NEW CLASSES OF RANDOM SEQUENCES FOR CODING AND CRYPTOGRAPHY APPLICATIONS

By

KIRTHI KRISHNAMURTHY VASUDEVA MURTHY

Bachelor of Engineering in Instrumentation Technology

Visvesvaraya Technological University

Bangalore, Karnataka

2013

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2016

NEW CLASSES OF RANDOM SEQUENCES FOR

CODING AND CRYPTOGRAPHY APPLICATIONS

Thesis Approved:

Dr. Subhash Kak

Thesis Adviser

Dr. Keith A Teague

Dr. George Scheets

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my master's thesis advisor Dr. Subhash Kak. He continually and convincingly conveyed a spirit of adventure and motivation in regard to research with profound patience handling me in all my tough situations. Without his persistent support, this thesis would not have been possible.

I render my sincere thanks to my committee members, Dr. Keith Teague and Dr. George Scheets for their support and guidance in final stages of thesis presentation and document review. I thank Oklahoma State University for giving opportunity to utilize and enhance my technical knowledge. Lastly, I would like to thank my parents and friends for constant encouragement emotionally and financially to pursue masters and complete thesis research and document at OSU.

Name: KIRTHI KRISHNAMURTHY VASUDEVA MURTHY

Date of Degree: MAY, 2016

Title of Study: NEW CLASSES OF RANDOM SEQUENCES FOR CODING AND CRYPTOGRAPHY APPLICATIONS

Major Field: ELECTRICAL ENGINEERING

Abstract: Cryptography is required for securing data in a digital or analog medium and there exists a variety of protocols to encode the data and decrypt them without third party interference. Random numbers must be used to generate keys so that they cannot be guessed easily.

This thesis investigates new classes of random numbers, including Gopala-Hemachandra (GH) and Narayana sequences, which are variants of the well-known Fibonacci sequences. Various mathematical properties of GH and Narayana sequences modulo prime have been found including their periods. Considering GH sequences modulo prime p, the periods are shown to be either (p-1) (or a divisor) or (2p+2) (or a divisor) while the Narayana sequence for prime modulo have either p^2+p+1 (or a divisor) or p^2-1 (or a divisor) as their periods. New results on the use of the Narayana sequence as a universal code have been obtained.

It is shown that the autocorrelation and cross correlation properties of GH and Narayana sequences justify their use as random sequences. The signal to noise ratio values are calculated based on the use of delayed sequences to carry different sets of data in wireless applications. The thesis shows that GH and Narayana sequences are suitable for many encoding and decoding applications including key generation and securing transmission of data.

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION 1

1.1) Introduction to Cryptography 1
1.2) Random Sequences 3
1.3) Spread Spectrum 4
1.4) Universal Codes 6
1.5) Thesis Organization 8

II. REVIEW OF LITERATURE 10

2.1) Secure Multiparty Communication 10
2.2) CDMA Communication 11
2.3) Electronic Commerce 12
2.4) Fibonacci Sequences 13

III. G-H SEQUENCES 15

3.1) Autocorrelation properties 15
3.1.1) Introduction 15
3.1.2) Periods of Fibonacci and GH sequences mod m 16
3.1.3) Generating the sequence 17
3.1.4) Mapping into random binary sequence 18
3.1.5) Autocorrelation properties of GH sequences 20

3.2) Cross correlation properties 22
3.2.1) Introduction 22
3.1.2) Periods of shift registers and Fibonacci sequences 22
3.1.3) Cross correlation of GH and PN sequences 23
3.1.4) Comparison between binary GH and PN sequences 27
3.1.4) Signal-to-noise ratio for binary GH sequences 28

Chapter Page

IV. NARAYANA SEQUENCES AND VARIANTS 32

4.1) Narayana sequences.....32
4.1.1) Introduction 32
4.1.2) Generation of Narayana series.....33
4.1.3) Periods of Narayana series modulo p 34
4.1.4) Autocorrelation properties.....36
4.1.5) Cross correlation properties.....40
4.1.5) Signal-to-noise ratio for Narayana sequences 42

4.2) Narayana variants 43
4.2.1) Introduction 43
4.2.2) Variants of Narayana sequence 43
4.2.3) Periods of variants of Narayana series modulo p 44
4.2.2) Autocorrelation properties.....47
4.2.2) Cross correlation properties.....49

V. NARAYANA UNIVERSAL CODE 52

5.1) Introduction 52
5.2) Narayana series.....53
5.3) Narayana universal code.....55
5.4) Implementation of Narayana series for Encryption and Hashing.....60

VII.CONCLUSION 61

REFERENCES 62

LIST OF TABLES

Table Page

1. Binary mapping of prime periods for GH sequences.....19
2. Peak CCF values for binary GH and PN sequences for variable length.....27
3. SNR for varying sequence lengths of GH sequences 29
4. Primes from 3 to 151 for Narayana sequence.....34
5. Primes from 157 to 233 for Narayana sequence.....35
6. SNR for varying sequence lengths of Narayana sequences.....41
7. Primes from 3 to 233 for Narayana variants.....44
8. Generation of Narayana sequence 52
9. Mapping of Narayana series to J series.....55
10. List of codewords for first 15 natural numbers.....57

LIST OF FIGURES

Figure Page

1. Block diagram of encryption and decryption algorithms 1
2. Different Encryption Algorithms.....2
3. Direct Sequence Spread Spectrum.....5
4. Frequency Hoping Spread Spectrum 6
5. Lossless source coding.....7
6. Channel coding 7
7. Centralized and Decentralized system 11
8. Architecture of CDMA WSN gateway node 12
9. Key generation.....15
10. F(n) generator.....18
11. Autocorrelation of B(n) for sequence length 175 20
12. Autocorrelation of B(n) for sequence length 300 21
13. PN sequence fragment of length 63.....23
14. CCF of binary GH sequences for 100 bits.....24
15. CCF of binary GH sequences for 200 bits.....25
16. CCF of PN sequences for sequence length 100.....26
17. CCF of PN sequences for sequence length 200.....26
18. CCF of GH and PN sequences for lengths up to 200 28
19. Cross correlation of GH sequences for sequence length of 20 29
20. Signal-to-noise ratio for GH sequences modulo prime.....30
21. ACF of binary sequence B(n) for Narayana series of length 80 bits.....36
22. ACF of binary sequence B(n) for Narayana series of length 150 bits.....37
23. ACF of C(n) sequence for Narayana series of length 100 bits 37
24. ACF of C(n) sequence for Narayana series of length 140 bits 38
25. CCF of binary sequence B(n) for Narayana series of length 50 bits 40
26. CCF of binary sequence B(n) for Narayana series of length 80 bits 40
27. Signal-to-noise ratio for Narayana sequences modulo prime.....41
28. ACF of binary sequence B(n) for Narayana variants of length 50 bits 47
29. ACF of binary sequence B(n) for Narayana variants of length 80 bits 47
30. CCF of binary sequence B(n) for Narayana variants of length 50 bits.....49
31. CCF of binary sequence B(n) for Narayana variants of length 60 bits.....49
32. Ratio between first 100 consecutive terms of Narayana series.....54
33. Required number of bits for first 1000 natural numbers.....58

CHAPTER I

INTRODUCTION

1.1) Introduction to Cryptography

In the current information society, securing data is of paramount importance. Cryptography refers to secure transmission of data through networks so that security of data is not compromised. The plaintext (payload) is turned into ciphertext or a code word using different encryption algorithms. In order to retrieve the original plain text, ciphertext is decrypted with the same or different key used for encryption [1],[2]. This may be described by the block diagram shown in Figure 1.

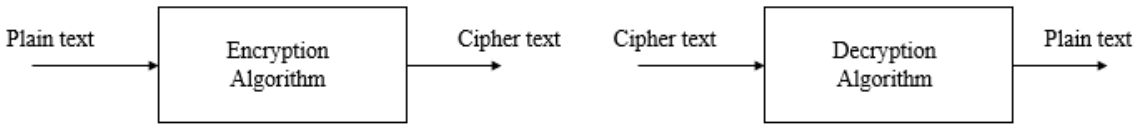

Figure 1. Block diagram of encryption and decryption algorithms

The keys are chosen in such a way that, intruders between source and destination are unable to decrypt the original text and the keys are required to be random in nature. The level of security obtained also depends on how keys are exchanged and how a communication is set up by an initial protocol.

In general, encryption algorithms include symmetric encryption, asymmetric encryption, keyless cipher, substitution cipher and transposition cipher [4][5]. Different encryption algorithms are shown in Figure 2. In symmetric encryption, encryption and decryption keys are same and can be represented by the equation as below

$$P = D(K, E(K, P)) \quad (1)$$

where P is the plaintext and K is the key used for encryption E and decryption D. However, the keys used for encryption and decryption in asymmetric encryption are different and can be represented by the equation

$$P = D(K_D, E(K_E, P)) \quad (2)$$

where K_D and K_E are decryption and encryption keys respectively [5].

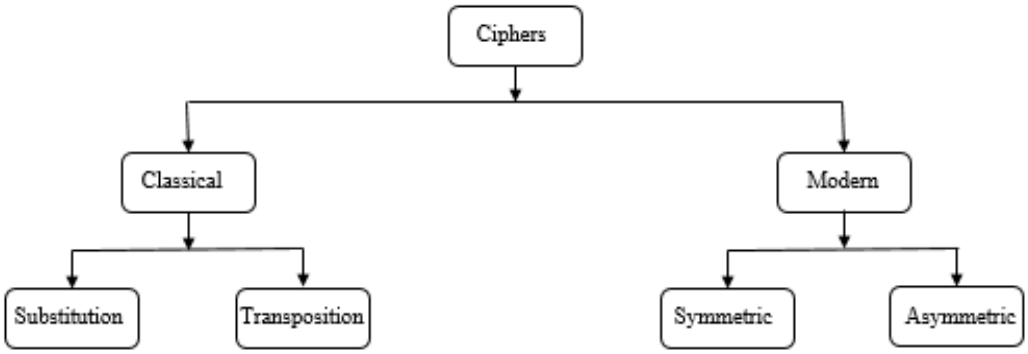

Figure 2. Different Encryption Algorithms [adapted from [3]]

Cryptography has wide range of applications in various fields, because of secure transmission of data. The fields of cryptographic applications include; quantum key distribution to establish secure communication, probability, image processing for secure processing and transmission of image data between different platforms, digital signature and authentication used in electronic mail [6][7]. Ensuring privacy of original data over a network is the primary aim of cryptography

and thus can be used in monitoring communication, transferring files, password authentication and for secure message transmission over low power computers [6].

1.2) Random sequences

Random sequences are sequences generated whose outcome is random in nature, do not have a standard pattern and thus are not easily predictable. The random numbers can be obtained through random number generator, shift registers or any other approaches [8]. Random number generators produce random numbers or pseudorandom numbers which are periodic with certain distribution properties and also exhibit hidden correlations [9]. Since a key is required to be more random in nature and not easily decryptable, the random numbers generated using different approaches are used as keys for encryption and decryption of information to avoid third party interference and have their main functions in cryptography, key generation and multiparty communication [8][10]. The random numbers, when used as a key, is shared only between trusted parties using different protocols such that cyber criminals intruding them are not able to decrypt the message [11][12].

The protocols used between two parties might involve probability events, in which the receiver has to predict a probability of sender sharing the required key that also involves verification of the procedure followed by the two parties [11]. The protocols for sharing random numbers used as keys for multiparty communication and verification involving oblivious protocol are also developed in which the process consists of three steps: Initial set-up, for sharing random number as a key between different parties, input mapping in the range [0, 1] with uniform probability followed by verification process [12].

The randomness measure of a sequence can be obtained using autocorrelation and cross-correlation properties. Sequences with higher autocorrelation and lower cross-correlation properties are said to be more random in nature and preferred as a key. Thus, random numbers play a vital role in the field of cryptography to maintain information secure and ensure secure

transmission of data. Random numbers are used in statistical cryptography, deterministic cryptography and quantum cryptography [13].

1.3) Spread spectrum

Spread spectrum technique is one of the significant technique in the applications related to communication and networking. The frequency of the signal is intentionally varied in order to obtain a resultant signal with random sequence, ensuring secure communication [14][15]. It is important to spread the spectrum prior to transmission of data sequence using a code or sequence which doesn't depend on the original code [14][15]. This can be achieved with the help of Pseudo-Noise (PN) sequences [15].

PN sequences are maximal length binary sequences which satisfy important properties of random numbers and recur in regular intervals and can be generated using linear feedback shift register (LFSR) [16]. The sequence can be spread through two basic techniques which include direct sequence spreading and frequency hopping method [14][16].

Multiplying the original bit sequence and PN sequence, results into a hopped signal which when modulated gives us a Direct Sequence Spread Spectrum (DSSS) [17].

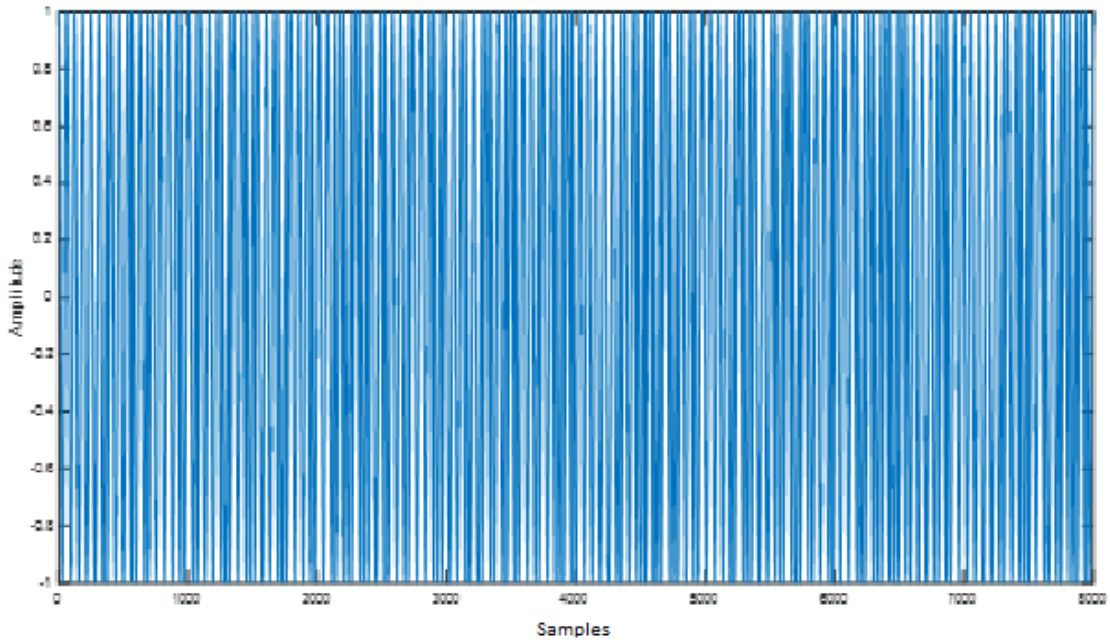

Figure 3. Direct Sequence Spread Spectrum

Figure 3 shows the spectrum spread by PN sequence for the original bit stream defined by 1 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0. By dispreading the above spectrum with the same PN sequence used at the transmitter, original bit stream is obtained at the receiver.

Frequency Hopping Spread Spectrum (FHSS) hops every bit of the signal to a new frequency which can be easily generated with Binary Phase Shift Keying (BPSK) signal [17]. Binary phase shift keying being an elementary modulation technique, generates sequences with a 180 degree phase shift [17]-[18].

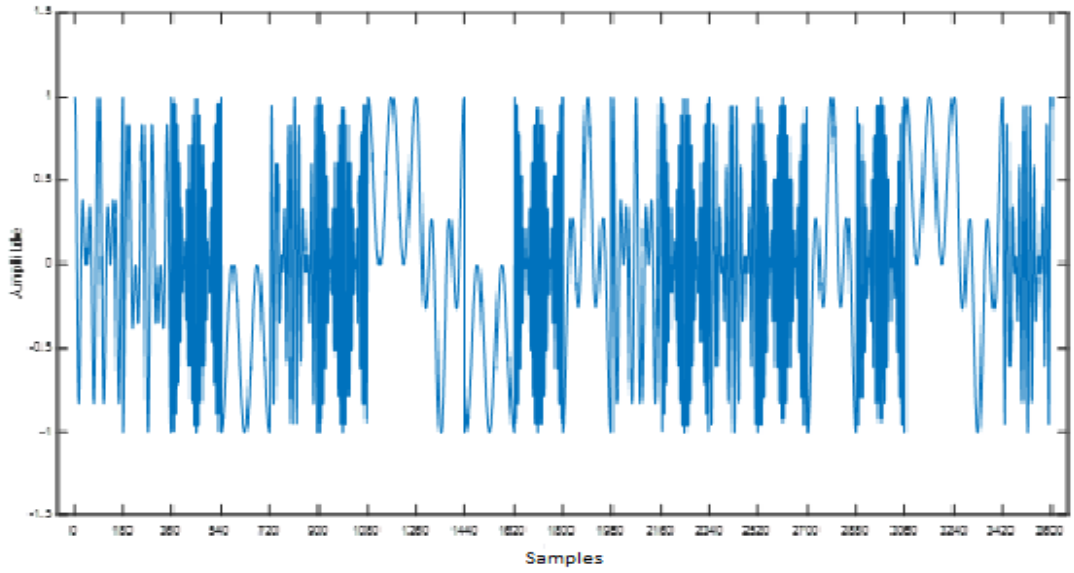

Figure 4. Frequency Hoping Spread Spectrum

Figure 4. shows the FHSS generated by spreading the BPSK spectrum with 12 different frequencies. The original sequence can be retrieved by demodulating FH spectrum with the same 12 frequencies and BPSK signal [17]-[18].

