A New Application of Generalized $k$-Horadam Sequence in Coding Theory

G. Srividhya$^1$, E. Kavitha Rani$^2$

$^1$ Assistant Professor, Department of Mathematics, Government Arts College. Trichy, Tamil Nadu, South India.
$^2$ Guest lecturer, Department of Mathematics, Government Arts College. Trichy -22, Tamil Nadu, South India.

Received 2022 March 15; Revised 2022 April 20; Accepted 2022 May 10.

Abstract

This paper describes a brand new coding and decoding technique using the generalized $k$-Horadam sequence. Our version is based on a blocked message matrix and encryption of every message matrix with a special key. Experiment with this result in the Jacobsthal sequence.

Keywords: Coding and decoding algorithm, Generalized $k$-Horadam Sequence, Jacobsthal Sequence.

AMS subject classification (2010): 68P30, 11B39, 11B37, 14G50, 11T71

Introduction

Encoding and deciphering algorithms are very crucial to help protection. Because of the reality facts protection is an extra great hassle in modern years.

This paper delineate about the application of generalized $k$-Horadam sequences in coding theory. Yasin Yazlik [8] discussed the generalized $k$-Horadam sequence. There are many authors, such as Hoggat, Koshy, Fikri Koken and Durmus Bozkart [1] Uterate about Fibonacci, Jacobsthal sequence.

Nihal Tas, Sumerya Ucar, Nihal Yilmaz Ozgur [3], [4] provides brief literature on the encoding and decoding methods of various sequences of such as Pell, Fibonacci and Lucas. The study by Stakhove [6] has been a great inspiration for introducing this paper.

In this study we instigate an algorithm for enciphering and deciphering the generalized $k$-Horadam sequence. We extend this result more concretely for Jacobsthal sequence.

Basic Definitions:

Generalized $k$-Horadam sequences are defined by Yasin Yazlik, Necati Taskara [8] as

$$H_{k,n+2} = f(k)H_{k,n+1} + g(k)H_{k,n}$$

With $H_{k,0} = a, H_{k,1} = b$

The equation (1) representing the linear difference equation of second order whose characteristic equation can be written as $\lambda^2 = f(k)\lambda + g(k)$. The real roots of this equation are

$$r_1 = \frac{f(k) + \sqrt{f^2(k) + 4g(k)}}{2}, r_2 = \frac{f(k) - \sqrt{f^2(k) + 4g(k)}}{2}, (r_1 > r_2)$$

$$r_1 + r_2 = f(k), r_1 - r_2 = \sqrt{f^2(k) + 4g(k)}, r_1r_2 = -g(k)$$

(2)
This definition can be reduced to certain special cases depending on the choice of parameters \((a, b, f(k), g(k))\).

### Table – 1

<table>
<thead>
<tr>
<th>S.no</th>
<th>Name of the sequences</th>
<th>((a, b, f(k), g(k)))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Generalized k-Fibonacci sequence</td>
<td>((a, b, 1, k, 1))</td>
</tr>
<tr>
<td>2.</td>
<td>(k)-Fibonacci sequence</td>
<td>((0, 1, 1, k, 1))</td>
</tr>
<tr>
<td>3.</td>
<td>(k)-Lucas sequence</td>
<td>((2, k, 1, k, 1))</td>
</tr>
<tr>
<td>4.</td>
<td>Horadam sequence</td>
<td>((a, b, 1, p, q))</td>
</tr>
<tr>
<td>5.</td>
<td>Fibonacci sequence</td>
<td>((0, 1, 1, 1, 1))</td>
</tr>
<tr>
<td>6.</td>
<td>Lucas sequence</td>
<td>((2, 2, 1, 1, 1))</td>
</tr>
<tr>
<td>7.</td>
<td>Pell sequence</td>
<td>((0, 1, 1, 2, 1))</td>
</tr>
<tr>
<td>8.</td>
<td>Jacobsthal sequence</td>
<td>((0, 1, 1, 1, 2))</td>
</tr>
<tr>
<td>9.</td>
<td>Jacobsthal Lucas sequence</td>
<td>((2, 1, 1, 1, 2))</td>
</tr>
<tr>
<td>10.</td>
<td>Mersenne sequence</td>
<td>((0, 1, 1, 3, 2))</td>
</tr>
<tr>
<td>11.</td>
<td>Fermat sequence</td>
<td>((1, 1, 3, 1, 2))</td>
</tr>
</tbody>
</table>

### Matrix Representation:

By using the concepts of Yasin Yazlik, Necati Taskara \[8\] we can take the matrix as

\[H_n = \left( H_{k,n-1} \ H_{k,n} \ H_{k,n} \ H_{k,n+1} \right) \text{ for } n \geq 1,\]

We can write \( |H_n| = (-g(k))^{n-1} \left( a^2 g(k) + ab f(k) - b^2 \right) \) \hspace{1cm} (3)

### Main Results:

We put our message in an even size matrix by adding zero between the two words and the end of the message until we receive the message grid size to be even.

