THE THİRD ORDER VARIATIONS ON THE FIBONACCI UNIVERSAL CODE AND AN APPLICATION TO CRYPTOGRAPHY

Cagla Ozyilmaz, Ayse Nalli

Faculty of Sciences, Department of Mathematics, Karabük University, Turkey

Abstract

In this paper, we have studied the third order variations on the Fibonacci universal code and we have defined $VF_a^{(3)}$ and then we have displayed tables $GH_a^{(3)}(n)$ we have defined for $-20 \le a \le -2$ and $1 \le n \le 50$. We have found that there is no *Gopala-Hemachandra* code for a particular positive integer n and for a particular value of $a \in Z$ and we conclude that for $n = 1,2,3,4,5,6,7,8,9,10$ *Gopala-Hemachandra* code exists for $a = -2,-3,\ldots,-20$ .We have compared with the third order variations on the Fibonacci universal code and the second order variations on the Fibonacci universal code in terms of cryptography and we have found that, the third order variations on the Fibonacci universal code is more advantageous than the second order variations on the Fibonacci universal code.

1.Introduction

Fibonacci coding is a universal code which encodes positive integers into binary codewords and Fibonacci coding is based on Fibonacci numbers. Every positive integer has a unique representation as the sum of nonconsecutive Fibonacci numbers according to Zeckendorf's theorem[4]. Let us first consider the standard Fibonacci numbers of order $m = 2$. Any positive integer B can be represented by a binary string of length r, $c_1c_2\ldots c_{r-1}c_r$, such that $B=\sum_{i=1}^{r} c_i F_i^{(2)}$ .The representation will be unique if one uses the following procedure to produce it: given the integer B, find the largest Fibonacci number $F_r^{(2)}$ smaller or equal to B; then continue recursively with $B-F_r^{(2)}$. For example $16 = 3 + 13$, so its binary Fibonacci representation would be 001001.As a result of this encoding procedure , there are never consecutive Fibonacci numbers in any of these sums, implying that in the corresponding binary representation, there are no adjacent 1 bits.

The generalization to higher order seems at first sight straightforward: any integer B can be uniquely represented by the string $d_1d_2\ldots d_{s-1}d_s$ such that $B = \sum_{i=1}^{s} d_i F_i^{(m)}$ using the iterative encoding procedure mentioned above. In this representation, there are no consecutive substrings of m 1 bits[3]. We append $(m-1)$ 1 bits to the Fibonacci representation of n so as to construct the Fibonacci code of n whose order is m.

2.*Gopala-Hemachandra (G-H)* sequence and codes

Daykin proved that only the Standard Fibonacci sequence gives a unique Zeckendorf's representation for all positive integers . But variant Fibonacci sequences allow for multiple Zeckendorf's representations of the same integer. James Harold Thomas showed that for the second order variant Fibonacci sequence $VF_{-5}^{(2)}(n)=\{-5,6,1,7,8,15,23,38,\ldots\}$ there is no Zeckendorf's representation for integer $n = 5,12$ [5].

*Corresponding author: Address: Karabük Üniverstiesi Mühendislik Fakültesi Karabük 12345 Turkey. E-mail address: casevfesey@hotmail.com

In this paper, we have studied the third order variant Fibonacci sequence , and we have defined $VF_a^{(3)}$ as the *GH* sequence $\{a, b, a+b, 2a+2b, 3a+4b, 6a+7b, 11a+13b, \ldots\}$ where $b = 1 - a$, That is, $VF_a^{(3)}(1) = a\ (a \in Z)$; $VF_a^{(3)}(2) = 1 - a$, $VF_a^{(3)}(3) = 1$; and for $n \geq 4$, $VF_a^{(3)}(n) = VF_a^{(3)}(n-1) + VF_a^{(3)}(n-2) + VF_a^{(3)}(n-3)$ .For the above definition for $a = -2$, we have $\{-2, 3, 1, 2, 6, 9, 17, \ldots\}$. We obtain for the third order variant Fibonacci sequence, for $a = -11$, $VF_{-11}^{(3)}(n) = \{-11, 12, 1, 2, 15, 18, \ldots\}$. It is seen that there is no Zeckendorf's representation for integer 11. With these variant Fibonacci sequences, we can obtain a new universal code , which we call *the Gopala-Hemachandra (G-H)* code . For the third order variations on the Fibonacci universal code ,we have displayed $GH_a^{(3)}(n)$ we have defined for $-20 \leq a \leq -2$ and $1 \leq n \leq 50$ *table 3-4.*

2.1 Abbreviations:

$VF_a^{(m)}$: the m-th order variant Fibonacci sequence

$GH_a^{(m)}$: the family of *GH* codes

$F_i^{(m)}$:the m-th order Fibonacci code.

3.An application of *Gopala-Hemachandra(GH)* codes to cryptography

In this paper, we, particularly, will use stream cipher for an application of *GH* codes to cryptography .

Definition 3.1 : A stream cipher is a tuple (P, C, K, L, E, D) together with a function g , such that the following conditions are satisfied:

1. P is a finite set of possible *plaintexts*

2. C is a finite set of possible *ciphertexts*

3. K , the *keyspace*, is a finite set of possible *keys*

4. L is a finite set called the *keystream alphabet*

5. g is the *keystream generator*. G takes a key K as input , and generates an infinite string $z_1 z_2 \ldots$ called the *keystream* , where $z_i \in L$ for all $i \geq 1$.

6.For each $z \in L$, there is an *encryption rule* $e_z \in E$ and a corresponding *decryption rule* $d_z \in D$. $e_z : P \to C$ and $d_z : C \to P$ are functions such that $d_z(e_z(x))$ for every plaintext element $x \in P$.

To illustrate this definition, suppose that d is the keyword lenght of a stream cipher . Define $K = Z_2$; and define the keystream $z_1 z_2 \ldots$ as follows:

$$z_i = \begin{cases} k_i & , \text{ if } 1 \le i \le d \\ z_{i-d} & , \text{ if } i \ge d+1 \end{cases}$$

where $K = (k_1, k_2, ..., k_d)$. This generates the keystream $k_1 k_2 ... k_d k_1 k_2 ... k_d k_1 k_2 ...$ from the key $K = (k_1, k_2, ..., k_d)$.

Stream ciphers are often described in terms of binary alphabet , i.e., $P = C = L = Z_2$.

In this situation , the encryption and decryption operations are just addition *modulo 2* :

$$e_z(x) = (x + z) \bmod 2 \qquad x = (x_1, x_2, ..., x_d) \in P \tag{3.3}$$

and

$$d_z(x) = (x + z) \bmod 2 \qquad y = (y_1, y_2, ..., y_d) \in C \tag{3.4}$$

In this paper, we will construct a method while applying *GH* codes to cryptography. Now, we will explain the method. While we are encrypting a text message , firstly, we will associate a number to every letter in text message. To easiest way to do that use to encrypt ordinary English text by setting up a correspondence between alphabetic characters and residues *modulo 26* as follows: $A \to 1,\ B \to 2,\ \dots,\ Z \to 26$. Since we will be using this correspondence in several examples , let's record by *table 1.*

Then we will obtain *GH* code of the number corresponding to every letter in text message from *table3* (if $m = 3$) or *table 2* in[2] (if $m = 2$). Later we will add '0' (s) to the end of *GH* codes of the numbers such that the length of *GH* code of every number in text message is equal to the length of the longest *GH* code in text message . Thus we have obtained P in definition 3.1.

Then , we will obtain K in definition 3.1. To do this, firstly , from *table 2* we will obtain the Standard Fibonacci code of $-a$ which is in the family of *GH* codes $GH_a^{(m)}$ which we have used while encrypting the text message .

Later we will add '0' (s) to the end of Standard Fibonacci code of $-a$ such that the length of standard Fibonacci code of $-a$ is equal to the lenght of the lenght of the longest *GH* code in text message. Thus we will have obtained K. Finally , we will encrypt according to (3.3) the text message and we will have obtained C in definition 3.1.

Let's Bob is the person receiving ciphertext. He will decrypt it according to (3.4) and will reconstruct the plaintext. Then he will make sense of plaintext. To do this, firstly, he will allocate parts the plaintext such that the length of every part will be equal to the lenght of K from definition 3.1. Secondly, he will delete '0' (s) at the end of every part including K. Then he will obtain the number corresponding to keyword code from *table 2* so that he can understand which the message have sent *GH* code . Later when he multiple the number by minus, he will have obtained which the message have sent *GH* code. Thus, he can obtain the number corresponding to every part from *table 3*(if $m = 3$) or from *table 2* in [2] (if $m = 2$). Finally, he convert the sequence of integers to alphabetic characters from *table 1* . Let's do several examples.

Example 3.1.1: Suppose $m = 2$ and $a = -4$ .So, while we are encrypting the text message , we will use $GH_{-4}^{(2)}$. Consider the text message 'EMERGENCY' .

We first convert the text message to a sequence of integers.

E	M	E	R	G	E
5	13	5	18	7	5
N	C	Y			
14	3	25			

From *table 2* in [2], we will obtain

011 0000011 011 0100011 000011 011

0010011 100011 01000011.

The lenght of the longest *GH* code in text message is 8. So,we add '0'(s) to the end of *GH* codes such that the lenght of *GH* code of every number in text message is 8 and we will obtain

01100000 00000110 01100000 01000110 00001100 01100000

00100110 10001100 01000011.

Thus, the plaintext is :

011000000000011001100000010001100000110001100000001001101000110001000011.

Now, let's obtain K .

$-a = 4$ and we know $m = 2$. From *table 2,*we will obtain 1011 is the Standard Fibonacci code of 4 . The lenght of the longest *GH* code in text message is 8. So, the key is 10110000 .

The keystream is as follows :

101100001011000010110000101100001011000010110000101100001011000010110000.

Now, we will encrypt plaintext according to (3.3) and we will obtain the ciphertext is :

110100001011011011010000111101101011110011010000100101100011110011110011.

Now, let's look at how the person receiving ciphertext will decrypt .

Thus , the ciphertext is :

110100001011011011010000111101101011110011010000100101100011110011110011 and K is 10110000.

The keystream is as follows :

101100001011000010110000101100001011000010110000101100001011000010110000.

Now, Bob will decrypt ciphertext according to (3.4). Bob will obtain the plaintext is :

01100000000001100110000001000110000011000110000000010011010001100010000011

and K is 10110000.

Now, he will make sense of plaintext. To do this, firstly, Bob allocate parts such that the lenght of every part will be 8 since the lenght of K is 8. Thus,he will obtain

01100000 00000110 01100000 01000110 00001100 01100000 00100110 10001100 01000011.

Then he will delete '0' (s) at the end of every part including K. So, he will obtain

011 0000011 011 0100011 000011 011 0010011 100011 01000011 and K is 1011.

From *table 2*, for $m = 2$, 1011 is the Standard Fibonacci code of 4. When 4 is multiplied by minus, -4 is obtained. So, the text message was sent by using $GH_{-4}^{(2)}$. From *table 2*in [2], the number corresponding to every part is obtained. So, Bob will obtain

5 13 5 18 7 5

14 3 25

Finally, when Bob convert the sequence of integers to alphabetic characters from *table 1*, the text message is obtained

E M E R G E N C Y.

Example 3.1.2 : Suppose $m = 3$ and $a = -10$. So,while we are encrypting the text message , we will use $GH_{-10}^{(3)}$. Consider the text message 'EMERGENCY' . We first convert the text message to a sequence of integers.

E	M	E	R	G	E
5	13	5	18	7	5
N	C	Y			
14	3	25			

From *table 3*, we will obtain

1010111 010111 1010111 11000111 10000111 1010111

0000111 0011011 100100111.