Spread spectrum is used in information security, key generation, data transfer and many other cryptographic applications since it avoids third party interference.

1.4) Universal codes

There are several techniques for compressing, collecting and transfer of data obtained from a source. It is important to gather the information about source and develop an algorithm to compress the original information for gathering and transmission of data. In order to develop an algorithm, it must consist of rules relevant to source parameters such that the algorithm is capable to compress any digital sequence [19]. This creates a necessity for the universal coding scheme to ensure secure and easy communication which is disclosed only to the source and destination parties.

The encoding and decoding codes while developing algorithms can be different and constructed in numerous ways. In order to minimize losses during transmission of information, there are fixed-rate lossless universal source coding and fixed-rate universal channel coding theorems [20]. By means of sparse matrices in construction of universal coding, the encoding error can be minimized and negligible [20].

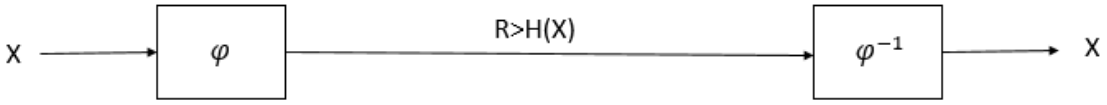

Figure 5. Lossless source coding [adapted from [20]]

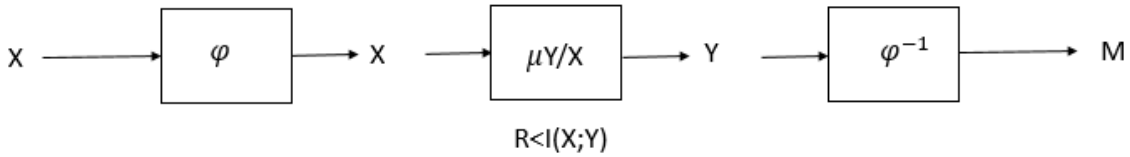

Figure 6. Channel coding [adapted from [20]]

Figure 5 and Figure 6 represent lossless source coding and channel coding respectively. Multiple universal coding schemes exist for ease of coding algorithm. They include Elias coding, Huffman coding and Levenshtein codes [21]. Elias gamma and delta coding was developed to reduce the number of bits for codeword representation. Elias gamma and delta codes are also implemented in prefix coding obtained through (2,3) representation of numbers in order to reduce the length of a codeword compared to original Elias coding and Fibonacci coding [21].

Huffman coding uses minimum redundancy coding technique to obtain codeword from the source code [22]. The average length of source code depends on probability of each message in source code and length of each message. To construct the codeword for the source code, we ensemble the code of each message which transmits the source code with minimum message length defining the optimum method of coding [22].

Levenshtein coding is a universal code for encoding and decoding of non-negative integers. There are a set of rules to be followed in order to perform encoding and decoding [23]. The codeword will be a set of binary numbers which depends on the source integer [23]. The codeword which uses Levenshtein method has its length as a bit longer compared to the one which employs Elias omega coding [23].

1.5) Thesis organization

This thesis aims at obtaining new classes of random sequences for cryptographic applications and we describe the proposed binary GH and Narayana sequences for use as random sequences for key generation and other applications.

Chapter 2 summarizes previous work in the field of random numbers which include CDMA communication, E-commerce, Fibonacci sequences, Fibonacci universal code and golden ratio along with a brief introduction to secure multiparty communication.

Chapter 3 details randomness properties of sequences derived from Fibonacci and Gopala-Hemachandra sequences modulo m for use in key distribution applications which includes autocorrelation and cross-correlation properties of sequences and presents comparison with cross-correlation properties of pseudo noise sequences.

Chapter 4 investigates randomness and cryptographic properties of the Narayana series modulo p, where p is a prime number. It is shown that the period of the Narayana series modulo p is either p^2+p+1 (or a divisor) or p^2-1 (or a divisor).

Chapter 5 presents a method of universal coding based on the Narayana series. The rules necessary to make such coding possible have been found and the length of the resulting code has been determined to follow the Narayana count.

Chapter 6 gives a description of the implementation of the various random sequences to engineering applications by investigating signal-to-noise ratio and other parameters along with applications of the sequences. Chapter 7 concludes the thesis with multiple classes of random sequences used in cryptographic applications and suggestions for further implementation and modification. All data processing was performed off-line using a commercial software package (MATLAB 6.1, The MathWorks Inc., Natick, MA, 2000).

CHAPTER II

PREVIOUS WORK

2.1) Secure multiparty communication

Privacy of data between multiple parties in the process of certain communication is a vital concept in cryptography. Suppose an access code needed to enter a room or unlock a vehicle can be activated only with collaboration of two or limited number of users, it is necessary to maintain privacy of the code and this can be achieved through secret sharing [24]. There are many protocols on multiparty communication whose applications extend to the fields of physics, intrusion detection, data protection and telecommunications.

Certain difficulties in multiparty communication may include cooperative scientific computations, database query, intrusion detection, data mining, geometric computation and statistical analysis [25]. There are certain approaches to solve problems in secure multiparty communication which includes oblivious transfer, secret sharing and threshold cryptography [25].

Oblivious transfer is performed by generation of probability of an event, based on sharing a single random number to solve multiparty communication problems [12]. The probability event might be generated by multiple parties through decentralized oblivious transfer protocol. In centralized system, the trusted authority evaluates the data received from different parties and parties are unaware of the randomization nature of transformation used for evaluating data [12].

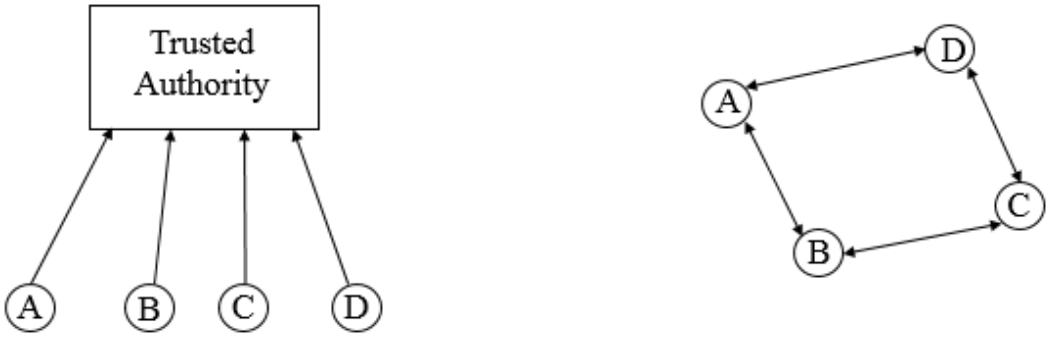

Figure 7. Centralized system with trusted authority (Left); decentralized system (Right)

The random numbers in the form of data is exchanged between multiple authorities after authentication of a protocol by users in a decentralized system [12]. Figure 7 shows centralized and decentralized systems for multiparty communication between three parties. In decentralized oblivious transfer protocol, multiple parties reach at a probability event in order to communicate without revealing the random numbers to each other [11]. This avoids third party interference and dishonesty by the users and thus, has been possible solution for problems in multiparty communication.

2.2) CDMA communication

Wireless Sensor Networks (WSN) is one of the new generation networks emerging to transmit data over long distances with the help of internet and other networks [26]. The gateway nodes used in WSN acts as an intermediate between sensor nodes (sender) and sink nodes (receiver) [26]. The sender and receiver have different types of network. The three gateways used in WSN include Ethernet WSN gateway, WLAN WSN gateway and GPRS/CDMA WSN gateway [26].

Code Division Multiple Access (CDMA) WSN gateway follows CDMA protocol which is schedule-based MAC (Multiple Access) protocol. The schedule-based MAC protocol divides the channel or gateway into smaller pieces and allocates pieces to the node for exclusive use [27]. In CDMA protocol, unique code is assigned to each user which refers to code set partitioning [27]. Although all users share same frequency, each user has own code sequence called, chipping

sequence to encode data. Thus, encoded signal is the product of original data and chipping sequence [27]. The data can be decoded through inner product of encoded signal and chipping sequence [27]. This is mostly used in cellular networks since it allows multiple users to co-exist and transmit simultaneously with minimal interference if codes are orthogonal [27].

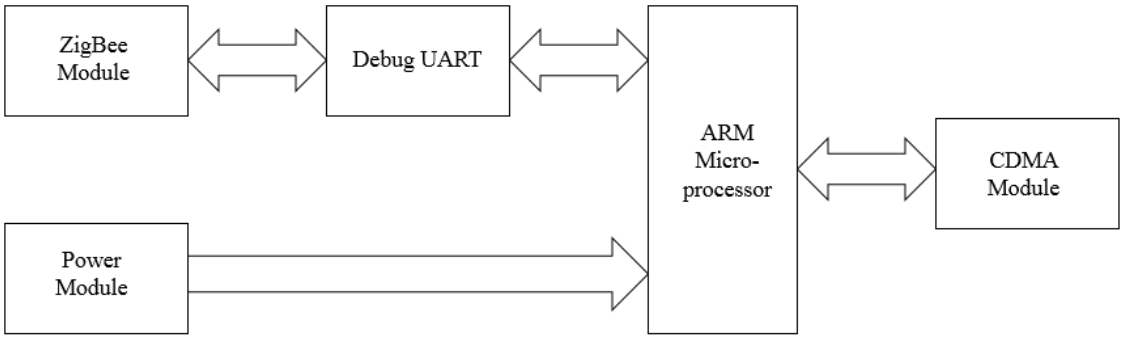

Figure 8. Architecture of CDMA WSN gateway node [adapted from [26]]

The entire architecture of WSN gateway node is shown in the figure 8. The gateway node receives source code information through wireless ZigBee module. The gateway node also accomplishes data transmission between ZigBee and ARM microprocessor [26]. In order to control the delivery of CDMA module, CDMA gateway runs the commands written in ARM processor and source code information sensed by WSN are transmitted to CDMA wireless network [26].

The CDMA communication reduces the collisions during transmission of data and the energy-efficiency of this technology is good for communication. This establishes strong applications of CDMA communication in cryptography, multiparty communication and wireless communication.

2.3) Electronic commerce

Electronic commerce is a business in which importing and exporting of products are performed with the help of internet [28]. It is easy to procure goods online on behalf of onsite purchase and

hence E-commerce is invasive in current world [28]. There are different types of E-commerce which comprises B2B E-commerce, B2C E-commerce, C2C E-commerce and others including G2G, G2E, G2B and B2G [28].

In order to use E-commerce technology for business, it is important to secure information on both, transmitter and receiver end [29]. Cryptography comes to play to ensure appropriate transmission of data between multiple parties [29]. It is necessary to certify the security with the help of keys for encryption and decryption of data over internet since the number of users are indefinite [29]. Use of secret key encryption, public key encryption or digital signature can solve the confidentiality issues for transmission [29].

Symmetric encryption is also called as secret key encryption which uses same keys during encryption and decryption of message. In public key encryption, each user is provided with two keys; public key and private key in which message is transmitted from one user to another using public key and private key is used for decrypting the message on user ends [29]. It is important for the users to maintain the secrecy of private key from others in this method of encryption. The digital signature uses message digest, a smaller piece of information, at both ends source and destination [29]. The message digest is obtained in a way that suffers radical changes with very little changes in the original message or information and this is contained only between source and receiver thus maintaining the privacy of data [29].

Applications of cryptography in E-commerce ensures secure email, secure web access, secure payments and reliable downloading of code and data [29].

2.4) Fibonacci sequence

Fibonacci, a mathematician of Europe, proposed a solution to the problem defined below:

"A pair of rabbits gives birth to a new pair. When the new pair is two months old, each pair produces another pair at the beginning of each month. What is the number of pairs of rabbits produced from the original pair at the end of 12 months? [30]"

The solution to this problem is the resulting Fibonacci sequence given by 1,1,2,3,5,8,13,.. and so on. The Fibonacci sequence is generated using the below equation:

$$F_{n+1} = F_n + F_{n-1} \tag{3}$$

Where F(0)=F(1)=1. The ratio between next term and previous term of Fibonacci sequence is termed as golden ratio which is given by a recursive number 1.61803… The golden ratio φ is defined as summation of reciprocal of φ with 1 which is $\varphi = \frac{1}{\varphi} + 1$ and is widely used in many applications [30]. The Fibonacci numbers can be evidenced in various forms in nature, some examples include, leaves of plants, pinecones, sunflowers, pineapples, flowers, bees and the palm of human hands [30].

Based on Fibonacci sequence, a unique Fibonacci coding which is a universal coding technique was developed [35]. The Fibonacci universal coding is a binary coding in which each positive integer has one unique representation based on certain set of defined rules [35]. The performance of this coding technique using Fibonacci sequence at encryption and decryption is better than other universal codes such as Elias coding [35].

The properties of Fibonacci numbers are innumerable [31]. Fibonacci numbers are used in various fields including physics, chemistry, mathematics and it is mainly applied to the field of telecommunications and cryptography for data hiding, key generation and information sharing [32].

CHAPTER III

G-H SEQUENCES

3.1) Autocorrelation properties

3.1.1) Introduction

A good key sequence for cryptographic applications must have excellent randomness properties but also be easy to generate (Figure 9). Lacking this, keys can become the weak point of an otherwise strong cryptographic system. Here we propose the use of Fibonacci and the related Gopala-Hemachandra (GH) sequences [33]-[35] for this purpose. These sequences have applications in coding and cryptography that are well known. We consider the randomness properties of the residues of the Fibonacci sequence F(n) = 0, 1, 1, 2, 3, 5, 8, …. and the related Gopala- Hemachandra sequence $GH_{a,b}(n)$ = a, b, a+b, a+2b, 2a+3b, … modulo m (The sequence $GH_{a,b}$ will also be called (a,b)-GH). We do so by considering m to be either a prime or as a composite number. Note that $GH_{a,b}(n) = GH_{a,b-1}(n) + F(n)$.

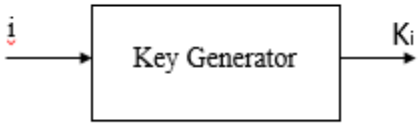

Figure 9: Key Generation

The general period properties of such sequences are well known [36]-[38] but how these numbers indexed by m may be mapped into random sequences has not been investigated. We need good mappings to map these into a binary sequence that has excellent autocorrelation properties. For different perspectives on randomness, that includes physical and algorithmic aspects, see [39]-[43].

Fibonacci and GH sequences are iterative. Another related iterative mapping is the algorithm to generate 1/p for prime p, the binary expansion a(n) of which is given by the formula [44]-[47] $a(n)=2^n \bmod p \bmod 2$. Other related sequences of interest to the computer scientist and to the student of dynamical system theory, include those obtained from general iterative maps [48]-[50].

In this section of the chapter, we summarize general properties of the periods of Fibonacci sequences mod m. Then, we present the mapping to transform these sequences into binary sequences that have excellent randomness properties.

3.1.2) Periods of Fibonacci and GH sequences mod m

As mentioned before GH sequences can be written in terms of the Fibonacci sequence. Thus:

$$GH_{2,1}(n) = F(n)+2F(n-1), \text{ for } n > 1 \quad (4)$$

As example, the sequence $GH_{2,1}(n)$ can be written as:

2, 1, 3, 4, 7, 11, … = 0, 1, 1, 2, 3, 5, 8, …

\+ 0, 0, 2, 2, 4, 6, 10, …

This indicates that the period of the GH sequence modulo m will be identical to that of the corresponding Fibonacci sequence.

Consider an F sequence to mod *m* series where *m* is a prime number. In this case, we can see that there can at most be p^2-1 pairs of consecutive residues in a period. For example, for m=p=3, the sequence will be 0, 1, 1, 2, 0, 2, 2, 1. This consists of the pairs 01, 02, 10, 11, 12, 20, 21, 22 which is all the possible pairs excepting 00, since that cannot be in such a sequence for it will lead to the next number being 0 that is impossible in a periodic residue sequence. Since p^2-1 is (p-1)(p+1), the period of the residue sequence will either be a divisor of p-1 or p+1 (or equivalently of 2p+2).

The periods of Fibonacci sequence mod m, with m as a prime, has been shown to be either p-1 if p≡1 or 9 (mod 10) or 2p+2 if p≡3 or 7(mod 10). The periods of GH sequences likewise follow the same behavior.

The periods of generalized (a,b)-GH sequence mod p are [38]:

(i) (p-1) or a divisor thereof if the prime number p ends with 1 or 9.

(ii) (2p+2) or a divisor thereof if the prime number *p* ends with 3 or 7.

(iii) 20 for *p* =5.

The pertinent result for the period, N, of GH sequence for non-prime modulo *m* is:

N(m) ≤ 6m with equality iff $m = 2*5^n$, for n=1, 2, 3, …

This includes the value when the modulus is 5.

Thus, the periods can be grouped into 2 types: p-1 or 2p+2.

3.1.3) Generating the sequence

One can generate an arbitrary element of the F sequence by means of the following formula:

$$F(n) = \frac{1}{\sqrt{5}}(u^n - v^n) mod\ m \quad (5)$$

Where

$$u = \frac{1+\sqrt{5}}{2} \tag{6}$$

And

$$v = \frac{1-\sqrt{5}}{2} \tag{7}$$

This result is easily proven by noting that $u^2 = u + 1$ and $v^2 = v + 1$. We get the sequence:

$F(0) = 0\,;\ F(1) = (u - v)/\sqrt{5} = 1\,;\ F(2) = (u^2 - v^2)/\sqrt{5} = 1\,;\ F(3) = (u^3 - v^3)/\sqrt{5} = 2\,;$ $F(4) = (u^4 - v^4)/\sqrt{5} = 3$; and so on.

If n is a whole number, F(n) is generated and modulus of F(n) is calculated. Thus, F(n) mod m can be generated easily. On the other hand, obtaining n from F(n) mod m is difficult for large m, even if sufficient number of consecutive digits are available.

