ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 13, October 2016

A Novel Graph Based Encryption Technique for Information Security

Mandira Sen [1], Swarnajit Ray [2]

Department of Information Technology, Kalyani Government Engineering College, Kalyani, Nadia, W.B, India [1]

Department of Information Technology, Kalyani Government Engineering College, Kalyani, Nadia, W.B, India [2]

ABSTRACT: Information Security has always been considered to be the prime concern for any kind of data/information driven event. Eavesdroppers always try to snoop into the message without having any access rights. It is the responsibility of the sender to deliver the message securely to the destination. In this paper a graph based information hiding technique has been proposed. In this technique an input technique has been proposed. In this technique an input indexed by a directed graph. The graph is used as a cover to index by a directed graph. The graph is used as a cover to encryption technique. It has been found that the algorithm encryption technique. It has been found that the algorithm posses time complexity of the order of O (n^2) but it can counter Brute-force attack as the encrypted text is not any direct mapping of the input text.

KEYWORDS: Graph, Directed Graph, Adjacency matrix.

I. INTRODUCTION

Cryptography is processes where data are keep secret when data are sending from sender side to receiver side [1-2]. There are mainly two type of cryptography, one is private key cryptography and another is public key cryptography [3-4]. In cryptography the encoded message can't understand by other, but in Steganography we hide the message such a way that there is no knowledge of existence of the message [5]. That mean in Cryptography hacker might be aware that the secret data are forwarding form sender side to receiver side, but in the case of Steganography the secret message doesn't make any attention of hacker [6]. This is the advantage of Steganography over cryptography. In Steganography there are mainly 3 types of cover are used namely Text, Image [7], Audio [8-9]. The data, image or audio in Steganography hide with in a cover like another message file or image. In our approach we are sending date with help of graph [10-12]. Here graph is used as cover. The main system consists of sender side and receiver side algorithm. Sender side first encode the data with help of our GBE algorithms and at sending time we hide data within a graph. On the receiver side receiver retrieve the information by the help of receiver side secret key.

II. PROPOSED ALGORITHM

In this section we will describe about the sender side algorithm. There are three parts in sender side algorithm. In the first part take an input from user and generate a binary String by applying some procedure say into algorithm A.1, and then create a directed graph by using this binary string say into algorithm A.2, and finally for secure purpose change the data of adjacency matrix. And then store the changing result into the receiver side. Here **GBE algorithm** (Graph Based Encryption) is the combination of sender side algorithm and receiver side algorithm.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 13, October 2016

A. ***SENDER SIDE ALGORITHM***

A.1 ALGORITHM COVERT STRING TO BINARY

INPUT: String S.
OUTPUT: Binary string and the Quotient [] array.

Step 1: Convert each letter of the input string into ASCII value and then convert into corresponding binary value.

Step 2: Group this binary value into some blocks where each Block contains 4 bit binary digit. In the time of grouping if the length of binary bits is not divisible by 4 then extra zero will be added in MSB position.

Step 3: Now convert the value of each block to its corresponding decimal value. And again grouping this decimal value into some blocks where each block contains 4 decimal Values. If number of digit in any block is less than 4 then store Values. If number of digit in any block is less than 4 then store and for each decimal block find the corresponding index from the existing database.

Step 4: When index is found, then index divided by 16 and store the quotient into an array (say quotient [] array) and store the remainder into MSB position of graph [] array.

Step 5: Convert all values of graph [] array into corresponding binary value and store into an array (say binary [] array).

A.2 ALGORITHM CREATE_GRAPH

INPUT: The Binary [] array.
OUTPUT: A connected graph G= (V, E).

Step 1: Let the vertex set be V= {v1, v2, v3, v4……vn}, where n is the size of binary [] array.

Step 2: Now create a directed graph from binary [] array. For each bit of binary array follow the following rules:
If binary [i] =1 then
2.1. Draw (n-1) edges from vi to all remaining (n-1) vertices up to vn.
2.2. Also check if previously one connection is already established between two vertices of any one direction then no need to connect them again and can't connect the same node means self loop is not occur for any node.

Step 3: Repeat step 2 until i=n, where n is the size of binary [] array.

Step 4: Return the adjacency matrix of graph G.

A.3 ALGORITHM CHANGE_GRAPH

INPUT: Adjacency matrix of graph G and the quotient [] array.
OUTPUT: Row [] array, Column [] array.

Step 1: Firstly complement each value of adjacency matrix except diagonal value and generate a matrix (say G‟‟).

