International Journal of Applied Engineering & Technology

Distribution of Multi-Reverse Primes within the Given Interval & Their Application in Asymmetric Cryptographic Algorithm

Deepak Kumar Sharma
Research Scholar, M.B. (Govt.) P.G. College, Haldwani, sharmadeepak0111209@gmail.com

Shubham Agarwal
Associate Professor, New Delhi Institute of Management, New Delhi, meshubhamagarwal@gmail.com

Anand Singh Uniyal
Professor, M.B. (Govt.) P.G. College, Haldwani, asuniyal0111@gmail.com

Date of Submission: 24th December 2020 Revised: 18th January 2021 Accepted: 03rd March 2021

How to Cite: **Sharma, D.K., Agarwal, S. and Uniyal A.S. (2021). Distribution of Multi-Reverse Primes within the Given Interval & Their Application in Asymmetric Cryptographic Algorithm. International Journal of Applied Engineering & Technology 3(1), pp.29-33.**

Abstract **- The distribution of prime numbers has fascinated mathematicians for over 2000 years. Ancient Greek mathematicians were the first to study the mathematical properties of prime numbers. The Prime Number Theorem is the hallmark theorem of prime distribution theory. It describes the asymptotic distribution of the prime numbers among the positive integers. The prime number theorem gives the idea to approximate the number of primes less than or equal to a given number n. Several researchers have given different results about the distribution of prime numbers but the study about the distribution of prime is still in continuation. In this paper we have given the new definition of primes, called multi-reverse primes and designed the java program for finding the Multi-reverse primes within the given interval and found their distribution. Also an algorithm for data encryption & decryption has been proposed for secure communication using multi-reverse primes in RSA Algorithm.**

Index Terms - Multi-reverse Primes, Prime Number Theorem, Prime numbers, Encryption & Decryption.

INTRODUCTION

The prime numbers are building blocks of arithmetic; every number is either a prime number or a number that equals the product of primes. Euclid proved 2000 years ago in his Elements that there were infinitely many prime numbers. That is, the sequence of prime numbers 2, 3, 5, 7, 11, 13, 17, 19, ……. is endless.

For example, 2, 3, 5 are the first three prime numbers, whereas $2^{57,885,161} - 1$ is the largest prime number to date, it has 17,425,170 digits and was found on 25 January 2013 by Dr. Curtis Cooper, An American Mathematician.

Let $\pi(x)$ denote the prime numbers up to x (Table I below gives some values of $\pi(x)$ for some large x), then Euclid's theorem of infinitude of primes actually says that $\pi(x) \to \infty$, as $x \to \infty$.

TABLE I
VALUES OF Π(X) FOR SOME LARGE X

X	Π(X)
10	4
100	25
1000	168
10000	1229
100000	9592
1000000	78498
10000000	664579
100000000	5761455
1000000000	50847534
10000000000	455052511
100000000000	4118054813
1000000000000	37609912018
10000000000000	346065536839
100000000000000	3204941750802

A much better result about the distribution of prime numbers is the Prime Number theorem, stating that

$$\pi(x) \sim x/\log_e x \quad (1)$$

In other words,

$$\lim_{x \to \infty} [\pi(x)/(x/\log_e x)] = 1 \quad (2)$$

where, e = 2.7182818……

Cryptography is also termed as the "science of secret writing", i.e. the method and ability of communicating or deciphering secret messages, or ciphers. In traditional cryptography, such as was available prior to the 1970s, the encryption and decryption operations are performed with the same key, called private key.

The face of cryptography was thoroughly changed when two Stanford University researchers, W. Diffie, et.al (1976 & 1979), invented an entirely new type of cryptography. In this type of cryptography the encryption and decryption could be done with a pair of different keys rather than with the same key. The decryption key would still have to be kept secret, but the encryption key could be made public without compromising the security of the decryption key. This concept was called public-key cryptography because of the fact that the encryption key could be made known to anyone.

LITERATURE REVIEW

Agarwal A., et.al (2020) introduced the application of Fibonacci primes as a method to secure any type of files / data for transmission and proposed the use of Fibonacci primes for data encryption and decryption.