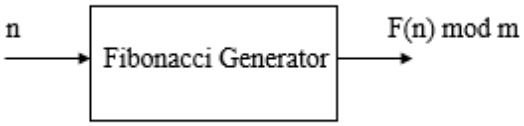

Figure 10. F(n) generator

3.1.4) Mapping into random binary sequence

As mentioned before, we propose to divide the sequence of periods for modulo p based on the property whether the period is a divisor of p-1 or 2p+2. We will include p=5 in the class p-1 since the period 20= 5x(5-1). We assign periods with multiples of (p-1) or divisor as binary value +1 and periods with multiples of (2p+2) or divisor as binary value -1.

Table 1 provides the first 25 prime numbers for easy reference.

Table 1: Binary mapping of prime periods for GH sequences

Prime numbers	Periods	In terms of p	Binary value
3	8	2p+2	-1
5	20	5(p-1)	1
7	16	2p+2	-1
11	10	p-1	1
13	28	2p+2	-1
17	36	2p+2	-1
19	18	p-1	1
23	48	2p+2	-1
29	14	(p-1)/2	1
31	30	p-1	1
37	76	2p+2	-1
41	40	p-1	1
43	88	2p+2	-1
47	32	(2p+2)/3	-1
53	108	2p+2	-1
59	58	p-1	1
61	60	p-1	1
67	136	2p+2	-1
71	70	p-1	1
73	148	2p+2	-1
79	78	p-1	1
83	168	2p+2	-1
89	44	(p-1)/2	1
97	196	2p+2	-1
101	50	(p-1)/2	1

We call the resulting binary sequence B(n). The first 20 bits of B(n) are -1,1,-1,1,-1,-1,1,-1,1,1,-1,1,-1,-1,-1,1,1,-1,1 and -1.

3.1.5) Autocorrelation properties of GH sequences

We first consider the prime moduli and determine autocorrelation properties of B(n) to determine how good they are from the point of view of randomness [47].

The autocorrelation function is calculated using the formula:

$$C(k) = \frac{1}{n}\sum_{j=0}^{n-1} B_j B_{j+k} \quad (8)$$

Where B_j and B_{j+k} are the binary values of the sequence generated by the above process and n is the length of the sequence.

Figure 11 and 12 present the normalized autocorrelation function of the B(n) sequence for 175 and 300 points.

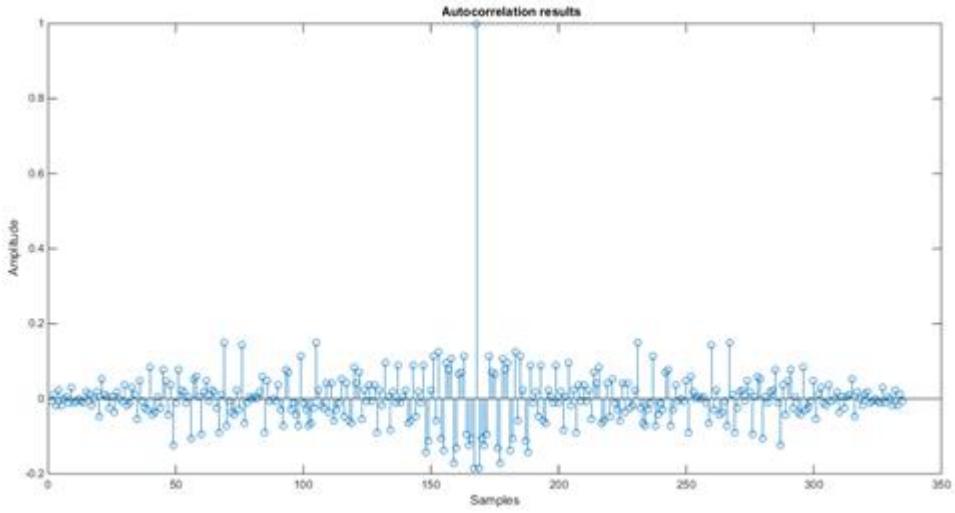

Figure 11. Autocorrelation of B(n) for sequence length 175

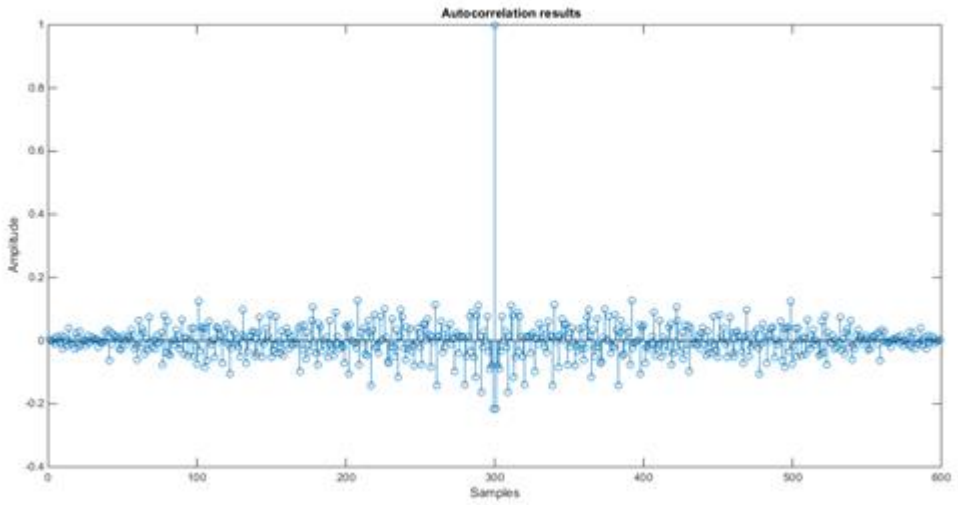

Figure 12. Autocorrelation of B(n) for sequence length 300

Looking at Figure 11 and Figure 12, their randomness is apparent from the effective two-valued character of the function.

Randomness may be calculated using the randomness measure, R(x), of a discrete sequence x by the expression below [40]:

$$R(x) = 1 - \frac{\sum_{k=1}^{n-1} |C(k)|}{n-1} \tag{9}$$

According to this measure, a constant sequence will have the measure of 0 whereas a fully random sequence will have the measure of 1. The randomness measure values for Figure 11 and Figure 12 using the above formula is found to be 0.9516 and 0.9631.

General moduli m. The challenge is to find the property that helps map the period information into two classes that lead to a random binary sequence like B(n). For length of 300, we found the property whether the period is a multiple of 8 to effectively put the period values into two classes. But the randomness measure of such a sequence was much inferior compared to that of Figures 11 and 12 with a value of 0.8988.

3.2) Cross-correlation properties

3.2.1) Introduction

Spread spectrum systems provide secure communications by spreading a signal over large frequency band. They are implemented using short periodic pseudo-random sequences with good correlation properties [51]-[54]. The sequences necessary for direct-sequence spreading and spread spectrum analysis must have low cross correlation characteristics. Peak cross-correlation is used when measuring the difference between two sequences of different time series [55].

As mentioned in the previous section of the chapter, GH sequences that are related to Fibonacci sequences [38] were shown to have good pseudo-randomness properties. As is well-known randomness can be examined both from a physics perspective [42][43][57][58] as well as an algorithmic one [44][45][59][60]. The family of GH sequences presents a new take on an old mathematical structure and is therefore of much interest. These sequences can be used also in sending side information that can help in authentication in cryptography systems which is one of the central problems in a networked society [64]-[72]. In particular, good random sequences can be of help in frustrating man-in-the-middle attacks in P2P systems [62] [63].

In this section of the chapter, we present the cross-correlation of binary GH sequences and compare these to that of PN sequences. For this comparison, we use peak cross-correlation function as a measure and show that GH sequences score over PN sequences. These sequences can have applications in cryptography in authentication of parties, especially in P2P systems.

3.2.2) Periods of shift register and Fibonacci sequences

A PN sequence is a periodic binary sequence generated using linear feedback shift register (LFSR) structure [56]. Figure 13 shows a fragment of a PN sequence generated using LFSR

structure for the polynomial $p(z)=z^{45}+z^4+z^3+z+1$. Here we show 63 bits of a sequence whose period is $3.52x10^{13}$.

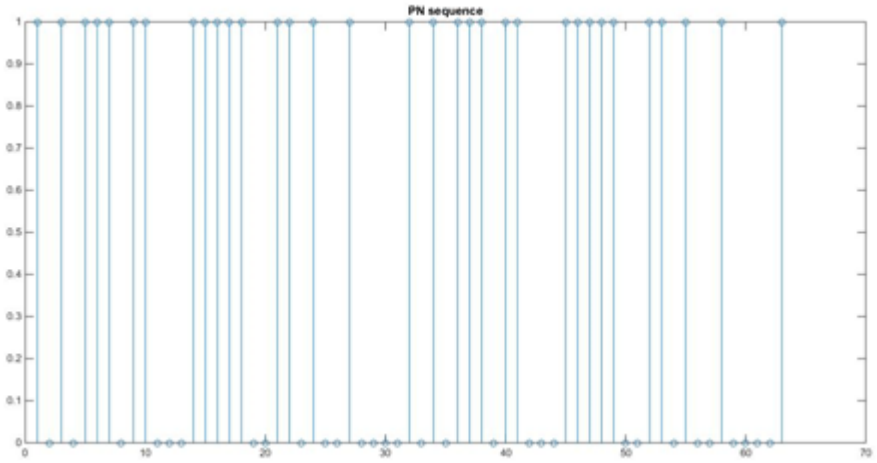

Figure 13. PN sequence fragment of length 63

The period of a PN sequence produced by a linear feedback shift register with m stages cannot exceed 2^m-1. When the period is exactly 2^m-1, the PN sequence is maximal length sequence or m-sequence [55] [56].

The periods of the GH sequences modulo m will be identical to that of the corresponding Fibonacci sequence [37]. Consider a GH sequence to mod p where p is a prime number. It is quite clear that there can at most be p^2-1 pairs of consecutive residues in a period. Since p^2-1 is (p-1)(p+1), the period of the residue sequence will either be a divisor of p-1 or p+1 (or equivalently of 2p+2). It was necessary to go beyond the results of autocorrelation properties and determine the cross-correlation properties of fragments of this sequence obtained from different regions.

3.2.3) Cross-correlation properties of GH and PN sequences

The cross-correlation between two sequences is complex inner product of the first sequence with the shifted version of the second sequence which indicates if two sequences are distinct. The correlation properties of the sequences are used to detect and synchronize the communication process.

We assign periods of GH sequence mod p, with multiples of (p-1) or divisors, as binary value +1 and periods with multiples of (2p+2) or divisor, as binary value -1 and thus call the resulting binary sequence as B(n). The first 20 bits of B(n) are - 1,1,-1,1,-1,-1,1,-1,1,1,-1,1,-1,-1,-1,1,1,-1,1 and -1.

Let us consider prime moduli and determine periodic cross-correlation properties of B(n) to determine how good they are from the point of view of randomness. The cross correlation function is calculated using the formula:

$$CCF(k) = \frac{1}{N}\sum_{j=0}^{N-1} A_j B_{j+k} \quad (10)$$

Where A_j and B_{j+k} are the binary values of two sequences at different time intervals and N is the length of sequence or period of sequence. The peak cross correlation function value of a cross correlated sequence will be denoted by CCF_{peak}.

Figures 14 and 15 present the normalized cross-correlation function of the B(n) sequence for 100 and 200 bits.

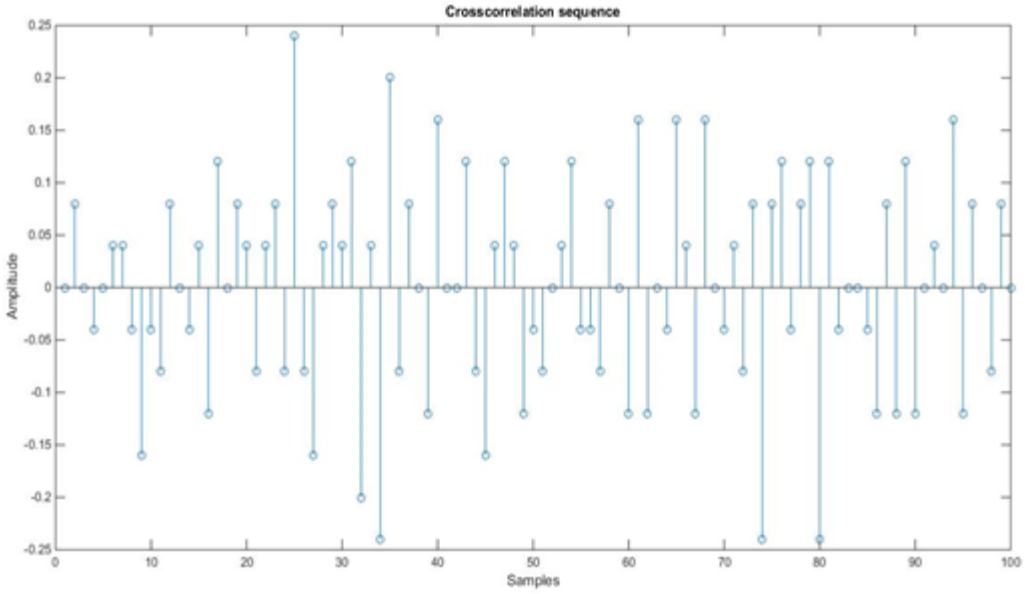

Figure 14. CCF of binary GH sequences for 100 bits

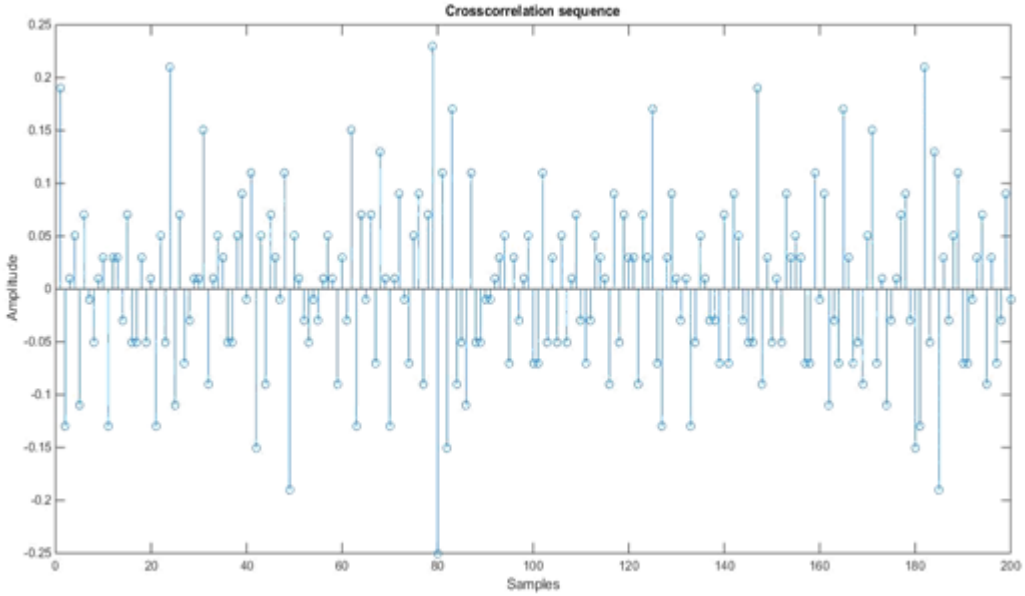

Figure 15. CCF of binary GH sequences for 200 bits

Looking at Figure 14 and Figure 15, their peak cross-correlation values are noted to be 0.25 in both cases.

Figures 16 and 17 present the normalized cross correlation function characteristics of the PN sequences for 100 and 200 points which peak cross-correlation values of 0.9 and 0.94 respectively. The PN sequence fragments in these figures are from the expansion of the polynomial $p(z)=z^{45}+z^4+z^3+z+1$.

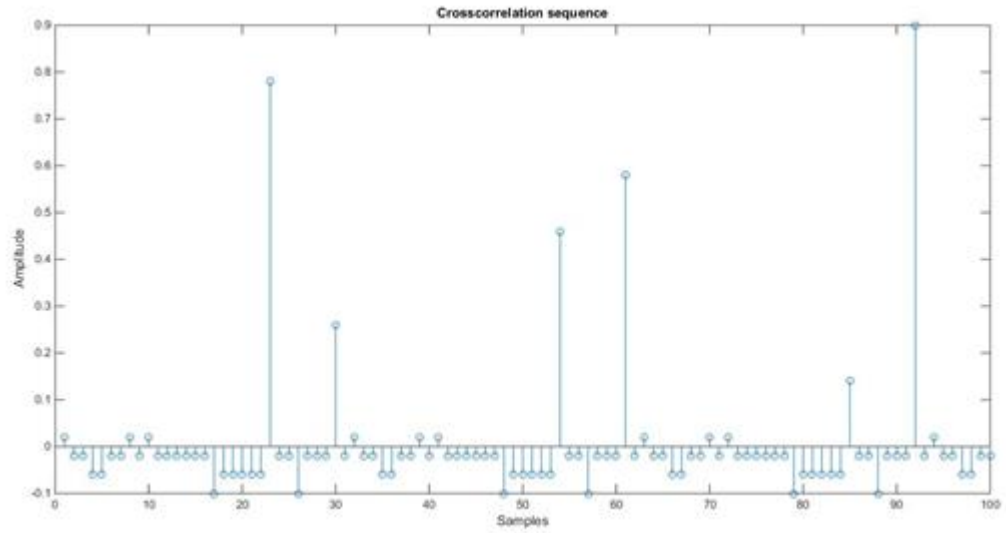

Figure 16. CCF of PN sequences for sequence length 100

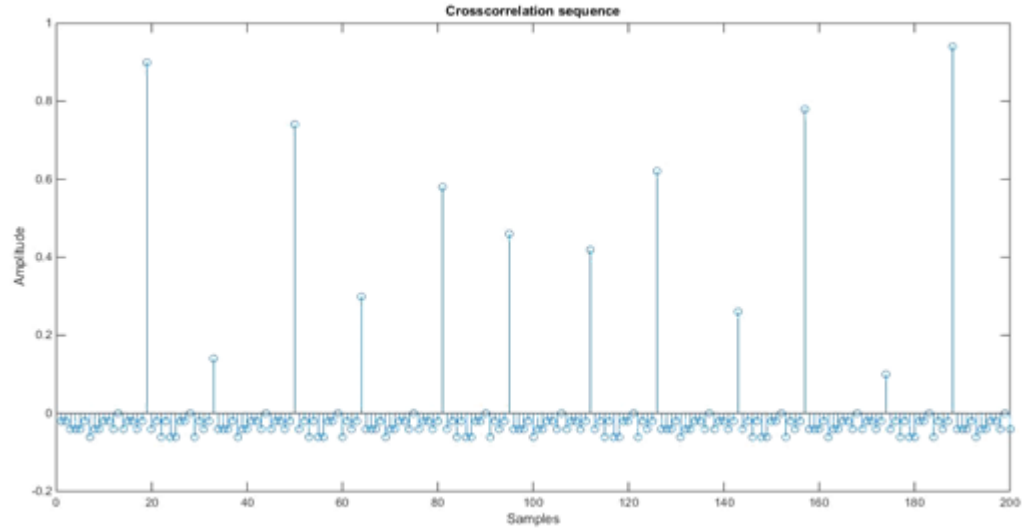

Figure 17. CCF of PN sequences of sequence length 200

Randomness may be calculated using the randomness measure, R (x), of a discrete sequence x by the expression below:

$$R(x) = 1 - \frac{\sum_{k=1}^{N-1} |CCF(k)|}{N-1} \tag{11}$$

where CCF(k) is the cross-correlation function value for k and N is the period of sequence to characterize the randomness of a sequence. According to above formula, the randomness measure of 1 indicates that the sequence is fully random whereas randomness measure of 0 indicates a constant sequence. The randomness measure for Figure 16 and Figure 17 are found to be 0.9212 and 0.9342 respectively.

3.2.4) Comparison between binary GH sequences and PN sequences

The following table gives a comparison between peak cross-correlation function (CCF) value of Binary GH Sequences and PN Sequences. Here we consider two types of PN sequences: (i) where the fragments are taken from the same long PN sequence; (ii) where the fragments come from different PN sequences.

Table 2. Peak CCF values for binary GH and PN sequences for variable length

Length of bits	Peak CCF value for binary GH sequences	Peak CCF value for PN sequences of same generator polynomials	Peak CCF value for PN sequences of different generator polynomials
25	0.52	0.6	0.36
50	0.32	0.72	0.32
100	0.25	0.9	0.36
150	0.24	0.9	0.32
200	0.24	0.94	0.25

As expected, the peak cross-correlation function of the PN sequences obtained from different generators is lower than that obtained from the same generator. The graph that represents the above table is shown below in Figure 18.

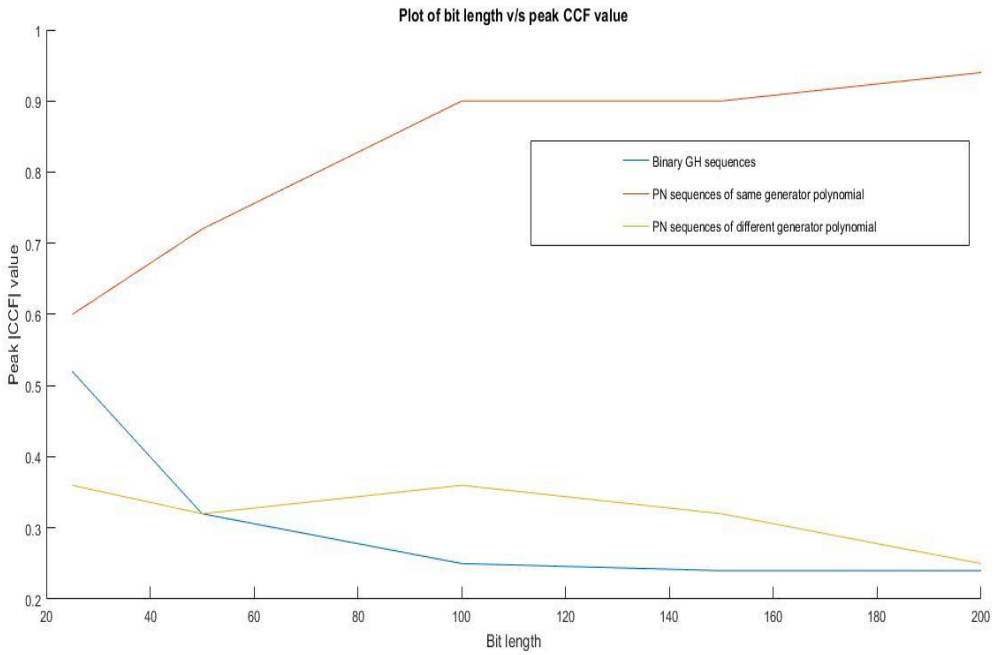

Figure 18. CCF of GH and PN sequences for lengths up to 200

Hence, comparing the peak cross-correlation function (CCF) value of GH series mod p with that of pseudo-noise sequences, it is found that GH series mod p has excellent cross-correlation properties that are comparable to PN sequences obtained from different generators.

3.2.5) Signal-to-noise ratio for binary GH sequences

Signal-to-noise ratio parameter helps in analysis and validation of sequences for cryptographic applications. Signal-to-noise ratio (SNR) with high reliability can be used in speech

enhancement, speech detection and speech recognition [74]. It is generally defined as ratio of signal power to noise power as given below.

$$SNR = \frac{Signal\ Power}{Noise\ Power} \tag{12}$$

The equation for signal to noise ratio in dB is given below:

$$SNR_{dB} = Signal\ Power\ in\ dB - Noise\ Power\ in\ dB \tag{13}$$

SNR for a sequence can be obtained using cross-correlation property of sequence at different time intervals. There are several metrics to measure signal to noise ratio of cross correlated sequence [76]. Here, SNR is calculated for cross-correlation function of GH sequences using peak-to-root mean square ratio.

The peak to root mean square ratio (PRMSR) is obtained by ratio of signal, which is square of fundamental peak cross correlation function value CCF_{max} to noise, which is square of root mean square value of cross correlation function of a cross correlated sequence CCF_{rms} [76].

$$PRMSR = \frac{|CCF_{max}|^2}{{CCF_{rms}}^2} \tag{14}$$

CCF_{rms} is calculated for cross correlation function whose values are less than or equal to half of fundamental peak cross correlation function value which is given by [76]:

$${CCF_{rms}}^2 = \sqrt{1/N_m \sum_{i \subset m} |CCF(i)|^2}^2 \tag{15}$$

Where m is the number of points of CCF value with lower than half of peak primary value [76]. Consider cross correlation function of GH sequences for length of first 20 points as one sequence and next 20 points in consecutive time interval as second sequence. The resulting graph is shown in figure 19.

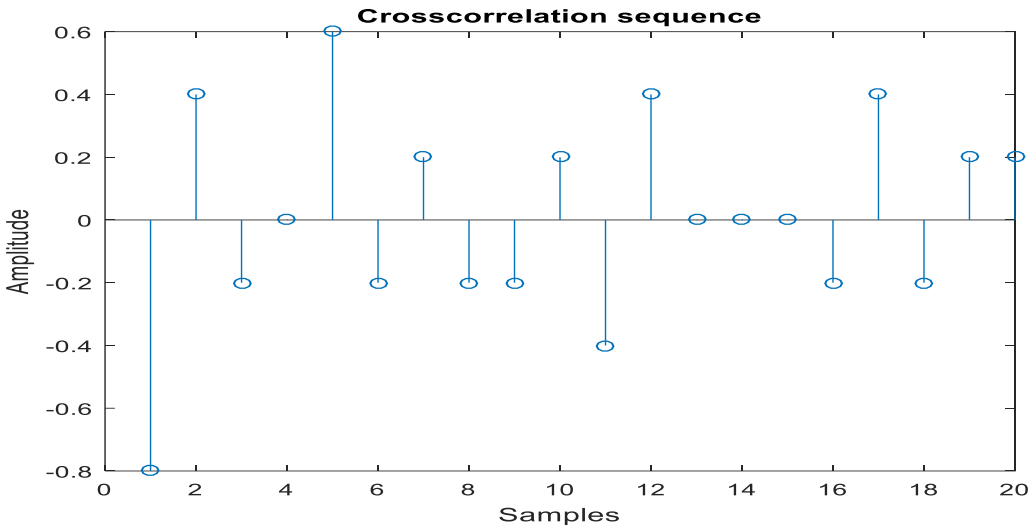

Figure 19. Cross correlation of GH sequences for sequence length of 20

According to PRMSR metric, the signal to noise ratio is calculated as shown below.

$$SNR = \frac{(0.6)^2+(0.8)^2}{\frac{1}{18}*((0.2)^2*10+(0.4)^2*4)} \quad (16)$$

In equation (16), it is seen that fundamental peak value square of 0.8 is considered as signal and values lower than half of fundamental peak value are considered as noise. The values above half of primary peak value are also considered as signal to obtain signal to noise ratio value.

We perform similar calculations for cross correlation function of GH sequences for a sequence length of 10, 20, 40, 50, 100 and 120 with consecutive time intervals in order to obtain a set of signal to noise ratio values. Table 3 shows SNR for varying sequence lengths of GH sequences.

Sequence Lengths	SNR Values
10	7.7815
20	12.3824
40	20.86
50	23.004
100	22.95
120	20.7387

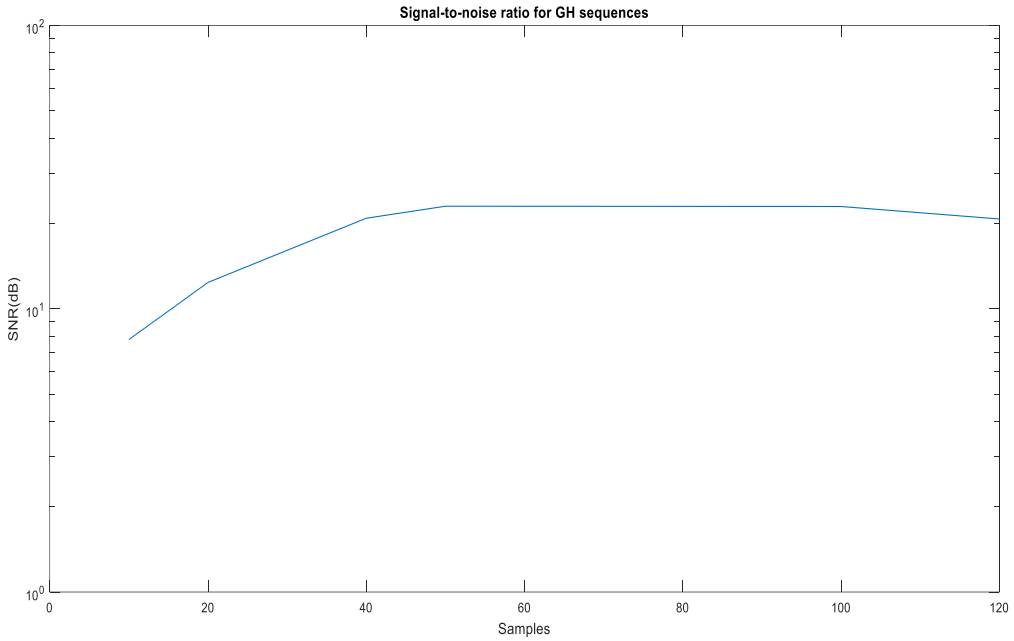

Figure 20. Signal-to-noise ratio for GH sequences modulo prime

Figure 20 shows SNR values for GH sequences modulo prime in which x-axis is mapped to 10, 20, 40, 50, 100 and 120 sequence lengths and y-axis is mapped to different SNR values in dB. This validates the randomness of sequence and its applications in cryptography.

CHAPTER IV

NARAYANA SEQUENCES AND VARIANTS

4.1) Narayana sequences

4.1.1) Introduction

There has been considerable recent interest in the Narayana sequences (e.g. [75]-[79]). Since they are closely related to Fibonacci sequences [34]-[36] [80], one would expect many applications in data coding and cryptography, especially multiparty computation [2] [11] [12] [44] [71] [72]. The properties of Fibonacci sequences modulo a prime have also been investigated [37]-[38]. Fibonacci sequences are also important in entropy problems of physics [43] [66] [82]-[85]. It is worthwhile then to determine if the use of the Narayana sequences can replace that of Fibonacci sequences in certain settings. Narayana, who lived in the 14th century, proposed the following problem [75]: "A cow gives birth to a calf every year. When the calf is three years old, each calf gives birth to a calf at the beginning of each year. What is the number of progeny produced during twenty years by one cow". The sequence resulting from this problem is 1,1,1,2,3,4,6,9,13,19, ... and so on. Each number in the sequence is calculated by the summation of previous number and number three places before that in the sequence:

$$u_{n+1} = u_n + u_{n-2} \tag{17}$$

In this section of the chapter, the period of Narayana series modulo p, where p is a prime number, is investigated and found to be either p^2+p+1 (or a divisor) or p^2-1 (or a divisor). Some other characteristics of the Narayana sequence are presented. The investigation of the autocorrelation and cross-correlation properties of the sequence reveals that they are good candidates for cryptographic and key generation applications.

4.1.2) Generation of Narayana series

Narayana, an outstanding Indian mathematician of the 14th century, who was interested in summation of arithmetic series and magic squares, proved a more general summation in the middle of 14th century [77].

$$S_n^{(m)} = \frac{n(n+1)(n+2)\ldots(n+m)}{1\cdot 2\cdot 3\cdot \ldots\ldots \cdot (m+1)} \tag{18}$$

Narayana applied the above equation to the problem of a herd of cows and calves which is famous as Narayana's problem. Thus, using the above equation, he obtained [77]:

$$n = 1 + 20 + \frac{17\cdot 18}{1\cdot 2} + \frac{14\cdot 15\cdot 16}{1\cdot 2\cdot 3} + \frac{2\cdot 3\cdot 4\cdot 5\cdot 6\cdot 7\cdot 8}{1\cdot 2\cdot 3\cdot 4\cdot 5\cdot 6\cdot 7} = 2745$$

Narayana's problem can also be solved by the similar method that Fibonacci solved his rabbit problem. In the beginning of the first year, there were two heads since one cow produced one calf. In the beginning of the second and third year, the number of heads increased by one and therefore, the number of heads is 3 and 4 respectively. From the fourth year, the number of heads is defined as follows:

$$x_4 = x_3 + x_1, x_5 = x_4 + x_2\ , \ldots, x_n = x_{n-1} + x_{n-3}, \tag{19}$$

since the number of cows for any year is a summation of number of cows of previous year and number of calves which was born (= number of heads that were three years ago).

We have the sequence

$$2,3,4,6,9,\ldots,u_{n+1} = u_n + u_{n-2} \quad (20)$$

Thus, we obtain u_{20}=2745 by computation. Now, we can consider the sequence

$$1,1,1,2,3,4,6,9,\ldots,u_{n+1} = u_n + u_{n-2} \quad (21)$$

with $n \geq 2$, $u_0 = 0$, $u_1 = 1$, $u_2 = 1$. These numbers are called Fibonacci Narayana numbers [77].

4.1.3) Periods of Narayana series modulo m

Consider Narayana sequence modulo 3, the sequence is obtained as follows: 1,1,1,2,0,1,0,0,1,1,1,2,… and so on. Here, if three consecutive zeroes appear during sequence generation, the next number will be a zero. Since each of the three preceding digits can take values from 0 to p-1, the maximum period can only be p^3-1. Given a maximum length Narayana sequence, the digit-wise multiplication by 1 through p-1 would leave the sequence unchanged. Therefore, we obtain the result that the maximum period is $(p^3-1)/(p-1) = p^2+p+1$. When we consider only the preceding two digits, a similar argument would establish that the period can be p^2-1. Thus, we have our central result:

Theorem 1:

Given Narayana sequence modulo p, where p is a prime number, the periods of the sequence will either be p^2+p+1 (or the divisor) or p^2-1 (or the divisor).

The table below provides the list of periods for first 50 prime numbers. Periods with multiples of p^2+p+1 (or the divisor) are assigned binary value -1 and periods with multiples of p^2-1 (or the divisor) are assigned binary value +1 and the resulting sequence is binary sequence B(n). Also, primes with even periods are assigned binary value 1 and primes with odd periods are assigned binary value 0 and the resulting sequence is binary sequence C(n).

