ENCRYPTION AND DECRYPTION TECHNIQUE USING GRAPH THEORY

L. Vinokthkumar, V. Balaji
1Research scholar, 2Assistant professor
1,2Department of Mathematics, Sacred Heart College, Tirupattur - 635601, Vellore District, Tamil Nadu, S.India.

Abstract: In this paper, we provided an numerical example for encryption and decryption technique using complete graph. Finally, we used matrix properties to decrypt the received message.

Index Terms - Encryption, Decryption, Adjacency matrix, Complete graph and Hamiltonian path.

2010 AMS Subject classification : 05C78.

I. INTRODUCTION

The information needs to be secured from attack in this competitive world. The process of encryption consists of an algorithm and a key. The key is the value independent of the plaintext. The algorithm will produce a different output depending on the specific key being used at the time. Once the ciphertext is produced, it may be transmitted upon reception, the ciphertext can be transformed back to the original plaintext by using a decryption algorithm and the same key that was used for decryption. The following preliminaries are used in this paper.

II. PRELIMINARIES

Definition 2.1 Walk A walk in a graph G is a non-empty alternating sequence \( v_0e_0v_1, \ldots, e_{k-1}v_k \) of vertices and edges in G, such that \( e_i = \{ v_i, v_{i+1} \} \quad \forall \quad i < k \).

Definition 2.2 Path A path is a walk that visits each vertex atmost once.

Definition 2.3 Hamiltonian path A path in G that contains every vertex of G is said to be hamilton path.

Definition 2.4 Complete graph A simple graph in which every two vertices are adjacent is called a complete graph. A complete graph with n vertices is denoted by \( K_n \). The graph \( K_n \) has n-vertices and \( (n(n-1)/2) \) edges.

Definition 2.5 Weighted graph A graph G with numerical labels on the edges is said to be weighted graph.

Definition 2.6 Adjacency matrix Let \( G = (V, X) \) be a \( (p, q) \) graph. Let \( V = \{v_1, v_2, \ldots, v_p\} \). Then \( p \times p \) matrix \( A = (a_{ij}) \).

Where \( a_{ij} = \begin{cases} 1, & \text{if } v_i, v_j \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases} \)

III. ALGORITHMS

3.1 Encryption Algorithm

Step 1: Encode the message using encoding chart.
Step 2: Apply the encoded information in the graph.
Step 3: If the encoded information contains n elements then we need \( (n-1) \) nodes.
Step 4: Choose n-nodes in a hamilton path and label the beginning point as 1.
Step 5: Assign the weight on its edges, which is computed in step 1 using the encoding chart.
Step 6: Then form the adjacency matrix using the hamilton path and name the matrix.
Step 7: Now, transform the hamiltonian path into a complete graph by joining each of the nodes.
Step 8: Assign false weighted information to the new links.
Step 9: Again form the adjacency matrix using complete graph and name the matrix.
Step 10: Now, multiply the sender matrix with a key matrix. So, the data will not go in a wrong way.
Step 11: Encrypted data is produced in the final matrix.
Step 12: Finally, we decide to send the final matrix and the adjacency matrix to the receiver in a linear format. Either can be in a row or columnwise, but separated by the space.

i.e., <a> <n> <final matrix data> <adjacency matrix of \( K_n \)>
3.2 Decryption Algorithm

Step 1: Form the corresponding matrix by reading the encryption data.
Step 2: Then multiply the inverse of key matrix with the information matrix and name the matrix as A.
Step 3: Obtain the inverse matrix for complete graph.
Step 4: Then matrix A is multiplied with it.
Step 5: Form a hamiltonian path after multiplying from the matrix.
Step 6: Label the length of the edge from its beginning point to end.
Step 7: Decode the message using encoding chart table in order to generate the original message.

Encoding chart

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

IV. Worked Examples

Suppose we are sending the message as SUPER. Let us replace the message by 19, 21, 16, 5, 18 as in encoding system. Now, label that numerical values on a hamiltonian path with 6 vertices as shown in the following graph.

NOTE: We have used R-Programming for matrix multiplication.

Let \( D_i \) be the adjacency matrix for the above graph.

\[
D_i = \begin{bmatrix}
0 & 19 & 0 & 0 & 0 & 0 \\
19 & 0 & 21 & 0 & 0 & 0 \\
0 & 21 & 0 & 16 & 0 & 0 \\
0 & 0 & 16 & 0 & 5 & 0 \\
0 & 0 & 0 & 5 & 0 & 18 \\
0 & 0 & 0 & 0 & 18 & 0 \\
\end{bmatrix}
\]

Since the Hamilton path has 6 nodes.