Step 2: Now within G‟‟ performs some operations which illustrated below:
2.1. Perform „OR‟‟ operation for each row of G‟‟.
2.2. Now calculate decimal value of the following information.
2.2.1. Result of the „OR‟‟ operation (say dec_or).

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 13, October 2016

2.2.2. Each row of the adjacency matrix and store into an array (say dec_row [] array).
2.2.3. Each column of the adjacency matrix and store into an array (say dec_column [] array).

Step 3: Now add the value of dec_or with every element of dec_row [] array and dec_column [] array and store the values into corresponding array (say row [] array and column [] array) and send this row [] array, column [] array and quotient [] array into the receiver side.

B. ***RECEIVER SIDE ALGORITHM***

Here firstly receive encoded information from sender side and then decoding into the reverse procedure and finally get the actual information.

ALGORITHM RETRIEVE_INFO

INPUT: row [] array, column [] array and the quotient [] array.
OUTPUT: Actual Information.

Step 1: From the row [] array find the smallest number.

Step 2: Now subtract that smallest number from other content of row [] array and column [] array.

Step 3: After subtractions convert each content of row [] array into corresponding binary value which is nothing but an n*n matrix (say row matrix). Same thing performs for the column [] array and again generate another matrix (say column matrix).

Step 4: Now check row matrix and column matrix. If both are not same then stop the procedure otherwise perform the following steps.

4.1. Complement each value of row matrix or column matrix except diagonal and generate a matrix say G. Now from G we get a binary string applying some rules:

4.1.1. Firstly check each position of first row of G. If the value of any position is 1, then store 1 into the first position of an array (say binary [] array). Same procedure performs for each row of the matrix G.
4.1.2. Continue this procedure up to n times, where the matrix is n*n. Finally get a binary string from G.

4.2. Now convert this binary string into equivalent decimal number starting from right side and store into an array say decimal [] array.

4.3. Split this decimal [] array according to the size of quotient [] array (say decimal1 [] array and decimal2 [] array).

4.4. Now multiply 16 with the first element of the quotient []array and add this result with the first element of decimal1 [] array and store the result into a new array (say index [] array). Continue this procedure until quotient [] array is not empty.

4.5. After getting the index array, find the content of first element of the index [] array from the existing database and store the content into an array (say decimal_ascii [] array). Continue this procedure until the index [] array is not empty.

4.6. Now convert each decimal value of decimal_ascii [] array into corresponding Character and get the actual information.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 13, October 2016

III. RESULT & EXPLANATION

Now show an example by using this algorithm step by step.

Sender side:

INPUT: Information.
OUTPUT: row [] array, column [] array and the quotient [] array according to the information.

Step 1: Input string say "kgec". Now corresponding binary string for "kgec" is:

1.1 ASCII table for "kgec":

k= 107, g=103, e=101, c=99

1.2 Binary value of corresponding ASCII values:

107=1101011, 103=1100111, 101=1100101, and 99=1100011

Step 2: Now grouping this binary string in following way. And convert each block into corresponding decimal value.

1101	0111	1001	1111	0010	1110	0011
⇩	⇩	⇩	⇩	⇩	⇩	⇩
13	7	9	15	2	14	3

Here the total length of the binary bit is divisible by 4 so padding is not required in the MSB position.

Step 3: Now group this decimal values starting from right side. Where each group contains 4 decimal values. Like here first group contain 13, 7, 9, 15. But in the second group three element is present so store this three element into an array (say graph [] array).

Step 4: Now find the index of 13, 7, 9, and 15 from database. We have an existing database like this:

Index	Value	Index	Value	Index	Value
1	0,0,0,0	100	0,0,6,3	55000	13,6,13,7
2	0,0,0,1	200	0,0,12,7	55200	13,7,9,15
3	0,0,0,2	300	0,1,2,11	:	:
4	0,0,0,3	400	0,1,8,15	:	:
5	0,0,0,4	:	:	65533	15,15,15,12
6	0,0,0,5	:	:	65534	15,15,15,13
7	0,0,0,6	1000	0,3,14,7	65535	15,15,15,14
8	0,0,0,7	1001	0,3,14,8	65536	15,15,15,15

So the index for 13, 7,9,15 is 55200. Now divide this index by16. And store the quotient 3450 into an array say quotient [] array, and store the remainder 0 into the graph [] array.

Step 5: Now the elements of the graph [] array is 0, 2, 14, and 3.Create a directed graph by using the binary value of this decimal value which is: 0000001011100011.