Table 4. Primes from 3 to 151 for Narayana sequence

Prime Numbers	Period		In terms of p	B(n)
	Number	C(n)		
3	8	1	p^2-1	1
5	31	0	p^2+p+1	-1
7	57	0	p^2+p+1	-1
11	60	1	$(p^2-1)/2$	1
13	168	1	p^2-1	1
17	288	1	p^2-1	1
19	381	0	p^2+p+1	-1
23	528	1	p^2-1	1
29	840	1	p^2-1	1
31	930	1	p^2-1	1
37	342	1	$(p^2-1)/4$	1
41	1723	0	p^2+p+1	-1
43	1848	1	p^2-1	1
47	46	1	$(p^2-1)/48$	1
53	468	1	$(p^2-1)/6$	1
59	3541	0	p^2+p+1	-1
61	1240	1	$(p^2-1)/3$	1
67	33	0	$(p^2-1)/136$	1
71	5113	0	p^2+p+1	-1
73	2664	1	$(p^2-1)/2$	1
79	6240	1	p^2-1	1
83	3444	1	$(p^2-1)/2$	1
89	7920	1	p^2-1	1
97	3169	0	$(p^2+p+1)/3$	-1
101	10303	0	p^2+p+1	-1
103	10713	0	p^2+p+1	-1
107	11557	0	p^2+p+1	-1
109	11991	0	p^2+p+1	-1
113	991	0	$(p^2+p+1)/13$	-1
127	2016	1	$(p^2-1)/8$	1
131	130	1	$(p^2-1)/132$	1
137	6256	1	$(p^2-1)/3$	1
139	1610	1	$(p^2-1)/12$	1
149	148	1	$(p^2-1)/148$	1
151	22800	1	p^2-1	1

Table 5. Primes from 157 to 233 for Narayana sequence

Prime numbers	Period		In terms of p	B(n)
	Number	C(n)		
157	24807	0	p^2+p+1	-1
163	26733	0	p^2+p+1	-1
167	4648	1	$(p^2-1)/6$	1
173	172	1	$(p^2-1)/174$	1
179	10680	1	$(p^2-1)/3$	1
181	32760	1	p^2-1	1
191	36673	0	p^2+p+1	-1
193	37443	0	p^2+p+1	-1
197	2156	1	$(p^2-1)/18$	1
199	3960	1	$(p^2-1)/10$	1
211	481	0	$(p^2+p+1)/93$	-1
223	12432	1	$(p^2-1)/4$	1
227	226	1	$(p^2-1)/228$	1
229	26220	1	$(p^2-1)/2$	1
233	54523	0	p^2+p+1	-1

As argued before, since the factors of p^3-1 are p-1 and p^2+p+1, the period of the Narayana sequence modulo p will either be p^2+p+1 (or the divisor) or (p-1)(p+1) (or the divisor).

4.1.4) Autocorrelation properties

Autocorrelation is a measure of similarity between a sequence and time shifted replica of the sequence. Ideally, the autocorrelation function (ACF) should be impulsive i.e. peak value at zero time shift and zero values at all other time-shifts (i.e. side-lobes).

The first 20 bits of resulting binary sequence B(n) obtained from periods of the Narayana series modulo prime based on p^2+p+1 (or the divisor) or p^2-1 (or the divisor) are 1,-1,-1,1,1,1,-1,1,1,1,1,-1,1,1,1,-1,1,1,1,-1 and 1. Similarly, the first 20 bits of resulting sequence C(n) obtained from periods of the Narayana series modulo prime based on evens and odds are 1,0,0,1,1,1,0,1,1,1,1,0,1,1,1,0,1,0,0 and 1.

We first consider prime moduli and determine periodic autocorrelation properties of B(n) and C(n) to determine how good they are from the point of view of randomness. For convenience, the zeroes in C(n) sequence is changed to -1 so as to make off-peak autocorrelation as small as possible.

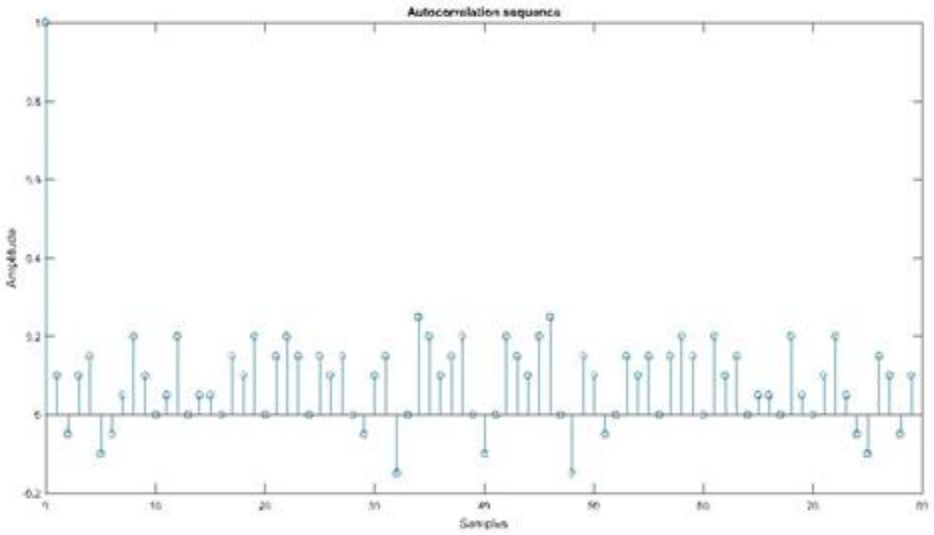

Figure 21. ACF of binary sequence B(n) for Narayana series of length 80 bits

The autocorrelation function is calculated using expression (8) represented below:

$$ACF(k) = \frac{1}{N}\sum_{j=0}^{N-1} B_j B_{j+k}$$

Where B_j and B_{j+k} are the binary values of sequence and time shifted version of the sequence and N is the length of sequence or period of sequence.

Figures 21 and 22 present the normalized autocorrelation function of B(n) sequence for 80 and 150 bits respectively.

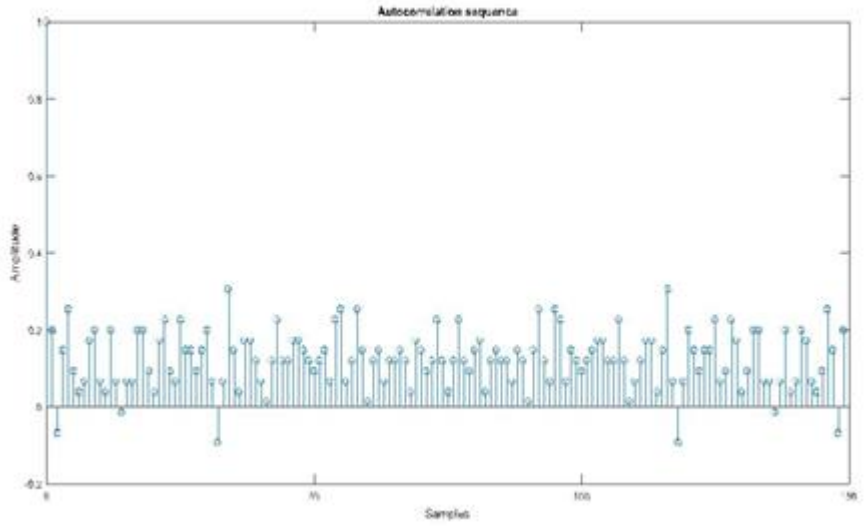

Figure 22. ACF of binary sequence B(n) for Narayana series of length 150 bits

Looking at Figure 21 and Figure 22, it is evident that their lobes are close to ideal autocorrelation function (ACF) with peak value at zero time shift and values close to zero at all other time-shifts (i.e. side-lobes). Figures 23 and 24 present the normalized autocorrelation function of C(n) sequence for 100 and 140 bits respectively.

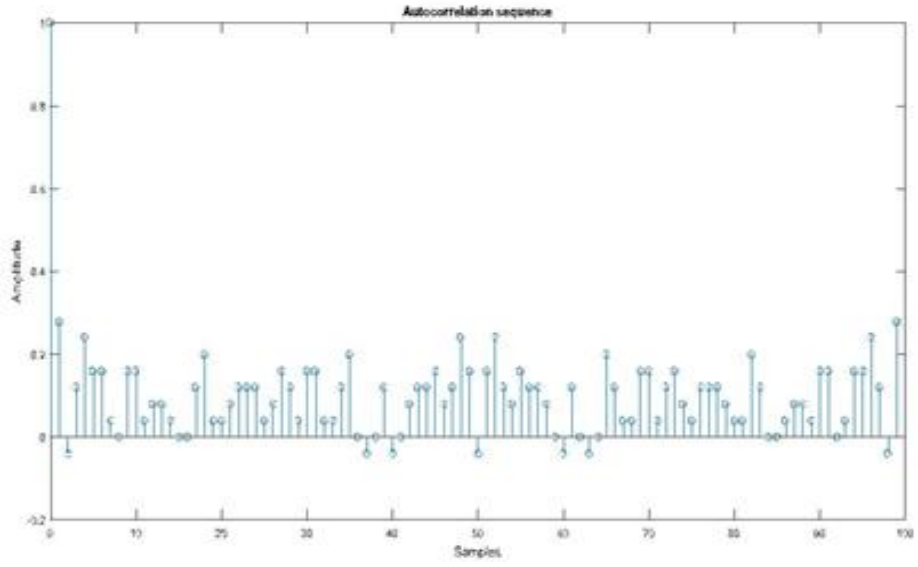

Figure 23. ACF of C(n) sequence for Narayana series of length 100 bits

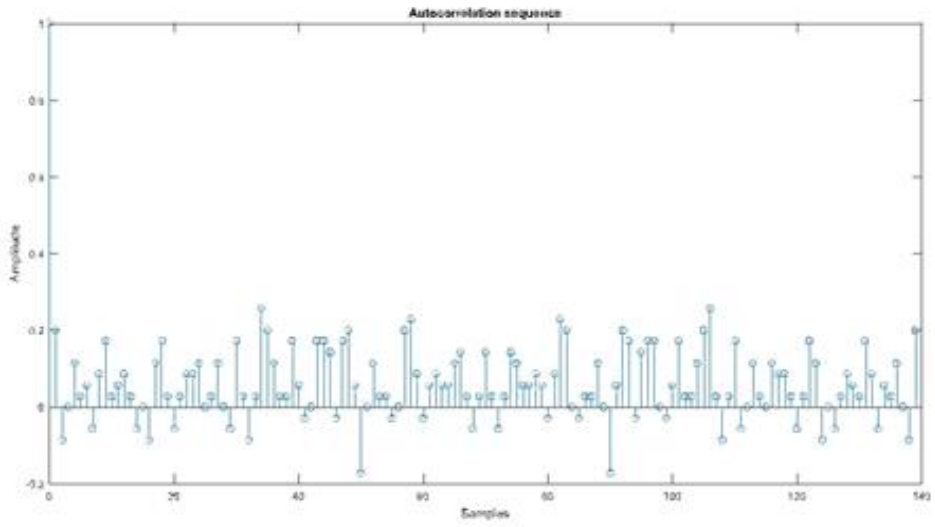

Figure 24. ACF of C(n) sequence for Narayana series of length 140 bits

Looking at Figure 23 and Figure 24, it is evident that their lobes are close to ideal autocorrelation function (ACF) with peak value at zero time shift and values close to zero at all other time-shifts (i.e. side-lobes).

Randomness may be calculated using the randomness measure, R(x), of a discrete sequence x using expression (9) represented below:

$$R(x) = 1 - \frac{\sum_{k=1}^{n-1} |ACF(k)|}{N-1}$$

where ACF(k) is the autocorrelation function value for k and N is the period of sequence to characterize the randomness of a sequence.

According to above formula, the randomness measure of 1 indicates that the sequence is fully random whereas randomness measure of 0 indicates a constant sequence. The randomness measure for Figure 21, Figure 22, Figure 23 and Figure 24 are found to be 0.8867, 0.8662, 0.8937 and 0.9110 respectively.

4.1.5) Cross-correlation properties

Cross correlation is the measure of similarity between two different sequences. The cross-correlation between two sequences is the complex inner product of the first sequence with a shifted version of the second sequence which indicates if the two sequences are distinct. Ideally, it is desirable to have sequences with zero cross-correlation value at all time shifts [37]. The correlation properties of the sequences are used to detect and synchronize the communication.

Now, we consider prime moduli and determine periodic cross-correlation properties of B(n) to determine how good they are from the point of view of randomness. The cross correlation function is calculated using the expression (10):

$$CCF(k)=\frac{1}{N}\sum_{j=0}^{N-1}A_j B_{j+k}$$

Where A_j and B_{j+k} are the binary values of two sequences at different time intervals and N is the length of sequence or period of sequence. The peak cross-correlation function value of a cross-correlated sequence is calculated using the formula:

$$CCF_{\text{peak}}=\frac{1}{N}\sum_{k=1}^{N}|CCF(k)| \qquad (22)$$

Where CCF(k) is the cross-correlation function value for k and N is the period of sequence. Figures 25 and 26 present the normalized cross-correlation function of the B(n) sequence for 50 and 80 bits.

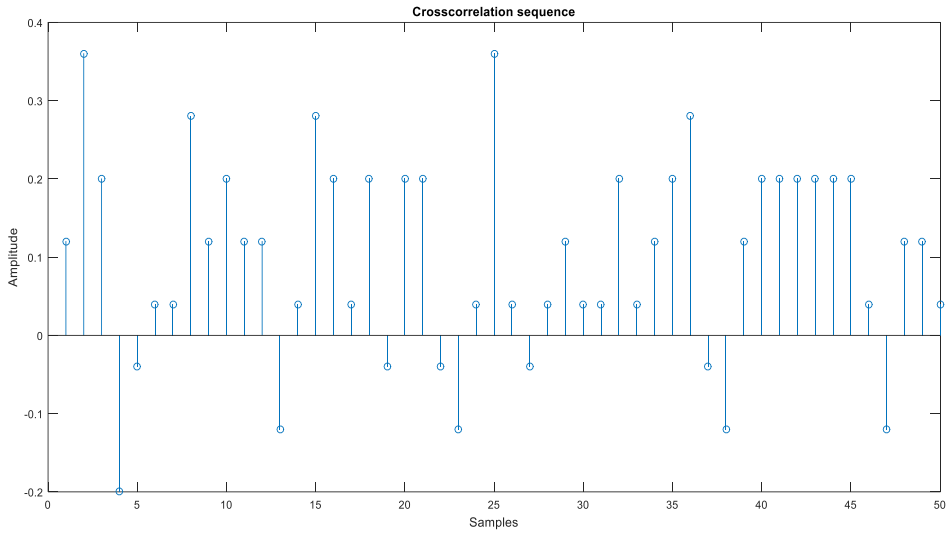

Figure 25. CCF of binary sequence B(n) for Narayana series of length 50 bits

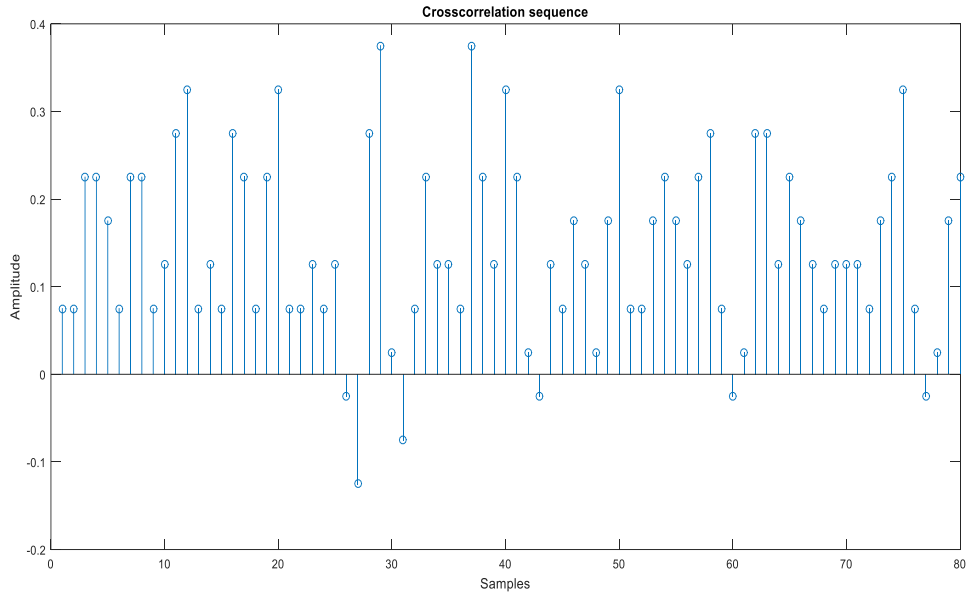

Figure 26. CCF of binary sequence B(n) for Narayana series of length 80 bits

Looking at Figure 25 and Figure 26, their peak cross-correlation values are noted to be 0.375 in both cases. This value compares favorably with the peak cross-correlation value of other pseudorandom sequences.

4.1.6) Signal-to-noise ratio for Narayana sequences

Signal-to-noise ratio (SNR) for the Narayana sequence modulo prime is calculated using equations (12)-(15). SNR is obtained using cross correlation function of the Narayana sequence for sequence length 20, 50, 70, 100, 110, 120 and 140 with consecutive time intervals. The table containing SNR values at different sequence length is shown below.