The lenght of the longest *GH* code in text message is 9. So, we add '0'(s) to the end of *GH* codes such that the lenght of *GH* code of every number in text message is 9 and we will obtain

101011100 010111000 101011100 110001110 100001110 101011100

000011100 001101100 100100111.

Thus , the plaintex is :

10101110001011100010101110011000111010000111010101110000000111000011011001001
100111.

Now, let's obtain K. $-a=10$ and we know $m=3$. From *table 2*, we will obtain 110111 is the Standard Fibonacci code of 10. The lenght of the longest *GH* code in text message is 9. So, the key is 110111000 .

The keystream is as follows :

11011100011011100011011100011011100011011100011011100011011100011011100011011100110
111000 . Now, we will encrypt plaintext according to (3.3) and we will obtain the ciphertext is :

01110010010000000000111001000001101100101101100111001001101001001110101000100
011111.

Now, let's look at how the person receiving ciphertext will decrypt .

Thus the ciphertex is :

01110010010000000000111001000001101100101101100111001001101001001110101000100
011111 and K is 110111000.

The keystream is as follows :

11011100011011100011011100011011100011011100011011100011011100011011100011011100110
111000 .

Now , he will decrypt ciphertext according to (3.4). So, he will obtain the plaintext is :

10101110001011100010101110011000111010000111010101110000000111000011011001001
100111and K is 110111000.

Now, he will make sense of plaintext. To do this,firstly, he allocate parts such that the lenght of every part will be 9 since the lenght of K is 9. So, he will obtain

101011100 010111000 101011100 110001110 100001110 101011100

000011100 001101100 100100111.

Then he will delete '0' (s) at the end of every part including K. So, he will obtain

1010111 010111 1010111 11000111 10000111 1010111

0000111 0011011 100100111

and K is 110111.From *table 2*, for $m = 3$, 110111 is the Standard Fibonacci code of 10. When 10 is multiplied by minus, -10 is obtained. So, the text message was sent by using $GH_{-10}^{(3)}$. From *table 3*, the number corresponding to every part is obtained.Thus, he will obtain

5 13 5 18 7 5

14 3 25

Finally, when he convert the sequence of integers to alphabetic characters from *table 1*, the text message is obtain

E M E R G E
N C Y

3.2 Tables

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Table 3.2.1

n	For $m = 2$ Standard Fibonacci code	For $m = 3$ Standard Fibonacci code
1	11	111
2	011	0111
3	0011	1111
4	1011	00111
5	00011	10111
6	10011	01111
7	01011	000111
8	000011	100111
9	100011	010111
10	010011	110111
11	001011	001111
12	101011	101111
13	0000011	0000111
14	1000011	1000111
15	0100011	0100111

Table 3.2.2

	$GH_{-2}^{(3)}$	$GH_{-3}^{(3)}$	$GH_{-4}^{(3)}$	$GH_{-5}^{(3)}$	$GH_{-6}^{(3)}$	$GH_{-7}^{(3)}$	$GH_{-8}^{(3)}$	$GH_{-9}^{(3)}$	$GH_{-10}^{(3)}$
1	00111	00111	00111	00111	00111	00111	00111	00111	00111
2	000111	000111	000111	000111	000111	000111	000111	000111	000111
3	0111	0011011	0011011	0011011	0011011	0011011	0011011	0011011	0011011
4	1000111	0111	1000111	1000111	1000111	1000111	1000111	1000111	1000111
5	1010111	1010111	0111	1010111	1010111	1010111	1010111	1010111	1010111
6	0000111	1001111	1001111	0111	1001111	1001111	1001111	1001111	1001111
7	10000111	0000111	10000111	10000111	0111	10000111	10000111	10000111	10000111
8	10100111	10100111	0000111	10100111	10100111	0111	10100111	10100111	10100111
9	00000111	10010111	10010111	0000111	10010111	10010111	0111	10010111	10010111
10	11000111	00000111	10110111	10110111	0000111	10110111	10110111	0111	10110111
11	00010111	11000111	00000111	0001111	1100111	0000111	010111	011011	0111
12	11010111	00010111	11000111	00000111	0001111	1100111	0000111	010111	01111
13	10001111	11010111	00010111	11000111	00000111	0001111	1100111	0000111	010111
14	10101111	10001111	11010111	00010111	11000111	00000111	0001111	1100111	0000111
15	100000111	10101111	10001111	11010111	00010111	11000111	00000111	0001111	1100111
16	101000111	100000111	10101111	10001111	11010111	00010111	11000111	00000111	0001111
17	000000111	101000111	100000111	10101111	10001111	11010111	00010111	11000111	00000111
18	110000111	100100111	101000111	100000111	10101111	10001111	11010111	00010111	11000111
19	000100111	000000111	100100111	101000111	100000111	10101111	10001111	11010111	00010111
20	110100111	110000111	101100111	100100111	101000111	100000111	10101111	10001111	11010111
21	100010111	000100111	000000111	101100111	100100111	101000111	100000111	10101111	10001111
22	101010111	110100111	110000111	11001111	101100111	100100111	101000111	100000111	10101111
23	100110111	100010111	000100111	000000111	00001111	101100111	100100111	101000111	100000111
24	100001111	101010111	110100111	110000111	11001111	01010111	101100111	100100111	101000111
25	101001111	100110111	100010111	000100111	000000111	00001111	01100111	101100111	100100111
26	100101111	100001111	101010111	110100111	110000111	11001111	01010111	01000111	101100111
27	110001111	101001111	100110111	100010111	000100111	000000111	00001111	01100111	0101111
28	000101111	100101111	100001111	101010111	110100111	110000111	11001111	01010111	01000111
29	110101111	101101111	101001111	100110111	100010111	000100111	000000111	00001111	01100111
30	1000000111	110001111	100101111	100001111	101010111	110100111	110000111	11001111	01010111
31	1010000111	000101111	101101111	101001111	100110111	100010111	000100111	000000111	00001111
32	0000000111	110101111	000001111	100101111	100001111	101010111	110100111	110000111	11001111
33	1100000111	1000000111	110001111	101101111	101001111	100110111	100010111	000100111	000000111
34	0001000111	1010000111	000101111	000110111	100101111	100001111	101010111	110100111	110000111
35	1101000111	1001000111	110101111	000001111	101101111	101001111	100110111	100010111	000100111
36	1000100111	0000000111	1000000111	110001111	001010111	100101111	100001111	101010111	110100111
37	1010100111	1100000111	1010000111	000101111	000110111	101101111	101001111	100110111	100010111
38	1001100111	0001000111	1001000111	110101111	000001111	000010111	100101111	100001111	101010111
39	1000010111	1101000111	1011000111	1000000111	110001111	001010111	101101111	101001111	100110111
40	1010010111	1000100111	0000000111	1010000111	000101111	000110111	010100111	100101111	100001111
41	1001010111	1010100111	1100000111	1001000111	110101111	000001111	000010111	101101111	101001111
42	1100010111	1001100111	0001000111	1011000111	1000000111	110001111	001010111	011000111	100101111
43	0001010111	1000010111	1101000111	010101111	1010000111	000101111	000110111	010100111	101101111
44	1101010111	1010010111	1000100111	0000000111	1001000111	110101111	000001111	000010111	010000111
45	1000110111	1001010111	1010100111	1100000111	1011000111	1000000111	110001111	001010111	011000111
46	1010110111	1011010111	1001100111	0001000111	011001111	1010000111	000101111	000110111	010100111
47	1000001111	1100010111	1000010111	1101000111	010101111	1001000111	110101111	000001111	000010111
48	1010001111	0001010111	1010010111	1000100111	0000000111	1011000111	1000000111	110001111	001010111
49	1001001111	1101010111	1001010111	1010100111	1100000111	010001111	1010000111	000101111	000110111
50	1100001111	1000110111	1011010111	1001100111	0001000111	011001111	1001000111	110101111	000001111

