CHAPTER-5

SUPER ENCRYPTION OF AFFINE CIPHER AND VIGENERE CIPHER WITH FIBONACCI-LUCAS MATRICES
CHAPTER-5

SUPER ENCRYPTION OF AFFINE CIPHER AND VIGENERE CIPHER WITH FIBONACCI-LUCAS MATRICES

5.1 Introduction:
Multiple encryptions in a practical system refer to encrypt the data more than once i.e., twice or thrice to increase the security levels. As long as the cipher is unbreakable the encryption schemes remains strong. In view of the known attacks encrypting the data more than once will strengthen the security levels. In this chapter we proposed a multiple encryption using Affine cipher with Fibonacci-Lucas transformation.

Fibonacci-Lucas [20],[25],[26] is a series of integers that both Fibonacci and Lucas numbers are inter relationship with each other by \( L_n = F_{n-1} + F_{n+1} \) for all integers \( n \) also \( F_n = L_{n-1} + L_{n+1} \) for all integers \( n \).

In general
\[
F_{n-k} + F_{n+k} = F_n L_k \text{ for all integers 'n' if } k \text{ is even}
\]
\[
F_{n-k} + F_{n+k} = F_k L_n \text{ for all integers 'n' if } k \text{ is odd}
\]
Due to this inter relationship of Fibonacci, Lucas numbers they played vital role in constructing privative and public key cryptosystems.

In chapter-4, we discussed the multiple encryption of Affine cipher and Vigenere cipher with Pell, Lucas numbers using multiple ciphers.

In this chapter we introduced the super encryption of Affine cipher and Vigenere cipher with Fibonacci-Lucas the matrices by
the method of multi-level level encryption using Fibonacci-Lucas cipher.

5.2 Key Generation of Affine cipher, Vigenere cipher with Fibonacci matrices

Let A(Alice) and B(Bob) are two communicating parties. For multiple encryptions initially both the parties agree upon Fibonacci-Lucas matrix of order n and the Affine cipher or Vigenere cipher. The value of n and the values of a,b of the Affine cipher or Vigenere key will be sent to B as the private keys.

1\textsuperscript{st} level of encryption: A encrypt the plaintext P=(p\textsubscript{1}, p\textsubscript{2},p\textsubscript{3},... p\textsubscript{n}) with Fibonacci-Lucas matrices with key K\textsubscript{1} of order n. A gets the first encrypted message C\textsubscript{1} and A sends (the key K\textsubscript{1} through the secure channel) P×(FL)^n=C\textsubscript{1} to B.

2\textsuperscript{nd} level of encryption or Super-encryption: As a second layer of encryption, A encrypts the first encrypted message C\textsubscript{1} as super-encryption with Affine cipher E(x)=(ax+b) mod26 or Vigenere cipher with key K\textsubscript{2} (the values of a,b).

1\textsuperscript{st} level of decryption: B decrypts the Super-encrypted message C\textsubscript{2} by using the inverse of a,b in K\textsubscript{2}, which is the inverse of affine cipher or using reverse offset rule with Vingere key to get the first decrypted message P\textsubscript{1}

2\textsuperscript{nd} level of decryption: Again B decrypts the first decrypted message P\textsubscript{1} using K\textsubscript{1} which is the inverse Fibonacci-Lucas matrix of order n to get plaintext message P.
5.3 Algorithm:

Algorithm of enciphering:

Step-1: Let the plaintext be P = p₁ p₂, p₃ ... pₘ.

Step-2: A computes P×(FL)ⁿ and gets 1st ciphertext C₁.

Step-3: Now, again A performs super-encryption with the Affine cipher or Vigenere cipher to get super-encrypted message C₂.

Step-4: A sends the super-encrypted message C₂ to B.

Algorithm of deciphering:

Step-1: B receives the super-encrypted message C₂.

Step-2: B decrypts the super-encrypted message C₂ by using $E^{-1}(y) = a^{-1}(y-b) \mod 26$ or reverse offset rule with Vigenere key to obtain the first decrypted message P₁.

Step-3: B computes P₁×(FL)⁻¹ to obtain the original plaintext message P.