By joining each vertex, we let form the complete graph \( K_6 \). We may assign false weight to the remaining links of \( K_6 \) graph.
Let $D_2$ be the adjacency matrix for the $K_6$ graph.

$$D_2 = \begin{bmatrix}
0 & 19 & 22 & 10 & 9 & 1 \\
19 & 0 & 21 & 24 & 23 & 7 \\
22 & 21 & 0 & 16 & 3 & 12 \\
10 & 24 & 16 & 0 & 5 & 17 \\
9 & 23 & 3 & 5 & 0 & 18 \\
1 & 7 & 12 & 17 & 18 & 0
\end{bmatrix}$$

Let $B = D_1D_2 = \begin{bmatrix}
361 & 0 & 399 & 456 & 437 & 133 \\
462 & 802 & 418 & 526 & 234 & 271 \\
559 & 384 & 697 & 504 & 563 & 419 \\
397 & 451 & 15 & 281 & 48 & 282 \\
68 & 246 & 296 & 306 & 349 & 85 \\
162 & 414 & 54 & 90 & 0 & 324
\end{bmatrix}$

Let us choose the key matrix $A$ as, $A = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1
\end{bmatrix}$
Then,\
\[
BA = \begin{bmatrix}
1786 & 1425 & 1425 & 1026 & 570 & 133 \\
2713 & 2251 & 1449 & 1031 & 505 & 271 \\
3126 & 2567 & 2183 & 1486 & 982 & 419 \\
1474 & 1077 & 626 & 611 & 330 & 282 \\
1350 & 1282 & 1036 & 740 & 434 & 85 \\
1044 & 882 & 468 & 414 & 324 & 324
\end{bmatrix}
\]

Finally, we decide to send the final matrix and the adjacency matrix to the receiver in a linear format as row-wise.

\[
1 6 1786 1425 1425 1026 570 133 2713 2251 1449 1031 505 271 3126 2567 2183 1486 982 419 1474 1077 626 611 330 282 1350 1282 1036 740 434 85 1044 882 468 414 324 324
\]

Form the corresponding matrix by reading the encryption data.

\[
D_2 = \begin{bmatrix}
0 & 2 & 22 & 10 & 9 & 1 \\
2 & 0 & 18 & 24 & 23 & 7 \\
22 & 18 & 0 & 1 & 3 & 12 \\
10 & 24 & 1 & 0 & 330 & 282 \\
1350 & 1282 & 1036 & 740 & 434 & 85 \\
1044 & 882 & 468 & 414 & 324 & 324
\end{bmatrix}
\]

and

\[
BA = \begin{bmatrix}
148 & 144 & 144 & 108 & 60 & 14 \\
1096 & 700 & 372 & 328 & 290 & 218 \\
1398 & 1352 & 1328 & 1003 & 571 & 143 \\
798 & 650 & 310 & 268 & 71 & 68 \\
1088 & 944 & 580 & 518 & 450 & 238 \\
212 & 176 & 84 & 72 & 16 & 16
\end{bmatrix}
\]

Suppose the data received is,

\[
1 6 148 144 108 60 14 1096 700 372 328 290 218 1398 1352 1328 1003 571 143 798 650 310 268 71 68 1088 944 580 518 450 238 212 176 84 72 16 16 0 2 22 10 9 1 2 0 18 24 23 7 22 18 0 1 3 12 10 24 1 0 14 17 9 23 3 14 17 9 23 3 14 0 4 1 7 2 12 17 4 0
\]

\[
A^{-1} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & -1 & 1
\end{bmatrix}
\]

By using inverse of key matrix A we have,

\[
B = BAA^{-1} = \begin{bmatrix}
4 & 0 & 36 & 48 & 46 & 14 \\
396 & 328 & 44 & 38 & 72 & 218 \\
46 & 24 & 325 & 432 & 428 & 143 \\
148 & 340 & 42 & 197 & 3 & 68 \\
144 & 364 & 62 & 68 & 212 & 238 \\
36 & 92 & 12 & 56 & 0 & 16
\end{bmatrix}
\]
\[
D_2^{-1} = \begin{bmatrix}
0.061 & 0.064 & 0.038 & -0.087 & 0.126 & -0.233 \\
0.064 & 0.022 & -0.047 & -0.014 & 0.146 & -0.186 \\
0.038 & -0.047 & 0.019 & 0.020 & -0.069 & 0.100 \\
-0.087 & -0.014 & 0.020 & 0.017 & -0.087 & 0.200 \\
0.126 & 0.146 & -0.069 & -0.086 & 0.291 & -0.515 \\
-0.233 & -0.186 & 0.100 & 0.199 & -0.515 & 0.818
\end{bmatrix}
\]

So, the decoded message is \(2,18,1,14,4\). Then, we may track the spanning tree from the above matrix and the decoded message is \(2,18,1,14,4\).

\[
D_1 = D_2D_2D_2^{-1} = \begin{bmatrix}
0 & 2 & 0 & 0 & 0 & 0 \\
2 & 0 & 18 & 0 & 0 & 0 \\
0 & 18 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 14 & 0 \\
0 & 0 & 0 & 14 & 0 & 4 \\
0 & 0 & 0 & 0 & 4 & 0
\end{bmatrix}
\]

Hence, the original plain text is BRAND. Therefore, we decrypt the message as BRAND.
V. CONCLUSION

In this method, the data we send is encrypted twice for security purpose. Encrypting first by using the hamiltonian path and second by using the complete graph. We used key matrix for better decryption. Decryption becomes more difficult, if we fail to track the spanning tree of graph. Similarly, encryption can be done in the same way for larger words and all the data’s can be transmitted.

VI. ACKNOWLEDGMENT

The corresponding author (Dr. V. Balaji) for financial assistance No. FMRP5766/15 (SERO/UGC).

REFERENCES

[1] Harary, Graph theory, Narosa publishing House.