Partition the message square matrix \(M\) of size \(2m\) into block matrices named \(D_i\) \((1 \leq i \leq m^2)\) of size \(2 \times 2\) from left to right. We assemble a brand new coding method and provide an explanation for the symbol of our coding method. Let us assume that the matrices \(D_i, L_i, H_n\) are following

\[D_i = (d_{i1}^1 \ d_{i1}^2 \ d_{i2}^1 \ d_{i2}^2), \quad L_i = (L_{i1}^1 \ L_{i1}^2 \ L_{i2}^1 \ L_{i2}^2), \quad H_n = (H_{k,n-1} \ H_{k,n} \ H_{k,n} \ H_{k,n+1})\]

Number of block matrix \(D_i\) can be represented by \(b\). In accordance with \(b\) we choose the number \(n\) as below

\[n = (b) \quad b \leq 3 \left[ \frac{n}{2} \right] \quad b > 3\]

By the help of chosen \(n\), we define alphabets table according to \(mod\ 34\) (This table can be expanded depending on the characters used in the message matrix)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n)</td>
<td>(n + 1)</td>
<td>(n + 2)</td>
<td>(n + 3)</td>
<td>(n + 4)</td>
</tr>
<tr>
<td>(F)</td>
<td>(G)</td>
<td>(H)</td>
<td>(I)</td>
<td>(J)</td>
</tr>
<tr>
<td>(n + 5)</td>
<td>(n + 6)</td>
<td>(n + 7)</td>
<td>(n + 8)</td>
<td>(n + 9)</td>
</tr>
<tr>
<td>(K)</td>
<td>(L)</td>
<td>(M)</td>
<td>(N)</td>
<td>(O)</td>
</tr>
<tr>
<td>(n + 10)</td>
<td>(n + 11)</td>
<td>(n + 12)</td>
<td>(n + 13)</td>
<td>(n + 14)</td>
</tr>
<tr>
<td>(P)</td>
<td>(Q)</td>
<td>(R)</td>
<td>(S)</td>
<td>(T)</td>
</tr>
<tr>
<td>(n + 15)</td>
<td>(n + 16)</td>
<td>(n + 17)</td>
<td>(n + 18)</td>
<td>(n + 19)</td>
</tr>
<tr>
<td>(U)</td>
<td>(V)</td>
<td>(W)</td>
<td>(X)</td>
<td>(Y)</td>
</tr>
<tr>
<td>(n + 20)</td>
<td>(n + 21)</td>
<td>(n + 22)</td>
<td>(n + 23)</td>
<td>(n + 24)</td>
</tr>
</tbody>
</table>
We can introduce generalized $k$-Horadam sequence coding and decoding with the transformation.

\[ D_i \times H_n = L_i \]

\[ \text{Det}(L_i) = \text{Det}(D_i \times H_n) = \text{Det} D_i \times \text{Det} H_n [L_1^i L_2^i L_3^i L_4^i] \]

\[ = [d_1^i d_2^i d_3^i d_4^i | H_{k,n-1} H_{k,n} H_{k,n+1}] \]

\[ (4)L_i^1 = d_1^i H_{k,n-1} + d_2^i H_{k,n} (1 \leq i \leq m^2) L_2^i \]

\[ = d_1^i H_{k,n} + d_2^i H_{k,n+1} L_3^i = d_1^i H_{k,n-1} + d_3^i H_{k,n} + d_4^i H_{k,n+1} \]

Let us name it as $\text{Det} D_i = r_i$ and from (3)

\[ \text{Det} H_n = (-g(k))^{-n} (a^2g(k) + abf(k) - b^2) \]

(5)

Let $d_3^i = x_i$ Equation (4) also can be written as below for decoding by the substitution of (5)

\[ [L_1^i L_2^i x_i H_{k,n-1} + d_4^i H_{k,n} x_i H_{k,n} + d_4^i H_{k,n+1}] \]

\[ = r_i (-g(k))^{-n-1} (a^2g(k) + abf(k) - b^2) r_i (-g(k))^{-n-1} (a^2g(k) + abf(k) - b^2) \]

\[ = L_1^i (x_i H_{k,n} + d_4^i H_{k,n+1}) - L_2^i (x_i H_{k,n-1} + d_4^i H_{k,n}) \]

(6)

**Blocking Algorithm**

**Coding Algorithm:**

Step 1: Divide the message matrix $M$ into block $D_i$ ($1 \leq i \leq m^2$)

Step 2: Select $n$.