Bairola M., et.al (2019) formulated new primes & proposed an algorithm for encryption & decryption for more secure cryptographic system.

Bairola M., et.al (2019) proposed an algorithm of data encryption & decryption using Fibonacci numbers. The proposed method provides two levels of security from an unauthorized excess as compared to the existing encryption methods.

Bairola M., et.al (2018) defined some new definition of primes like Sam prime, Adherent prime, Extreme prime and Reverse prime and use them in cryptographic system for secure communication.

Bairola M., et.al (2018) designed the java program for finding the Fibonacci primes within the given norms and find the distribution of Fibonacci primes.

Agarwal S., et.al (2017) defined multi-dimensional tree and proposed an encryption scheme using multi-dimensional tree in public key cryptography for security of ATM password.

Agarwal S., et.al (2015) showed that like RSA, ECC offers the highest strength-per-key-bit of any known public-key system of first generation techniques. With less bits required to give the same security, ECC has fared favorably compared to RSA.

Agarwal S., et.al (2015) defined the prime weighted graph and proposed an efficient encryption scheme using prime weighted graph in cryptographic system for secure communication and also designed java program for the encryption/decryption algorithm.

Agarwal S., et.al (2015) generated the role of primes in a different manner because fundamental theorem says that every number can be factorized as the product of primes. Our study will be about the pairs of composite numbers which can be factorized as a product of non-repeating primes. With the help of this we are going to develop certain distributions and their applications in the field of cryptography key system.

Raghu M.E., et.al (2015) proposed a method using classical cryptography to protect the data in faster way in which encryption and decryption are done in parallel using threads with the help of underlying hardware and also analyzed the time taken by sequential and parallel method.

Khadri, S.K.A., et.al (2014) highlighted the problem and provides some possible approach to solve the problem of secure data transmission using Fibonacci series. The Encryption of data is done by combining the original data with Fibonacci numbers to get a Cipher text which is non-understandable to any intruder. Only the receiver knows the logic to do so.

Mukherjee M., et.al (2014) proposed a novel technique that encrypted the message such a ways that the message encoded as well as hidden in an image. The proposed solution is to use image cryptography to hide textual message. The proposed technique use of an encryption technique that is based on Fibonacci series & image encryption and a secret key generated from the image.

Jamgekar R.S., et.al (2013) showed that the MREA algorithm is used to encrypt files and transmit encrypted files to other where it is decrypted. The algorithm works efficiently for small size files while it consumes time for large size of files, at an instant only one files can be encrypted and transmitted.

Sahu A., et.al (2012) proposed a new key generation algorithm based on palm print, which is used for encryption and decryption of an image. The scheme allows one party to send a secret image to another party over the open network, even if many eavesdroppers listen, this scheme gives reliable security.

Landge I., et.al (2012) described that both color and black & white images of any size saved in tagged image file format can be encrypted and decrypted using blowfish algorithm. Histogram of encrypted images is less dynamic and significantly different from the respective histogram of the original image.

Rajalakshmi P., et.al (2010) has presented a compact hardware-software co-design of Advanced Encryption Standards (AES) on the Field Programmable Gate Arrays (FPGA) designed for low-cost embedded systems.

MULTI-REVERSE PRIMES

Let p, q, and r be three different primes having α, β, and γ digits respectively, then two numbers $P_{\alpha,\beta,\gamma}$ = pqr and $P_{\gamma,\beta,\alpha}$ = rqp such that $P_{\alpha,\beta,\gamma} \neq P_{\gamma,\beta,\alpha}$ are called Multi-reverse primes.

