enter the value of p: ";
cin >> p;

cout << "Enter the value of q: "; cin
>> q;

}

string msg;
cout << "Enter the plain text: ";
fflush(stdin);
getline(cin, msg);
```

```
vector<int> message; for
(char ch : msg)
{
message.push_back(static_cast<int>(ch));
}

int n, phi_n; n = p
* q;
phi_n = (p - 1) * (q - 1);

// calculate e int e,
d;
for (e = 2; e < phi_n; e++)
{
if (calculateGCD(e, phi_n) == 1)
{
break;
}
}

// calculate d
d = modInverse(e, phi_n);

cout << "Public Encryption Key {e, n}: " << "{" << e << "," << n <<
"}" << endl;
cout << "Public Decryption Key {d, n}: " << "{" << d << "," <<
```

```
n << "}" << endl;

// Encryption vector<int>
ciphertext;
cout << "\nEncrypted message: "<<endl; int
CipherText=0;
for (int ch : message)
{
ciphertext.push_back(encrypt(ch, e, n));
cout<<"'"<<static_cast<char>(ch)<<"': "<<encrypt(ch, e, n)<<endl;
}
cout<<"Encrypted Text"<<endl; for
(int ch : message)
{
cout<<static_cast<char>(encrypt(ch, e, n));
}
// Decryption
vector<int> decryptedMessage; cout
<< "\n\nDecrypted message: "; for (int
ch : ciphertext)
{
decryptedMessage.push_back(decrypt(ch, d, n)); cout
<< static_cast<char>(decrypt(ch, d, n));
}
cout<<endl;
return 0;
}
```

INPUT:

OUTPUT: