IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

# ANALYSIS OF FACTORING ALGORITHMS FOR NUMBER FACTORIZATION

[1]Seema Kute,[2] Dr. Chitra Desai,[3]Dr. Mukti Jadhav
[1]Research Student, [2]Professor, [3]Assistant Professor
[1]Dept. of CS & IT,
[1]Dr. B. A. M. U., Aurangabad, India

***Abstract:*** Information exchange on internet is increased in very high level because of this reason the security risk is also increased on the user's side. Cryptography is used to decrease the risk level. Public and private keys are the base of the Asymmetric cryptography. The time required for encryption and decryption improve the capability of information security. The foundation of public key cryptographic security therefore relies on this hardness of the RSA algorithm. This is a secure factorization based on modern classical cryptographic algorithms. Various techniques are available to find the numerical factors using different algorithms. The best-known set of techniques includes probabilistic integer factorization-based cryptography and geometric elliptic curve cryptography. Both techniques approach the factorization problem differently by generating at least one factor.

This paper examines Pollard Rho and Lenstra's algorithm used to decompose integer factorization-based cryptography, finds the number factors, and compares the factorization times for different key lengths. The experimental results will be beneficial to recognize the impact of computation complexity implicated in factoring process of Pollard Rho as well as Lenstra algorithms regarding to time taken in each case.

***Index Terms* - Pollard Rho, Lenstra's Algorithm, Result and Discussion.**

## I. INTRODUCTION

In today's internet world, security is very important aspect that ensures our information is available to only authenticate receiver and it protects our information from any kind of modifications in the information. For providing this security to our information on internet various cryptographic algorithms are used like RSA. RSA is one of the methods based on integer factorization. Cryptography is the science-based magical technique of transforming information into an encoded form. It is a very necessary aspect for a human being in today's world. Encryption is a very important aspect for practically doing any online task like cloud storage, emails, etc. From the start of the Modern Classical Cryptographic era, the Integer Factorization method has been used for encryption. RSA is the most popular integer factorization-based algorithm. It is generally used in the encryption process which was designed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977. [3] In 1985, Elliptic Curve Cryptography, a new type of encryption technique, was introduced by Miller and Kolblitz. NIIST implemented this Elliptic Curve Cryptography firstly from 161 bits. The security is given by a large key size in RSA Security. The same kind of security is achieved with the help of Elliptic Curve Cryptography using a very small key.[1,7] RSA like integer factorization techniques are break by using different techniques like Pollard rho integer factorization based algorithmic technique and Lenstra's Elliptic Curve based algorithmic technique.[13]

Indeed, every classical algorithm that is linked to the RSA and ECC algorithms can still be executed efficiently on a digital computer. This means that Lenstra's Elliptic Curve algorithm – which operates using computers – remains an efficient method for calculating long modulus values in this era.

In this paper, we present the comparative study between the Elliptic Curve-based Lenstra's algorithm and the Pollard Rho algorithm for finding the factors of the number which is the reverse process of the RSA algorithm.

## II. METHDOLOGY

In order to begin to calculate the factoring time of 'Number' on the basis of its length by using Pollard Rho algorithm and Lenstra's Elliptic Curve algorithm. Here we consider length of the 'Number' in bits that are 8, 16, 32, and 64. For this testing process Python Version 3.9.5 is used. These programs are running on a 14-inch Lenovo system, with 8 Gigabytes of memory, running OS Windows 10. Excel is used for data tables.

This research is to essentially examine the time required for 'Number' bit length 8, 16, 32 and 64 factored by Pollard rho and Lenstra's Elliptic Curve Algorithm.

