Stochastic Modeling and Applications
Vol. 25 No. 2 (July-December, 2021)
ISSN: 0972-3641

Received: 07th July 2021 *Revised: 28th August 2021* *Selected: 20th September 2021*

ANALYSIS OF FIBONACCI PRIMES & THEIR APPLICATION IN CRYPTOGRAPHY

A. AGARWAL[1], S. AGARWAL[2*], B.K. SINGH[3]

ABSTRACT. In this paper some novel recurrence relation between Fibonacci numbers has been generated. We have developed the JAVA program for finding the Fibonacci primes within the given interval and found the distribution of Fibonacci primes. An algorithm of data encryption & decryption has also been proposed using Fibonacci primes. The proposed method provides high level of security from an unauthorized excess as compared to the existing encryption methods.

1. Introduction

According to Hardy, G.H. *et al.* (1960) a number p is called a prime number if

(i) $p > 1$,

(ii) Except 1 and p, p has no positive divisors.

Otherwise it is called a composite number.

Fibonacci numbers can be defined as,

$$u_{n+1} = u_n + u_{n-1}, (n \geq 1) \quad \text{……. (1.1)}$$

with $u_0 = 0$, $u_1 = 1$, and $u_2 = 1$

A Fibonacci prime is a Fibonacci number that is both prime and Fibonacci. With 17103 digits, the largest known Fibonacci prime is u_{81839}, which was confirmed to be a prime by Broadhurst, D. *et al.* (2001). Lifchitz, H. (2009) discovered $u_{1968721}$, the largest known probable Fibonacci prime with 411439 digits.

Individuals and organizations are now facing security dangers as a result of technological advancements that allow them to channelize or transfer data or information across a transmission medium. Cryptography allows for the exchange or transmission of data or information while preventing unwanted access to the original data. The exchange of information can be accomplished using a variety of methods and algorithms. Three cryptographic approaches are commonly employed to achieve cryptographic goals: symmetric cryptography, public-key cryptography, and hash functions.

[*] ***Corresponding Author***

Key words and phrases*: Fibonacci Primes; Recurrence Relation; Distribution of Fibonacci Primes; Encryption; Decryption; Security*

Agarwal, S. *et al.* (2015) defined the prime weighted graph and proposed an efficient encryption/decryption algorithm for secure communication. Agarwal, S. *et al.* (2015) generated the pairs of composite numbers which can be factorized as a product of non-repeating primes and developed certain distributions and applied them in cryptography key system. Khadri, S.K.A. *et al.* (2014) proposed a method of encoding the data in which encryption of data is done by combining the original data with Fibonacci numbers. Agarwal, S. (2019) defined non-singular prime matrix and developed encryption and decryption algorithm in public key cryptography for secure communication. Mukherjee, M., *et al.* (2014) suggested a new method based on the Fibonacci sequence and image encryption. Bairola, M., *et al.* (2018) created several new prime definitions and applied them to cryptographic systems for secure data exchange. Joseph R. *et al.* (2012) developed an extraction algorithm that converts Unicode symbols to encrypted text and then back to ordinary text.

In this paper we have generated some new recurrence relation between Fibonacci numbers & found the distribution of Fibonacci primes. We have also designed the JAVA program for finding the Fibonacci primes within the given interval. An encryption & decryption algorithm has also been designed for security of data using Fibonacci primes.

2. Representation of Fibonacci Numbers

The Fibonacci number u_n, $n \geq 2$ can be expressed as the sum or the difference of squares of two Fibonacci numbers as

$$u_n = (u_\lambda)^2 \pm (u_\mu)^2 \qquad \text{……. (2.1)}$$

where n, λ, μ are indices of Fibonacci numbers.

Pythagorean-Fibonacci numbers are Fibonacci numbers that can be represented as the sum of squares of two different numbers, and Pythagorean-Fibonacci primes are Fibonacci primes that can be written as the sum of squares of two different primes. Because the Fibonacci numbers 5, 13, and 34 may be represented as the sum of squares of two other numbers, they are known as Pythagorean-Fibonacci numbers.