Table 3.2.3

	$GH^{(3)}_{-11}$	$GH^{(3)}_{-12}$	$GH^{(3)}_{-13}$	$GH^{(3)}_{-14}$	$GH^{(3)}_{-15}$	$GH^{(3)}_{-16}$	$GH^{(3)}_{-17}$	$GH^{(3)}_{-18}$	$GH^{(3)}_{-19}$	$GH^{(3)}_{-20}$
1	00111	00111	00111	00111	00111	00111	00111	00111	00111	00111
2	000111	000111	000111	000111	000111	000111	000111	000111	000111	000111
3	0011011	0011011	0011011	0011011	0011011	0011011	0011011	0011011	0011011	0011011
4	1000111	1000111	1000111	1000111	1000111	1000111	1000111	1000111	1000111	1000111
5	1010111	1010111	1010111	1010111	1010111	1010111	1010111	1010111	1010111	1010111
6	1001111	1001111	1001111	1001111	1001111	1001111	1001111	1001111	1001111	1001111
7	10000111	10000111	10000111	10000111	10000111	10000111	10000111	10000111	10000111	10000111
8	10100111	10100111	10100111	10100111	10100111	10100111	10100111	10100111	10100111	10100111
9	10010111	10010111	10010111	10010111	10010111	10010111	10010111	10010111	10010111	10010111
10	10110111	10110111	10110111	10110111	10110111	10110111	10110111	10110111	10110111	10110111
11	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
12	0111	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
13	011011	0111	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
14	010111	011011	0111	N/A	N/A	N/A	N/A	N/A	N/A	N/A
15	0000111	010111	011011	0111	N/A	N/A	N/A	N/A	N/A	N/A
16	1100111	0000111	010111	011011	0111	N/A	N/A	N/A	N/A	N/A
17	0001111	1100111	0000111	010111	011011	0111	N/A	N/A	N/A	N/A
18	00000111	0001111	1100111	0000111	010111	011011	0111	N/A	N/A	N/A
19	11000111	00000111	0001111	1100111	0000111	010111	011011	0111	N/A	N/A
20	00010111	11000111	00000111	0001111	1100111	0000111	010111	011011	0111	N/A
21	11010111	00010111	11000111	00000111	0001111	1100111	0000111	010111	011011	0111
22	10001111	11010111	00010111	11000111	00000111	0001111	1100111	0000111	010111	011011
23	10101111	10001111	11010111	00010111	11000111	00000111	0001111	1100111	0000111	010111
24	100000111	10101111	10001111	11010111	00010111	11000111	00000111	0001111	1100111	0000111
25	101000111	100000111	10101111	10001111	11010111	00010111	11000111	00000111	0001111	1100111
26	100100111	101000111	100000111	10101111	10001111	11010111	00010111	11000111	00000111	0001111
27	101100111	100100111	101000111	100000111	10101111	10001111	11010111	00010111	11000111	00000111
28	0110111	101100111	100100111	101000111	100000111	10101111	10001111	11010111	00010111	11000111
29	0101111	0100111	101100111	100100111	101000111	100000111	10101111	10001111	11010111	00010111
30	01000111	0110111	N/A	101100111	100100111	101000111	100000111	10101111	10001111	11010111
31	01100111	0101111	0100111	N/A	101100111	100100111	101000111	100000111	10101111	10001111
32	01010111	01000111	0110111	N/A	N/A	101100111	100100111	101000111	100000111	10101111
33	00001111	01100111	0101111	0100111	N/A	N/A	101100111	100100111	101000111	100000111
34	11001111	01010111	01000111	0110111	N/A	N/A	N/A	101100111	100100111	101000111
35	000000111	00001111	01100111	0101111	0100111	N/A	N/A	N/A	101100111	100100111
36	110000111	11001111	01010111	01000111	0110111	N/A	N/A	N/A	N/A	101100111
37	000100111	000000111	00001111	01100111	0101111	0100111	N/A	N/A	N/A	N/A
38	110100111	110000111	11001111	01010111	01000111	0110111	N/A	N/A	N/A	N/A
39	100010111	000100111	000000111	00001111	01100111	0101111	0100111	N/A	N/A	N/A
40	101010111	110100111	110000111	11001111	01010111	01000111	0110111	N/A	N/A	N/A
41	100110111	100010111	000100111	000000111	00001111	01100111	0101111	0100111	N/A	N/A
42	100001111	101010111	110100111	110000111	11001111	01010111	01000111	0110111	N/A	N/A
43	101001111	100110111	100010111	000100111	000000111	00001111	01100111	0101111	0100111	N/A
44	100101111	100001111	101010111	110100111	110000111	11001111	01010111	01000111	0110111	N/A
45	101101111	101001111	100110111	100010111	000100111	000000111	00001111	01100111	0101111	0100111
46	01101111	100101111	100001111	101010111	110100111	110000111	11001111	01010111	01000111	0110111
47	010000111	101101111	101001111	100110111	100010111	000100111	000000111	00001111	01100111	0101111
48	011000111	01001111	100101111	100001111	101010111	110100111	110000111	11001111	01010111	01000111
49	010100111	01101111	101101111	101001111	100110111	100010111	000100111	000000111	00001111	01100111
50	000010111	010000111	N/A	100101111	100001111	101010111	110100111	110000111	11001111	01010111