## III. LITERATURE REVIEW

We are using various techniques to give success to the cryptography implementation concept. Key is the very main aspect of cryptography. Two different keys are used in asymmetric key cryptography for both the encryption and decryption process.[6] Many methods have been invented by various scientists like Integer factorization, Discrete Logarithms, and Elliptic Curves. On the basis of these methods, algorithms like RSA, Elgamal, and ECElgamal are developed. The implementation criteria of these algorithms are different. The RSA algorithm is very popular in integer factorization methods. The core concept of RSA is hardness of factoring two bigger numbers. The reliability of this algorithm totally depends on the hardness of the number.[8] If the public key (e,n) is known and the size of decryption key d is small then it is easy to place attack on RSA algorithm. A Trial division method has been initiated, to factorize RSA mod N. Simple arithmetic operations are used to find the factors which are closer to √N.[14]The Trial division technique require so much time (in days or months) to factor large numbers that don't have small prime factors. In such, instances other techniques are utilized like quadratic sieve and the General Number Field Sieve (GNFS).[15] The Quadratic Sieve is the second fastest technique after GNFS. It is Suitable for numbers less than 100 decimal digits or so on.[16] The most structured GNFS classical algorithm is used to factor the numbers larger than $10^{100}$ .[17] Quadratic Sieve and General Number Field Sieve algorithms have super polynomial time development a practical limit of 'N' digits is extended very rapidly. Just because of that in asymmetric key cryptography, values for number are selected to have large prime factors of same size in order that they are unable to factor by any publicly well-known technique in a functional time period on any accessible computer system such as super computers. By using the GNFS and resources of several supercomputers RSA-250 has been factored. The execution time was 2700 core years.[15]In 1975 John Pollard was invented Pollard's Rho algorithm for number factorization.[18] It is one of the most popular and used number factorization algorithm. As it's time complexity is corresponding to the square root of the size of the small prime factor and it requires very less amount of space for algorithm execution than others.[20]In 1985 Miller and Koblitz presented the utilization of the elliptic curve additive group for creating an asymmetric key cryptosystem. In 1988 Lenstra presented elliptic curve arithmetic based Integer Factorization method. General Number field sieve technique requires the testing of smoothness proportional to small composite numbers.In this scenario, Lenstra elliptic curve technique is the foremost choice for achieving smoothness testing. It is also required the less memory space.[19]Discrete logarithm is difficult to compute as compared to RSA because its base is to find the exponent given power in a well-known multiplicative group[9]. The Elgamal algorithm is based on this discrete logarithm problem. On other side, the small key size is the core point of Elliptic Curve Cryptography. With a small key size, it gives more security in less time.[10] Pollard and Lenstra's algorithms are integer factorization-based and elliptic curve-based algorithms used to find the factors of numbers, which is the base of our RSA integer factorization algorithm. Floyd's cycle-detecting algorithm is the base of Pollard Rho. [12]. Lenstra used an elliptic curve method to factorize the number.[5]

## IV. RESULT AND DISCUSSION

### 4.1 Pollard Rho Algorithm

The Pollard Rho algorithm is a probabilistic type of algorithm. Its work is based on the consecutive iterations of a random quadratic function. For a standard quadratic function, it produces numbers that are decreased modulo the number. Arbitrarily, the selected initial number is the main core part of consecutive iterations of this function, which produce a series that establishes looping after a certain point.

The numbers are congruent to each other modulo a factor of the number if two points are in the identical position of the loop. In pollard rho algorithm as the loop is actually a subgroup produced by the starting element as the identity and with the random function as the group operation. Thus, if two points are associated with the same class in the subgroup. They are identical to each other modulo the order of the subgroup, which divides the order of the whole group, which is the composite number. Subtraction gives a multiple of the order of the subgroup and returns the gcd of this and the modulus gives a factor of the modulus.[11]

**Algorithm**

```
1.  Select any random value of x and c
2.  Select  y = x and $f(x) = x^2-c$
3.  While divisor is not acquired
    i)      Update  x to f(x) % n
    ii)     Update y to f(y) % n
    iii)    Evaluate GCD of x – y and n
4.  If GCD $\neq 1$
    i)      If GCD is n, repeat  from step2 with other set of x,y and c
    ii)     Else GCD is our answer.
```