Example:

$$5 = 2^2 + 1^2$$
$$13 = 2^2 + 3^2$$
$$34 = 3^2 + 5^2$$

The Fibonacci number 13 is a Pythagorean-Fibonacci prime because it may be expressed as the sum of squares of two different primes.

3. Related Fibonacci Numbers (R_n)

If a number can be stated as the sum or difference of two Fibonacci numbers, it is referred to as a related Fibonacci number. Related Fibonacci number is denoted by R_n and is defined as,

$$R_n = u_{n+1} \pm u_{n-1} \qquad \ldots\ldots\ldots (3.1)$$

3.1 Related Fibonacci Numbers of First Kind (λ)

Related Fibonacci numbers of first kind are denoted by λ_n and are defined by,

$$\lambda_n = u_{n+1} + u_{n-1}, \quad \text{where } u_0 = 0,\ n \geq 1 \qquad \ldots\ldots\ldots (3.2)$$

$$\lambda_1 = u_2 + u_0 = 1$$
$$\lambda_2 = u_3 + u_1 = 3$$
$$\lambda_3 = u_4 + u_2 = 4$$
$$\lambda_4 = u_5 + u_3 = 7$$
$$\lambda_5 = u_6 + u_4 = 11$$
$$\lambda_6 = u_7 + u_8 = 18$$

The set of related Fibonacci numbers of first kind is given by,

$$\lambda = \{1, 3, 4, 7, 11, 18, 29, \ldots\}$$

3.2 Related Fibonacci Numbers of Second Kind (μ)

Related Fibonacci numbers of second kind are actual Fibonacci numbers.

$$\mu_n = u_{n+1} - u_{n-1}, \text{ where } u_0 = 0,\ n \geq 1 \qquad \ldots\ldots\ldots (3.3)$$

$$\mu_1 = u_2 - u_0 = 1$$
$$\mu_2 = u_2 - u_0 = 1$$
$$\mu_3 = u_4 - u_2 = 2$$
$$\mu_4 = u_5 - u_3 = 3$$
$$\mu_5 = u_6 - u_4 = 5$$
$$\mu_6 = u_7 - u_5 = 8$$

The set of related Fibonacci numbers of second kind is given by,

$$\mu = \{1, 1, 2, 3, 5, 8, 13, 21, \ldots\}$$

***Theorem*-1:** If u_n is a Fibonacci number such that, $u_{2n} = (u_{n+1})^2 - (u_{n-1})^2$ then the recurrence relation between Fibonacci numbers & related Fibonacci numbers is,

$$u_{2n} = \lambda_n \mu_n, \text{ for every } n \in N \qquad \ldots\ldots\ldots (3.4)$$

Proof: Given, $u_{2n} = (u_{n+1})^2 - (u_{n-1})^2$

Also we have,

$$\lambda_n = u_{n+1} + u_{n-1} \text{ and } \lambda = \{1, 3, 4, 7, 11, 18, 29, \ldots\}$$

$$\mu_n = u_{n+1} - u_{n-1} \text{ and } \mu = \{1, 1, 2, 3, 5, 8, 13, 21, \ldots\}$$

for n = 1, $u_{2n} = \lambda_n\mu_n$ gives

$$u_2 = \lambda_1\mu_1$$

$$1 = 1.1 = 1$$

Therefore, the condition is true for n = 1.

for n = 2, $u_{2n} = \lambda_n\mu_n$ gives

$$u_4 = \lambda_2\mu_2$$

$$3 = 3.1 = 3$$

Therefore, the condition is true for n = 2.

