South East Asian J. of Mathematics and Mathematical Sciences
Vol. 17, No. 3 (2021), pp. 17-28

ISSN (Online): 2582-0850

ISSN (Print): 0972-7752

CRYPTOGRAPHIC ALGORITHMS USING FINITE STATE MACHINE, BERNOULLI AND LUCAS NUMBERS

Rachna Navalakhe and Harsha Atre*

Department of Applied Mathematics and Computational Science,
Shri Govindram Seksaria Institute of Technology and Science,
Vallabh Nagar, Indore - 452003, Madhya Pradesh, INDIA

E-mail : sgsits.rachna@gmail.com

*Department of Applied Science,
SAGE University, Indore - 452020, Madhya Pradesh, INDIA

E-mail : atre.harsha.p22@gmail.com

(**Received:** *Jan. 29, 2021* **Accepted:** *Sep. 25, 2021* **Published:** *Dec. 30, 2021)*

Abstract: The aim of this paper is to propose a new cryptographic encryption and decryption algorithms using finite state machine and explicit forms of recurrence relations that is Bernoulli numbers and Lucas numbers. The efficiency of the proposed algorithm has been analyzed and the analysis shows an improved cryptographic protection in digital communications. The authenticity of algorithms is assured because in these algorithms we have used multiple set of keys to encipher the original message and its inverse to decipher the original message again. The algorithms has different levels of security which enhance the chances to keep our data or information confidential and secure for long time. There are many states in finite state machine to calculate the appropriate output. In every state it takes a new recurrence relation which depends upon the input. To compute the output we apply mathematical operation which is our cipher text. At each level we have number of cipher text which increases the data security. With the use of this algorithms, we can send information securely through the communication channel because we have used different recurrence relations at each level of input.

Keywords and Phrases: Cryptography, Lucas Numbers, Moore machine, Mealy

Machine, cipher text.

2020 Mathematics Subject Classification: 03D05, 11T71, 15A15.

1. Introduction

The fundamental objective of cryptography [7], [3] is to enable two people to communicate over an insecure channel in such a way that any opponent cannot understand what is being said. Communications security is gaining importance due to increase in electronic communications in our every day activity. Cryptography is the practice and study of hiding information. It is a critical part of secured communication. Cryptography is an art of achieving security by encoding messages to make them non-readable. The cryptography consists of two techniques one is encryption and another one is decryption [10], [11]. Encryption is a method of conversion of plain text into cipher text. Decryption is the reverse process of encryption that is conversion of cipher text into plain text. The technological advancement in today's world have made the cryptographic algorithms more prone to attacks.

2. Finite State Automata

A finite state automata or machine is a mathematical model of computation. It has finite internal memory with input feature that reads symbols one at a time and an output feature that produces output which can be understood by user, once the model is created. It is an abstract machine that can be in exactly one of a finite number of states at any given time. The FSM can change from one state to another in response to some inputs, the change from one state to another is called a transition. Recently, in cryptography [6] finite state machines are used to encrypt the message as well as maintain secrecy of the message.
There are two types of Finite state automata without output.

2.1. Deterministic Finite Automata (DFA)

Definition 2.1. *A deterministic finite automaton is defined by quintuple* $M = (Q, \sum, \delta, q_0, F)$ *where* Q *is a finite set of internal states,* $\sum$ *is a finite state of symbols called the input alphabet,* $\delta : Q \times \sum \to Q$ *is a transition function,* $q_0 \in Q$ *is the initial state,* $F \subseteq Q$ *is a set of final state.*

2.2. Non- Deterministic Finite Automata (NDFA)

Definition 2.2. *A Non- Deterministic finite automata* [4], [5] *is defined by quintuple* $M = (Q, \sum, \delta, q_0, F)$ *where* $Q, \sum, q_0, F$ *are defined as for deterministic finite accepters, but* $\delta : Q \times (\sum \cup \lambda) \to 2^Q$.
The difference between ***deterministic*** *and* ***nondeterministic automata*** *is only*

in δ.For deterministic automaton (DFA) the outcome is a state that is an element of Q; for nondeterministic automaton the outcome is a subset of Q.

