Available online @ https://jjem.jnnce.ac.in
https:www.doi.org/10.37314/JJEM.2022.060114
Indexed in International Scientific Indexing (ISI)
Impact factor: 1.395 for 2021-22
Published on: 31July 2022

A Comprehensive study on RSA Prime Factorization Algorithms

Murthy D H R

Presidency University, Bangalore

murthydhr@presidencyuniversity.in

Abstract

RSA is the well-known Public Key Cryptographic Algorithm invented in 1985 which is more powerful even today because of its hardness in factoring the public modulus. The algorithm begins by considering two large prime numbers and all the remaining calculations are based on these two numbers. Once we are able to find any one prime factor successfully from the Public modulus which leads to cracking RSA. Many Cryptographers working hard to find a novel approaches for factorization. This paper deals with the study of different algorithms used for factorization and their key features in RSA. In the concluding part of this paper, the Author would like to explain a novel approach which looks fair for Carmichael Numbers

Keywords: *Public Key Cryptography, RSA, Prime Numbers, Factorization*

1. Introduction

Public Key Cryptography is a Cryptographic scheme uses two different keys in which Receiver's public key used for Encryption and only the intended receiver can decrypt it using his own private key. This is shown in Figure 1.

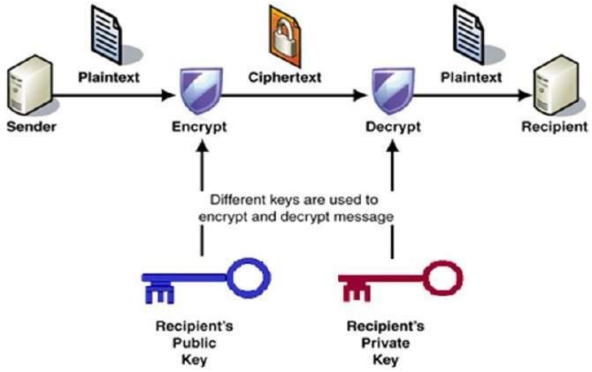

Figure 1: Public Key Cryptography model

Algorithm is well known popular Public Key Cryptography Algorithm designed by R. Rivest, A. Shamir and L. Adleman in 1977[1]. It is used widely in most of the e-commerce applications. Few examples of RSA Applications are web browser, e-mail, chat, Master Card, VISA, e-banking and many more. The main reason behind most usage of RSA Algorithm is the hardness of finding the two prime factors of a large composite number. The implementation steps for RSA algorithm is as given below.

1. Select two large prime numbers, x and y
2. Calculate N = x*y
3. Calculate the totient function
Phi(n) = (x-1)(y-1)
4. Select an integer e, such that e is co-prime to Phi(n) and 1 < e < Phi(n).
5. Calculate d such that e.d = 1 mod Phi(n).
6. Encryption : Given plain text 'M'
Cipher text $C = M^e \bmod N$
7. Decryption : $M = C^d \bmod N$

The key strength point in RSA Algorithm is in finding the two prime factors from public modulus 'N'. RSA laboratories has put forward many challenges for factorization since

1991[2]. The later part of this paper explains the complexity behind some of the popular factorization algorithms.

2. RSA Factoring challenge

Many Cryptographers and Mathematicians working hard to find the better and fast approach for Factorization. Security of RSA relies on the computational difficulty of factoring large integers. As computing power increases and more efficient factoring algorithms are discovered, the ability to factor larger and larger numbers also increases. Encryption strength is directly tied to key size, and doubling key length delivers an exponential increase in strength, although it does impair performance. RSA keys are typically 1024- or 2048-bits long, but experts believe that 1024-bit keys could be broken in the near future, which is why government and industry are moving to a minimum key length of 2048-bits

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. Brief history of factorization challenge offered and their details are given in Table 1 and the next milestone is RSA 896 and the prize money worth US$75,000.

Table 1 : RSA factorization history [2]

Decimal digits	**Binary digits**	**Factored on**	**Factored by**
100	330	Apr-91	Arjen K. Lenstra
110	364	Apr-92	Arjen K. Lenstra and Manasse
120	397	Jul-93	T. Denny et al.
129	426	Apr-94	Arjen K. Lenstra et al.
130	430	Apr-96	Arjen K. Lenstra et al.
140	463	Feb-99	Herman te Riele et al.
150	496	Apr-04	Kazumaro Aoki et al.
155	512	Aug-99	Herman te Riele et al.
160	530	Apr-03	Jens Franke et al.,
170	563	Dec-09	D. Bonenberger and M. Krone
174	576	Dec-03	Jens Franke et al.,
180	596	May-10	S. A. Danilov and I. A. Popovyan
190	629	Nov-10	A. Timofeev and I. A. Popovyan
193	640	Nov-05	Jens Franke et al.,
200	663	May-05	Jens Franke et al.,
210	696	Sep-13	Ryan Propper
212	704	Jul-12	Shi Bai et al
220	729	May-16	Shi Bai et al
230	762	Aug-18	Samuel S. Gross, Noblis, Inc.
232	768	Feb-20	Zamarashkin, et al
232	768	Dec-09	T. Kleinjung et al.
240	795	Dec-19	F. Boudot, et al
250	829	Feb-20	F. Boudot, et al

Some of the factorization algorithms discussing in this paper are :

- Trial Division
- Fermat’s factorization
- Polard (p-1)
- Polard Rho

1. RSA Factorization Algorithms

This chapter describes the working principles and mathematical background of few popular factorization Algorithms and also their time complexity

3.1 Trial Division

Trial division is the easiest of the algorithms to understand, but least efficient. The run-time of this Algorithm is exponential[3]. The algorithm simply iterates through a list of primes p1 to pk and tries to divide N. If it divides without a remainder, the algorithm has successfully factored N. If not, the algorithm repeats for next prime numbers in the series. It is very much necessary to try all the primes upto sqrt(N) in worst case.

3.2 Fermat's Factorization Method

This Algorithm is based on how you can represent an odd integer as the difference of two squares. Lets discuss with an example.

Take composite number N = 6557

Step (1) : try for a is ceil value of square root of 6557, which is 81.

Step (2): Perform the following task untill you get perfect square
b2 = 812 - 6557 = 4,
as it is a perfect square.
So, b = 2

Step (3). factors of 6557 are:
(a - b) = 81 -2 = 79 and
(a + b) = 81 + 2 = 83.

The Complexity of this Algorithm will be in the $O(N^{1/4})$. The disadvantage of this Algorithm is in step(2) we may not get perfect square and loop may go infinite.

3.3 Pollard p-1 algorithm

This is a better approach than trial division by making use of Modular Exponentiation and GCD[6].

Step (1): Set a=2

Step (2): for all i=2 to B perform following
$a=a^i$ mod N
d=gcd(a-1,N)
where B is prescribed bound

Step(3): if 1 < d < N then return (d)
Else
Failure

- There are B-1 modular exponentiations each requiring at most 2log2B modular multiplications
- The gcd can be computed in O(log2n)3 using the Extended Euclidean algorithm.
- Overall complexity will be O(BlogB(log n)2+(log n)3).

3.4 Pollard Rho algorithm

Named the Pollard rho method because of its similarity with the Greek letter ρ (rho) the Pollards rho method performs a random search using the cyclical properties of some Sequence Steps shown below.
(1). i =1
(2). x1= f(0, n-1) while true
(3). do i = i + 1
(4). Xi = f(0, n - 1)
(5). d = gcd(| xi+1- xi|, n)
(6). if d = 1 or d = n Output d

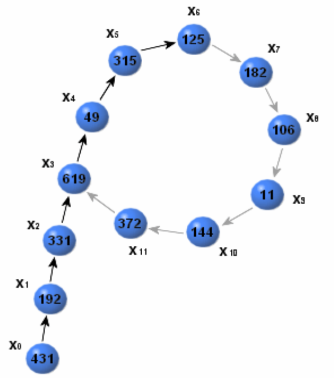

Figure 2: Pollard – Rho factors in a periodic cycle

3. Results and Discussion

RSA Prime factorization is a big challenge for Mathematicias and Cryptographers. Four different Algorithms for factorization are discussed in this paper with their important isues.
A Carmichael number is an Odd Composite Number which satisfies Fermat's Little Theorem and having 3 factors.
An integer 'N' is said to Carmichael if it

satisfies Fermat’s little theorem and three factors (a,b,c) such that
(a-1)/(N-1)
(b-1)/(N-1)
(c-1)/(N-1).
Where a, band c are the three factors of a given Carmichael Numbers.

Now the author would like to discuss a novel approach for factorization and found suitable for Carmichael Numbers. not promising for all odd composite numbers.
Basically these carnichael numbers are called as pseudoprime or false prime because Fermat’s little theorem is used to test whether given number is prime or not. Carmichael numbers are special cases because it is having three factors.

With the help of Fibonacci Series and matrix method of expanding the fibonacci series, finding the factors of Carmichael numbers seems to give fair results.

GMP Arithmetic Library [10] is used to handle big integer. For expanding the Fibonacci series in a faster way, following method can be used

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$$

The equation to generate Fibonacci Series is

Fn = Fn-2 + r Fn-1 ……..(1)

Where ‘r’ is defined as order of ‘N’ choosen from AKS Algorithm[9] and also ordr(n)>log^{2}N

By customising the fibonacci series in this fashion, good results came out for Carmichael Numbers..

6. Conclusion and Scope for future work

Four different approaches for prime factorization have been discussed with their complecity with an examples. A novel approach for factoring a Carmichael Number also discussed in thie paper Further this approach can be enhanced to get the factors of all odd composite numbers.

References

1. William Stallings, Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson Publisher, 2017, ISBN : 10:1292-15858-1

2. Wikipedia page - RSA Factoring Challenge, https://en.wikipedia.org/wiki/RSA_Factoring_Challenge, March 2021

3. Noah Zemel, Integer Factorization and RSA Encryption, https://digitalccbeta.coloradocollege.edu/pid/coccc:25808/datastream/OBJ

4. R. C. Daileda, The Fermat Factorization Method, Trinity University, http://ramanujan.math.trinity.edu/rdaileda/teach/s18/m3341/lectures/fermat_factor.pdf

5. Debdeep Mukhopadhyay, Presentation on The RSA Cryptosystem: Factoring the public modulus, IIT Kharagpur, https://cse.iitkgp.ac.in/~debdeep/courses_iitkgp/Crypto/slides/Factorization.pdf

6. Arpita Patra, IISc notes on Algorithms for factoring. https://www.csa.iisc.ac.in/~arpita/Cryptography15/CT9.pdf

7. Joakim Nilsson, Integer Factorization Algorithms, Bachelor Degree thesis, Umea University, https://www.divaportal.org/smash/get/diva2:1460632/FULLTEXT01.pdf

8. Eric W. Weisstein, Web Resource on Carmichael Numbers. https://archive.lib.msu.edu/crcmath/math/math/c/c060.htm

9.Agarwal, Kayal and Saxena, “Primes in P”, Research article on Primality testing. https://www.cse.iitk.ac.in/users/manindra/algebra/primality_v6.pdf

10.https://gmplib.org/