Example: $P_{1,2,3}$ = 717103 and $P_{3,2,1}$ = 103177 are Multi-reverse primes as 717103 ≠ 103177

DISTRIBUTION OF MULTI-REVERSE PRIMES

I. Distribution of Multi-reverse Primes within the interval [1-25]

p(1,1,2) [3,2,17] = 3217 p(2,1,1) [17,2,3] = 1723
p(1,1,2) [3,5,11] = 3511 p(2,1,1) [11,5,3] = 1153
p(1,1,2) [3,5,17] = 3517 p(2,1,1) [17,5,3] = 1753
p(1,1,2) [3,7,19] = 3719 p(2,1,1) [19,7,3] = 1973
p(1,2,2) [3,17,23] = 31723 p(2,2,1) [23,17,3] = 23173
p(1,1,2) [7,2,13] = 7213 p(2,1,1) [13,2,7] = 1327
p(1,1,2) [7,5,23] = 7523 p(2,1,1) [23,5,7] = 2357
p(1,2,2) [7,13,17] = 71317 p(2,2,1) [17,13,7] = 17137
p(1,2,2) [7,17,11] = 71711 p(2,2,1) [11,17,7] = 11177
p(1,2,2) [7,17,13] = 71713 p(2,2,1) [13,17,7] = 13177
p(2,1,1) [11,5,3] = 1153 p(1,1,2) [3,5,11] = 3511
p(2,2,1) [11,17,7] = 11177 p(1,2,2) [7,17,11] = 71711
p(2,1,1) [13,2,7] = 1327 p(1,1,2) [7,2,13] = 7213
p(2,1,2) [13,2,19] = 13219 p(2,1,2) [19,2,13] = 19213
p(2,2,1) [13,17,7] = 13177 p(1,2,2) [7,17,13] = 71713
p(2,1,1) [17,2,3] = 1723 p(1,1,2) [3,2,17] = 3217
p(2,1,1) [17,5,3] = 1753 p(1,1,2) [3,5,17] = 3517
p(2,2,1) [17,13,7] = 17137 p(1,2,2) [7,13,17] = 71317
p(2,1,2) [19,2,13] = 19213 p(2,1,2) [13,2,19] = 13219
p(2,1,1) [19,7,3] = 1973 p(1,1,2) [3,7,19] = 3719
p(2,1,1) [23,5,7] = 2357 p(1,1,2) [7,5,23] = 7523
p(2,2,1) [23,17,3] = 23173 p(1,2,2) [3,17,23] = 31723

Total pairs of Multi-Reverse prime within the interval [1-25] = 22

II. Distribution of Multi-reverse Primes within the interval [25-50]

p(2,2,2) [29,41,31] = 294131 p(2,2,2) [31,41,29] = 314129
p(2,2,2) [29,43,41] = 294341 p(2,2,2) [41,43,29] = 414329
p(2,2,2) [31,29,43] = 312943 p(2,2,2) [43,29,31] = 432931
p(2,2,2) [31,41,29] = 314129 p(2,2,2) [29,41,31] = 294131
p(2,2,2) [37,43,47] = 374347 p(2,2,2) [47,43,37] = 474337
p(2,2,2) [37,47,41] = 374741 p(2,2,2) [41,47,37] = 414737
p(2,2,2) [41,31,43] = 413143 p(2,2,2) [43,31,41] = 433141
p(2,2,2) [41,43,29] = 414329 p(2,2,2) [29,43,41] = 294341
p(2,2,2) [41,47,37] = 414737 p(2,2,2) [37,47,41] = 374741
p(2,2,2) [43,29,31] = 432931 p(2,2,2) [31,29,43] = 312943
p(2,2,2) [43,31,41] = 433141 p(2,2,2) [41,31,43] = 413143
p(2,2,2) [43,37,47] = 433747 p(2,2,2) [47,37,43] = 473743
p(2,2,2) [47,37,43] = 473743 p(2,2,2) [43,37,47] = 433747
p(2,2,2) [47,43,37] = 474337 p(2,2,2) [37,43,47] = 374347

Total pairs of Multi-Reverse prime within the interval [25-50] = 14

III. Distribution of Multi-reverse Primes within the interval [50-75]

p(2,2,2) [53,59,73] = 535973 p(2,2,2) [73,59,53] = 735953
p(2,2,2) [59,53,73] = 595373 p(2,2,2) [73,53,59] = 735359
p(2,2,2) [59,73,61] = 597361 p(2,2,2) [61,73,59] = 617359
p(2,2,2) [61,59,71] = 615971 p(2,2,2) [71,59,61] = 715961
p(2,2,2) [61,73,59] = 617359 p(2,2,2) [59,73,61] = 597361
p(2,2,2) [71,59,61] = 715961 p(2,2,2) [61,59,71] = 615971
p(2,2,2) [73,53,59] = 735359 p(2,2,2) [59,53,73] = 595373
p(2,2,2) [73,59,53] = 735953 p(2,2,2) [53,59,73] = 535973

Total pairs of Multi-Reverse prime within the interval [50-75] = 8

IV. Distribution of Multi-reverse Primes within the interval [75-100]

p(2,2,2) [79,97,89] = 799789 p(2,2,2) [89,97,79] = 899779
p(2,2,2) [89,97,79] = 899779 p(2,2,2) [79,97,89] = 799789

Total pairs of Multi-Reverse prime within the interval [75-100] = 2

V. Graphical Representation of Distribution of Multi-reverse Primes

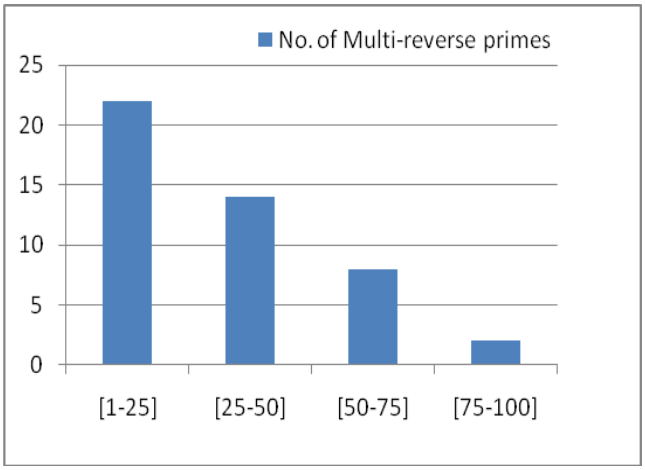

FIGURE 1
GRAPHICAL REPRESENTATION OF DISTRIBUTION OF MULTI-REVERSE PRIMES

Similarly, we can find the Multi-Reverse primes within any specific interval.

JAVA PROGRAM TO FIND THE MULTI-REVERSE PRIMES WITHIN ANY SPECIFIC INTERVAL

```
import java.io.*;
class multireverseprimes
{
 public static void main(String args[]) throws IOException
        {
        InputStreamReader read=new
        InputStreamReader(System.in);
        BufferedReader in=new BufferedReader(read);
        long min,max,n,num,rev;
        int i,j,k,c=0,d1,d2,d3,count=0;
        long prime[]=new long[10000];
        boolean flag,flag1,flag2;
        System.out.println("Enter lower limit");
        min=Long.parseLong(in.readLine());
        System.out.println("Enter upper limit");
        max=Long.parseLong(in.readLine());
                for(n=min;n<=max;n++)
                  {
                   flag=isPrime(n);
                   if(flag==true)
                    prime[c++]=n;
                }

                for(i=0;i<c;i++)
                {
                  for(j=0;j<c;j++)
                    {
                      for(k=0;k<c;k++)
                       {
                        if(i!=j && j!=k && k!=i)
                         {
                            d1=countDigits(prime[i]);
                            d2=countDigits(prime[j]);
                            d3=countDigits(prime[k]);
        num=prime[i]*(int)Math.pow(10,d2+d3)+prime[j]*(
int)Math.pow(10,d3)+prime[k];

        rev=prime[k]*(int)Math.pow(10,d2+d1)+prime[j]*(i
nt)Math.pow(10,d1)+prime[i];
                            if(num!=rev)
                            {
                            flag1=isPrime(num);
                            flag2=isPrime(rev);
                            if(flag1==true && flag2==true)
                                    {
                                    count++;
   System.out.print("p("+d1+","+d2+","+d3+")
["+prime[i]+","+prime[j]+","+prime[k]+"] \t= "+num+"\t");
   System.out.println("\tp("+d3+","+d2+","+d1+")
["+prime[k]+","+prime[j]+","+prime[i]+"]  \t= "+rev);
                                    }
                            }
                        }
                    }
                }
        }
        System.out.println("Total Multi-Reverse
prime from "+min+" to "+max+" are= "+count);
        }
        public static int countDigits(long n)
        {
                int c=0;
                while(n!=0)
                {
                        c++;
                        n=n/10;
                }
                return c;
        }
        public static boolean isPrime(long n)
        {
                long i;
                if(n==1)
                return false;
                for(i=2;i<=n/2;i++)
                {
                        if(n%i==0)
                        return false;
                }
                return true;
        }
}
```

