**Marathwada Mitra Mandal’s**

**COLLEGE OF ENGINEERING, PUNE**

**DEPARTMENT OF COMPUTER ENGINEERING**

**BE Comp-II 2020-21**

**SUBJECT: ICS SEMESTER-II**

**Lab Assignment3**

**Implementation of RSA algorithm**

**Theory:**

RSA (Rivest – Shamir - Adleman) is one of the first public-key cryptosystems and is widely used for secure communication. The RSA algorithm first generates two large random prime numbers, and then use them to generate public and private key pairs, which can be used to do encryption, decryption, digital signature generation, and digital signature verification. The RSA algorithm is built upon number theories, and it can be quite easily implemented with the support of libraries.

RSA is probably the most widely used encryption schema in the world today. RSA is a public-key cipher : anyone can encrypt messages using the public key; however, knowledge of the private key is required in order to decrypt messages, and knowing the public key does not help crack the private key. Public-key cryptography makes the secure Internet possible. Before public-key cryptography, keys had to be carefully exchanged between people who wanted to communicate, often by non- electronic means. Now RSA is routinely used to exchange keys without allowing anyone snooping on the channel to understand what has been communicated. RSA is believed to be very secure, based on the widely held assumption that no one has an efficient algorithm for factoring large numbers; deriving the private key from the public key appears to be as hard as factoring.

**RSA algorithm steps:**

**RSA Key Setup:**

* each user generates a public/private key pair by:
* selecting two large primes at random - p,q
* computing their system modulus N = p.q
  + note ø(N)=(p-1)(q-1)
* selecting at random the encryption key e
  + - where 1<e<ø(N), gcd(e,ø(N))=1 (e,ø(N)are co-prime)
* solve following equation to find decryption key d
  + e.d = 1 mod ø(N) and 0≤d≤N or
  + e.d mod ø(N) = 1
* publish their public encryption key: KU = {e,N}

keep secret private decryption key: KR = {d,p,q}

* to encrypt a message M the sender:
  + obtains **public key** of recipient KU={e,N}
  + computes: **C=Me mod N,** where 0≤M<N
* to decrypt the ciphertext C the owner:
  + uses their private key KR={d,p,q}
  + computes**: M=Cd mod N**

**RSA Example:**

1. Select primes: *p*=17 & *q*=11
2. Compute *n* = *pq* =17×11=187
3. Compute ø(*n*)=(*p–*1)(*q-*1)=16×10=160
4. Select e *:* gcd(e,160)=1; choose *e*=7
5. Determine d*: de=*1 mod 160 and *d* < 160 Value is d=23 since 23×7=161= 10×160+1
6. Publish public key KU={7,187}
7. Keep secret private key KR={23,17,11}

* 8. sample RSA encryption/decryption is:
* given message M = 88 (nb. 88<187)

encryption: C = 887 mod 187 = 11

decryption: M = 1123 mod 187 = 88

**RSA pseudocode for implementation:**

Use C/C++/Java programming to code RSA as per algorithm steps.

**Implementation steps:**

C++ Program:

#include<iostream>

using namespace std;

#include<math.h>

long long calculate(long long base, long long exp, long long n) {

long long y = 1;

while(exp > 0) {

long long r = fmod(exp,2);

if(r == 1) {

y = y \* base;

y = fmod(y, n);

}

base = base \* base;

base = fmod(base,n);

exp /= 2;

}

return y;

}

int gcd(int a, int h) {

int temp;

while(1) {

temp = a % h;

if(temp == 0)

return h;

a = h;

h = temp;

}

}

int main() {

long long p, q;

cout << "\nEnter p (first prime number) : "; // Get first prime no

cin >> p;

cout << "\nEnter q (second prime number) : "; // Get second prime no

cin >> q;

long long n = p \* q; // Calculate system modulus

cout << "\nN = " << n;

long long phi = (p - 1) \* (q - 1); // Calculate phi

cout << "\nphi = " << phi;

// Calculate encryption key e

long long e = 2;

while (e < phi) {

if(gcd(e,phi) == 1)

break;

else

e++;

}

cout << "\ne = " << e;

// Caclulate decryption key d such that e \* d = 1 mod phi

// e \* d = 1 + k \* phi where k is min

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

long long d;

long long k = 2;

while(1) {

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

if(fmod(d,1) == 0)

break;

k++;

}

cout << "\nd = " << d;

// Publish encryption key

cout << "\n\nEncryption key KU = {e,N} is {" << e << "," << n << "} \n";

long long msg;