Encryption

K₁ is Fibonacci Lucas matrix  
K₂: Affine cipher keys with a and b or Vigenere key
**5.4 Example:**

**Case-1:** For key $n=1$ we get $FL = \begin{pmatrix} F_1 & F_2 \\ L_1 & L_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$

**Encryption:**

Step-1: Let the Plaintext be $P = \begin{pmatrix} H & A \\ C & K \end{pmatrix} = \begin{pmatrix} 7 & 0 \\ 2 & 10 \end{pmatrix}$

Step-2: A computes $P \times (FL)^1$ to get first encrypted message as $C_1$

\[
\begin{pmatrix} 7 & 0 \\ 2 & 10 \end{pmatrix} \times \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 7 & 7 \\ 22 & 12 \end{pmatrix}
\]

First encrypted message as $C_1 = \begin{pmatrix} 7 & 7 \\ 22 & 12 \end{pmatrix}$

Step-3: Now A applies super-encryption with the Affine cipher $E(x) = (ax+b) \mod 26$ for keys are $a = 5$ & $b = 25$ to get the second encryption message as $C_2$. 
Step-3: A sends super-encrypted message C₂ to B as IIFH

**Decryption:**

Step-1: B receives super-encrypted message as IIFH

Step-2: B use inverse Affine cipher $E^{-1}(y) = a^{-1}(y-b) \mod 26$ to super encrypted message to get the first decrypted message as $P₁$.

<table>
<thead>
<tr>
<th>Message</th>
<th>I</th>
<th>I</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>$y-25$</td>
<td>-17</td>
<td>-17</td>
<td>-20</td>
<td>-18</td>
</tr>
<tr>
<td>$21(y-25)$</td>
<td>-357</td>
<td>-357</td>
<td>-420</td>
<td>-378</td>
</tr>
<tr>
<td>$21(y-25) \mod 26$</td>
<td>7</td>
<td>7</td>
<td>22</td>
<td>12</td>
</tr>
</tbody>
</table>

First Decrypted text as $P₁ = \begin{pmatrix} H & H \\ W & M \end{pmatrix} = \begin{pmatrix} 7 & 7 \\ 22 & 12 \end{pmatrix}$

Step-3: Again B computes $P₁ \times (FL)^{-1}$ to get original message as $P$.

\[
\begin{pmatrix} 7 & 7 \\ 22 & 12 \end{pmatrix} \times \begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix} = \begin{pmatrix} 7 & 0 \\ 2 & 10 \end{pmatrix}
\]

<table>
<thead>
<tr>
<th>Mod 26</th>
<th>7</th>
<th>0</th>
<th>2</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second Decrypted message</td>
<td>H</td>
<td>A</td>
<td>C</td>
<td>K</td>
</tr>
</tbody>
</table>

B gets the plaintext message $P = \begin{pmatrix} H & A \\ C & K \end{pmatrix}$

**Case-2:** For key $n=2$ we get $FL = \begin{pmatrix} F₂ & F₃ \\ L₂ & L₃ \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix}$

**Encryption:**
Step-1: Let the Plaintext be \( P = \begin{pmatrix} H & A \\ C & K \end{pmatrix} = \begin{pmatrix} 7 & 0 \\ 2 & 10 \end{pmatrix} \)

Step-2: A computes \( P \times (FL) \) to get first encrypted message as \( C_1 \)

\[
\begin{pmatrix} 7 & 0 \\ 2 & 10 \end{pmatrix} \times \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 7 & 14 \\ 12 & 34 \end{pmatrix}
\]

First encrypted message as \( C_1 = \begin{pmatrix} 7 & 14 \\ 12 & 34 \end{pmatrix} \)

Step-3: Now A applies super-encryption with the Affine cipher \( E(x) = (ax+b) \mod 26 \) for keys are \( a = 5 \) & \( b = 30 \) to get the second encryption message as \( C_2 \).

\[
\begin{array}{c|c|c|c}
 x & 7 & 14 & 12 \\
\hline
 5x+30 & 65 & 100 & 90 \\
\hline
 (5x+30) \mod 26 & 13 & 22 & 12 \\
\hline
 Second Encrypted message & N & W & M & S \\
\end{array}
\]