APPLICATION OF MULTI-REVERSE PRIMES IN RSA CRYPTOSYSTEM

The RSA public-key cryptosystem was invented at MIT in 1977 by Rivest R., et.al (1978). The RSA algorithm is an asymmetric cryptography algorithm, uses a public key and a private key. A public key is shared publicly, while a private key is secret and must not be shared with anyone. One of the special numbers generated and used in RSA encryption is the modulus, which is obtained by multiplying two large primes. In order to break this cryptosystem, one must calculate the prime factorization of the modulus, which results in the two different primes. The complexity of RSA encryption depends on the difficultly to obtain this prime factorization.

- The public key in this cryptosystem consists of the value n, which is called the modulus, and the value e, which is called the public exponent.
- The private key consists of the modulus n and the value d, which is called the private exponent.

In the proposed method, Multi-reverse primes have been used, in place of large primes, to find the modulus. i.e, use two different primes $P_{\alpha,\beta,\gamma} \neq P_{\gamma,\beta,\alpha}$ to calculate the modulus. Application of Multi-reverse primes in RSA algorithm will help to protect the data with high level of security as there are number of Multi-reverse primes exist within the given interval, so each time if we consider different multi-reverse prime to find the modulus, it will be difficult for an

unauthorized person to obtain the private key. So the proposed method increases the level of security as well as it also increases the time to generate the keys, which will result in more secure cryptosystem.

I. Algorithm to Generate Keys for RSA Cryptosystem using Multi-reverse Primes

Step-1: Select a pair of large Multi-reverse primes $P_{\alpha,\beta,\gamma} \neq P_{\gamma,\beta,\alpha}$.
Step-2: Calculate modulus, $n = P_{\alpha,\beta,\gamma} * P_{\gamma,\beta,\alpha}$
Step-3: Calculate the **totient** function;

$$\phi(n) = (P_{\alpha,\beta,\gamma}-1)(P_{\gamma,\beta,\alpha}-1)$$

Step-4: Select an integer e, such that e is **co-prime** to $\phi(n)$ and $1<e<\phi(n)$.
The pair of numbers **(n,e)** makes up the **public key**.
Step-5: Calculate d such that $e.d = 1 \bmod \phi(n)$.
d can be calculated using the **extended euclidean algorithm**.
The **private key** can be represented as the pair **(n,d)**.

II. Encryption

Given a plaintext P, represented the text as numbers using ASCII values, then the ciphertext C is calculated using public key (n,e) as, $C = P^e \bmod n$.

III. Decryption

Using the private key (n,d), the plaintext can be found using, $P = C^d \bmod n$.
Represent the numbers as alphabets using ASCII values to find the original text.

CONCLUSION

The distribution of Multi-reverse primes presents a paradigm that can be useful to find the further distribution and analysis of primes within the given interval. New algorithms can be obtained from the proposed encryption & decryption method, which will be helpful in providing greater level of security.

REFERENCES

[1] Agarwal, A., Agarwal, S. & Singh, B.K., 2020. Algorithm For Data Encryption & Decryption Using Fibonacci Primes. *Journal of Mathematical Control Science and Applications (JMCSA)*, ISSN: 0974-0570, Vol. 6, No. 1, pp. 63-71.

[2] Bairola, M., Agarwal, S. & Uniyal, A.S., 2019. Application of Prime Numbers in Cryptographic System. Book Chapter in *Paradigm Shift in Management Practices for Fostering Excellence*, New Delhi Publishers, New Delhi, ISBN: 978-93-86453-92-1, pp. 246-251.

[3] Bairola, M., Agarwal, S. & Uniyal, A.S., 2019. Fibonacci Numbers in Data Security. *Mathematical Sciences International Research Journal*, ISSN: 2278-8697, Volume 8, Issue 2, pp. 11-15.

[4] Bairola, M. & Uniyal, A.S., 2018. Application of Advanced Cryptographic System. *International Journal of Mathematics Trends and Technology (IJMTT)*, ISSN: 2231-5373, Vol.55 (4), pp. 311-316.

[5] Bairola, M., Uniyal, A.S., 2018. Distribution of Fibonacci Numbers within the Given Norms. *Mathematical Sciences International Research Journal*, ISSN: 2278-8697, Volume 7, Spl. Issue, pp: 56-62.

[6] Agarwal, S., Uniyal, A.S., 2017. Enhancing the Security of ATM Password using Multi-dimensional Tree. *International Journal of Mathematics Research (IJMR)*, ISSN: 0976-5840, Volume 9, No. 1, pp. 53-58.

[7] Agarwal, S., Uniyal, A.S., 2015. Elliptic Curves: An Efficient and Secure Encryption Scheme in Modern Cryptography. *International Journal of Advance Research in Science & Engineering (IJARSE)*, (ISSN 2319-8354 (E)), Volume 04, Issue 03, pp. 134-143.

[8] Agarwal, S., Uniyal, A.S., 2015. Prime Weighted Graph in Cryptographic System for Secure Communication. *International Journal of Pure and Applied Mathematics (IJPAM)*, (ISSN: 1311-8080), Volume 105, No. 3, pp. 325-338.

[9] Agarwal, S., Uniyal, A.S. 2015. Multiprimes Distribution within a Given Norms. *International Journal of Applied Mathematical Sciences (JAMS)*, (ISSN 0973-0176), Vol. 8, No. 2 pp. 126-132.

[10] Raghu, M.E. & Ravishankar, K.C., 2015. Application of Classical Encryption Techniques for Securing Data – A Threaded Approach. *International Journal on Cybernetics and Informatics (IJCI)*, Vol.4, No.2, pp. 125-132.

[11] Khadri, S.K.A., Samanta, D. & Paul, M., 2014. Approach of Message Communication Using Fibonacci series: In Cryptology. *Engineering and Technology Publications*, Vol.2, No.2, pp. 168-171.

[12] Mukherjee, M. & Samanta, D., 2014. Fibonacci Based Text Hiding Using Image Cryptography. *Lecture Notes on Information Theory*, Vol.2, No.2, pp. 172-176.

[13] Jamgekar, R.S. & Joshi, G.S., 2013. File Encryption and Decryption using Secure RSA. *International Journal of Emerging Science and Engineering (IJESE)*, Vol.1, Issue-4, pp. 11-14.

[14] Sahu, A., Bahendwar, Y., Verma, S. & Verma, P., 2012. Proposed Method of cryptographic Key Generation for Securing Digital Image. *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol.2, Issue 10.

[15] Landge, I., Contractor, B., Patel, A. & Choudhary, R., 2012. Image Encryption and Decryption using Blowfish Algorithm. *World Journal of Science and Technology*, 2(3), pp. 151-156.

[16] Rajalakshmi, P., 2010. Hardware-software co-design of AES on FPGA. *International Conference of Advanced Study in Computing, Communications and Informatics*, pp. 1118–1122.

[17] Hellman, M. E., 1979. The Mathematics of Public Key Cryptography. *Scientific American*, 241, pp. 146-157.

[18] Diffie, W. & Hellman, M., 1976. New Directions in Cryptography. *IEEE Trans. Information Theory* 22, pp. 644-654.

[19] Diffie, W. & Hellman, M., 1976. Multi-user Cryptographic Techniques. *IEEE Trans. Information Theory*.

[20] Rivest, R.; Shamir, A. & Adleman, L., 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, Vol. 21 (2), pp. 120–126.

AUTHOR INFORMATION

Deepak Kumar Sharma, Research Scholar, Department of Mathematics, M.B. (Govt.) P.G. College, Haldwani, India - 263139.
Shubham Agarwal, Associate Professor, Department of Mathematics, New Delhi Institute of Management, New Delhi, India - 110062.
Anand Singh Uniyal, Professor, Department of Mathematics, M.B. (Govt.) P.G. College, Haldwani, India - 263139.