Table 3.2.4

4.Discussion

When an application of *Gopala-Hemachandra (GH)* codes to cryptography is done, for the third order variations on the Fibonacci universal code $GH_{-2}^{(3)}(n), GH_{-3}^{(3)}(n), \ldots, GH_{-10}^{(3)}(n)$ can be used , whereas for the second order variations on the Fibonacci universal code, only $GH_{-2}^{(2)}(n), GH_{-3}^{(2)}(n), GH_{-4}^{(2)}(n)$ can be used. Therefore, as order m increases in variations on the Fibonacci universal code, cryptographic advantages increase , too .

Conclusion

We have defined $VF_a^{(3)}$ and then we have displayed tables $GH_a^{(3)}(n)$ we have defined for $-20 \le a \le -2$ and $1 \le n \le 50$. We have found that there is no *Gopala-Hemachandra* code for a particular positive integer n and for a particular value of $a \in Z$ and we conclude that for $n = 1,2,3,4,5,6,7,8,9,10$ *Gopala-Hemachandra* code exists for $a = -2, -3, \ldots, -20$ and for $1 \le n \le 50$, there is at most k consecutive not available *(N/A) GH* code in $GH_{-(10+k)}^{(3)}(n)$ column where $1 \le k \le 10$ and finally for $1 \le n \le 50$, as k increases , the availability of *GH* code decreases in $GH_{-(10+k)}^{(3)}(n)$ column where $1 \le k \le 10$.

We have compared with the third order variations on the Fibonacci universal code and the second order variations on the Fibonacci universal code in terms of cryptography and we have found that, the third order variations on the Fibonacci universal code is more advantageous than the second order variations on the Fibonacci universal code.

References

[1] A. Apostolico, A. Fraenkel , Robust transmission of strings using Fibonacci representations. IEEE Trans. On Information Theory , 33,238-245, 1987 .

[2] M. Basu , B. Prasad , Long range variations on the Fibonacci universal code, Journal of Number Theory , 130 1925 – 1931 , 2010 .

[3] S. T.Klein, M. K Ben-Nissan, The Computer Journal , On the Usefulness of Fibonacci Compression Codes, 53, 6 , 701 – 716 , 2010 .

[4] E. Zeckendorf, Representation des nombres naturels par une somme des nombres de Fibonacci ou de nombres de Lucas , Bull. Soc. Roy . Sci. Liege 41 179-182 , 1972 .

[5] James Harold Thomas, Variation on the Fibonacci universal code , arxiv:cs/0701085v2 , 2007.

[6] Douglas R. Stinson , *Cryptography Theory and Practice*, Chapman & Hall / CRC ,2002.