2.3. Types of Finite State Machine with Outputs

i **Moore Machine**
A Moore machine [5] is a six-tuple $M = (Q, \sum, \Delta, \delta, \lambda, q_0)$ where Q is a finite set of internal states. $\sum$ is a finite state of symbols called the input alphabet, Δ is an output alphabet, δ; $\sum \times Q \to Q$ is a transition function, λ is the output function mapping Q into Δ, q_0 is the initial state.

ii **Mealy Machine**
A Mealy machine [5] is a six-tuple $M = (Q, \sum, \Delta, \delta, \lambda, q_0)$ where Q is a finite set of internal states. $\sum$ is a finite state of symbols called the input alphabet, Δ is an output alphabet,$\delta : \sum \times Q \to Q$ is a transition function, λ is the output function mapping $\sum \times Q$ into Δ, q_0 is the initial state.
If in a finite state machine, output depends on the present state as well as the present input, then this type of machine is called Mealy machine, while in the Moore machine the outputs depended only on the present state. Here we have used Moore machine.

3. Recurrence Relation

A recurrence relation [1] [2] is an equation which is defined in term of itself. The recurrence relation is of the form $A_0x_n + A_1x_{n-1} + A_2x_{n-2} + ... + A_kx_{n-k} = Y_n$. A recurrence relation relates the n^{th} element of a sequence to its predecessors. Initial conditions for the sequence $a_0, a_1, a_2, ...$ are explicitly given values for a finite number of the terms of the sequence. Matrix which is derived from the recurrence relation is known as recurrence matrix.

3.1. Bernoulli Numbers

The Bernoulli numbers [8] are the sequence of rational numbers which is defined by exponential generating functions

$$\frac{x}{e^x - 1} = \sum_{n=0}^{\infty} B_n \frac{x^n}{n!}$$

These numbers generally arise in the series expansions of trigonometric functions and very important in number theory and its analysis.

3.1.1. Recursion formula for Bernoulli numbers

The recursive formula for Bernoulli number is

$$B_n = \sum_{k=0}^{n} \binom{n}{k} B_k ...\text{for } n \geq 2$$

For every n other than $0, B_n$ is negative if it is divisible by 4 and for every n other than even numbers, $B_n = 0$. The values of first 5 Bernoulli numbers are as follows:

No	Fraction
0	1
1	$-\frac{1}{2}$
2	$\frac{1}{6}$
3	0
4	$-\frac{1}{30}$
5	0

Table 1: Values of Bernoulli numbers.

On using the Bernoulli recursion [5] the matrix will be $B_n = \begin{bmatrix} B_{n-1} & B_n \\ B_n & B_{n+1} \end{bmatrix}$

For $n = 1$, the matrix elements are $B_1 = \begin{bmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{6} \end{bmatrix}$

For $n = 2$, the extension of matrix is as follows: $B_2 = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{6} & 0 \\ 0 & 0 & 1 \end{bmatrix}$

The inverse of above matrix is $B_2^{-1} = \begin{bmatrix} -2 & -6 & 0 \\ -6 & -12 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Then other explicit forms of B_2 are calculated recursively (in terms of multiplicative inverse) which is shown in table:

X	1	2	3	4
B_2^x	$\begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{6} & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{5}{4} & -\frac{7}{12} & 0 \\ -\frac{7}{12} & \frac{10}{36} & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{37}{24} & -\frac{52}{72} & 0 \\ -\frac{52}{72} & \frac{73}{216} & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{274}{144} & -\frac{385}{432} & 0 \\ -\frac{385}{432} & \frac{541}{1296} & 0 \\ 0 & 0 & 1 \end{bmatrix}$
B_2^{-x}	$\begin{bmatrix} -2 & -6 & 0 \\ -6 & -12 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 40 & 84 & 0 \\ 84 & 180 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -584 & -1284 & 0 \\ -1284 & -2664 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 8656 & 18480 & 0 \\ 18480 & 39456 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table 2: Explicit forms of B_2^x.