Let the condition is true for n = m, i.e.

$$u_{2m} = \lambda_m\mu_m$$

Now, for n = m+1

$$u_{2(m+1)} = (u_{m+2})^2 - (u_m)^2$$

$$= (u_{m+2} + u_m)(u_{m+2} - u_m)$$

From (3.2) and (3.3) we can write,

$$u_{2(m+1)} = \lambda_{m+1}\mu_{m+1}$$

Therefore, the statement is also true for n = m+1.

i.e., $u_{2n} = \lambda_n\mu_n$, for every $n \in N$

Hence the theorem is true for all values of n.

4. Java Program for Finding the Fibonacci Primes within the Interval

```
importjava.util.Scanner;
importjava.math.BigInteger;
public class fibonacci
{
        public static void main(Stringargs[])
        {
                Scanner in=new Scanner(System.in);
                long i;
                int z;
                boolean res1,res2;
                System.out.println("Enter starting value of interval…..");
```

```
		String a=in.next();
		BigInteger start=new BigInteger(a);
		System.out.println("Enter ending value of interval…..");
		String b=in.next();
		BigInteger end=new BigInteger(b);
		longfibo[]=new long[1000];
		fibo[0]=0;
		fibo[1]=1;
		int c=2;
		for(c=2;c<=998;c++)
		{
			fibo[c]=fibo[c-1]+fibo[c-2];
		}
		BigInteger one=new BigInteger(a);
		BigInteger o=new BigInteger("1");
		while(true)
		{
			i=one.longValue();
			res1=isPrime(i);
			res2=isFibo(i,fibo);
			if(res1==true && res2==true)
			{
				System.out.print(" "+i);
			}
			if(one.compareTo(end)==0)
			break;
			one=one.add(o);
		}
}
public static booleanisFibo(long n,longfibo[])
{
		int i;
		for(i=0;i<fibo.length;i++)
		{
			if(n==fibo[i])
```

```
                return true;
            }
            return false;
        }
        public static booleanisPrime(long n)
        {
            long i;
            boolean flag=true;
            if(n==1)
            return false;
            for(i=2;i<=n/2;i++)
            {
                if(n%i==0)
                {
                    flag=false;
                    break;
                }
            }
            return flag;
        }
}
```

5. Dead Zone

The dead zone is defined as the interval in which no Fibonacci prime number exists and is denoted by D_z. There is no Fibonacci prime in the range [501-1000], hence [501-1000] is a dead zone.

6. Distribution of Fibonacci Primes

Table 1 shows the distribution of Fibonacci primes within the given intervals.

Table 1- Distribution of Fibonacci primes

Interval	Fibonacci primes						No. of Fibonacci Primes
[1-500]	2	3	5	13	89	233	6
[501-1000]	-	-	-	-	-	-	0
[1001-5000]	1597	-	-	-	-	-	1

[5001-10000]	-	-	-	-	-	-	0
[10001-50000]	28657	-	-	-	-	-	1
[50001-100000]	-	-	-	-	-	-	0
[100001-500000]	-	-	-	-	-	-	0
[500001-1000000]	514229	-	-	-	-	-	1
[1000001-5000000]	-	-	-	-	-	-	0
[5000001-10000000]	-	-	-	-	-	-	0
[10000001-50000000]	-	-	-	-	-	-	0
[50000001-100000000]	-	-	-	-	-	-	0
[100000001-500000000]	433494437	-	-	-	-	-	1
[500000001-1000000000]	-	-	-	-	-	-	0
[1000000001-5000000000]	2971215073	-	-	-	-	-	1

7. Graphical Representation of Distribution of Fibonacci Primes

The graphical representation of the distribution of Fibonacci primes within the intervals is shown in Figure 1.

