Name: Sohel Momin

Roll No:1513126/a4

CSS EXPERIMENT 8

**Program:**

**%% RABIN CRYPTOSYSTEM**

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

% public key of 1st party -> n (= p \* q)

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

% n should be chosen according to the padding scheme, here it should be

% larger than binary replication of m

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

% some large primes congruent to 3mod4 => 9007,9043,9067,9127,9187,9227,9283,9319,9787

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. p such that mod(p,4)=3 ->');

if checkPrime(temp) && mod(p,4)==3

p = temp;

end

elseif q == -1

temp = input('Enter another distinct prime no. q such that mod(q,4)=3 ->');

if checkPrime(temp) && mod(p,4)==3 && (temp~=p)

q = temp;

end

end

end

**%% KEY GENERATION**

n = p\*q;

**%% KEY DISTRIBUTION**

% The second party gets the public key (n)

**%% ENCRYPTION**

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

%padding message by replication since we need to select correct message from 4 different outputs !

pad\_msg = padding(msg);

if any(pad\_msg > n)

fprintf('The chosen primes should be larger for encrypting this message\n')

return

end

enc\_msg = modexpo(pad\_msg,2,n);

**%% DECRYPTION**

dec\_msg = zeros(size(enc\_msg));

for i=1:length(enc\_msg)

a1 = modexpo(enc\_msg(i),(p+1)/4,p);

b1 = modexpo(enc\_msg(i),(q+1)/4,q);

a2 = -a1;

b2 = -b1;

p1 = CRT([a1,b1],[p,q]);

p2 = CRT([a1,b2],[p,q]);

p3 = CRT([a2,b1],[p,q]);

p4 = CRT([a2,b2],[p,q]);

choices = [p1,p2,p3,p4];

for j=1:4

m\_bin = dec2bin(choices(j));

pl = length(m\_bin);

if mod(pl,2)==0

if m\_bin(1:end/2) == m\_bin(end/2 +1 : end)

dec\_msg(i) = bin2dec(m\_bin(1:end/2));

end

end

end

end

disp(char(dec\_msg));

**OUTPUT:**

Enter a distinct prime no. p such that mod(p,4)=3 ->

9043

Enter another distinct prime no. q such that mod(q,4)=3 ->

9067

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