International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

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.

Manuscript published on 30 September 2019
* Correspondence Author

P. Bala Manoj Kumar*, Department of Software and Systems, School of Information Technology, VIT, Vellore -632014, India.Email: pasalabalamanojkumar@gmail.com

K. Sai Teja, Department of Software and Systems, School of Information Technology, VIT, Vellore -632014, India. Email: tejasssh@gmail.com

A. Manimaran, Department of Mathematics, School of Advanced Sciences, VIT, Vellore – 632014, India. Email: marans2011@gmail.com

G. Deepa, Department of Mathematics, School of Advanced Sciences, VIT, Vellore – 632014, India. Email: deepa.g@gmail.com

B. Praba, SSN College of Engineering, Chennai. Email: prabab@ssn.edu.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

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 digited 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 $n^{n-2} = 4^{4-2}$ (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

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

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

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

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

$$= 6\ [4-2] = 12$$

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

$$= \binom{4}{1} [(-1)^{0} \binom{1}{0} (1-0)]^{1}$$

$$= 4$$

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

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

Hence, we get spanning trees when n=4 like Figure 1 **s**hows 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 a total to meet encryption and decryption. For that we choose a graph with 5 vertices. Hence, we get 48 spanning as possible.

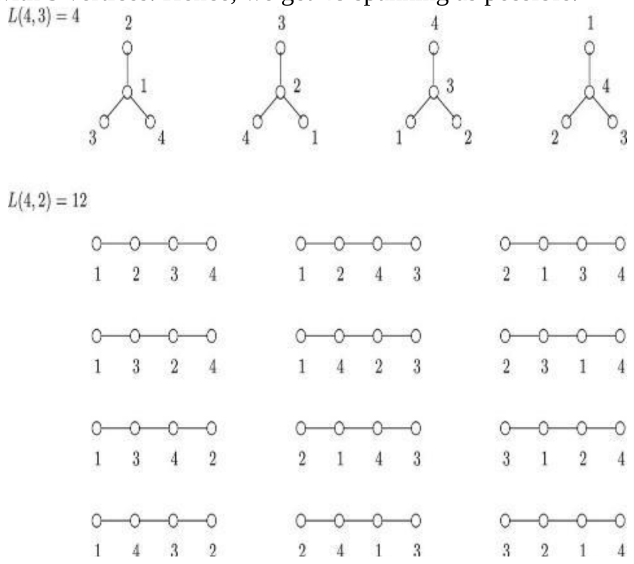


Fig. 1

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

$= 5^3$

= 125 (i.e., 5*5*5)

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

$$L(5,3) = \binom{5}{2}\sum_{i=0}^{2}(-1)^i\binom{3}{i}(3-1)^3 = \binom{5}{3}[(-1)^0\binom{3}{0}(3-0) + (-1)^1\binom{3}{1}(3-1)^3 + (-1)^2\binom{3}{2}(3-2)^3] = 10[27-24+3] = 60$$

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

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

$$= 10[8-2] = 60$$

$$L(5,4) = \binom{5}{1}\sum_{i=0}^{0}(-1)^i\binom{1}{i}(i-1)^3 = \binom{5}{1}[(-1)^0\binom{1}{0}(1-0)^3] = 5$$

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

= 60 + 60 + 5

= 125

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 a_1. Let there is a vertex b_1 which is adjacent to a_1. Remove the edge (a_1,b_1). Now there are n-1 vertices. Follow the same procedure for (a_2, b_2). 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 $(b_1, b_2, b_3, b_4, b_5, b_6, \ldots, b_{n-2})$.

Let us take the tree with 9 vertices and its sequence. We get sequence about n-2 i.e., 7 digited sequence. Figure 2 is a nine vertex tree yields the sequence 1135559 i.e., (1, 1, 3, 5, 5, 5, 9).

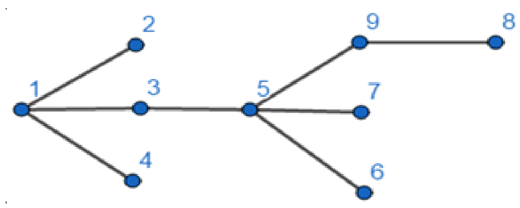
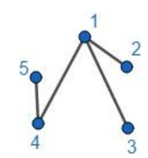
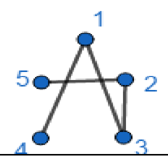
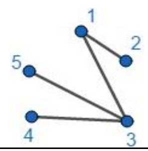
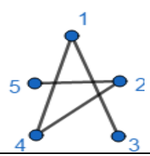
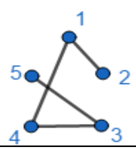