Step-3: A sends super-encrypted message \( C_2 \) to B as NWMS

**Decryption:**

Step-1: B receives super-encrypted message as NWMS

Step-2: B use inverse Affine cipher \( E^{-1}(y) = a^{-1}(y-b) \mod 26 \) to superencrypted message to get the first decrypted message as \( P_1 \).

\[
\begin{array}{c|c|c|c|c}
 Message & N & W & M & S \\
\hline
 y & 13 & 22 & 12 & 18 \\
\hline
 y-30 & -17 & -8 & -18 & -12 \\
\hline
 21(y-30) & -357 & -168 & -378 & -252 \\
\hline
 21(y-30) \mod 26 & 7 & 14 & 12 & 8 \\
\hline
 First Decrypted text & H & O & M & I \\
\end{array}
\]

First decrypted text as \( P_1 = \begin{pmatrix} H & O \\ M & I \end{pmatrix} = \begin{pmatrix} 7 & 14 \\ 12 & 8 \end{pmatrix} \)
Step-3: Again B computes $P_1 \times (FL)^{-1}$ to get original message as $P$.

\[
\begin{pmatrix} 7 & 14 \\ 12 & 8 \end{pmatrix} \times \begin{pmatrix} 3 & -2 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 7 & 0 \\ 28 & -16 \end{pmatrix}
\]

<table>
<thead>
<tr>
<th>Mod 26</th>
<th>7</th>
<th>0</th>
<th>28</th>
<th>-16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second Decrypted message</td>
<td>H</td>
<td>A</td>
<td>C</td>
<td>K</td>
</tr>
</tbody>
</table>

B gets the plaintext message $P = \begin{pmatrix} H & A \\ C & K \end{pmatrix}$

**Case-3:** For key $n=3$ we get $FL = \begin{pmatrix} F_3 & F_4 \\ L_3 & L_4 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}$

**Encryption:**

Step-1: Let the Plaintext be $P = \begin{pmatrix} H & A \\ C & K \end{pmatrix} = \begin{pmatrix} 7 & 0 \\ 2 & 10 \end{pmatrix}$

Step-2: A computes $P \times (FL)$ to get first encrypted message as $C_1$

\[
\begin{pmatrix} 7 & 0 \\ 2 & 10 \end{pmatrix} \times \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 14 & 21 \\ 34 & 46 \end{pmatrix}
\]

First encrypted message as $C_1 = \begin{pmatrix} 14 & 21 \\ 34 & 46 \end{pmatrix}$

Step-3: Now A applies super-encryption with the Affine cipher $E(x) = (ax + b) \mod 26$ for keys $a = 5$ & $b = 29$ to get the second encryption message as $C_2$.

<table>
<thead>
<tr>
<th>x</th>
<th>14</th>
<th>21</th>
<th>34</th>
<th>46</th>
</tr>
</thead>
<tbody>
<tr>
<td>$5x+29$</td>
<td>99</td>
<td>134</td>
<td>199</td>
<td>259</td>
</tr>
<tr>
<td>$(5x+29) \mod 26$</td>
<td>21</td>
<td>4</td>
<td>17</td>
<td>25</td>
</tr>
</tbody>
</table>
Step-3: A sends super-encrypted message C₂ to B as VERZ

**Decryption:**

Step-1: B receives super-encrypted message as VERZ

Step-2: B use inverse Affine cipher $E^{-1}(y)=a^{-1}(y-b)\mod 26$ to super encrypted message to get the first decrypted message as $P₁$.

<table>
<thead>
<tr>
<th>Message</th>
<th>V</th>
<th>E</th>
<th>R</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>21</td>
<td>4</td>
<td>17</td>
<td>25</td>
</tr>
<tr>
<td>y-29</td>
<td>-8</td>
<td>-25</td>
<td>-12</td>
<td>-4</td>
</tr>
<tr>
<td>21(y-29)</td>
<td>-168</td>
<td>-525</td>
<td>-252</td>
<td>-84</td>
</tr>
<tr>
<td>21(y-29) \mod 26</td>
<td>14</td>
<td>21</td>
<td>8</td>
<td>20</td>
</tr>
</tbody>
</table>

First decrypted text as $P₁ = \begin{pmatrix} O & V \\ I & U \end{pmatrix} = \begin{pmatrix} 14 & 21 \\ 8 & 20 \end{pmatrix}$

Step-3: Again B computes $P₁ \times (FL)^{-1}$ to get original message as $P$.

$$\begin{pmatrix} 14 & 21 \\ 8 & 20 \end{pmatrix} \times \begin{pmatrix} -4 & 3 \\ 3 & -2 \end{pmatrix} = \begin{pmatrix} 7 & 0 \\ 28 & -16 \end{pmatrix}$$

$$\begin{array}{cccc} 7 & 0 & 28 & -16 \\ \text{Mod 26} & 7 & 0 & 2 & 10 \\ \text{Second Decrypted message} & H & A & C & K \end{array}$$

B gets the plaintext message $P = \begin{pmatrix} H & A \\ C & K \end{pmatrix}$

**Example for Vigenere cipher**

**Case: 1** For key n=1 we get $FL = \begin{pmatrix} F₁ & F₂ \\ L₁ & L₂ \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$

**Encryption:**
Step-1: Let the Plaintext be \( P = \begin{pmatrix} R & A \\ M & U \end{pmatrix} = \begin{pmatrix} 17 & 0 \\ 12 & 20 \end{pmatrix} \)

Step-2: A computes \( P \times (FL) \) to get the first encrypted message as \( C_1 \)

\[
\begin{pmatrix} 17 & 0 \\ 12 & 20 \end{pmatrix} \times \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 17 & 17 \\ 52 & 32 \end{pmatrix}
\]

First encrypted message as \( C_1 = \begin{pmatrix} 17 & 17 \\ 52 & 32 \end{pmatrix} \)

A using vigenere key as second key

\[
\begin{array}{cccc}
P & A & S & S \\
15 & 0 & 18 & 18 \\
\end{array}
\]

Step-3: Again A applies super-encryption with vigenere key using offset rule to the first encrypted message to get the second encrypted message as \( C_2 \).

\[
\begin{array}{cccc}
17 & 17 & 52 & 32 \\
17 & 17 & 32 & 12 \\
+ & + & + & + \\
15 & 0 & 18 & 18 \\
32 & 17 & 70 & 50 \\
6 & 17 & 18 & 24 \\
G & R & S & Y \\
\end{array}
\]

Step-4: A send super-encrypted message \( C_2 \) to B as GRSY

**Decryption:**

Step-1: B receives super-encrypted message \( C_2 \) as GRSY
Step-2: B use reverse offset rule with vigenere key to getting the first decrypted message as $P_1$.

<table>
<thead>
<tr>
<th>Message</th>
<th>G</th>
<th>R</th>
<th>S</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>17</td>
<td>18</td>
<td>24</td>
</tr>
<tr>
<td>Reverse offset rule with key</td>
<td>6</td>
<td>17</td>
<td>18</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>-9</td>
<td>17</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Mod 26</td>
<td>17</td>
<td>17</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

| First Decryption message | R | R | A | G |

First decrypted text as $P_1 = \begin{pmatrix} R & R \\ A & G \end{pmatrix} = \begin{pmatrix} 17 & 17 \\ 0 & 6 \end{pmatrix}$

Step-3: Again B computes $P_1 \times (FL)^{-1}$ to get the original message as $P$.

$$\begin{pmatrix} 17 & 17 \\ 0 & 6 \end{pmatrix} \times \begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix} = \begin{pmatrix} 17 & 0 \\ 12 & -6 \end{pmatrix}$$

<table>
<thead>
<tr>
<th>Mod 26</th>
<th>17</th>
<th>0</th>
<th>12</th>
<th>-6</th>
</tr>
</thead>
</table>

| Second Decrypted message | R | A | M | U |

B gets the plaintext message $P = \begin{pmatrix} R & A \\ M & U \end{pmatrix}$

Case-2: For key $n=2$ we get $FL = \begin{pmatrix} F_2 & F_3 \\ L_2 & L_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix}$