// Get message

cout << "\nEnter message : ";

cin >> msg;

// Encrypt message

long long enc = calculate(msg,e,n);

cout << "\nEncrypted message : " << enc;

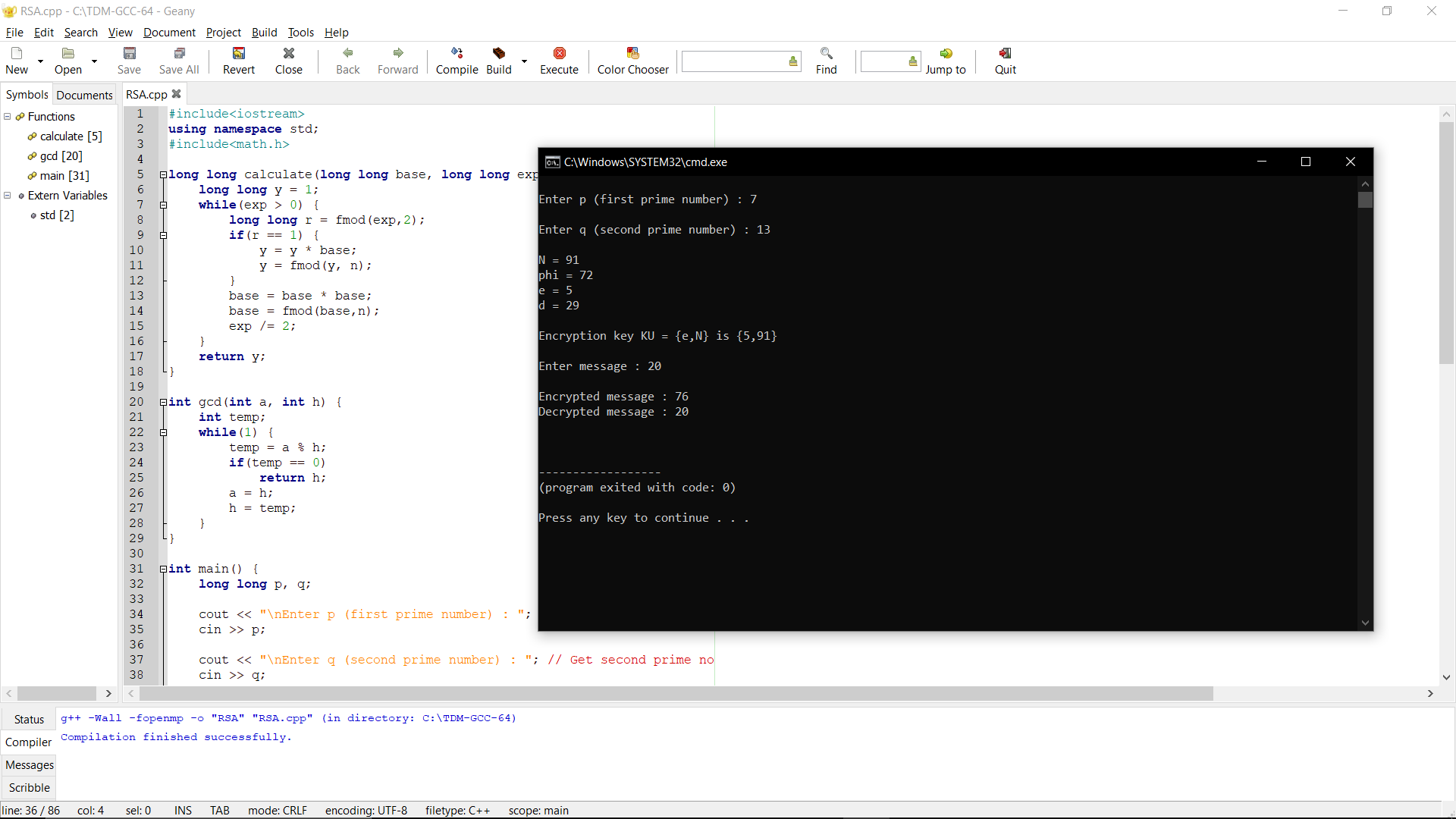
// Decrypt message

long long dec = calculate(enc,d,n);

cout << "\nDecrypted message : " << dec << " \n\n";

}

**Result with screenshot:**

****

**Outcome:**

We are able to

1. Identify need of information security, Public key Encryption Standards and RSA Algorithm

2. Design & implement Public key Encryption Standards and RSA algorithm