3.2. Lucas Numbers

The Lucas numbers [8] are an integer sequence named after the mathematician Francois Eduard Anatole Lucas. Lucas numbers are also defined as the sum of its

two preceding terms. The Lucas numbers may be defined as

$$L_n = \begin{cases} 2 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ L_{n-1} + L_{n-2} & \text{if } n > 1. \end{cases}$$

The sequence of Lucas numbers is $2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, ...$ The Lucas numbers can have following recurrence relation $L_{n+1} = L_n + L_{n-1}$.
With the initial conditions $L_0 = 2$ and $L_1 = 1$. Here is a 2*2 matrix below

$$L_n = \begin{bmatrix} L_{n-1} & L_n \\ L_n & L_{n+1} \end{bmatrix}$$

When $n = 1$, matrix the values are $L_1 = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$

When $n = 2$, and the extension of the matrix is $L_2 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

The additive inverse of the above matrix is $L_2^{-1} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & -3 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

Then other explicit forms of L_2 are calculated recursively (in terms of additive inverse) which is shown in table:

X	1	2	3	4
L_2^x	$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 0 \\ 5 & 10 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 15 & 20 & 0 \\ 20 & 35 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 50 & 75 & 0 \\ 75 & 125 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
L_2^{-x}	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & -3 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -5 & -5 & 0 \\ -5 & -10 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -15 & -20 & 0 \\ -20 & -35 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -50 & -75 & 0 \\ -75 & -125 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

Table 3: Explicit forms of L_2^x.

4. Algorithm for Encryption and Decryption

In Cryptography, we have plain text, key and cipher text; to hide the originality of plain text we add a key into it which converts the plain text into the cipher text. Cipher text is changed at each level so it will surely enhance the security level of the original message and it is very hard to decode it.

4.1. Encryption Algorithm

Step 1: Consider a plain text P.

Step 2: Define a Moore machine through public channel.

Step 3: Define input.

Step 4: Get output through Moore machine.

Step 5: Now define recurrence matrix and recurrence relation.

Step 6: Define the value n of recurrence matrix.

Step 7: Get cipher text for all the plain text.

Step 8: Send the cipher text to the receiver.

4.2. Decryption Algorithm

On getting the finite state machine, cipher text and recurrence matrix decode the plaintext either by using multiplicative or additive inverse else we can use inverse operation of the recurrence relation or matrix, to get the plain text or unique information. For a finite state machine with n states, we require n multiplicative or additive inverse.

4.3. Implementation of Algorithm

Example 1. Let the plain text be $P = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

We want to encrypt the plain text $P = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

Solution : Encryption Process

Step 1: $P = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$.

Step 2: Moore machine is published through public channel.

Step 3: Let the input binary key is = 10101 .

Step 4: The output of the above key is found by Moore machine shown in Figure 1.

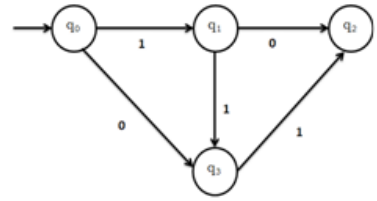

Figure 1: Moore Machine to calculate output for 10101

Step 5: The Recurrence matrix is defined by extension of Bernoulli numbers.

$$B_n = \begin{bmatrix} B_{n-1} & B_n & 0 \\ B_n & B_{n+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{with initial conditions } B_0 = 0, B_1 = -1/2$$

$$B_2^1 = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{6} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_2^2 = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{6} & 0 \\ 0 & 0 & 1 \end{bmatrix}^2 = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{6} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{6} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{5}{4} & -\frac{7}{12} & 0 \\ -\frac{7}{12} & \frac{10}{36} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_2^3 = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{6} & 0 \\ 0 & 0 & 1 \end{bmatrix}^3 = B_2^2 \begin{bmatrix} \frac{5}{4} & -\frac{7}{12} & 0 \\ -\frac{7}{12} & \frac{10}{36} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{37}{24} & -\frac{52}{72} & 0 \\ -\frac{52}{72} & \frac{73}{216} & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and so on .}$$

Step 6: Assume that the cipher text at q_{i+1} is $q_{i+1} = q_i B_2^x$ ... (1)
where q_{i+1} = shows the output,q_i = shows the input, B_2^x = Recurrence matrix key. By eq (1).