Step 3: Find $d_j^i (1 \leq j \leq 4)$

Step 4: Determine $\text{det det}(D_i) \rightarrow r_i$

Step 5: Constructing $E = [r_i d_j^i]_{p=1,2,4}$

Step 6: The algorithm is now complete.

**Decoding Algorithm:**

Step 1: Compute $H_n$ for chosen $n$

Step 2: Compute $L_1^i, L_2^i$ to construct $L_i$

\[ L_1^i = d_1^i H_{k,n-1} + d_2^i H_{k,n}, L_2^i = d_1^i H_{k,n} + d_2^i H_{k,n+1} (1 \leq i \leq m^2) \]

Step 3: Solve

\[ r_i (-g(k))^{-n-1}(a^2g(k) + abf(k) - b^2) = L_1^i (x_i H_{k,n} + d_4^i H_{k,n+1}) - L_2^i (x_i H_{k,n-1} + d_4^i H_{k,n}) \]

Step 4:

Substitute for $x_i = d_3^i$

Step 5:
Construct $D_i$
Step 6:
Construct message matrix $M$
Step 7:
End of Algorithm.

We can explain the above algorithm more specifically by choosing any of the special cases defined in table 1.

Here we are considering Jacobsthal Sequence.

Consider $f(k) = 1, g(k) = 2, a = 0, b = 1$ and $k = 1$

By the help of (1) we can rewrite equation of Jacobsthal sequence as $J_{1,n+2} = J_{1,n+1} + 2J_{1,n}$

From (3) $J_2 = \begin{bmatrix} J_{1,1} & J_{1,2} & J_{1,3} \end{bmatrix}$

(7)

Since $J_{1,0} = 0, J_{1,1} = 1, J_{1,2} = 1, J_{1,3} = 3$. Hence $J_2 = [1 \ 1 \ 3]$

Using Equation (4)

$L_i = d_i^1(1) + d_i^2(1)$
$L_i = d_i^1(1) + d_i^2(3)$ where $(1 \leq i \leq 4)$

Equation (6) can be simplified here as

$r_i(-2)^{n-1}(-1) = L_i^1(1) + d_i^2(3) - L_i^2(1) = L_i(1 + 3d_i^4) - L_i^2(1 + d_i^4)r_i(-1)^{n-1}$

(9)

Application:

Here we can discuss an example using above algorithm. Let us consider the message matrix for the message text “MATH SYMBOL + - *”

Coding Algorithm:

Step 1:
Partition the message matrix $M$ of size $4 \times 4$ matrices with the name label $D_i$ $(1 \leq i \leq 4)$ from left to right, each of size $2 \times 2$

$M = [M \ A \ T \ H \ 0 \ S \ Y \ M \ B \ + \ O \ - \ L \ * \ 0 \ ]._{4\times 4}$

$D_1 = [d_1^1 \ d_1^2 \ d_1^3 \ d_1^4] = [M \ A \ 0 \ S \ ]., D_2 = [d_2^1 \ d_2^2 \ d_2^3 \ d_2^4] = [T \ H \ Y \ M \ ]D_3 = [d_3^1 \ d_3^2 \ d_3^3 \ d_3^4] = [B \ O \ + \ - \ ]., D_4 = [d_4^1 \ d_4^2 \ d_4^3 \ d_4^4] = [L \ 0 \ * \ ].$

Step 2:
Since $b$ represent number of Block Matrix $'D_i'$ hence $b = 4 \geq 3$

We find $n = \begin{bmatrix} 4 \ 2 \end{bmatrix} = 2$.

We can use the following character table for the message matrix $M$.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>T</th>
<th>H</th>
<th>O</th>
<th>S</th>
<th>Y</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>2</td>
<td>21</td>
<td>9</td>
<td>28</td>
<td>20</td>
<td>26</td>
<td>14</td>
</tr>
<tr>
<td>0</td>
<td>L</td>
<td>0</td>
<td>+</td>
<td>-</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 3:
We build components of the Block $D_i$ ($1 \leq i \leq 4$)

<table>
<thead>
<tr>
<th>$\downarrow$</th>
<th>$d_1^1 = 14$</th>
<th>$d_1^2 = 2$</th>
<th>$d_1^3 = 28$</th>
<th>$d_1^4 = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_2^1 = 21$</td>
<td>$d_2^2 = 9$</td>
<td>$d_2^3 = 26$</td>
<td>$d_2^4 = 14$</td>
<td></td>
</tr>
<tr>
<td>$d_3^1 = 03$</td>
<td>$d_3^2 = 16$</td>
<td>$d_3^3 = 32$</td>
<td>$d_3^4 = 33$</td>
<td></td>
</tr>
<tr>
<td>$d_4^1 = 13$</td>
<td>$d_4^2 = 28$</td>
<td>$d_4^3 = 34$</td>
<td>$d_4^4 = 31$</td>
<td></td>
</tr>
</tbody>
</table>

