Final Year B. Tech, Sem VII 2022-23
PRN – 2020BTECS00211
Name – Aashita Narendra Gupta
Cryptography And Network Security Lab
Batch: B4
Practical No – 10

Title: Implementation of RSA Factorization challenge.

Theory:
The RSA Factoring Challenge was a challenge put forward by RSA Laboratories on March 18, 1991 to encourage research into computational number theory and the practical difficulty of factoring large integers and cracking RSA keys used in cryptography. They published a list of semiprimes (numbers with exactly two prime factors) known as the RSA numbers, with a cash prize for the successful factorization of some of them. The smallest of them, a 100-decimal digit number called RSA-100 was factored by April 1, 1991. Many of the bigger numbers have still not been factored and are expected to remain unfactored for quite some time, however advances in quantum computers make this prediction uncertain due to Shor's algorithm.

The RSA Factoring Challenges ended in 2007.[5] RSA Laboratories stated: "Now that the industry has a considerably more advanced understanding of the cryptanalytic strength of common symmetric-key and public-key algorithms, these challenges are no longer active."[6] When the challenge ended in 2007, only RSA-576 and RSA-640 had been factored from the 2001 challenge numbers.

The factoring challenge was intended to track the cutting edge in integer factorization. A primary application is for choosing the key length of the RSA public-key encryption scheme. Progress in this challenge should give an insight into which key sizes are still safe and for how long. As RSA Laboratories is a provider of RSA-based products, the challenge was used by them as an incentive for the academic community to attack the core of their solutions — in order to prove its strength.

The RSA numbers were generated on a computer with no network connection of any kind. The computer's hard drive was subsequently destroyed so that no record would exist, anywhere, of the solution to the factoring challenge.

Example:

The mathematics [edit]

RSA Laboratories states that: for each RSA number *n*, there exists prime numbers *p* and *q* such that

$n = p \times q$.

The problem is to find these two primes, given only *n*.

The prizes and records [edit]

The following table gives an overview over all RSA numbers. Note that the RSA Factoring Challenge ended in 2007[5] and no further prizes will be awarded for factoring the higher numbers.

The challenge numbers in white lines are part of the original challenge and are expressed in base 10, while the challenge numbers in yellow lines are part of the 2001 expansion and are expressed in base 2

RSA number	Decimal digits	Binary digits	Cash prize offered	Factored on	Factored by
RSA100	100	330	US$1,000[6]	April 1, 1991[9]	Arjen K. Lenstra
RSA110	110	364	US$4,429[6]	April 14, 1992[9]	Arjen K. Lenstra and M.S. Manasse
RSA120	120	397	US$5,898[6]	July 9, 1993[10]	T. Denny *et al.*
RSA129 [a]	129	426	US$100	April 26, 1994[9]	Arjen K. Lenstra *et al.*
RSA130	130	430	US$14,527[6]	April 10, 1996	Arjen K. Lenstra *et al.*
RSA140	140	463	US$17,226	February 2, 1999	Herman te Riele *et al.*
RSA150	150	496		April 16, 2004	Kazumaro Aoki *et al.*
RSA155	155	512	US$9,383[6]	August 22, 1999	Herman te Riele *et al.*
RSA160	160	530		April 1, 2003	Jens Franke *et al.*, University of Bonn
RSA170 [b]	170	563		December 29, 2009	D. Bonenberger and M. Krone [c]
RSA576	174	576	US$10,000	December 3, 2003	Jens Franke *et al.*, University of Bonn
RSA180 [b]	180	596		May 8, 2010	S. A. Danilov and I. A. Popovyan, Moscow State University[11]
RSA190 [b]	190	629		November 8, 2010	A. Timofeev and I. A. Popovyan
RSA640	193	640	US$20,000	November 2, 2005	Jens Franke *et al.*, University of Bonn

Code Snapshots:

```cpp
#include <bits/stdc++.h>
#define ll long long
#define ul unsigned long long
#define pb emplace_back
#define po pop_back
#define vi vector<ll>
#define vii vector<vector<ll>>
using namespace std;
void file(){
     ios_base::sync_with_stdio(false);
     cin.tie(NULL);}
ll M = 1e9 + 7;
int rem;
string longDivision(string number, int divisor)
{
    string ans;

    int idx = 0;
    int temp = number[idx] - '0';
    while (temp < divisor && number.length()>1)
        temp = temp * 10 + (number[++idx] - '0');

    while (number.size() > idx) {
        rem = temp % divisor;
        ans += (temp / divisor) + '0';
        temp = (temp % divisor) * 10 + number[++idx] - '0';
```

```
    }

    if (ans.length() == 0)
        return "0";
    if(rem==0)
    return ans;
    else return number;
}
int main(){
    string num;
    cout<<"Prime Factors:\n";
    cout<<"Enter Number : ";
    cin>>num;
    rem=0;

    unordered_map<int,int> mp;
    int len = num.size();

    string ans = longDivision(num,2);
    while(rem == 0){
        mp[2]++;
        num =  ans;
        ans = longDivision(num,2);
    }

      for (int i = 3; i <= 1000000; i = i + 2)
    {   string ans = longDivision(num,i);
        while (ans!="0" && rem==0)
        {   mp[i]++;
            num =  ans;
            ans = longDivision(num,i);
        }
    }

    cout<<"\n";
    cout<<"Prime Factor"<<" - "<<"Power"<<"\n";
    for(auto x:mp) cout<<x.first<<" - "<<x.second<<"\n";
}
```

Output Snapshots:

```
PROBLEMS  OUTPUT  TERMINAL  GITLENS  DEBUG CONSOLE

PS C:\Users\Ashitra\OneDrive\Desktop\7th sem\Practicals\CNS\Programs> cd "c:\Users\Ashitra\OneDrive\De
sktop\7th sem\Practicals\CNS\Programs\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile }
 ; if ($?) { .\tempCodeRunnerFile }
Prime Factors:
Enter Number : 295

Prime Factor - Power
59 - 1
5 - 1
PS C:\Users\Ashitra\OneDrive\Desktop\7th sem\Practicals\CNS\Programs>
```

Conclusion:

1. RSA is breakable byfactoring the “N”, the security of RSA is often basedon the integer factorization problem.
2. To resolve the prime factors of RSA, we can use integer factorization algorithm.
3. The RSA breach can be resolved easily by doing factorization on the public key.