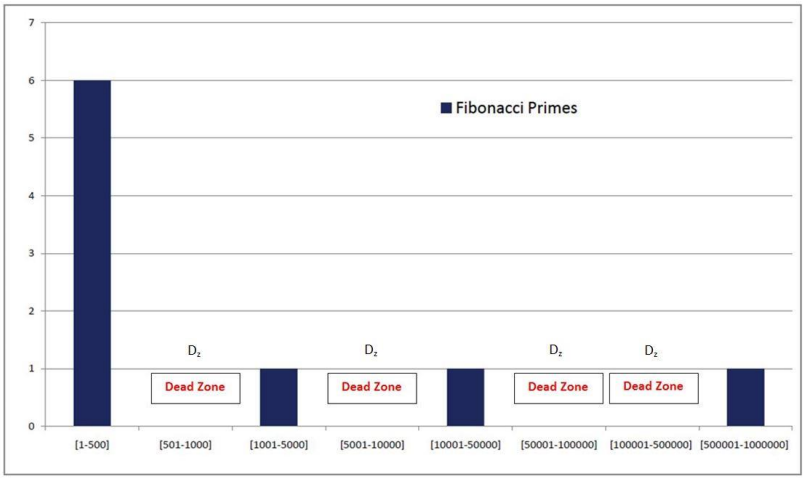

Figure 1- Graphical Representation of Distribution of Fibonacci Primes

8. Proposed Method for Data Encryption & Decryption

The suggested approach encrypts data by rearrangement using Fibonacci primes, rendering it unreadable to everyone except the individual who can decrypt it. Converting plain text to cipher text requires the usage of a security key, which is also used to convert cipher text back to plain text. It's tough to decipher the method,

which complicates the system's message retrieval for an unknown person. The benefit of utilizing Fibonacci primes is that it decreases suspicion while sending the data over an unsecured communication channel; otherwise, if an intruder tries to convert, nothing can be retrieved until the retrieval method and secret key are known.

8.1 Encryption Algorithm

Let the plain text (P) to be encrypted: **abcd**
Secret key (K): **xyz**
Length (L) of secret key: **3**

Step-1: Reverse the plain text (P) i.e. **abcd** is written as

dcba

Step-2: The secret key (K) is affixed i.e. the cipher text (C) becomes

dcbaxyz

Step-3: The Fibonacci prime series, up to the length (L) of the secret key is generated.
Here, the secret key length (L) is 3, so the first three Fibonacci primes viz. **2, 3,** and **5** are generated.

Step-4: The generated Fibonacci prime accelerates every element in odd position in the cipher text (C), so in the 1st iteration, all odd indexed elements would be advanced by 2, in the 2nd iteration by 3, in the 3rd iteration by 5.
(Note that once we reach "z", we wrap around, so if we forward "z" or "Z" by 2, the outcome is "b" or "B". Similarly, the generated Fibonacci prime inverts every element in even position in the cipher text, i.e. in the first iteration; all even indexed elements would be reversed by 2, in the second iteration by 3, in the third by 5. Wrap around once we've reached "a" i.e., if we reverse "a" or "A" by 2, the outcome is "y" or "Y".)
So, for the cipher text **dcbaxyz**, the encryption iteration would be,
1st iteration: fadyzwb when the first Fibonacci prime is 2
2nd iteration: ixgvcte when the second Fibonacci prime is 3
3rd iteration: nslqhoj when the third Fibonacci prime is 5
Therefore, the resultant encrypted text would be **nslqhoj**.

8.2 Decryption Algorithm

The encrypted text: **nslqhoj**
Secret key: **xyz**
Length (L) of secret key: **3**

Step-1: Secret key length (L) is 3, hence number of Fibonacci primes are 3 viz. **2, 3,** and **5**.

