RSA Algorithm

|  |
| --- |
| #include <stdio.h> |
|  | #include <stdlib.h> |
|  | #include <math.h> |
|  |  |
|  | int checkPrime(int n) { |
|  | int i; |
|  | int m = n / 2; |
|  |  |
|  | for (i = 2; i <= m; i++) { |
|  | if (n % i == 0) { |
|  | return 0; // Not Prime |
|  | } |
|  | } |
|  |  |
|  | return 1; // Prime |
|  | } |
|  |  |
|  | int findGCD(int n1, int n2) { |
|  | int i, gcd; |
|  |  |
|  | for(i = 1; i <= n1 && i <= n2; ++i) { |
|  | if(n1 % i == 0 && n2 % i == 0) |
|  | gcd = i; |
|  | } |
|  |  |
|  | return gcd; |
|  | } |
|  |  |
|  | int powMod(int a, int b, int n) { |
|  | long long x = 1, y = a; |
|  |  |
|  | while (b > 0) { |
|  | if (b % 2 == 1) |
|  | x = (x \* y) % n; |
|  | y = (y \* y) % n; // Squaring the base |
|  | b /= 2; |
|  | } |
|  |  |
|  | return x % n; |
|  | } |
|  |  |
|  | int main(int argc, char\* argv[]) { |
|  | int p, q; |
|  | int n, phin; |
|  |  |
|  | int data, cipher, decrypt; |
|  |  |
|  | while (1) { |
|  | printf("Enter any two prime numbers: "); |
|  | scanf("%d %d", &p, &q); |
|  |  |
|  | if (!(checkPrime(p) && checkPrime(q))) |
|  | printf("Both numbers are not prime. Please enter prime numbers only...\n"); |
|  | else if (!checkPrime(p)) |
|  | printf("The first prime number you entered is not prime, please try again...\n"); |
|  | else if (!checkPrime(q)) |
|  | printf("The second prime number you entered is not prime, please try again...\n"); |
|  | else |
|  | break; |
|  | } |
|  |  |
|  | n = p \* q; |
|  |  |
|  | phin = (p - 1) \* (q - 1); |
|  |  |
|  | int e = 0; |
|  | for (e = 5; e <= 100; e++) { |
|  | if (findGCD(phin, e) == 1) |
|  | break; |
|  | } |
|  |  |
|  | int d = 0; |
|  | for (d = e + 1; d <= 100; d++) { |
|  | if ( ((d \* e) % phin) == 1) |
|  | break; |
|  | } |
|  |  |
|  | printf("Value of e: %d\nValue of d: %d\n", e, d); |
|  |  |
|  | printf("Enter some numerical data: "); |
|  | scanf("%d", &data); |
|  |  |
|  | cipher = powMod(data, e, n); |
|  | printf("The cipher text is: %d\n", cipher); |
|  |  |
|  | decrypt = powMod(cipher, d, n); |
|  | printf("The decrypted text is: %d\n", decrypt); |
|  | return 0; |
|  | } |