Sequence Lengths	SNR Values
20	8.01632346
50	14.79300905
70	19.5853058
100	20.17849357
110	22.6634349
120	24.1198343296
140	21.7038757037

Table 6. SNR for varying sequence lengths of Narayana sequences

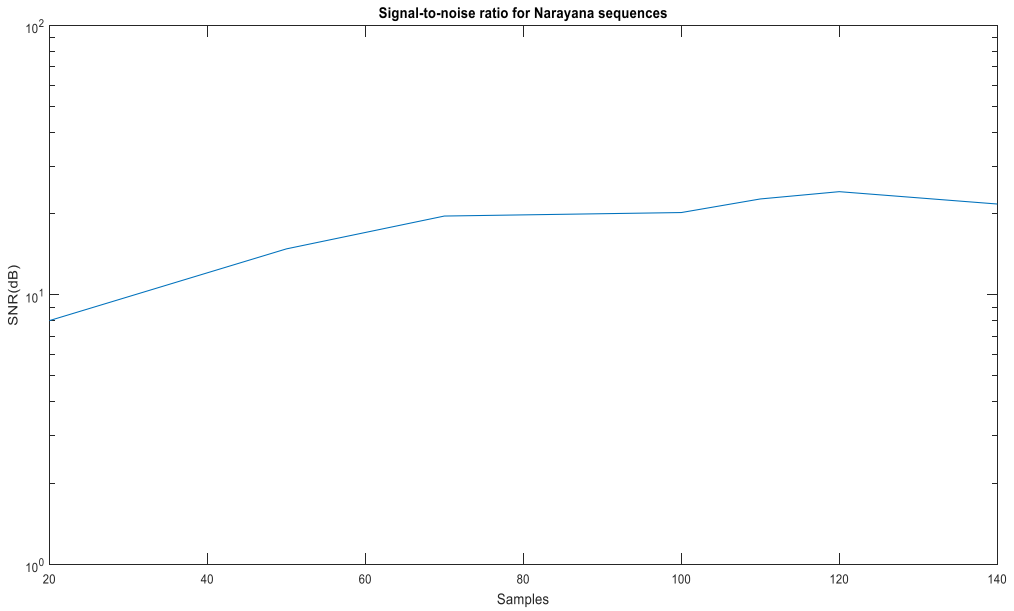

Figure 27. Signal-to-noise ratio for Narayana sequences modulo prime

Figure 27 shows a set of signal-to-noise ratio values for Narayana sequences modulo prime on y-axis plotted with respect to sequence lengths 20, 50, 70, 100, 110, 120 and 140 on y-axis. Thus, Narayana sequences are suitable for cryptographic applications.

4.2) Narayana Variants

4.2.1) Introduction

Fibonacci sequence has wide applications in the field of mathematics, science, computer science, botany, elementary number theory and many other fields [30] [86]-[88]. The main application of Fibonacci numbers is in the field of cryptography for secure communication and transmission of data [89]-[92].

The Narayana sequence derived from the Narayana problem has many properties in close relation with properties of Fibonacci sequence [76] [77] [93]. Hence, research in the field of cryptography with the help of Narayana Series is considerably large.

The periods of the Fibonacci sequence modulo p and the Narayana sequence modulo p, where p is a prime number are already investigated in the previous sections of this thesis. In this section of chapter, periods of variants of the Narayana series is investigated which are found to be either $(p^2-1)(p^2+1)$ (or a divisor) or $(p^2+p+1)(p-1)$ (or a divisor).

4.2.2) Variants of Narayana sequence

According to Narayana problem described in previous section, each term is the sum of penultimate term and term 3 places before that in the Narayana sequence [94]-[95]. Hence, Narayana sequence is given by 1,1,1,2,3,4,6,9,13,19,28 and so on.

For n terms in the sequence, Narayana series as a result of Narayana problem can be defined as:

$$N_a(k+1) = N_a(k) + N_a(k-2) \tag{23}$$

with $N_a(0) = N_a(1) = N_a(2) = 1$ and $k \geq 2$. A more general Narayana sequence $N_a(n)$ is given by {*a, b, c, a+c, a+b+c, a+b+2c, 2a+b+3c, 3a+2b+4c, and so on*} with *a=1, b=2 and c=3.*

We consider a special case of Narayana problem in which cows had to wait until fifth year to give birth to calves. Beginning in its fifth year, we investigate the number of cows and calves produced after 20 years, when each calf gives birth to a calf at the beginning of each year [75]. Now, the number of cows in any year is the summation of number of cows of previous year and number of calves which were born. The number of calves which were born is number of heads that were 4 years ago.

The sequence which results from the above problem is termed as variants of Narayana sequence and is given by 1,1,1,1,2,3,4,5,7,10,14,19,26 and so on.

For n terms in the sequence, variants of Narayana sequence can be defined by the below formula:

$$vN_a(k+1) = vN_a(k) + vN_a(k-3) \tag{24}$$

With $vN_a(0) = vN_a(1) = vN_a(2) = vN_a(3) = 1$ and $k \geq 3$. A more general variants of Narayana sequence is given by {a, b, c, d, a+d, a+b+d, a+b+c+d, a+b+c+2d, 2a+b+c+3d, and so on} with a=1, b=2 and c=3.

4.2.3) Periods of variants of Narayana series modulo p

Consider variants of the Narayana series modulo 3, where 3 is a prime number. The sequence obtained is as follows: 1,1,1,1,2,0,1,2,1,1,2,1,2,0,2,… and so on. Here, the next number will be zero if four consecutive zeroes occur during sequence generation. The maximum period can only be (p^4-1) (which is equal to $(p^2-1)(p^2+1)$) since each of the preceding terms in the sequence can range between 0 and p-1. Since the sequence remains unchanged for digit-wise multiplication by

1 through p-1 for a given maximum length variant Narayana sequence, the period is obtained as $(p^4-1)/(p-1) = (p^2+1)(p+1)$. When we consider only the preceding three digits, a similar argument establishes that the period can be (p^3-1) (which is equal to $(p^2+p+1)(p-1)$). Thus, we have our central result.

Theorem 2:

Given Variant Narayana sequence modulo p, where p is a prime number, the periods can be either $(p^2-1)(p^2+1)$ (or a divisor) or $(p^2+p+1)(p-1)$ (or a divisor).

The table below provides the list of periods for first 50 prime numbers. Periods with multiples of $(p^2-1)(p^2+1)$ (or the divisor) are assigned binary value of +1 and periods with multiples of $(p^2+p+1)(p-1)$ (or the divisor) are assigned binary value of -1. The resulting sequence is binary sequence B(n).

Table 7. Primes from 3 to 233 for Narayana variants

Primes	Periods	In terms of p	B(n)
3	80	$(p^2-1)(p^2+1)$	1
5	312	$(p^2-1)(p^2+1)/2$	1
7	342	$(p^2+p+1)(p-1)$	-1
11	1330	$(p^2+p+1)(p-1)$	-1
13	2196	$(p^2+p+1)(p-1)$	-1
17	96	$(p^2-1)(p^2+1)/870$	1
19	14480	$2(p^2-1)(p^2+1)/(p-1)$	1
23	12166	$(p^2+p+1)(p-1)$	-1
29	12194	$(p^2+p+1)(p-1)/2$	-1
31	61568	$2(p^2-1)(p^2+1)/(p-1)$	1
37	1368	$(p^2-1)(p^2+1)/1370$	1
41	68920	$(p^2+p+1)(p-1)$	-1
43	162800	$2(p^2-1)(p^2+1)/(p-1)$	1
47	212160	$2(p^2-1)(p^2+1)/(p-1)$	1

53	1404	$(p^2-1)(p^2+1)/5620$	1
59	205378	$(p^2+p+1)(p-1)/3364$	-1
61	75660	$(p^2+p+1)(p-1)/3$	-1
67	4488	$(p^2-1)(p^2+1)/4490$	1
71	1008	$(p^2-1)(p^2+1)/25210$	1
73	1332	$(p^2-1)(p^2+1)/21320$	1
79	208	$(p^2-1)(p^2+1)/187260$	1
83	82	$(p^2-1)(p^2+1)/578760$	1
89	704968	$(p^2+p+1)(p-1)$	-1
97	304224	$(p^2+p+1)(p-1)/3$	-1
101	5100	$(p^2-1)(p^2+1)/20404$	1
103	1092726	$(p^2+p+1)(p-1)$	-1
107	954	$(p^2-1)(p^2+1)/137400$	1
109	2614040	$(p^2-1)(p^2+1)/54$	1
113	3192	$(p^2-1)(p^2+1)/51080$	1
127	16128	$(p^2-1)(p^2+1)/16130$	1
131	5720	$(p^2-1)(p^2+1)/51486$	1
137	642838	$(p^2+p+1)(p-1)/4$	-1
139	5410160	$2(p^2-1)(p^2+1)/(p-1)$	1
149	6660600	$2(p^2-1)(p^2+1)/(p-1)$	1
151	491850	$(p^2+p+1)(p-1)/7$	-1
157	3869892	$(p^2+p+1)(p-1)$	-1
163	1443582	$(p^2+p+1)(p-1)/3$	-1
167	1874208	$(p^2-1)(p^2+1)/415$	1
173	10415640	$2(p^2-1)(p^2+1)/(p-1)$	1
179	5735338	$(p^2+p+1)(p-1)$	-1
181	32760	$(p^2-1)(p^2+1)/32762$	1
191	4560	$(p^2-1)(p^2+1)/291856$	1
193	18624	$(p^2-1)(p^2+1)/74500$	1
197	12936	$(p^2+p+1)(p-1)$	-1
199	7880598	$(p^2+p+1)(p-1)$	-1
211	9393930	$(p^2+p+1)(p-1)$	-1
223	22279040	$2(p^2-1)(p^2+1)/(p-1)$	1
227	51528	(p2-1)(p2+1)/51530	1

229	13110	(p2-1)(p2+1)/209768	1
233	3162334	(p2+p+1)(p-1)/4	-1

From the above table, it is seen that periods of Variant Narayana sequence can be either $(p^2-1)(p^2+1)$ (or a divisor) or $(p^2+p+1)(p-1)$ (or a divisor).

4.2.4) Autocorrelation properties

Autocorrelation measures the similarity of a sequence with time-shifted replica of itself [96]. The autocorrelation function is calculated using expression (8) represented below:

$$ACF(k) = \frac{1}{N}\sum_{i=0}^{N-1} B_i B_{i+k}$$

Where B_i and B_{i+k} are the binary values of the sequence and time shifted version of the sequence respectively and N is the length of the sequence. Under ideal conditions, autocorrelation function has its peak value at zero time shift and zero values at all other time-shifts.

The first 20 bits of resulting binary sequence B(n) are 1,1,-1,-1,-1,1,1,-1,-1,1,1,-1,1,1,1,-1,-1,1,1 and 1.

Figure 28 and Figure 29 represents the normalized autocorrelation function of B(n) sequence for 50 and 80 bits respectively.

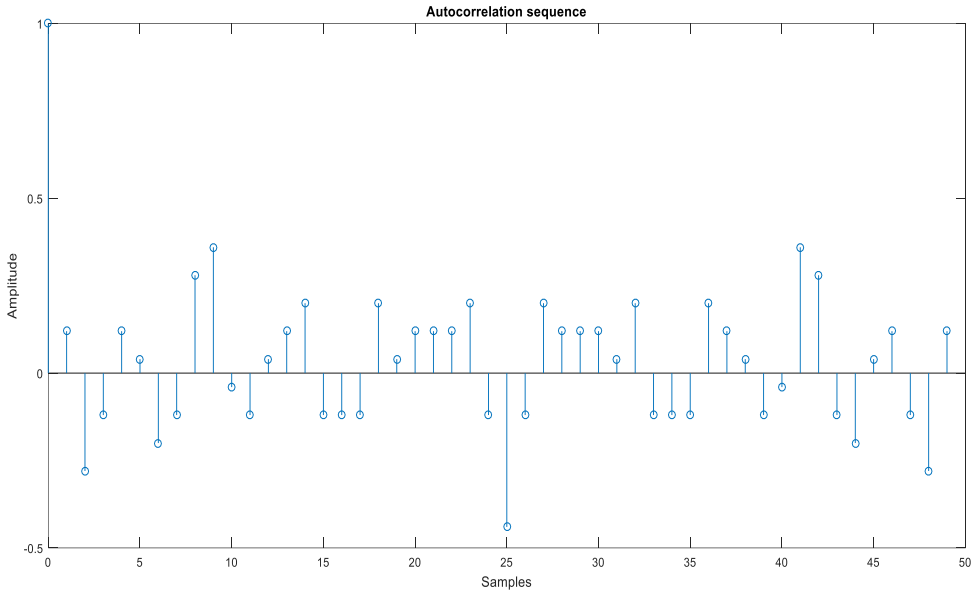

Figure 28. ACF of binary sequence B(n) for Narayana variants of length 50 bits

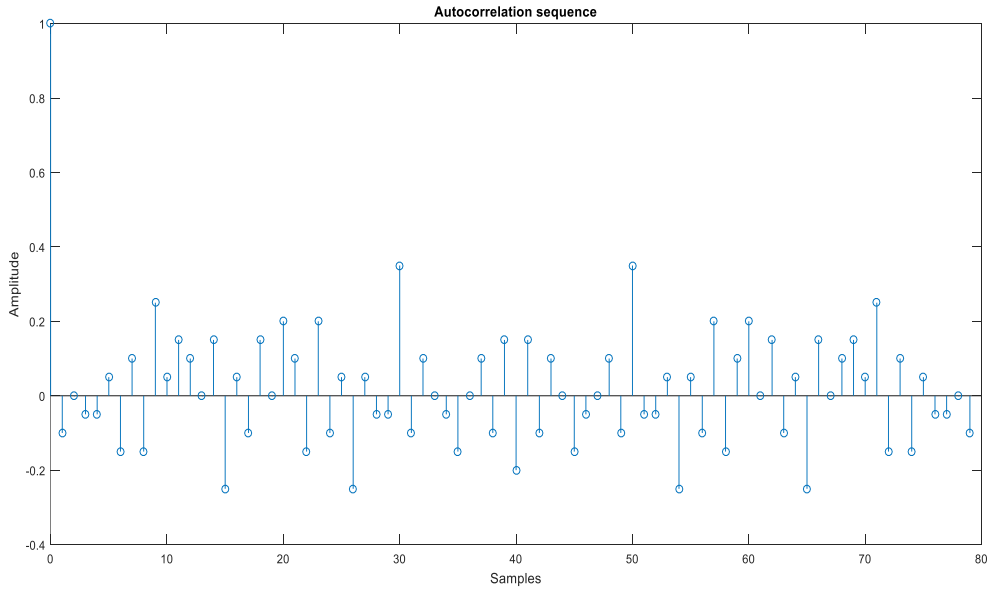

Figure 29. ACF of binary sequence B(n) for Narayana variants of length 80 bits

Looking at Figure 28 and Figure 29, it is evident that the lobes are close to ideal autocorrelation function (ACF) with peak value at zero time shift and values close to zero at all other time shifts.

Randomness is calculated using randomness measure R(x) of a discrete sequence x by the expression (9) represented below:

$$R(x) = 1 - \frac{\sum_{k=1}^{n-1} |ACF(k)|}{N-1}$$

where ACF(k) is the autocorrelation function value for k and N is the period of sequence to characterize the randomness of a sequence. The randomness measure of 1 indicates that the sequence is fully random and randomness measure of 0 indicates constant sequence. The randomness measure of Figure 28 and Figure 29 are found to be 0.83265 and 0.8797 respectively which are close to fully random sequence.

4.2.5) Cross-correlation properties

Cross-correlation describes the relationship between two different sequences. Here, we measure the cross-correlation using expected value of product of two sequences chosen from different time intervals [97]. The expression (10) for calculating the cross correlation function is represented by:

$$CCF(k) = \frac{1}{N} \sum_{i=0}^{N-1} A_i B_{i+k}$$

Where A_i and B_{i+k} are the binary values of two sequences at two different time intervals and N is the length of the sequence. Under ideal conditions, cross correlation function has a peak value of zero at all time shifts. The peak cross correlation function value is calculated using expression (22) represented below:

$$CCF_{peak} = \frac{1}{N} \sum_{k=0}^{N-1} |CCF(k)|$$

Where CCF(k) is the cross correlation function of binary sequence B(n). Now, we perform cross correlation on the binary sequence B(n) to determine whether they are applicable for random processes. Figure 30 and Figure 31 present the normalized cross correlation function of B(n) sequence for 50 and 60 bits respectively.

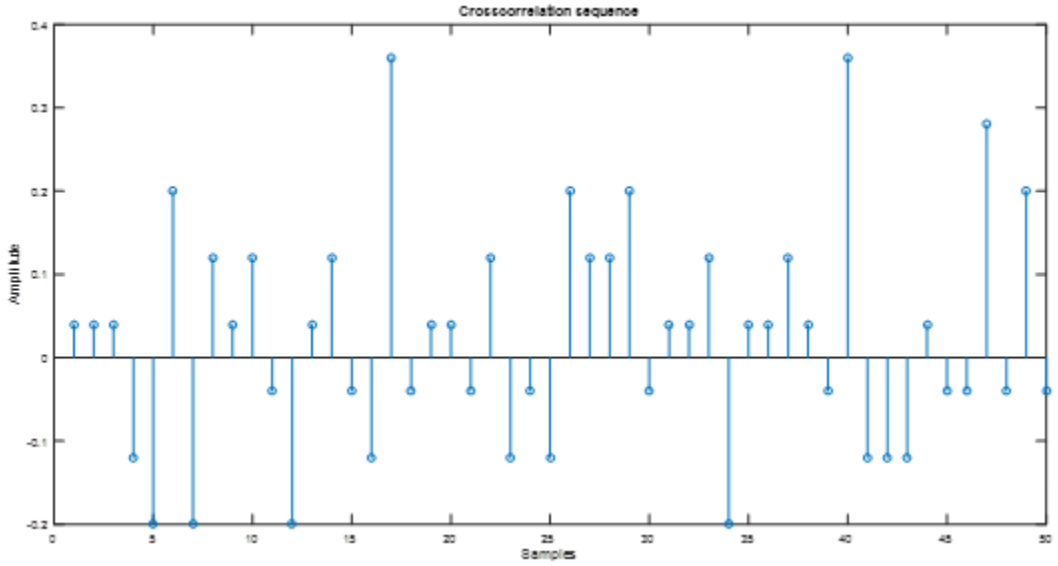

Figure 30. CCF of binary sequence B(n) for Narayana variants of length 50 bits

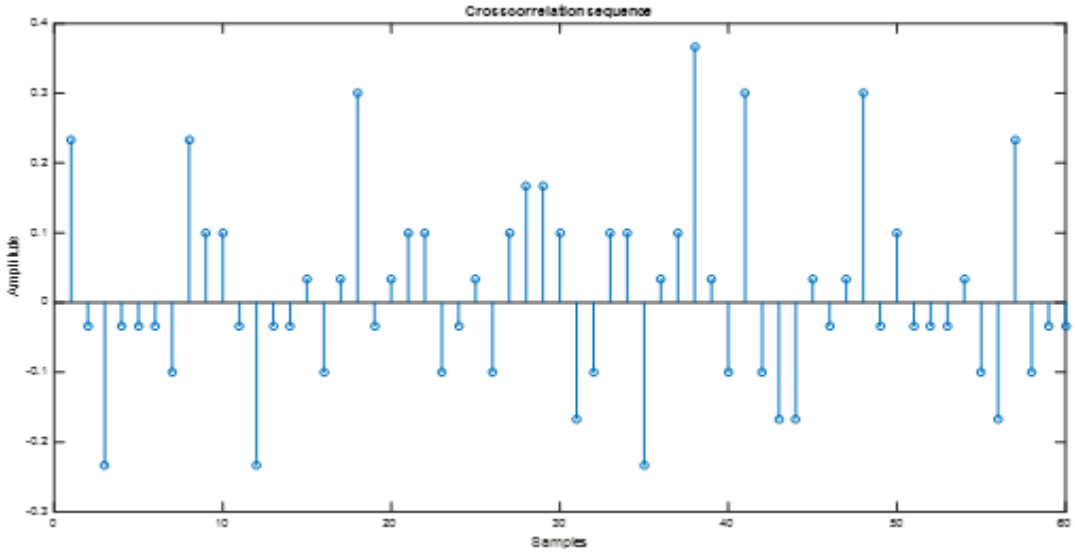

Figure 31. CCF of binary sequence B(n) for Narayana variants of length 60 bits

Looking at Figure 30 and Figure 31, their peak cross-correlation function values are noted to be 0.36 and 0.3667 respectively. These values are close to ideal conditions which are favorable for applications in random processes.