Present State = Previous State B_2^x
$q_1 = q_0.B_2^1$
$q_2 = q_0.B_2^2$
$q_2 = q_1.B_2^2$
$q_3 = q_1.B_2^3$
$q_3 = q_2.B_2^3$

Table 4: Previous and Present state of the system.

Step 7: Now we compute cipher text at each level. Then the cipher text at each level is

S.No.	Input	Present State	Previous State	Key	Cipher Text
1.	1	q_1	q_0	B_2^1	$q_1 = q_0.B_2^1 = \begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}.\begin{bmatrix}1&\frac{-1}{2}&0\\\frac{-1}{2}&\frac{1}{6}&0\\0&0&1\end{bmatrix} = \begin{bmatrix}0&\frac{-1}{6}&3\\\frac{3}{2}&\frac{-7}{6}&6\\3&\frac{-13}{6}&9\end{bmatrix}$
2.	0	q_2	q_0	B_2^2	$q_2 = q_0.B_2^2 = \begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}.\begin{bmatrix}\frac{5}{4}&\frac{-7}{12}&0\\\frac{-7}{12}&\frac{10}{36}&0\\0&0&1\end{bmatrix} = \begin{bmatrix}\frac{1}{12}&\frac{-1}{36}&3\\\frac{25}{12}&\frac{-34}{36}&6\\\frac{49}{12}&\frac{-67}{36}&9\end{bmatrix}$
3.	1	q_2	q_1	B_2^2	$q_2 = q_1.B_2^2 = \begin{bmatrix}0&\frac{-1}{6}&3\\\frac{3}{2}&\frac{-7}{6}&6\\3&\frac{-13}{6}&9\end{bmatrix}.\begin{bmatrix}\frac{5}{4}&\frac{-7}{12}&0\\\frac{-7}{12}&\frac{10}{36}&0\\0&0&1\end{bmatrix} = \begin{bmatrix}\frac{7}{12}&\frac{-10}{216}&3\\\frac{184}{72}&\frac{-259}{216}&6\\\frac{901}{216}&\frac{-508}{216}&9\end{bmatrix}$
4.	0	q_3	q_1	B_2^3	$q_3 = q_1.B_2^3 = \begin{bmatrix}0&\frac{-1}{6}&3\\\frac{3}{2}&\frac{-7}{6}&6\\3&\frac{-13}{6}&9\end{bmatrix}.\begin{bmatrix}\frac{37}{24}&\frac{-52}{72}&0\\\frac{-52}{72}&\frac{73}{216}&0\\0&0&1\end{bmatrix} = \begin{bmatrix}\frac{52}{432}&\frac{-73}{1296}&3\\\frac{1363}{432}&\frac{-1915}{1296}&6\\\frac{2674}{432}&\frac{3757}{1296}&9\end{bmatrix}$
5.	1	q_3	q_2	B_2^3	$q_3 = q_2.B_2^3 = \begin{bmatrix}\frac{7}{12}&\frac{-10}{216}&3\\\frac{184}{72}&\frac{-259}{216}&6\\\frac{901}{216}&\frac{-508}{216}&9\end{bmatrix}.\begin{bmatrix}\frac{37}{24}&\frac{-52}{72}&0\\\frac{-52}{72}&\frac{73}{216}&0\\0&0&1\end{bmatrix} = \begin{bmatrix}\frac{14506}{15552}&\frac{-73}{46656}&3\\\frac{1363}{15552}&\frac{-1915}{46656}&6\\\frac{2674}{15552}&\frac{3757}{46656}&9\end{bmatrix}$

Table 4.1: Cipher text for the given input

Step 8: Send cipher text $\begin{bmatrix}\frac{14506}{15552}&-\frac{73}{46656}&3\\\frac{1396}{15552}&-\frac{1915}{46656}&6\\\frac{2674}{15552}&\frac{3757}{46656}&9\end{bmatrix}$ to the receiver.