**Example**

N = 493; a = b = 4; k=1
F(a) = $a^2$ + 1(mod 493)
= $4^2$ + 1(mod 493)
= 17 mod 493
= 17
F(a) = $a^2$ + 1(mod 493)
= $(17)^2$ + 1(mod 493)
= 290 mod 493
= 290
F(a) = $a^2$ + 1(mod 493)
= $(290)^2$ + 1(mod 493)
= 841 mod 493
= 348
Now, Calculate GCD i.e. D
D = gcd (| 290 – 348 |, 493)
= gcd (58, 493)
= 29
One factor i.e prime1 = 29 now we find second factor i.e prime2 = 493÷29 = 17
Therefore, 493 = 29 * 17

### 4.2 Lenstra’s Elliptic Curve Algorithm

The geometric concept is the basis of Lenstra’s elliptical curve algorithm[4, 5] $y_2=ax_3+b$ function is the core part of this algorithm, which is a function of the arc length of an ellipse. When applying a multiplication operation on two points, that is ($x_1$, $y_1$) X ($x_2$, $y_2$) = ($x_3$, -$y_3$), where the point ($x_3$, $y_3$) indicates the third point on the line that passes through the points ($x_1$, $y_1$) and ($x_2$,$y_2$). Obtain Inverses such as (x, y) X (x, -y) = ∞, where ∞ is the point that all vertical lines intersect and the identity of the group [8]. A starting point and successive iterations of the group operation are applied. In the end, some point will be a generator of a subgroup, and its order will be a factor of the modulus. The best way to obtain the modulus factor is to take two initial points and successively operate on them in increasing numbers, so the kth iteration will have k! iterations of the group operation. Thus, if the current term is a generator of a subgroup, the GCD of k! and the modulus will be the order of the elements.

**Algorithm**

```
An integerN, we use the following steps to find factors of N. [2]
1 Examine that N isn’t divisible by 2 or 3, and that n isn’t a perfect power.
2 Select random integers A, x, y between 1 and N.
3 Let C = $y^2 − x^3 − Ax$ (mod N).
4 Compute T = GCD($4A^3 + 27C^2$; N).
    i)   If $1 < T < N$, we are done.
    ii)  If T = 1, move to Step 5.
    iii) If T = N, go back to Step 2 and pick a different A.
5 Let E be the Elliptic Curve E: $y^2= x^3 + Ax + C$, and let P = (x, y) ∈ E.
6 Select a number k which is a product of small prime numbers raised to small powers.
For example, a good selection is k = lcm(2, 3, . . . , C) for some integer C ≈ 100.
7 Calculate(k * P ) % N.
8 If k * P lies on E, go back to Step 2 and select different values for A, x, y. Otherwise, Step 7 generates a factor of N.
```

**Example**

N = 4453.

The Elliptic Curve E: $y^2 = x^3 + 10x - 2$, and point P = (1,3).

C =[ $y^2 – x^3 – 10x$] % 4453
= [$(3)^2 – (1)^3$ -10(1)] % 4453
= -2 % 4453
= 2
T = [$4A^3 + 27B^2$] % 4453
= [$4(10)^3 + 27(2)^2$] % 4453
= 4108 % 4453
= 4108
G = ($2y_1$) % 4453
= 2(3) % 4453
= 6 % 4453
= 6
GCD (6,4453) = 1
Compute 2P = (4332,3230)
$x_2$= 4332 $y_2$ = 3230
H = $2y_2$ * ($x_2$-$x_1$) % 4453
= [2(3230) * (4332 -1)] % 4453
= [27978260] % 4453
= 61
Since, GCD (61, 4453) = 61
Therefore, 61 ≠ 1
So, here we found one factor i.e. 61
Second factor = 4453 / 61 = 73
Therefore, 4453 = 61 * 73

