International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

Application of Super Magic Labeling in Cryptography

M. Giridaran[1]

Assistant Lecturer, Dept. of Mathematics, DMI – St. Eugene University, Lusaka, Zambia[1]

ABSTRACT:Let$G = (V, E)$ be a graph. $G$is said to be an magic graph if the edges of G can be labelled by positive integers, so that the sum over the edges incident with any vertex is the same, independent of the choice of vertex. If the integers are the first q positive integers, where q is the number of edges, then the labelling is are called super magic labeling. The graph which admits a super magic labeling is called super magic graph. In this paper, I have applied the concept of Super Magic labeling and propose a new technique to encrypt and decrypt messages.

KEYWORDS:MagicLabeling**,** Super Magic Labeling, Cipher Text, Encryption, Decryption.

I. INTRODUCTION

Cryptography means *art of writing* or *solving codes*. Cryptography can be classified into two types called Ancient cryptography and Modern cryptography. Ancient cryptography deals with the communication of secret codes whereas modern cryptography deals with digital cash, digital signatures, authentication protocols etc. In cryptography the process of encoding a message is called *Encryption*. The message is called plain text and the encoded text is called cipher text. The process of converting the cipher text into a plain text is known as *Decryption*[1].

Any real life problem can be modelled as a graph theoretic problem. Now a days graphs are also used to encrypt data's. Since there are so many graphs available online, it is not easy to find whether the graph is used for encryption process or not. In graph theory, graph labeling is an important problem and there are so many different types of graph labeling are available in literature. A Rosa presented the first paper on graph labeling in 1966 [4] and till today there are a lot of researches going on graph labeling. Graph labeling has many applications to mention a few, graph labeling techniques are used in frequency assignment problems, communication networks etc [2].

In the past, few works have been carried out in such a way that some graph theoretic concept has been used to encrypt a message [5, 6]. Having it as a base, in this paper we have tried to apply the concept of super magic labeling in graph theory to encrypt a secret message.

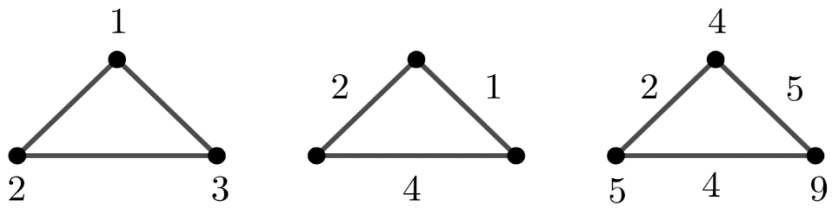
This paper is further organized as follows. In section 2, the preliminaries needed for the proposed method have been given. Section 3 deals with the encryption and decryption method along with the java program.

II. PRELIMINARIES

In this section, we give some elementary details required in the proposed work.

Definition 1. Graph Labeling

The process of assignment of integers to the vertices, edges or both based on some condition is known as Graph labeling [3]. In the following figure 1, (a) represents a vertex labelled graph, (b) represents a edge labelled graph and (c) represents a graph which is both vertex and edge labelled.



(a) (b) (c)

Figure 1. (a) Vertex Labelled Graph, (b) Edge Labelled graph, (c) Vertex and Edge Labelled graph

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

Definition 2. Magic Labeling

The process of assigning labels to the edges of G, such that the sum over the edges incident with any vertex is the same, independent of the choice of vertex is known as magic labeling [2].

Definition 3. Super Magic Labeling

In magic labeling, if the labels assigned are the first q positive integers, where q is the number of edges in the graph, then such a labelling is called super magic labeling [2]. An example of super magic labeling is given in figure 2.

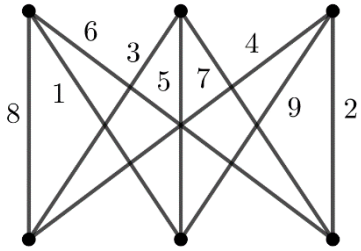


Figure 2. Super Magic labeling of $K_{3,3}$

III. METHODOLOGY

In this section, we have discussed the encryption and decryption algorithms along with Java program.

Here I have considered complete bipartite graphs with at least 5 vertices on each set so that the resultant matrix will have 25 values which can be assigned to different English alphabets and zero can be assigned to the 26th alphabet. If suppose we consider some other language, for instance Tamil where we have 247 words we need a complete bipartite graph with at least 15 vertices to encrypt which makes difficult for humans to encrypt or decrypt it. Hence, I have given a Java program which makes our work much more simpler. The edge weights in the following algorithm is computed using Super Magic labeling condition. Now the encryption process works as follows,

Encryption Algorithm:

Step 1. Consider a complete bipartite graph $K_{n,n}, n \geq 5$.