Encryption:
Step-1: Let the Plaintext be \( P = \begin{pmatrix} R & A \\ M & U \end{pmatrix} = \begin{pmatrix} 17 & 0 \\ 12 & 20 \end{pmatrix} \)

Step-2: A computes \( P \times (FL) \) to get the first encrypted message as \( C_1 \)

\[
\begin{pmatrix} 17 & 0 \\ 12 & 20 \end{pmatrix} \times \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 17 & 34 \\ 32 & 84 \end{pmatrix}
\]

First encrypted message as \( C_1 = \begin{pmatrix} 17 & 34 \\ 32 & 84 \end{pmatrix} \)

A using Vigenre key as second key

\[
\begin{array}{cccc}
F & A & I & L \\
5 & 0 & 8 & 11 \\
\end{array}
\]

Step-3: B applies super-encryption with vigenere key using offset rule to the first encrypted message to get the second encrypted message as \( C_2 \).

<table>
<thead>
<tr>
<th></th>
<th>17</th>
<th>34</th>
<th>32</th>
<th>84</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset rule with key</td>
<td>17</td>
<td>34</td>
<td>42</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>34</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>Mod 26</td>
<td>22</td>
<td>8</td>
<td>24</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>22</th>
<th>34</th>
<th>50</th>
<th>95</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second Encrypted message</td>
<td>( W )</td>
<td>( I )</td>
<td>( Y )</td>
<td>( R )</td>
</tr>
</tbody>
</table>

Step-4: A send super-encrypted message \( C_2 \) to B as WIYR

**Decryption:**

Step-1: B receives super-encrypted message as WIYR
Step-2: B use reverse offset rule with vigenere key to get the first decrypted message as $P_1$.

<table>
<thead>
<tr>
<th>Message</th>
<th>W</th>
<th>I</th>
<th>Y</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>22</td>
<td>8</td>
<td>24</td>
<td>17</td>
</tr>
</tbody>
</table>

Reverse offset rule with key

<table>
<thead>
<tr>
<th></th>
<th>22</th>
<th>8</th>
<th>24</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>8</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mod 26</th>
<th>17</th>
<th>8</th>
<th>6</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Decryption message</td>
<td>I</td>
<td>R</td>
<td>S</td>
<td>G</td>
</tr>
</tbody>
</table>

First decrypted text as $P_1 = \begin{pmatrix} R & I \\ G & G \end{pmatrix} = \begin{pmatrix} 17 & 8 \\ 6 & 6 \end{pmatrix}$

Step-3: Again B computes $P_1 \times (FL)^{-1}$ to get the original message as P.

$\begin{pmatrix} 17 & 8 \\ 6 & 6 \end{pmatrix} \times \begin{pmatrix} 3 & -2 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 43 & -26 \\ 12 & -6 \end{pmatrix}$

<table>
<thead>
<tr>
<th></th>
<th>43</th>
<th>-26</th>
<th>12</th>
<th>-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mod 26</td>
<td>17</td>
<td>0</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>Second Decrypted message</td>
<td>R</td>
<td>A</td>
<td>M</td>
<td>U</td>
</tr>
</tbody>
</table>

B gets the plaintext message $P = \begin{pmatrix} R & A \\ M & U \end{pmatrix}$

Case-3: For key $n=3$ we get $FL = \begin{pmatrix} F_3 & F_4 \\ L_3 & L_4 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}$

**Encryption:**
Step-1: Let the Plaintext be \( P = \begin{pmatrix} R & A \\ M & U \end{pmatrix} = \begin{pmatrix} 17 & 0 \\ 12 & 20 \end{pmatrix} \)

Step-2: A computes \( P \times (FL) \) to get the first encrypted message as \( C_1 \)

\[
\begin{pmatrix} 17 & 0 \\ 12 & 20 \end{pmatrix} \times \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 34 & 51 \\ 84 & 116 \end{pmatrix}
\]

First encrypted message as \( C_1 = \begin{pmatrix} 34 & 51 \\ 84 & 116 \end{pmatrix} \)