**4.3 Observations:**

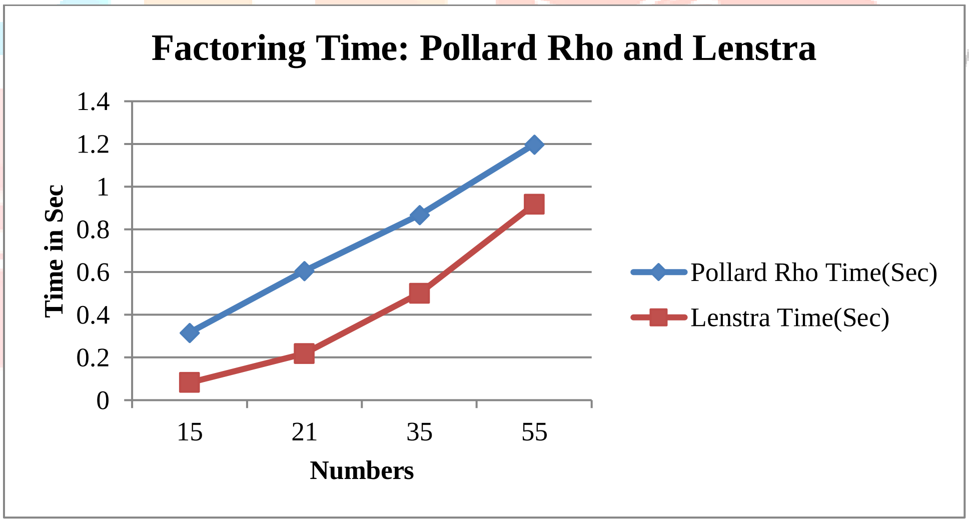

Fig.1Factoring time: Pollard rho and Lenstra Algorithm Timing Analysis

Figure. 1, shows that Lenstra’s Algorithm utilized less time during the 15, 21 35 and 55 numbers factorization process than Pollard rho algorithm.

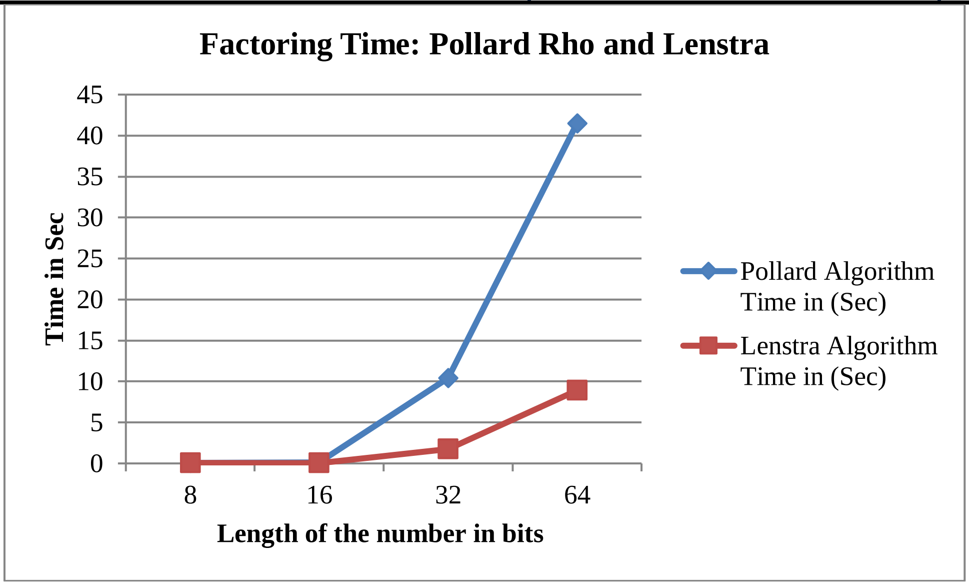

Fig2. Factoring time: Pollard rho and Lenstra's Algorithm with increasing number size Timing Analysis

Figure. 2, shows that Lenstra's Algorithm utilized less time during the number factorization process of different key lengths than the Pollard rho algorithm.