Step 2. Compute the edge weights of $K_{n,n}$ using MATLAB.

Step 3. Assign edge weight to English alphabets row wise and swap the alphabet. For example, if the edge weight of first row first column element is 5, then the alphabet A will be replaced by E. As each edge weight is distinct, every alphabet will be distinctly swapped.

Consider the following example given for a particular case. That is here we have considered the complete bipartite graph $K_{5,5}$. The below 5×5 matrix can be obtained from MATLAB.

```
>> magic(5)

ans =

    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

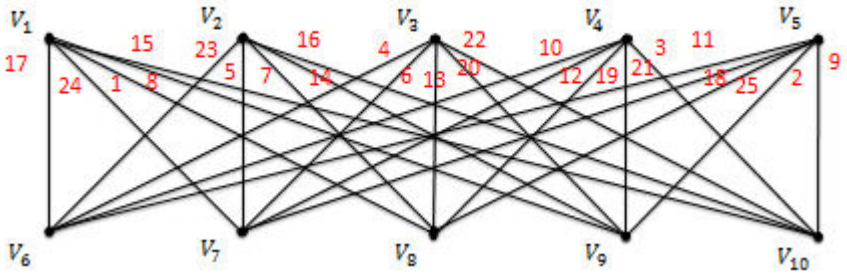


Figure 3. Super Magic labeling of $K_{5,5}$.

$A \to 17 \to Q, B \to 24 \to X, C \to 1 \to A, D \to 8 \to H, E \to 15 \to O, F \to 23 \to W, G \to 5 \to E, H \to 7 \to G, I \to 14 \to N, J \to 16 \to P, K \to 4 \to D, L \to 6 \to F, M \to 13 \to M, N \to 20 \to T, 0 \to 22 \to V, P \to 10 \to J, Q \to 12 \to L, R \to 19 \to S, S \to 21 \to U, T \to 3 \to C, U \to 11 \to K,$ V$\to 18 \to R, W \to 25 \to Y, X \to 2 \to B, Y \to 9 \to I, Z \to 0 \to Z.$

Encryption program:

```
importjava.util.*;

public class EncryptionAndDecryption{
static void process_userInput(String name){

        String encryptkey = "";

        // Convert Iuput string small to captial

        String input_uppercase=name.toUpperCase();
        //System.out.println("Array name ==>"+input_uppercase);

        //Convert string to array
        char[] array_name = input_uppercase.toCharArray();
        HashMap<String,String>hm=new HashMap<String,String>();
hm.put("A","Q");
hm.put("B","X");
hm.put("C","A");
hm.put("D","H");
hm.put("E","O");
hm.put("F","W");
hm.put("G","E");
hm.put("H","G");
hm.put("I","N");
hm.put("J","P");
hm.put("K","D");
hm.put("L","F");
hm.put("M","M");
hm.put("N","T");
hm.put("O","V");
hm.put("P","J");
hm.put("Q","L");
hm.put("R","S");
hm.put("S","U");
hm.put("T","C");
hm.put("U","K");
```

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

```
hm.put("V","R");
hm.put("W","Y");
hm.put("X","B");
hm.put("Y","I");
hm.put("Z","Z");


    // System.out.println("Initial list of elements:");


for (inti =0; i<array_name.length;i++){

            String array_value = Character.toString(array_name[i]);

for(Map.Entry m:hm.entrySet())
    {
if(array_value != " "){

            if(m.getKey().equals(array_value))
                    {

            encryptkey = encryptkey + m.getValue();
                        break;

                }

                }
            else
                {

                    encryptkey = " " + encryptkey;
                    break;

                }
    }

 }

            System.out.println(" Your encryptkey is ==>"+encryptkey.toLowerCase());

 }

public static void main(String args[]) {

            Scanner sc = new Scanner(System.in);

            System.out.print("Enter your Key ==> ");
            String name = sc.nextLine();
process_userInput(name);