CHAPTER V

NARAYANA UNIVERSAL CODE

5.1) Introduction

A universal code maps positive integers which represent the source messages into codewords of different lengths. The codeword elements are a set of digits that are constructed according to a specified rule, and they may be binary. There are various universal codes including the Elias codes, the Fibonacci universal code, Levenshtein coding and non-universal codes including unary coding, Rice coding, Huffman coding and Golomb coding [34] [98]-[100]. If one were to represent numbers as sum of two prime numbers using Goldbach conjecture, inverse sequence may also sequences may also be used to construct a universal code [44].

The simplest of Elias codes is the gamma code in which the binary representation of the source code is preceded by $[\log_2 x]$ zeroes indicate codeword for any natural number x, where $x \in N = \{1, 2, 3,..\}$. The time requirement for compression and decompression algorithms for cases where decompression time is a critical issue, is advantageous in this coding [101] [102].

The Fibonacci code has a useful property of easy recovery of data from damaged bit stream in comparison with other universal codes. The performance of Fibonacci universal code is better than that of Elias coding [35]. Fibonacci and GH universal codes are obtained using the Zeckendorf representation. In this representation, every positive integer can be represented uniquely as a sum of non-adjacent Fibonacci numbers [103]. This helps in unique representation of codewords without two consecutive 1s, and this may be used for generalization of any coding.

Narayana (short for Narayana Pandit) wrote his famous book Gaṇita Kaumudi in 1356. His eponymous sequences [75]-[77], which are related to Fibonacci and GH sequences, have potential applications in cryptography and data coding. The properties of Fibonacci sequences together with applications in cryptography and coding have been presented in several studies [2] [80] [81] [104] [105]. Here, we present a variant of Fibonacci universal code based on Narayana series with the help of a representation procedure that leads to the Narayana universal code.

5.2) Narayana sequence

The Narayana sequence is derived from the following problem that was proposed by Narayana: "A cow gives birth to a calf every year. In turn, the calf gives birth to another calf when it is three years old. What is the number of progeny produced during twenty years by one cow?" We assume that we begin with a new-born calf, who is shown in the first row of the matrix below. After three years, in each successive year, there is a new calf born to this one and additional calves are born to the 3-year or older calves, leading to second and additional rows in the matrix every 3 steps. This may be represented in the matrix below:

Table 8. Generation of Narayana sequence

1	1	1	1	1	1	1	1	1	1	1	1	1 …
			1	2	3	4	5	6	7	8	9	10…
						1	3	6	10	15	21	28…
									1	4	10	20…
												1…
												….
1	1	1	2	3	4	6	9	13	19	28	41	60…

The last row that adds up the numbers in the previous rows represents the count of the Narayana sequence. This sequence is the sum of previous term and term 2 places before. It is given by

1,1,1,2,3,4,6,9,13,19,28, ….

The $(k+1)^{st}$ term of the Narayana series may be defined using expression (23) represented below:

$$N(k+1)=N(k)+N(k-2)$$

with $N(0)=N(1)=N(2)=1$ and $k\geq 2$. A more general Narayana sequence $N_a(n)$ is given by

{a, b, c, a+c, a+b+c, a+b+2c, 2a+b+3c, 3a+2b+4c, and so on} with a=1, b=2 and c=3 (25)

Consider the ratio of two consecutive terms in Narayana series. In the limit where n goes to infinity, we have

$$\lim_{n\to\infty}\left(\frac{N_a(n+1)}{N_a(n)}\right)=1+\lim_{n\to\infty}\left(\frac{N_a(n-2)}{N_a(n)}\right) \quad (26)$$

Equation (26) may be written as

$$\lim_{n\to\infty}\left(\frac{N_a(n+1)}{N_a(n)}\right)=1+\lim_{n\to\infty}\left(\frac{N_a(n-2)}{N_a(n-1)}\right)*\lim_{n\to\infty}\left(\frac{N_a(n-1)}{N_a(n)}\right) \quad (27)$$

With $\lim_{n\to\infty}\left(\frac{N_a(n+1)}{N_a(n)}\right)=L$, we obtain the equation $L^3-L^2-1=0$. This leads to the following theorem:

Theorem 3:

The real positive solution of equation $L^3-L^2-1=0$ characterizes the relation between two consecutive terms in Narayana sequence, and the Narayana ratio approaches 1.4655712318767669…

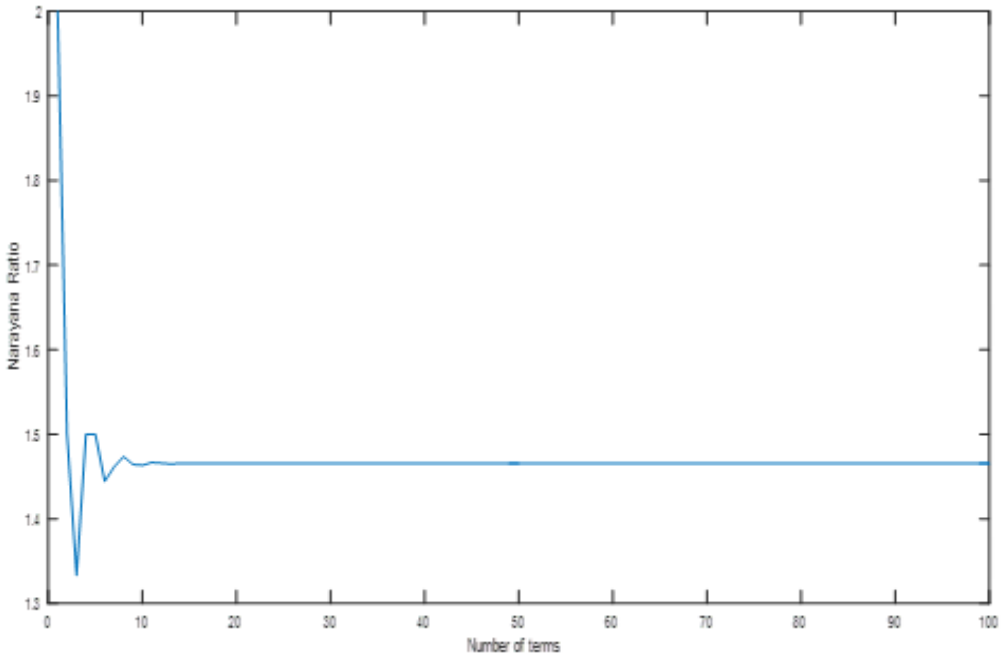

Figure 32. Ratio between first 100 consecutive terms of Narayana series

This constant of 1.4655712318767669 may be termed the Narayana ratio.

5.3) Narayana universal code

To generate Narayana code as a generalization of Fibonacci universal code, we need to be able to map any given positive integer representing source code into variable length codeword in a manner used earlier by Thomas [34].

Consider Narayana series N(k) given by 1,1,1,2,3,4,6,9,… for generating variable length source code for any given positive integer. Since the series contains 3 consecutive 1s, we obtain various codewords for any given positive integer and the resulting codeword fails to comply with the requirement of universal coding.

Hence, to it is essential to modify Narayana series N(k) for k terms such and we define a new series J(k) as N(k+2)=J(k). The series N(k) is mapped to the J series as shown in Table 9.

Table 9. Mapping of Narayana series to J series

Narayana Series N(k)		J(k) Series
1	N(0)	
1	N(1)	
1	N(2)	J(0)
2	N(3)	J(1)
3	N(4)	J(2)
4	N(5)	J(3)
6	N(6)	J(4)
9	N(7)	J(5)
13	N(8)	J(6)
19	N(9)	J(7)
28	N(10)	J(8)

The conditions for unique representation of Narayana universal code in terms of J series to obtain binary set of codewords are now presented.

Rule 1: For a given positive integer *n*, construct a vector $A(n)$ such that $A(n)_i = J(i)$, *i=0,1,…,d,* where $J(d)$ is the largest number of J series less than or equal to *n.* A vector $B(n)$ of binary digits with dimension d is constructed such that

$$A(n)^T B(n) = n \text{ and } B(n)_d = 1. \quad (28)$$

The codeword $NB(n)$ for the positive integer *n*, is defined by a vector with dimension *d+1*, where $NB(n)_k = B(n)_k$ for $1 \leq k \leq d$, and $NB(n)_{d+1} = 1$ [34].

Consider an example of constructing codeword for integer 10. Since $J(5) = 9$ is the largest number of J series less than or equal to 10, vectors $A(n), B(n)$ and $NB(n)$ are given as (*d=5* in this example):

$$A(n) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 6 \\ 9 \end{pmatrix}, B(n) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, NB(n) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad (29)$$

While the recursive nature of Narayana series allows to have more than one representation for some integers using above scheme, $B(n)$ is chosen not to have two consecutive ones [34]. In the above example, integer 10 can be represented as $J(3) + J(4)$ by $B(10) = (0 \quad 0 \quad 0 \quad 1 \quad 1)^T$ or using Zeckendorf representation as $J(0) + J(5)$ by $B(10) = (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)^T$. Since $NB(10)_{d+1} = NB(10)_d = 1$, consecutive ones occur only at the termination of codeword $NB(n)$, when Zeckendorf representation is chosen. Thus, the prefix conditions for unique representation of codeword are found to be:

Rule 2: If the source code to be represented/decoded is a term in Narayana series, the codeword consists of binary set with all zeroes followed by two consecutive ones at the termination.

Rule 3: If the source code to be represented/decoded is not a term in Narayana series, the codeword consists of binary representation of summation of two or more terms in Narayana sequence such that $A(n)^T B(n) = n$ which includes two consecutive ones at the termination as a part of the Zeckendorf representation. We consider codeword which can be represented by summation of least number of terms in the Narayana series.

Consider a general the Narayana sequence which is given by {*a, b, c, a+c, a+b+c, a+b+2c, 2a+b+3c, and so on*}. Let *a+c=d, a+b+c=e, a+b+2c=f, 2a+b+3c=g* in the above sequence which represents *{a, b, c, d, e, f, g, and so on}.* Here, *g* is obtained by summation of *f* and *d* (that is, *g=d+f*) which in turn are summations of *e and c* and *c and a* respectively. Since any term of the Narayana series is obtained by summation of two different terms in the sequence and codeword for any given positive integer is binary representation of sum of two or more terms in

the sequence, the codeword obtained is said to be unique in Zeckendorf representation. Therefore, we are led to the following result:

Theorem 4:

The variable length codeword obtained in Zeckendorf representation using the Narayana sequence for any given positive integer n, which represents the source message, is unique.

Table 10 provides codewords for first 15 natural numbers which contain source messages.

N	Representation in terms of J series	Binary Representation in terms of J series	Narayana Code	Number of bits required for representation of Narayana Code
1	$J(0)$	1	11	2
2	$J(1)$	01	011	3
3	$J(2)$	001	0011	4
4	$J(3)$	0001	00011	5
5	$J(0)+J(3)$	1001	10011	5
6	$J(4)$	00001	000011	6
7	$J(0)+J(4)$	10001	100011	6
8	$J(1)+J(4)$	01001	010011	6
9	$J(5)$	000001	0000011	7
10	$J(0)+J(5)$	100001	1000011	7
11	$J(1)+J(5)$	010001	0100011	7
12	$J(2)+J(5)$	001001	0010011	7
13	$J(6)$	0000001	00000011	8
14	$J(0)+J(6)$	1000001	10000011	8
15	$J(1)+J(6)$	0100001	01000011	8

In order to decode the codeword, remove the last 1 in the codeword and assign the remaining bits with the values 1,2,3,4,6,9,13,19,… which are terms of the Narayana series (Narayana number) and add. Thus, the Narayana code can be used to encode any positive integer, which could be a portion of signal with source messages contained in it.

A variant of Narayana coding scheme can be obtained by defining second-order variant Narayana sequence, $VN_a(n)$, such that $b = 3 - a$ and $c = 1 - a$. This yields $VN_a(0) = a$ ($a \in Z$), $VN_a(1) = 3 - a$, $VN_a(2) = 1 - a$ and for $n \geq 3$, $VN_a(n) = VN_a(n-1) + VN_a(n-3)$.

With the above definition, we obtain the variant Narayana sequence $VN_{-2}(n)$, which starts with $a = -2$, as {*-2,5,3,1,6,9,10,16,25,..*}. However, certain codewords cannot be represented with the above definition since there is no Zeckendorf representation for integer 2 using the above sequence.

Similarly, we obtain $VN_{-1}(n)$ as {*-1,4,2,1,5,7,8,13,20,28,..*}, $VN_{-3}(n)$ as {*-3,5,4,1,6,10,11,17,27,38,..*} and there is no Zeckendorf representation for integers 3 and 15 using the sequence $VN_{-1}(n)$ and integers 2,13 and 19 cannot be represented using sequence $VN_{-3}(n)$. Although, codes obtained through variant Narayana sequence are not capable for encoding certain positive integers, they could be used for portions of source messages which they are able of encode. But, variant Narayana sequences cannot be considered for universal coding.

Figure 33 presents the number of bits required for representation of codewords, obtained through Narayana universal coding, for first 1000 natural numbers.

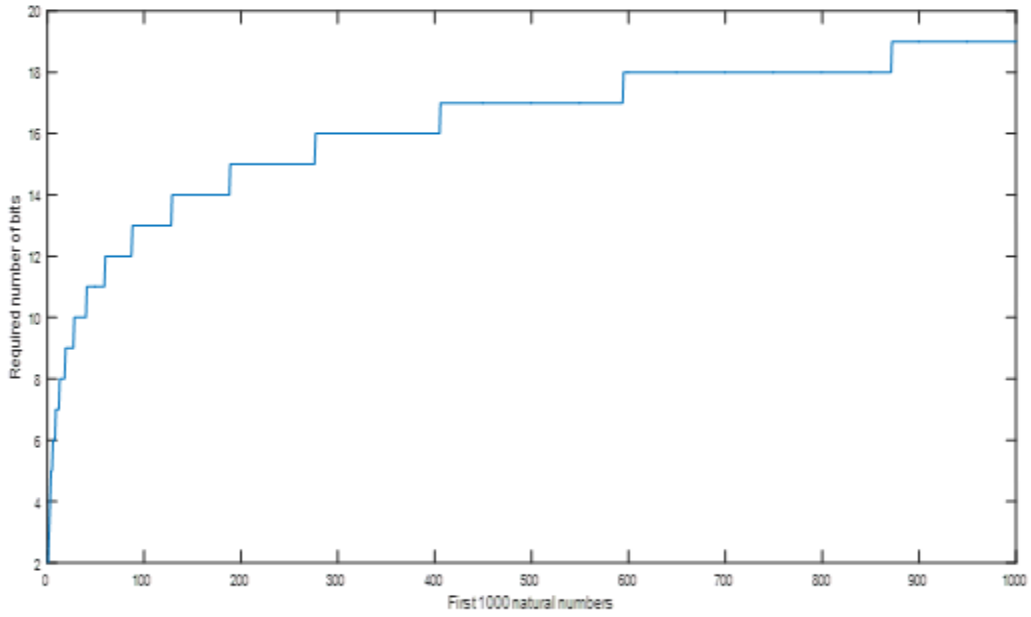

Figure 33. Required number of bits for first 1000 natural numbers

Let the number of elements with the same number of bits, in the codeword representation for natural numbers obtained through Narayana universal coding, for repeated count greater than or equal to 2, be represented by the sequence b(x).

With the above definition, we have b(1)=2, b(2)=3, b(3)=4, b(4)=6 and so on. We see that sequence b(x) represents Narayana series.

Theorem 5:

The sequence b(n) which represents the number of elements with the same number of bits for repeated count greater than or equal to 2, is also according to the Narayana series.

Proof. The above statement is true with respect to Rule 1 in which the codeword is defined by a vector with dimension d+1, where $NB(n)_k = B(n)_k$ for $1 \leq k \leq d$, and $NB(n)_{d+1} = 1$.