Fig. 2

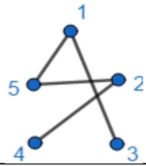
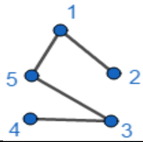
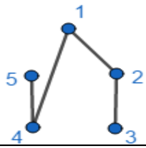
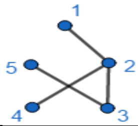
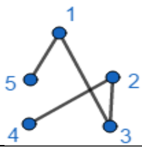
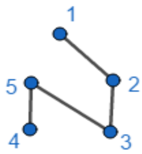
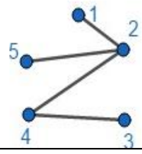
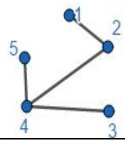
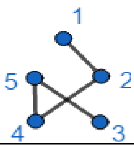
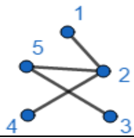
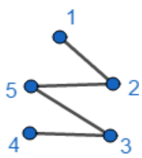
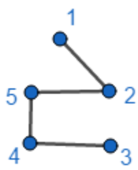
Similarly, we find the sequence for the 48 spanning trees from 5-vertex complete graph shown in Table 1. That individual sequence of spanning tree is assigned to corresponding A-Z (26 alphabets), 0-9 (10 digits) and the special characters like !, @, #, $, ^, &, *, {, }, :, <, > (we get 12 special characters).

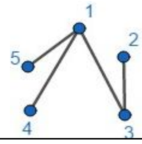
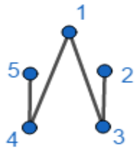
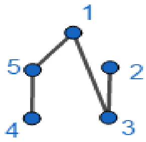
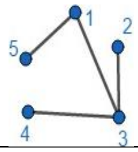
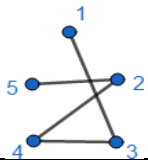
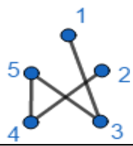
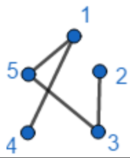
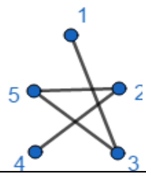
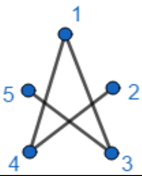
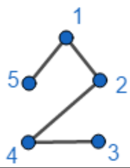
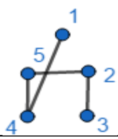
Table 1: Labelled sequence for 48 spanning trees for a complete graph with 5 vertices

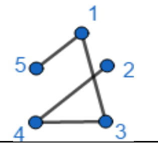
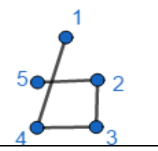
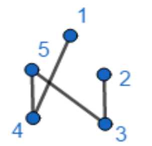
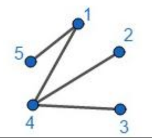
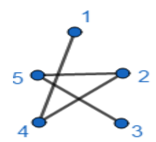
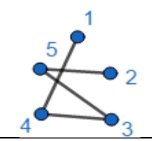
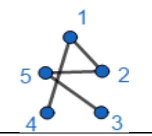
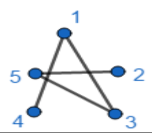
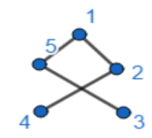
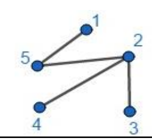
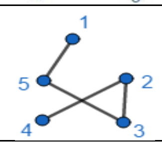
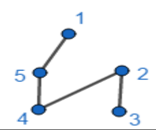
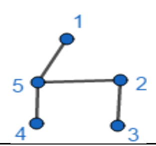
Letters	Trees and Sequence	
	Spanning Trees	Sequence
A		114
B		132
C		133
D		142
E		143

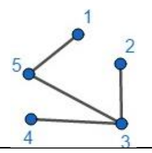
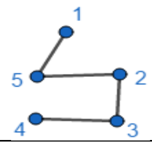
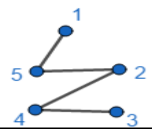
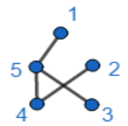
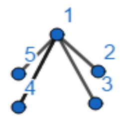
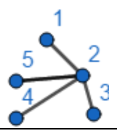
Retrieval Number: C5249098319/2019©BEIESP
DOI:10.35940/ijrte.C5249.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

F		152
G		153
H		214
I		223
J		231
K		235
L		242
M		244
N		245
O		252
P		253
Q		254

R		311
S		314
T		315
U		331
V		342
W		345
X		351
Y		352
Z		413
0		421
1		425

2		**431**
3		**432**
4		**435**
5		**441**
6		**451**
7		**452**
8		**453**
9		**513**
!		**521**
@		**522**
#		**523**
$		**524**
^		**525**

&		**533**
*		**532**
{		**542**
}		**545**
:		**111**
<		**222**
>		**333**

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;

case ' ': newStr=newStr+

break;

//for 0 to 9

case '0': newStr=newStr+ "421"

break;

.....

case '9': newStr=newStr+ "513"

break;

//for special characters

case '!':

newStr=newStr+ "521"

break;

```
        .....
        case '>': newStr=newStr+ "333"
                  break;
        default: newStr=newStr+ " "
      end switch
   end for
End
```