Step 4:
Calculate determinant of $r_i$ of the block matrix $D_i$

$D_1 = [M A 0 S] = [d_1^1 d_1^2 d_1^3 d_1^4] = [14 2 28 20], r_1 = \det (D_1) = 280 - 56 = 224$

$D_2 = [T H Y M] = [d_2^1 d_2^2 d_2^3 d_2^4] = [21 9 26 14], r_2 = \det (D_2) = 294 - 234 = 60$

$D_3 = [B 0 -1 -] = [d_3^1 d_3^2 d_3^3 d_3^4] = [3 16 32 33], r_3 = \det (D_3) = 99 - 512 = -413$

$D_4 = [L 0 . .] = [d_4^1 d_4^2 d_4^3 d_4^4] = [13 28 34 31], r_1 = \det (D_4) = 403 - 952 = -549$

Step 5:
Construct $E = [r_i, d_i^j], p \in (1,2,4)E = [224 142 20 60 21 9 14 - 413 - 549 3 13 16 28 33 31 ]$

Step 6:
End Algorithm

Decoding Algorithm:

Step 1:
From equation (7) $H_2 = [J_{1,1}, J_{1,2}, J_{1,2}, J_{1,3}] = [1 1 1 3 ]$

Step 2:
Compute $L_1^i, L_2^i$ to construct $L_i$

From equation (5) $L_1^i = d_1^i + d_2^i (1 \leq i \leq 4) L_1^1 = d_1^1 + d_1^2 = 16, L_2^2 = d_2^1 + d_2^2 = 30, L_3^3 = d_3^1 + d_3^2 = 19, L_4^4 = d_4^1 + d_4^2 = 41$

$L_1^2 = d_1^1 + 3d_2^2$

$L_1^3 = d_1^1 + 3d_2^3 = 20, L_2^2 = d_2^2 + 3d_2^3 = 48, L_3^3 = d_3^2 + 3d_3^3 = 51, L_4^4 = d_4^2 + 3d_4^3 = 97$

Step 3:
when $1 \leq i \leq 4, n = 2$, Equation (9) becomes

$r_1(-1)^2 \bar{z}^1 = L_1^1(x_1 + 3d_1^1) - L_2^2(x_1 + d_1^2) = 16(x_1 + 60) - 20(x_1 + 20) = 28$

$r_2(-1)^2 \bar{z}^2 = L_1^1(x_2 + 3d_1^1) - L_3^3(x_2 + d_2^3) = 20 + 1260 - 482 = 26$

$r_3(-1)^2 \bar{z}^3 = L_1^1(x_3 + 3d_1^1) - L_4^4(x_3 + d_2^4) = -826 = 19x_3 + 1881 - 51x_3 - 1683x_3 = 32$

$r_4(-1)^2 \bar{z}^4 = L_1^1(x_4 + 3d_1^1) - L_4^4(x_4 + d_2^4) = 1098 = 41x_4 + 3813 - 97x_4 - 3007x_4 = 34$
Step 4:
Substitute for $x_i = d^i_3 \quad i = 1 \text{ to } 4$
Hence $d^1_3 = 28, d^2_3 = 26, d^3_3 = 32, d^4_3 = 34,$

Step 5:
Construct $D_i \quad i = 1 \text{ to } 4$
(With the help of $E$ step 5 in coding algorithm)
$D_1 = (14 \ 28 \ 20), D_2 = (21 \ 9 \ 26 \ 14), D_3 = (3 \ 16 \ 32 \ 33), D_4 = (13 \ 28 \ 34 \ 31)$

Step 6:
Message matrix $M = [14 \ 28 \ 20 \ 14 \ 32 \ 33 \ 16 \ 33 \ 13 \ 34 \ 28 \ 31 ] = [\text{MATH} \ 0 \text{SYM} \ 0 \text{B} + \text{O} - \text{L} \ast \text{0}].$

Text Message “MATH SYMBOL + *.”

Step 7:
The algorithm is now complete.

Conclusion:
In this paper we introduced a new blocking algorithm to encrypt and decrypt a message. We explain these results more specifically for Jacobsthal. One can extend this result for other sequences like Oresmme, Fermat sequences. These new calculations won’t just build the security of data yet in addition has high right capacity.

References:
[7]. Sumeyra Ucar, Nihal Tas and Nihal Yilmaz Ozgur “A new application to coding theory via Fibonacci and Lucas Number” Mathematical Sciences and Applications E. Notes 7(1) 62-70 (2019)©MSAEN.