## V. CONCLUSIONS

Pollard-Rho is a probabilistic algorithm, so the limitation of this algorithm is that every time it is not possible to find the factor in the first run. Lenstra's algorithm is very difficult to implement because it is a geometric method-based algorithm. The major drawback of Lenstra algorithm is that we don't know which elliptic curve will create a factorization for mod number. Factoring time using Lenstra's Elliptic Curve Cryptographic algorithm takes less time to compute compared to Pollard rho algorithm.The experimental outcomes show that Lenstra's Elliptic Curve algorithm can complete the task faster than Pollard rho algorithm for all cases. Furthermore, the computation time is reduced by 51 to 68 percent.

## VI. REFERENCES

[1] Katie Groves,"Literature Review: Elliptic Curve Cryptography on Field Programmable Gate Array", Electrical and Computer Engineering Department,Tennessee Tech.
[2] Yuhan Yan," The Overview of Elliptic Curve Cryptography(ECC)", conf-CIAP 2022, IOP publishing journal of physics: conference series, 2386(2022) 012019, doi:10.1088/1742-6596/2386/1/012019.
[3] Mondal, M. & Ray, S. K., "Review on DNA Cryptography", (2019), Preprint rXiv:1904.05528.arxiv.org.
[4] Atkin, A. O. L. and F. Morain, " Elliptic curves and primality proving. Math. Comp.",
61(203):29--68, July 1993.
[5] Lenstra, H. W.,"Factoring integers with elliptic curves",Annals of Math, 126: 649-673, 1987.
[6] Anita Ganpati, NarenderTyagi, A Survey of Different Public-Key Cryptosystems, International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 6, Nov-Dec 2015.
[7] "Implementation of ElGamal Elliptic Curve Cryptography Over Prime Field Using C",DebabratBoruah,MonjulSaikia,ICICES2014, ISBN No.978-1-4799-3834-6/14.
[8]Abhishek, Dr. Vandana, "A Study On Modified Rsa Algorithm In Network Security", International Research Journal of Modernization in Engineering Technology and Science, Volume:04/Issue:04/April-2022.
[9]W. Stallings, "Cryptography and Network Security", Prentice Hall, Second Edition, 1998.
[10]ZarniSann, ThiThiSoe, KhaingMyatNwe "Comparison of Public Key Cryptography in Different Security Level", International Journal of Recent Development in Engineering and Technology, ISSN 2347-6435(Online) Volume 8, Issue 12, December 2019.
[11]Justin Moore "Runtime and Implementation of Factoring Algorithms: A Comparison", CSC290 Cryptology,December 20, 2003.
[12] Donald E. Knuth.,"The Art of Computer Programming Volume 2: Seminumer-ical Algorithms", Addison-Wesley, Boston, third edition, 1997.
[13] KritsanapongSomsuk,"The Improvement of Elliptic Curve Factorization Method to Recover RSA's Prime Factors",Symmetry 2021, 13, 1314.
[14] MalothBhavsingh, M. Sri Lakshmi, Dr. S. PremKumar,N. Parashuram," Improved Trial Division Algorithm by Lagrange"s Interpolation Function", International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169, Volume: 5 Issue: 5, pp: 1227 – 1231.
[15] https://en.wikipedia.org/wiki/Trial_division
[16] https://en.wikipedia.org/wiki/Quadratic_sieve
[17] https://en.wikipedia.org/wiki/General_number_field_sieve
[18] http://www.math.umd.edu/~immortal/ClassNotes/pollardrho.pdf

[19] Sa´ul Zapotecas-Mart´ınez, Cuauhtemoc Mancillas-L´opez, Francisco Rodr´ıguez-Henr´ıquez and Nareli Cruz-Cort´es,” Reconfigurable Hardware Implementation of the Lenstra Factorization Algorithm”, 3rd International Conference on Electrical and Electronics Engineering, Sep 2006, [ 4018024] ; https://doi.org/10.1109/ICEEE.2006.251939

[20] https://www.topcoder.com/thrive/articles/pollards-rho-algorithm