Decryption Process

From the above table, we get the cipher text $\begin{bmatrix}\frac{14506}{15552}&-\frac{73}{46656}&3\\\frac{1396}{15552}&-\frac{1915}{46656}&6\\\frac{2674}{15552}&\frac{3757}{46656}&9\end{bmatrix}$. For decryption process we need to multiply the cipher text with the inverse of the recurrence matrix used. So, on getting the finite state machine and recurrence matrix (which is our secret key) we can decode the cipher text. For a finite state machine with n states, we require n multiplicative inverse matrix. In this way.

Decryption at q_i^{th} state = Cipher text$*B_2^{-x}$

Example 2: Let the plain text $P = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$

Let us encrypt the plain text $P = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$

Solution: Encryption Process

Step 1: $P = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$

Step 2: Moore machine is defined through public channel.

Step 3: Let the input binary key is = 10111.

Step 4: The output of the above the key is found by Moore machine.

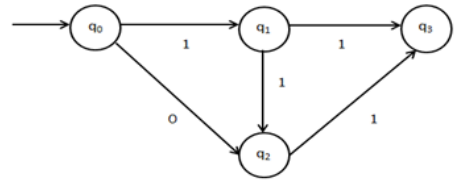

Figure 2: Moore Machine to calculate output for 10111

Step 5: The recurrence matrix is defined by extension of Lucas numbers.

$$L_2^x = \begin{bmatrix} L_{n-1} & L_n & 0 \\ L_n & L_{n+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ with initial conditions } L_0 = 2, L_1 = 1.$$

$$L_2^1 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$L_2^2 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}^2 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 5 & 0 \\ 5 & 10 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$L_2^3 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}^3 = L_2^2 \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 15 & 20 & 0 \\ 20 & 35 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 6: Assume the cipher text at q_{i+1} state is $q_{i+1} = q_0 + L_2^x$ ... (1)
where q_{i+1} = shows the output, q_i = shows the input, L_2^x = recurrence matrix key. By $eq^n(1)$.

Present State = Previous State $+L_2^x$
$q_1 = q_0 + L_2^1$
$q_2 = q_0 + L_2^2$
$q_1 = q_2 + L_2^1$
$q_3 = q_1 + L_2^3$
$q_2 = q_3 + L_2^2$

Table 5: Previous and Present state of the system.

Step 7: Now we compute cipher text at each level. Then the cipher text at each level is table 4.2.

S. No.	Input	Present State	Previous State	Key	Cipher Text
1.	1	q_1	q_0	L_2^1	$q_1 = q_0 + L_2^1 = \begin{bmatrix}9&8&7\\6&5&4\\3&2&1\end{bmatrix} + \begin{bmatrix}2&1&0\\1&3&0\\0&0&1\end{bmatrix} = \begin{bmatrix}11&9&7\\7&8&4\\3&2&2\end{bmatrix}$
2.	0	q_2	q_0	L_2^2	$q_2 = q_0 + L_2^2 = \begin{bmatrix}9&8&7\\6&5&4\\3&2&1\end{bmatrix} + \begin{bmatrix}5&5&0\\5&10&0\\0&0&1\end{bmatrix} = \begin{bmatrix}14&13&7\\11&15&4\\3&2&2\end{bmatrix}$
3.	1	q_1	q_2	L_2^1	$q_1 = q_2 + L_2^1 = \begin{bmatrix}14&13&7\\11&15&4\\3&2&2\end{bmatrix} + \begin{bmatrix}2&1&0\\1&3&0\\0&0&1\end{bmatrix} = \begin{bmatrix}16&14&7\\12&18&4\\3&2&3\end{bmatrix}$
4.	1	q_3	q_1	L_2^3	$q_3 = q_1 + L_2^3 = \begin{bmatrix}16&14&7\\12&18&4\\3&2&3\end{bmatrix} + \begin{bmatrix}15&20&0\\20&35&0\\0&0&1\end{bmatrix} = \begin{bmatrix}31&34&7\\32&53&4\\3&2&4\end{bmatrix}$
5.	1	q_2	q_3	L_2^2	$q_2 = q_3 + L_2^2 = \begin{bmatrix}31&34&7\\32&53&4\\3&2&4\end{bmatrix} + \begin{bmatrix}5&5&0\\5&10&0\\0&0&1\end{bmatrix} = \begin{bmatrix}36&39&7\\38&63&4\\3&2&5\end{bmatrix}$