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

```
Begin
   String str;
   Int sequence=Integer.parseInt(str)    //Input sequence
   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 '114': newStr=newStr+ "A"
                  break;
        .....
        case '413': newStr=newStr+ "Z"
                  break;
        case ' ': newStr=newStr+
                  break;
        //for 0 to 9
        case '421': newStr=newStr+ "0"
                  break;
        .....
        case '513': newStr=newStr+ "9"
                  break;
        //for special characters
        case '521': newStr=newStr+ "!"
                  break;
        .....
        case '333': newStr=newStr+ ">"
                  break;
        default: newStr=newStr+ " "
      end switch
   end for
End
```

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 **314-143-245-315 114 244-143-314-314-114-153-143** 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 244143314314114153143** 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 311143133143223342143142 **352421321254** 235152314314132154152 **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** 235-152-314-314-132-154-152 **114-315 425-431-111-421-425** 253-244 **315-252-142-114-352** 333

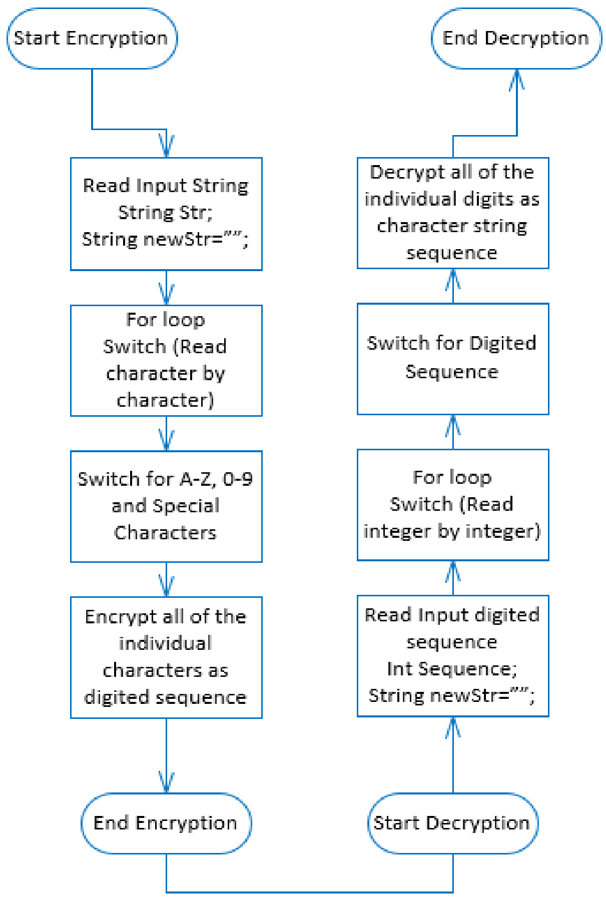
Step 7: Assign the received message to Decrypt

Step 8: Therefore, **DECRYPT** is the message received

DECRYPT =

< 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.

Retrieval Number: C5249098319/2019©BEIESP
DOI:10.35940/ijrte.C5249.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

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

1. T. Guo, J. Jiao, F. Liu and W. Wang. (2017). On the Pixel Expansion of Visual Cryptography Scheme, International Journal of Digital Crime and Forensics (IJDCF), 9(2), 38-44.
2. Rathore A K and Jain A (2017). A New Visual Cryptography Approach using Mosaic and Spread Spectrum Watermarking, International Journal of Computer Applications, 165(10), 44 – 48.
3. Yang, D., Doh, I., and Chae, K. (2017). Enhanced password processing scheme based on visual cryptography and OCR. International Conference on Information Networking (ICOIN), IEEE, 254-258.
4. Deo, N. (2017). Graph theory with applications to engineering and computer science. Courier Dover Publications. Chandrasekaran, V. M., Manimaran, A., & Ranjan, A. (2015).
5. Cryptography using a Pair of Dice. International Journal of PharmTech Research, 7(1), 85-89.
6. Manimaran A, Chandrasekaran V M and Archit. (2015). Dynamic Key Cryptography in sharing Medicine, International Journal of Applied Engineering Research, 10, 34068 – 34071.
7. Manimaran, A., Chandrasekaran, V. M., Gaur, M., Gupta, A., and Narwani, P. (2015). Data Encryption and Decryption Using Guitar Strings. International Journal of Pharmacy an Technology, 7(3), 9774-9778.
8. Manimaran A, Chandrasekaran V M, Vivek Mallineni, Gangireddy Koushik Reddy and Karthick P M. (2015), A New Approach For Encrypting And Decrypting Phone Numbers, International Journal of Pharmacy and Technology, 7(3), 9904 – 9908.
9. Al-Hazaimeh, Obida M. A. (2013). A new approach for complex encrypting and decrypting data. International Journal of Computer bNetworks & Communications, 5(2), 95 – 103.
10. Zimmerman, P. (1999). Introduction to Cryptography, 1-88.

AUTHORS PROFILE

P. Bala Manoj Kumar, is a student in M. Tech in Sofware 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 Sofware 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 ecurity (ISS), and Cryptography.s

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, Muthayammal 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.

Retrieval Number: C5249098319/2019©BEIESP
DOI:10.35940/ijrte.C5249.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication