Information Sharing By Spanning Trees of a Graph By Labelled Sequence as A Key

P. Bala Manoj Kumar, K. Sai Teja, A. Manimaran, G. Deepa, B. Praba

Abstract: Keeping the Information Sharing to manage cyber risks as a key, Security Intelligence speaks about secured data every day in today’s world. Hence it is considered that Cyber security is a Data Analytics challenge. For this reason, many researchers were effectively working on privacy protection and Liability protection. As a supporting hand for these global issues to secure data transfer, we propose a method to encrypt and decrypt the messages instantly by spanning trees of a graph with labelled tree sequences.

Keywords: encryption; decryption; spanning tree; labelled sequence; algorithm

I. INTRODUCTION

It is observed that there is exponential increase in cyber-attacks is making effective need for encrypting data. This is the reason that many researches are going through their way to encrypt and decrypt the information. This makes need of effective encryption and decryption algorithms. To make familiarity between encryption and decryption, the following terms must be known and they are cipher, encoding and decoding. Where the cipher is the conversation of information data into a particular form, that cannot be understood by unauthorized people. Decryption is the process of bringing the cipher to its back state which can be understood. Various authors like Guo et al. [1], Rathore et al. [2] Yang et al. [3] contributed their idea towards the visual cryptography in assorted aspects in 2017. Numerous ways are there to transfer the information more securely using cryptosystem. Manimaran et al. [5, 6, 7, 8] described those secure transformation more elaborately in 2015. Obaida [9] introduced about complex cryptosystem in 2013. The author Zimmerman [10] publicized the basic ideas of cryptography.

II. DEFINITIONS [4]

A. Graph (G)
A Graph G = (V, E) is said to be a graph where V is the no vertices and E is no of edges between two or more vertices.

B. Tree
A tree is a connected graph without any circuits (A graph must have one vertex to be a tree)

C. Spanning Tree
If a tree ‘T’ is a subgraph of ‘G’ and ‘T’ contains all vertices of ‘G’ then ‘T’ is a spanning tree.

D. Labelled Sequence
A tree with n vertices has n-2 digitized sequence as its labelled sequence as shown in fig 2 in methodology.

III. METHODOLOGY

In this paper w proposed labelled sequence of a spanning tree as a key which is purely based on graph theory applications. The detailed procedure regarding the spanning trees, and its labelled sequence is explained in the methodology after this section. The graph to be drawn is first selected i.e., the graph with 5 vertices or even more. Then after the selection of number of vertices we will come to know that how many spanning trees can be drawn by Cayley Theorem. The number of spanning trees for a Graph(G) with n vertices can be known by the Cayley theorem restricted to constraint that n>=2 is n^n-2.

For an example, consider a graph with 4 vertices, By Cayley, we get n^n-2 = 4^2 (4^2 = 16 spanning trees). In specific if need to know that how many trees with n-points and r-end points. This can be known by the formula shown below

L(n, r) = \[ \sum_{j=0}^{r-2} (-1)^{r-j} \binom{n-r}{j} (n-r-j) \]

Assume for graph with n=4 (n>=3 must be followed for this theorem)

L(4, 2) = \[ \frac{4}{2} \sum_{i=0}^{1} (-1)^{i} \binom{2}{i} (2-i)^2 \]

= \[ \frac{4}{2} \left\{ (-1)^{0} \binom{2}{0} (2-0)^2 + (-1)^{1} \binom{2}{1} (2-1)^2 \right\} \]

= 6 [4-2] = 12

L(4, 3) = \[ \frac{4}{1} \sum_{i=0}^{1} (-1)^{i} \binom{1}{i} (1-i)^1 \]

= \[ \frac{4}{1} \left\{ (-1)^{0} \binom{1}{0} (1-0)^1 + (-1)^{1} \binom{1}{1} (1-1)^1 \right\} \]

= 4

Therefore, this shows the equal result generated by both Cayley Theorem and by proof given in paper i.e., Let L(n, r) be the number of labelled trees with n ≥ 3 points and r end-points then
Information Sharing by Spanning Trees of a Graph by labelled sequence as a key

\[ L(n, r) = \binom{n}{n-r} \sum_{i=0}^{r-1} (-1)^i \binom{n-r-i}{i} \]

Hence, we get spanning trees when n=4 like Figure 1 shows the no of graphs that we get for r=2, 3 end points with n=4 vertices of a tree.

Similarly, we get the for a graph with n=5, 6, 7, …. Since, we need to have the alphabets from A-Z (26 letters), 0-9 (10 digits) and the special characters like !, @, $, ^, &, *, {, }, :, <, > (we get 12 special characters).

This shows we have to take 48 spanning trees as possible. Hence, we get 48 spanning as possible.

\[ L(5, r) = \binom{5}{5-r} \sum_{i=0}^{r-1} (-1)^i \binom{5-r-i}{i} \]

For n=5, by Cayley \(n^{n-2}\) (n>3) = 5^3

\[ = 5^3 = 125 \text{ (i.e., 5*5*5)} \]

For n=5, by the proposed formula where n>3

\[ L(5, 3) = \binom{5}{2} \sum_{i=0}^{2} (-1)^i \binom{3-i}{i}(3-1)^i = \binom{5}{2} \sum_{i=0}^{2} (-1)^i \binom{3-i}{i}(3-0)^i \]

\[ = \binom{5}{2} \left[ (-1)^0 \binom{3-0}{0} + (-1)^1 \binom{3-1}{1} + (-1)^2 \binom{3-2}{2} \right] \]

\[ = \binom{5}{2} \left[ 1 + (-1) \cdot \frac{3-1}{1} + \frac{3-2}{2} \cdot \frac{3-1}{1} \right] \]

\[ = \binom{5}{2} \left[ 1 + (-1) \cdot 2 + \frac{1}{2} \cdot 2 \right] \]

\[ = \binom{5}{2} \left[ 1 - 2 + 1 \right] \]

\[ = \binom{5}{2} \cdot 0 = 0 \]

\[ = 0 \]

For n=5, by the proposed formula where n>3

\[ L(5, 4) = \binom{5}{1} \sum_{i=0}^{1} (-1)^i \binom{1-1}{i} = \binom{5}{1} \left[ (-1)^0 \cdot 1 + (-1)^1 \cdot 0 \right] \]

\[ = \binom{5}{1} \left[ 1 - 0 \right] \]

\[ = 5 \]

Sum of all L (n, r) = L (5, 2) + L (5, 3) + L (5, 4)

\[ = 5 + 60 + 5 \]

\[ = 60 \]

After taking the spanning trees, we follow the spanning tree sequence.

A. Labelled Sequence of a Tree

Let the n vertices of a tree T be labelled 1, 2, 3, 4, ..., n. Remove the pendent vertex which has the smallest label. Assume that pendent vertex be a1. Let there is a vertex b1 which is adjacent to a1. Remove the edge (a1,b1). Now there are n-1 vertices. Follow the same procedure for (a2,b2). This method is repeated on remaining n-2 vertices, then n-3 vertices and so on. This entire step is eliminated after n-2 steps. Later only 2 vertices are left. This gives us a sequence of n-2 labels. Hence, the tree T defines the sequence (b1, b2, b3, b4, b5, ... , bn), {, }, :, <, > (we get 12 special characters).

Table 1: Labelled sequence for 48 spanning trees for a complete graph with 5 vertices

<table>
<thead>
<tr>
<th>Letter</th>
<th>Trees and Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td><img src="image1.png" alt="Image" /></td>
</tr>
<tr>
<td>B</td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>C</td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>D</td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>E</td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
<tr>
<td>F</td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Fig. 1

Fig. 2

Similarly, we find the sequence for the 48 spanning trees from 5-vertex complete graph shown in Table 1. That individual sequence of spanning tree is assigned to corresponding A-Z (26 alphabets), 0-9 (10 digits) and the special characters like !, @, $, ^, &, *, {, }, :, <, > (we get 12 special characters).
<table>
<thead>
<tr>
<th>G</th>
<th>153</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>214</td>
</tr>
<tr>
<td>I</td>
<td>223</td>
</tr>
<tr>
<td>J</td>
<td>231</td>
</tr>
<tr>
<td>K</td>
<td>235</td>
</tr>
<tr>
<td>L</td>
<td>242</td>
</tr>
<tr>
<td>M</td>
<td>244</td>
</tr>
<tr>
<td>N</td>
<td>245</td>
</tr>
<tr>
<td>O</td>
<td>252</td>
</tr>
<tr>
<td>P</td>
<td>253</td>
</tr>
<tr>
<td>Q</td>
<td>254</td>
</tr>
<tr>
<td>R</td>
<td>311</td>
</tr>
<tr>
<td>S</td>
<td>314</td>
</tr>
<tr>
<td>T</td>
<td>315</td>
</tr>
<tr>
<td>U</td>
<td>331</td>
</tr>
<tr>
<td>V</td>
<td>342</td>
</tr>
<tr>
<td>W</td>
<td>345</td>
</tr>
<tr>
<td>X</td>
<td>351</td>
</tr>
<tr>
<td>Y</td>
<td>352</td>
</tr>
<tr>
<td>Z</td>
<td>413</td>
</tr>
<tr>
<td>0</td>
<td>421</td>
</tr>
<tr>
<td>1</td>
<td>425</td>
</tr>
</tbody>
</table>
Information Sharing by Spanning Trees of a Graph by labelled sequence as a key

B. Algorithm
Algorithm uses switch cases for decision making for both encryption and decryption.

Encryption
Input: Combination of Letters[A-Z], Digits [0-9], Special Characters.
Output: Digited sequence

Begin
String str //Input String
String newStr= " "  //Output String
for (int i=0; i<str.length(); i++)
    char ch=Character.toUpperCase(str.charAt(i))
    switch(ch)
        //for A to Z
        case 'A': newStr=newStr+ "114"
            break;
        ......
        case 'Z':
            newStr=newStr+ "413"
            break;

...
IV. RESULTS AND DISCUSSION

A. Encryption and Decryption Process

Encryption

Step 1: Find your sentence or word which has its own meaning.

Let the sentence be

< I HAVE SENT A MESSAGE AT 12:00 PM TODAY >

Step 2: Find the 3 digited sequence from above table for individual letter/digit/special character.

222 223 214-114-342-143 314-143-245-315 114
244-143-314-114-153-143 114-315
425-431-111-421-421 253-244 315-252-142-114-352 333

Step 3: After mapping the sequence with the sentence, assign it to the variable ENCRYPT i.e.,

ENCRYPT =

222 223 214114342143 314143245315 114
24414331413114145314 114315 425431111421421
253244 315252142114352 333

Step 4: Send the ENCRYPT to the receiver

Decryption

Step 5: We can decrypt any message by reversing the encryption procedure.

For example, the receiver gets a message like

222 215 311433131432342143142 352421321254
23515231431312154152 114315 425431111421425
253244 315252142114352 333

Step 6: Divide the word into 3-3 segments with spacings following the exact order

222 215 311-143-133-143-223-342-143-142 352-2-321-254
235-152-314-132-154-152 114-315
425-431-111-421-425 253-244 315-252-142-114-352 333

Step 7: Assign the received message to Decrypt

Step 8: Therefore, DECRIPT is the message received

DECRIPT =

< I RECEIVED YOUR MESSAGE AT 12:01 PM TODAY >

B. Encryption and Decryption Process Flow
V. CONCLUSION

The number of spanning trees should be taken based on the requirement of the data. In this paper, we have given the encryption and decryption algorithm based on English alphabet, digits from 0-9, and some other special characters. For this purpose, we have chosen the complete graph with 5 vertices. For the complete graph with 5 vertices, we can draw 125 spanning trees. To add more complexity for the methodology, we can take any other language, its alphabets and so on. The no of spanning trees increases with the no of vertices in a complete graph. For a complete graph with 6, 7, 8 vertices, we get 1296, 16807, 262114 spanning trees and so on. Hence this provides efficient encryption and decryption algorithm for information sharing as a key.

REFERENCES


AUTHORS PROFILE

P. Bala Manoj Kumar, is a student in M. Tech in Software Engineering, department of Software and Systems, School of Information and Technology, Vellore Institute and Technology (VIT), Vellore – 632014, Tamil Nadu, India. He is now focusing on Data Science, Big Data and Hadoop. He is doing his research both in Data Science and Cloud Computing Security with Microservices and DevOps Fundamentals for Medical Diagnosis. He is currently working on Medical Analysis, Business Analytics in Expert Systems (AI).

K. Sai Teja, is a student in M. Tech in Software Engineering, department of Software and Systems, School of Information and Technology, Vellore Institute and Technology (VIT), Vellore – 632014, Tamil Nadu, India. He worked on a wide variety of projects that have allowed him to put what he learned in the classroom into use in a practical perspective. His research is on Information and System Security (ISS), and Cryptography.

Dr. A. Manimaran, has been working as an Assistant Professor in the Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology (VIT) Vellore, Tamil Nadu. India from July 2012. Previously, he worked as an Assistant Professor in the Department of Mathematics, Muthyamnam Engineering College, Rasipuram, Namakkal District. He received his Doctoral degree from VIT, Vellore in the year 2017, PG degree and UG degree in Mathematics from Government Arts College, Salem, India in 2007 and 2005 respectively. His research interests include Rough Sets, Fuzzy Sets, Medical Diagnosis problem, Semiring and Cryptography. His current research focus on Semiring, Rough Sets and its applications.