Table 4.2: Cipher text for the given input

Step 8: Send cipher text $\begin{bmatrix} 36 & 39 & 7 \\ 38 & 63 & 4 \\ 3 & 2 & 5 \end{bmatrix}$ to the receiver.

Decryption Process

From the above table, we get the cipher text $\begin{bmatrix} 36 & 39 & 7 \\ 38 & 63 & 4 \\ 3 & 2 & 5 \end{bmatrix}$. For decryption process we need to add the cipher text with the inverse of the recurrence matrix used. So, on getting the secret key, finite state machine and recurrence matrix (which is our secret key) we can decode the cipher text. For a finite state machine with n states, we require additive inverse matrix. So,

$$\textbf{Decryption at } q_i^{th} \textbf{ state = Cipher text } + L_2^{-x}$$

5. Conclusion

In the proposed work we have developed new cryptographic algorithms using finite state machine, recurrence matrix (Bernoulli numbers and Lucas number) and operations of matrix multiplication and matrix addition. The proposed algorithm increases the secrecy of message which is being sent between sender and receiver over an insecure channel. Explicit forms of Bernoulli numbers and Lucas numbers are very large in size and therefore they are more secured. Also if we increase the size of the matrix and number of rounds, more information can be sent securely at a time. Using the algorithm, the secrecy is protected at 5 levels :The input, Finite state machine, Multiple matrix operation, Recurrence matrix, Different Cipher text at each stage.

References

[1] Gandhi Krishna B., Shekhar Chandra A., Srilakshmi S., Cryptography Scheme For Digital Signals Using Finite State Machine, International Journal of Computer Applications, 29(6) (2011), 61-63.

[2] Jyotirmie P. A., Shekhar Chandra A., Devi Uma S., Application of Mealy Machine And Recurrence Relations in Cryptography, International Journal of Engineering Research and Technology, 2(5) (2013), 1286-1290.

[3] Kahate Atul, Cryptography And Network Security, Tata McGraw Hill (2003).

[4] Linz Peter, An Introduction to Formal Languages and Automata, (4th Edition) Narosa Publishing, (2009).

[5] Mishra K. L. P., Chandrashekhran N., Theory of Computer Science, (3rd Edition) PHI Learning, (2014).

[6] Schneier Bruce, Ferguson Niels, Practical Cryptography, Wiley Publishing Inc., (2004).

[7] Stallings William, Cryptography and Network Security: Principles and Practices, (4th Edition) Prentice Hall, (2007).

[8] Sudha K. R., Shekhar Chandra A., Reddy Prasad, Cryptography Protection of Digital Signals Using Some Recurrence Relations, International Journal of Computer Science and Network Security 7(5) (2007), 203-207.

[9] Taş N., Uçar S., Özgür N. Y., Pell Coding and Pell Decoding Methods with Some Applications, Contributions to Discrete Mathematics, Vol. 15, No. 1 (2020), 52-66.

[10] Uçar S., Taş N. and Ozgur N. Y., A New Application to Coding Theory via Fibonacci and Lucas Numbers, Mathematical Sciences and Applications E-Notes, Vol. 7, No. 1 (2019), 62-70.

[11] Taş N., Uçar S., . Özgür N. Y, and Kaymak Ö., A new coding/decoding algorithm using fibonacci numbers, Discrete Mathematics, Algorithms and Applications, Vol. 10, No. 2 (2018).