A using vigenere key as second key

\[
\begin{array}{cccc}
L & O & S & S \\
11 & 14 & 18 & 18 \\
\end{array}
\]

Step-3: Again A applies super-encryption with vigenere key using offset rule to the first encrypted message to get the second encrypted message as \( C_2 \).

\[
\begin{array}{cccc}
34 & 51 & 84 & 116 \\
\hline
34 & 51 & 84 & 116 \\
+ & + & + & + \\
11 & 14 & 18 & 18 \\
\hline
45 & 65 & 102 & 134 \\
\hline
19 & 13 & 24 & 4 \\
\hline
T & N & Y & E \\
\end{array}
\]

Step-4: A send super-encrypted message \( C_2 \) to B as TNYE

**Decryption:**
Step-1: B receives super-encrypted message as TNYE
Step-2: B use reverse offset rule with vigenere key to getting the first decrypted message as $P_1$.

<table>
<thead>
<tr>
<th>Message</th>
<th>T</th>
<th>N</th>
<th>Y</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>19</td>
<td>13</td>
<td>24</td>
<td>4</td>
</tr>
</tbody>
</table>

Reverse offset rule with key

<table>
<thead>
<tr>
<th>Key</th>
<th>19</th>
<th>-11</th>
<th>13</th>
<th>-14</th>
<th>24</th>
<th>-18</th>
<th>4</th>
<th>-18</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
<td>-1</td>
<td>6</td>
<td>-14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mod 26

<table>
<thead>
<tr>
<th>Mod 26</th>
<th>8</th>
<th>25</th>
<th>6</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Decryption message</td>
<td>I</td>
<td>Z</td>
<td>G</td>
<td>M</td>
</tr>
</tbody>
</table>

First decrypted text as $P_1 = \begin{pmatrix} I \\ G \\ M \end{pmatrix} = \begin{pmatrix} 8 \\ 25 \\ 6 \\ 12 \end{pmatrix}$

Step-3: Again B computes $P_1 \times (FL)^{-1}$ to get the original message as $P$.

\[
\begin{pmatrix} 8 & 25 \\ 6 & 12 \end{pmatrix} \times \begin{pmatrix} -4 & 3 \\ 3 & -2 \end{pmatrix} = \begin{pmatrix} 43 & -26 \\ 12 & -6 \end{pmatrix}
\]

<table>
<thead>
<tr>
<th></th>
<th>43</th>
<th>-26</th>
<th>12</th>
<th>-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mod 26</td>
<td>17</td>
<td>0</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>Second Decrypted message</td>
<td>R</td>
<td>A</td>
<td>M</td>
<td>U</td>
</tr>
</tbody>
</table>

B gets the plaintext message $P = \begin{pmatrix} R \\ M \\ A \\ U \end{pmatrix}$
5.5 Result Analysis:

For the chosen plaintext the space for memory occupation and time for executing the encryption were calculated using matlab software.

Result analysis of Fibonacci-Lucas matrix with Affine cipher

<table>
<thead>
<tr>
<th>S.No</th>
<th>First encryption</th>
<th>Super encryption</th>
<th>Space complexity</th>
<th>First encryption time</th>
<th>Super encryption Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fibonacci-Lucas matrix</td>
<td>Affine</td>
<td>60 bytes</td>
<td>8 MILLI SEC</td>
<td>11 MILLI SEC</td>
</tr>
</tbody>
</table>

Result analysis of Fibonacci-Lucas matrix with Vigenere cipher

<table>
<thead>
<tr>
<th>S.No</th>
<th>First encryption</th>
<th>Super encryption</th>
<th>Space complexity</th>
<th>First encryption time</th>
<th>Super encryption Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fibonacci-Lucas matrix</td>
<td>Vigenere</td>
<td>60 bytes</td>
<td>7 MILLI SEC</td>
<td>10 MILLI SEC</td>
</tr>
</tbody>
</table>

5.6 Conclusions: For this constructed cryptosystem the time complexity for encryption and decryption are same but is more secure than the symmetric cryptosystems with Fibonacci, Lucas, Pell numbers we can exiled this concept to public key cryptosystem also.