Step-2: The generated Fibonacci prime inverts every element in odd position in the cipher text (C), i.e. in the 1st iteration; all odd indexed items are inverted by 2, in the 2nd iteration by 3, in the 3rd iteration by 5. Similarly, the generated Fibonacci prime accelerates every element in even position in the cipher text, i.e. in the first iteration; all even indexed elements would be forwarded by 2, in the second iteration by 3, in the third by 5.
The decryption iteration for the encrypted text **nslqhoj** would be,
1st iteration: nslqhoj when the first Fibonacci prime is 2 will be lujsfqh
2nd iteration: lujsfqh when the second Fibonacci prime is 3 will be ixgvcte
3rd iteration: ixgvcte when the third Fibonacci prime is 5 will be **dcbaxyz**

Step-3: Extract last L characters from cipher text.
Here, the secret key length (L) is 3, therefore extracting 3 characters from cipher text gives **dcba**.

Step-4: Reverse the remaining text, which will be the original plain text, i.e. **abcd**, after extracting the key.

9. Conclusion

In this study, we have generated the recurrence relation between Fibonacci numbers and related Fibonacci numbers as $u_{2n} = \lambda_n \mu_n$, for every $n \in N$. We have investigated the Fibonacci prime distribution and discovered the dead zone, which will aid in determining the distribution of Fibonacci primes in the future. The suggested encryption and decryption strategy in this study provides the tools and methods for hiding data and preventing its unwanted alteration or usage. The proposed process is simple and straightforward to apply. As a secret message, any type of text data can be used, and it is conveyed through an open channel with a high level of security.

References

1. Hardy, G.H. and Wright, E.M., "An Introduction to the Theory of Numbers", 4th ed., Oxford, Clarendon Press (1960).
2. Broadhurst, D. and Water, B.D., "Number Theory Archives announcement".
3. Caldwell, C., "The Top Twenty: Fibonacci Numbers from the Prime Pages", Retrieved (2009-11-21).
4. Lifchitz, H., "PRP Top Records, Search for: F(n)", Retrieved (2009-11-21).
5. Agarwal, S. and Uniyal, A.S., "Prime Weighted Graph in Cryptographic System for Secure Communication", International Journal of Pure and Applied Mathematics, Vol. 105, No. 3 (2015), pp.325-338.
6. Agarwal, S. and Uniyal, A.S., "Multiprimes Distribution within a Given Norms", International Journal of Applied Mathematical Sciences, Vol. 8, No. 2 (2015), pp.126-132.
7. Khadri, S.K.A., Samanta, D. and Paul, M., "Approach of Message Communication Using Fibonacci Series: In Cryptology", Engineering and Technology Publications", Vol. 2, No. 2 (2014), pp.168-171.
8. Agarwal, S., "Encryption & Decryption Using Linear Algebra: Advancement in Public Key Cryptography", Indian Journal of Economics and Business, Vol. 18, No.1, (2019), pp.167-180.
9. Mukherjee, M. and Samanta, D., "Fibonacci Based Text Hiding Using Image Cryptography", Lecture Notes on Information Theory, Vol. 2, No. 2 (2014).
10. Bairola, M. and Uniyal, A.S., "Application of Advanced Cryptographic System", International Journal of Mathematics Trends and Technology, Vol. 55 (4) (2018), pp.311-316.
11. Joseph R. and Sundaram, V., "Secured Communication through Fibonacci Numbers and Unicode Symbols", International Journal of Scientific & Engineering Research, Vol. 3, Issue 4 (2012), pp.1-5.

[1]**AKANSHA AGARWAL**: RESEARCH SCHOLAR, DEPARTMENT OF MATHEMATICS, IFTM UNIVERSITY, MORADABAD, INDIA-244102

EMAIL: agw.akansha@gmail.com

[2]**SHUBHAM AGARWAL**: ASSOCIATE PROFESSOR, DEPARTMENT OF MATHEMATICS, NEW DELHI INSTITUTE OF MANAGEMENT, NEW DELHI, INDIA-110062

EMAIL: meshubhamagarwal@gmail.com

[3]**B.K. SINGH**: PROFESSOR, DEPARTMENT OF MATHEMATICS, IFTM UNIVERSITY, MORADABAD, INDIA-244102

EMAIL: drbksingh@iftmuniversity.ac.in