5.3) Implementation of Narayana series for Encryption and Hashing

The Narayana series can be used as a key for encryption of plain text in order to obtain cipher text. Suppose an alphabet is used as a key and the Narayana series is used to index alphabets, where index starts from key alphabet, a cipher text could be obtained by combination of ASCII values of key alphabet, the Narayana series indexed alphabets and plain text, dividing the resultant set of numbers into pairs and assigning a letter in the range of [0-25] starting from a to z based on modulus of the pair of numbers [91]. The set of numbers obtained by combining ASCII values can also be considered as cipher text. This provides cipher text whose length is nearly same as that of plain text and thus useful for encryption methods.

The Narayana series can also be used to generate hash table for hashing the plain text into cipher text. This provides various combinations to obtain cipher text leading to difficulty in decrypting and obtaining plain text. Hence, the Narayana series is used in hashing algorithms which ensures protection of data from intruders.

CHAPTER VI

CONCLUSION

The thesis presents new classes of random numbers for cryptographic applications. The new classes include binary GH and Narayana sequences for use in key generation and in wireless communications. Considering GH sequences modulo prime p, the periods are found to be either (p-1) (or a divisor) or (2p+2) (or a divisor) while the Narayana sequence for prime modulo have either p^2+p+1 (or a divisor) or p^2-1 (or a divisor) as their periods. By mapping the different periods to binary values we obtain a corresponding binary sequence. It is shown that the autocorrelation and cross correlation properties of GH and Narayana sequences justify their use as random sequences. The signal to noise ratio values are calculated based on the use of delayed sequences to carry different sets of data in wireless applications. The use of the Narayana sequence as universal code was also established. In summary, the newly discovered classes of random numbers described in the thesis can be used for encoding and decoding applications along with data hiding and information transmission.

REFERENCES

[1] Kak A, 'Computer and Network Security', Classical encryption techniques, Purdue University (2016).

[2] Kak S, 'The piggy bank cryptographic trope', Infocommunications Journal, vol. 6, pp. 22-25 (2014).

[3] Kak S, 'Authentication Using Piggy Bank Approach to Secure Double-Lock Cryptography', arXiv preprint arXiv:1411.3645 (2014).

[4] Mathur A, 'A Research paper: An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms' International Journal on Computer Science and Engineering 4, no. 9 (2012).

[5] Singh G, 'A study of encryption algorithms (RSA, DES, 3DES and AES) for information security' International Journal of Computer Applications 67, no. 19 (2013).

[6] Goyal S, 'A Survey on the Applications of Cryptography' International Journal of Engineering and Technology 2, no. 3 (2012).

[7] Saha S, Satyabrata M, Suman S, Rourab P, Chadrajit P, 'A brief experience on journey through hardware developments for image processing and its applications on Cryptography' arXiv preprint arXiv:1212.6303 (2012).

[8] Niu X, Yongting W, Di W, 'A Method to Generate Random Number for Cryptographic Application' In Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2014 Tenth International Conference on, pp. 235-238. IEEE (2014).

[9] Hellekalek P, 'Good random number generators are (not so) easy to find', Mathematics and Computers in Simulation 46, no. 5 (1998).

[10] Bouftass S, Abdelhak A, 'On a new properties of random number sequences, a randomness test and a new RC4's key scheduling algorithm' (2014).

[11] Kak S, 'Oblivious transfer protocol with verification' arXiv preprint arXiv: 1504.00601 (2015).

[12] Kak S, 'Multiparty Probability Computation and Verification' arXiv preprint arXiv: 1505.05081 (2015).

[13] Stipčević M, 'Quantum random number generators and their use in cryptography' In MIPRO, 2011 Proceedings of the 34th International Convention, pp. 1474-1479. IEEE (2011).

[14] Pickholtz R L, Donald L S, Laurence B M, 'Theory of spread-spectrum communications--a tutorial' Communications, IEEE Transactions on 30, no. 5 855-884 (1982).

[15] Sharma H L, Atul R D, Bawane N G, 'Spread Spectrum Pattern and PN Sequence Retrieval in Wireless Ad Hoc Network: Design Approach' In Emerging Applications of Information Technology (EAIT), 2011 Second International Conference on, pp. 391-394. IEEE (2011).

[16] Mitra A, 'On the properties of pseudo noise sequences with a simple proposal of randomness test' International Journal of Electrical and Computer Engineering 3, no. 3: 164-169 (2008).

[17] Flikkema P, 'Spread-spectrum techniques for wireless communication' Signal Processing Magazine, IEEE 14, no. 3: 26-36 (1997).

[18] Popescu S O, Gontean A S, Budura G, 'BPSK system on Spartan 3E FPGA' In Applied Machine Intelligence and Informatics (SAMI), 2012 IEEE 10th International Symposium on, pp. 301-306. IEEE (2012).

[19] Jacob Z, Lempel A, 'A universal algorithm for sequential data compression' IEEE Transactions on information theory 23, no. 3: 337-343 (1977).

[20] Muramatsu J, Miyake S, 'Hash property and fixed-rate universal coding theorems' arXiv preprint arXiv: 0804.1183 (2008).

[21] Anisimov A V, 'Prefix Encoding by Means of the-Representation of Numbers' Information Theory, IEEE Transactions on 59, no. 4: 2359-2374 (2013).

[22] Huffman D A, 'A method for the construction of minimum-redundancy codes' Proceedings of the IRE 40, no. 9: 1098-1101 (1952).

[23] Salomon D, 'Variable-length codes for data compression' Springer Science & Business Media (2007).

[24] Bogdanski J, Johan A, Nima R, Alma I, Mohamed B, 'Secure multiparty quantum communication over telecom fiber networks' In Lasers and Electro-Optics 2009 and the European Quantum Electronics Conference CLEO Europe-EQEC 2009 European Conference on, pp. 1-1. IEEE (2009).

[25] Du W, Atallah M J, 'Secure multi-party computation problems and their applications: a review and open problems' In Proceedings of the 2001 workshop on New security paradigms, pp. 13-22. ACM (2001).

[26] Rong-lin H, Xia-jun G, Jian-rong Y, Xiang-ping G, Chen Li-Qing, 'Research and design of gateway node based on CDMA for wireless sensor networks' In Information Science and Engineering (ICISE), 2010 2nd International Conference on, pp. 2285-2288. IEEE (2010).

[27] Willig H K A, 'Protocols and architectures for wireless sensor networks' England: John Wiley & Sons (2005).

[28] Gangeshwer D K, 'E-Commerce or Internet Marketing: A Business Review from Indian Context' International Journal of u-and e-Service, Science and Technology 6, no. 6: 187-194 (2013).

[29] Bond R, 'Encryption–use and control in E-commerce' Amicus Curiae 2000, no. 32: 4-10 (2000).

[30] Omotheinwa T O, Ramon S O, 'Fibonacci numbers and golden ratio in mathematics and science' International Journal of Computer and Information Technology (ISSN" 2279-0764) Volume (2013).

[31] Jastrzebska M, Grabowski A, 'Some properties of Fibonacci numbers' Formalized Mathematics 12, no. 3: 307-313 (2004).

[32] Dey S, Al-Qaheri H, Sane S, Sanyal S, 'A Note On the Bounds for the Generalized Fibonacci-p-Sequence and its Application in Data-Hiding' arXiv preprint arXiv:1005.1953 (2010).

[33] Wu J, 'Extended Fibonacci cubes' IEEE Trans. on Parallel and Distributed Systems 8, 1203-1210 (1997).

[34] Thomas J H, 'Variations on the Fibonacci universal code' arXiv: cs/0701085 (2007).

[35] Basu M, Prasad B, 'Long range variations on the Fibonacci universal code' Journal of Number Theory 130: 1925-1931 (2010).

[36] Nalli A, Ozyilmaz C, 'The third order variations on the Fibonacci universal code' Journal of Number Theory 149: 15-32 (2015).

[37] Gupta S, Rockstroh P, Su F E, 'Splitting fields and periods of Fibonacci sequences modulo primes', Math. Mag. 85: 130–135 (2012).

[38] Renault M, 'The period, rank, and order of the (a,b)-Fibonacci sequence mod m' Math. Mag. 86: 372-380 (2013).

[39] Kolmogorov A, 'Three approaches to the quantitative definition of information' Problems of Information Transmission. 1:1-17 (1965).

[40] Kak S, 'Classification of random binary sequences using Walsh-Fourier analysis' IEEE Trans. on Electromagnetic Compatibility, EMC-13: 74-77 (1971).

[41] Kak S, 'Information, physics and computation' Found. of Phys. 26: 127-137 (1996).

[42] Landauer R, 'The physical nature of information' Physics Letters A 217: 188-193 (1996).

[43] Kak S, 'Quantum information and entropy' Int. Journal of Theo. Phys. 46: 860-876 (2007).

[44] Kak S, Chatterjee A, 'On decimal sequences' IEEE Trans. on Information Theory IT-27: 647 – 652 (1981).

[45] Kak S, 'Encryption and error-correction coding using D sequences' IEEE Trans. on Computers C-34: 803-809 (1985).

[46] Mandhani N, Kak S, 'Watermarking using decimal sequences' Cryptologia 29: 50-58 (2005).

[47] Thippireddy S B, 'Binary random sequences obtained from decimal sequences' arXiv preprint arXiv: 0809.0676 (2008).

[48] Kak S, 'Feedback neural networks: new characteristics and a generalization' Circuits, Systems, and Signal Processing 12: 263-278 (1993).

[49] Korripati N S, 'Random sequences generated by recursive maps'. https://subhask.okstate.edu/sites/default/files/nikhil11.pdf

[50] Kak S, 'Goldbach partitions and sequences' Resonance 19: 1028-1037 (2014).

[51] Dixon R C, 'Spread Spectrum Systems' John Wiley (1994).

[52] Helleseth T, Kumar V P, 'Pseudonoise sequences' The Mobile Communications Handbook 264 (1999).

[53] Kedia D, 'Comparative Analysis of Peak Correlation Characteristics of Non-Orthogonal Spreading codes for Wireless Systems' International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.3 (2012).

[54] Jensen J M, Jensen H E, Høholdt T, 'The merit factor of binary sequences related to difference sets' IEEE Trans. Inform. Theory, IT-37, 617–626 (1991).

[55] Winterhof A, Yayla O, 'Family complexity and cross-correlation measure for families of binary sequences', arXiv: 1408.4980v1 (2014).

[56] Sharma M, Mathur R, 'A Method to Generation and Simulation of PN Sequence in MATLAB', ISSN 2250-2459, Volume 2, Issue 7 (2012).

[57] Kak S, 'The initialization problem in quantum computing' Foundations of Physics 29: 267-279 (1999).

[58] Kak S, 'A three-stage quantum cryptography protocol' Foundations of Physics Lett. 19: 293-296 (2006).

[59] Chaitin G, 'Randomness and mathematical proof' Scientific American. 232(5): 47-52 (1975).

[60] Marsaglia G, Tsay L H, 'Matrices and the structure of random number sequences' Linear Algebra Appl. 67: 147-156 (1985).

[61] Kak S, 'Multilayered array computing' Information Sciences 45: 347-365 (1988).

[62] Kim J T, Park H K, Paik E H, 'Security issues in peer-to-peer systems' The 7th International Conference on Advanced Communication Technology, 2005, ICACT 2005., vol.2, pp.1059-1063 (2005).

[63] Gheorghe G, Cigno R L, Montresor A, 'Security and privacy issues in P2P streaming systems: a survey' Peer-to-Peer Netw. Appl. 4: 75-91 (2011).

[64] Kak S, 'The Nature of Physical Reality' Peter Lang, New York, 1986 (2016).

[65] Castells M, 'The Rise of the Networked Society' Wiley-Blackwell (2010).

[66] Kak S, 'The Architecture of Knowledge' CSC, New Delhi (2004).

[67] Rabin M, 'Digitalized signatures and public key functions as intractable as factoring' Tech. Rep. MIT/LCS/TR-212, MIT (1979).

[68] Even S, Goldreich O, Lempel A, 'A randomized protocol for signing contracts' Comm. of the ACM 28: 637-647 (1985).

[69] Singh S, The Code Book: the Secret History of Codes and Code-breaking, FourthEstate, London (1999).

[70] Gnanaguruparan M, Kak S, 'Recursive hiding of secrets in visual cryptography' Cryptologia 26: 68-76 (2002).

[71] Washbourne L, 'A survey of P2P Network security' arXiv:1504.01358 (2015).

[72] Gangan S, ‘A review of man-in-the-middle attacks’ arXiv:1504.02115 (2015).

[73] Kim C, Stern R M, ‘Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis’ In Interspeech, pp. 2598-2601 (2008).

[74] Zhenyu X, Charonko J, Vlachos P, ‘Particle image velocimetry correlation signal-to-noise ratio metrics and measurement uncertainty quantification’ Measurement Science and Technology 25, no. 11: 115301 (2014).

[75] Allouche J P, Johnson T, ‘Narayana’s cows and delayed morphisms’ Cahiers du GREYC, Troisiemes Journées d’Informatique Musicale (JIM 96) 4 (1996).

[76] Flaut C, Shpakivskyi V, ‘On generalized Fibonacci quaternions and Fibonacci-Narayana quaternions’ Adv. Appl. Clifford Algebras 23: 673-688 (2013).

[77] Didkivska T V, Stopochkina M V, ‘Properties of Fibonacci-Narayana numbers’ In the World of Mathematics 9 (1): 29-36 (2003).

[78] Bona M, Sagan B E, ‘On divisibility of Narayana numbers by primes’ J. Integer Sequences (2005).

[79] Barry P, ‘On a generalization of the Narayana triangle’ J. Integer Sequences (2011).

[80] Singh P, ‘The so-called Fibonacci numbers in ancient and medieval India’ Historia Mathematia 12: 229-244 (1985).

[81] Kak S, ‘The Golden Mean and the Physics of Aesthetics’, arXiv: physics/0411195 (2004).

[82] Al F I E, ‘Entanglement entropy of aperiodic quantum spin chains’ Europhysics Letters (2007).

[83] Horibe Y, ‘An entropy view of Fibonacci trees’ The Fibonacci Quarterly (1982).

[84] Varn B, ‘Optimal variable length codes: arbitrary symbol cost and equal code word probability’ Information and Control 19: 289-301 (1971).

[85] Mertens S, Bauke H, ‘Entropy of pseudo-random number generators’ Physical Review E (2004).

[86] Roberts G E, Fun with Fibonacci Numbers: Applications in Nature and Music (2012).

[87] Howard F T, ‘Applications of Fibonacci Numbers’, Volume 8: Proceedings of the Eighth International Research Conference on Fibonacci Numbers and Their Applications (1999).

[88] Lin Y, Peng W, Chen H, Liu Y, Fibonacci Numbers in Daily Life.

[89] Sudha K R, Chandrasekhar A, Reddy P P V G D, Cryptography Protection of Digital Signals using Some Recurrence Relations (2007).

[90] Mishra M, Mishra P, Adhikary M C, Kumar S, Image Encryption Using Fibonacci-Lucas Transformation (2012).

[91] Raphael A J, Sundaram V, ‘Secure Communication through Fibonacci Numbers and Unicode Symbols’, ISSN 2229-5518 (2012).

[92] Elfard S S, Cryptography Based on the Linear Fibonacci Forms (2013).

[93] Leonesio J M, Fascinating Characteristics and Applications of the Fibonacci Sequence (2007).

[94] Hoos H H, Narayana’s Cows – Compositional Study #1 (1996).

[95] Waldschmidt M, Some arithmetic problems raised by rabbits, cows and the Da Vinci Code (2009).

[96] Michael H, ‘Autocorrelation of random processes’, Version 2.4, Connexions, Tech. Rep modulo m10676 (2002).

[97] Michael H, ‘Cross correlation of random processes’, Version 2.2, Connexions, Tech. Rep modulo m10686 (2003).

[98] Platos J, Baca R, Snasel V, Kratky M, El-Qawasmeh E, Fast ‘Fibonacci encoding algorithm’, arXiv: cs/0712.0811v2 (2007).

[99] Buschmann T, Bystrykh L V, Levenshtein error-correcting barcodes for multiplexed DNA sequencing, BMC Bioinformatics 14(1):272 (2013).

[100] Malvar H S, Adaptive Run-Length / Golomb-Rice encoding of quantized generalized Gaussian sources with unknown statistics.
http://researchsrv.microsoft.com/pubs/102069/Malvar_DCC06.pdf

[101] Filmus Y, ‘Universal codes of the natural numbers’ Logical Methods in Computer Science (3:7), 1–11 (2013).

[102] Elias P, ‘Universal codeword sets and representations of the integers’ IEEE Transactions on Information Theory 21, 194-203 (1975).

[103] Kologlu M, Kopp G, Miller S J, Wang Y, ‘On the number of summands in Zeckendorf decompositions’, arXiv:1008.3204v1 (2010).

[104] Filipponi P, Montolivo E, ‘Representation of natural numbers as sums of Fibonacci numbers’ In Applications of Fibonacci Numbers, pp. 89-99. Springer (1990).

[105] Daykin D E, ‘Representation of natural numbers as sums of generalized Fibonacci numbers’ J London Math Soc 35, 143-160 (1960).

VITA

Kirthi Krishnamurthy Vasudeva Murthy

Candidate for the Degree of

Master of Science

Thesis: NEW CLASSES OF RANDOM SEQUENCES FOR CRYPTOGRAPHIC APPLICATIONS

Major Field: Electrical and Computer Engineering

Biographical:

Education:

Received Bachelor of Engineering Degree in Instrumentation Technology from Visvesvaraya Technological University, Bangalore, Karnataka in June, 2013.

Completed the requirements for the Masters of Science in Electrical and Computer Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2016.

Experience: Employed by Oklahoma State University, Department of Electrical and Computer Engineering as Research and Teaching Assistant in the year 2015-2016.

Professional Memberships: Active member of Oklahoma State University Automation Society functioning as social chair.