}
}
```

The resultant cipher text can be send along with the magic constant so that the receiver can decrypt it easily. For each n there is a magic constant which can be computed by the formula, $\frac{n(n^2+1)}{2}$ by which we can find the value of n.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

Decryption Algorithm:

Step 1. First we compute the value of n using the formula $\frac{n(n^2+1)}{2}$.

Step 2. Next, we compute the edge weights of $K_{n,n}$ can be obtained using MATLAB.

Step 3. Now, we can decrypt the cipher text and can get the original plain text.

If we manually try to decrypt the cipher text which is large in size, it consumes more time and in order to reduce it, we have also given a Java program to decrypt the cipher text which saves our time. If we give the input as cipher text along with the n value in the following java program, it will decrypt and we can obtain the plain text.

Decryption program:

```
importjava.util.*;

public class EncryptionAndDecryption{

static void process_userInput(String name){

        String encryptkey = "";

        // Convert Iuput string small to captial

        String input_uppercase=name.toUpperCase();
        //System.out.println("Array name ==>"+input_uppercase);

        //Convert string to array
        char[] array_name = input_uppercase.toCharArray();
        HashMap<String,String>hm=new HashMap<String,String>();
hm.put("Q","A");
hm.put("X","B");
hm.put("A","C");
hm.put("H","D");
hm.put("O","E");
hm.put("W","F");
hm.put("E","G");
hm.put("G","H");
hm.put("N","I");
hm.put("P","J");
hm.put("D","K");
hm.put("F","L");
hm.put("M","M");
hm.put("T","N");
hm.put("V","O");
hm.put("J","P");
hm.put("L","Q");
hm.put("S","R");
hm.put("U","S");
hm.put("C","T");
hm.put("K","U");
hm.put("R","V");
hm.put("Y","W");
hm.put("B","X");
```

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

```
hm.put("I","Y");
hm.put("Z","Z");

    // System.out.println("Initial list of elements:");


for (inti =0; i<array_name.length;i++){

            String array_value = Character.toString(array_name[i]);

for(Map.Entry m:hm.entrySet())
    {
if(array_value != " "){

            if(m.getKey().equals(array_value))
                        {

            encryptkey = encryptkey + m.getValue();
                        break;

                    }

                    }
            else
                    {

                        encryptkey = " " + encryptkey;
                        break;

                    }
    }

 }

            System.out.println(" Your Decryption key is ==>"+encryptkey.toLowerCase());

 }

public static void main(String args[]) {

            Scanner sc = new Scanner(System.in);

            System.out.print("Enter your Key ==> ");
            String name = sc.nextLine();
process_userInput(name);

}
}
```

Example:

Consider the following illustration to see the encryption and decryption process using java program.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

Encryption:

Suppose if we want to send the message, supermagiclabelinggraphandmagicsquare, to encrypt it manually it consumes more time and also there is a chance of making mistakes. In order to avoid all those we use the java program. It automatically encrypts and gives us the cipher text which is to be send.
Input: supermagiclabelinggraphandmagicsquare
Output: rsjgvmchytqcxgqyihhvcjdcikmchytrascvg

Decryption:

In order to decrypt the cipher text, we give the input as the cipher text along with the value of n, so that the program decrypts it automatically and gives us the plain text.
Input: rsjgvmchytqcxgqyihhvcjdcikmchytrascvg5
Output: supermagic labelling graphand magicsquare

IV. CONCLUSION

Since there are many type of graph labeling techniques available in literature, it is very much safe to use graphs in cryptography as it is difficult to find the labeling technique and also to find the difference between the graphs which has been used for encryption. In this paper, we have applied the concept of super magic labeling to encrypt and decrypt a secret message. I am trying to extend this idea to encrypt messages using different graph theoretic technique.

REFERENCES

[1] Jonathan Katz, Yehuda Lindell. Introduction to modern cryptography. CRC press, 2014.
[2] Joseph A. Gallian, "A dynamic survey of graph labeling." The Electronic journal of combinatorics, Vol. 17, 2014.
[3] KinsYenoke, Arputha Jose T, Venugopal P, On The Radio Antipodal Mean Number Of Certain Types of Ladder Graphs, International Journal of Innovative Research in Science, Engineering and Technology, Vol. 9, Issue 6, pp. 4607 – 4614, 2020.
[4] A. Rosa, On certain valuations of the vertices of a graph, In Theory of Graphs (Internat. Symposium, Rome, pp. 349 – 355, 1966.
[5] WaelMahmoud AlEtaiwi, Encryption algorithm using graph theory, Journal of Scientific Research and Reports, pp. 2519-2527, 2014.
[6] M. Yamuna, A. Suwathi, Nikitha Krishnan, Four Digit Pin Number as a Digraph, International Journal of Computer Application, Vol. 3, Issue 4, pp. 100 – 107, 2014.