Name: Sohel Momin

Roll No:1513126/a4

CSS EXPERIMENT 7

**Program:**

**%% RSA CRYPTOSYSTEM**

% Two distinct prime nos. chosen by 1st party -> p and q

% modulus for public,private keys of 1st party -> n (= p \* q)

% private key of 1st party -> d

% public key of 1st party -> e

% the prime pairs selected should be such that 0 <= m < n ; where m is any plaintext !!!

% checkPrime.m, lcm.m, mulinv.m , modexpo.m should be in the same folder !!!

clc;clear variables;close all;

**%% Getting Valid primes p and q**

[p,q] = deal(-1,-1);

while (p == -1) || (q == -1)

if p == -1

temp = input('Enter a distinct prime no. of 1st party ->');

if checkPrime(temp)

p = temp;

end

elseif q == -1

temp = input('Enter another distinct prime no. of 1st party ->');

if checkPrime(temp) && (temp~=p) && (p\*temp>25)

q = temp;

end

end

end

**%% KEY GENERATION**

n = p\*q;

lambda = lcm(p-1, q-1);

for i=2:lambda-1

[mi,gcd] = mulinv(i,lambda);

if gcd == 1

[d,e] = deal(mi,i);

break

end

end

**%% KEY DISTRIBUTION**

% The second party gets the public key (n,e)

**%% ENCRYPTION**

msg = double(input('Enter the message to be encrypted by 2nd party -> ','s'));

enc\_msg = modexpo(msg,e,n);

**%% DECRYPTION**

dec\_msg = modexpo(enc\_msg,d,n);

fprintf('The message decrypted by the first party is -> %s \n',char(dec\_msg));

**OUTPUT :**

Enter a distinct prime no. of 1st party ->

23

Enter another distinct prime no. of 1st party ->

41

Enter the message to be encrypted by 2nd party ->

The quick brown fox jumps over lazy dog

The message decrypted by the first party is -> The quick brown fox jumps over lazy dog