Step 6: Create the directed graph by using this binary string in such a way that if the value of any position is 1 then draw an edge from that node to another nodes and don't connect same node again means Self loop is not occur. And the number of bit in binary string is equal to the number of node of the directed graph. We are counting 1 from MSB. Here first 1 occur in node 7.so we draw edges from node 7 to all other nodes except 7. Next 1 occurs at position 9. So draw edges from node 9 to all other nodes except 7 and 9. In this way we are creating the graph.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 13, October 2016

The adjacency matrix of the directed graph given below:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	0	1	0	0	1	1	1	1	1	1
1	1	1	1	1	1	0	1	0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	1	0	0	0	1	1	1	0	1
1	1	1	1	1	1	0	1	0	0	0	1	1	1	0	0

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 13, October 2016

Step 8: Now complement value of each position of this adj. matrix except diagonal after that performs row wise "OR" operation to generate a new binary string. And get another matrix shown below:

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	0	1	1	1	0	0	0	1	0

And result of row wise "OR" operation is:

1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1

Step 8: Now calculate decimal value for result of "OR" operation, each row of this adj. matrix and each column of this adj. matrix."OR" operation: 65023.

Column wise decimal value:
32028, 48412, 56604, 60700, 62748, 63772, 65023, 64540, 64895, 64831, 64799, 64780, 64788, 64792, 64797, 64796
Row wise decimal value:
32767, 49151, 57343, 61439, 63487, 64511, 0, 65279, 512, 640, 704, 65519, 65527, 65531, 736, 738

Step 9: Now add the decimal value of „OR" operation with each element of column wise decimal value and row wise decimal value and store the result into column [] array and row [] array. Get the following result:

Column wise decimal value after addition (content of column [] array):
97051, 113435 , 121627, 125723 , 127771, 128795 ,130046, 129563, 129918, 129854 , 129822, 129803 , 129811, 129815 , 129820, 129819

Row wise decimal value after addition (content of row [] array):
97790, 114174, 122366, 126462 , 128510, 129534, 65023, 130302, 65535, 65663, 65727, 130542, 130550, 130554, 65759, 65761

Step 10: Now send the quotient [] array (here content is 3450), the column [] array and the row [] array into the receiver side.

Receiver Side:

INPUT: row [] array, column [] array and quotient [] array. OUTPUT: Actual information.

Step 1:Firstly from the row [] array finds the smallest element, here smallest element is: 65023.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 13, October 2016

Step 2: Now subtract this smallest number (65023) from each element of row [] array as well as column [] array and get the following result.

After subtraction content of row [] array is:
32767, 49151, 57343, 61439, 63487, 64511, 0, 65279, 512, 640, 704, 65519, 65527, 65531, 736, 738

After subtraction content of column [] array is:
32028, 48412, 56604, 60700, 62748, 63772, 65023, 64540, 64895, 64831, 64799, 64780, 64788, 64792, 64797, 64796

Step 3: Now convert the content of row [] array and column [] array into equivalent binary number.
Binary representation of content of row [] array and get a matrix say row matrix and Binary representation of content of column [] array and get a matrix say column matrix:

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	0	1	1	1	0	0	0	1	0

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	0	1	1	1	0	0	0	1	0

Step 4: Now check if the value of row matrix and column matrix are not same (data is change or hacked by someone) then stop the whole procedure as early as possible otherwise continue the following steps. Here row matrix and column matrix both are same.

Step 5: Complement value of each position of any one matrix (row matrix or column matrix) except diagonal.

Step 6: Now create a binary string from this matrix by applying some rules show in following steps.

Step 7: Check first row of the matrix. If the value of any position is 1, then store 1 into the first position of corresponding array (say binary [] array). Continue this procedure for each row of the matrix. In this way get a binary string (here string is 0000001011100011).

Step 8: Now convert this binary string into decimal number.

0000	0010	1110	0011
⇩	⇩	⇩	⇩
0	2	14	3

Step 9: Now split this decimal number according to the size of quotient [] array. Here size is 1. So first part contains 0, and second part contains 2, 14, and 3.

Step 10: Now multiply 16 with the content of quotient [] array and add the result with decimal number which first part contain. Continue until quotient [] array is not empty. So we get here: (16 * 3450) +0=55200.

Step 11: Now find the content of index 55200 from the existing database. Content are: 13, 7,9,15. Now merge this value with the second part of decimal number say here 2, 14,3. So we get 13, 7,9,15,2, 14, and 3.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 13, October 2016

Step 12: Convert this decimal number into equivalent binary number and get the following binary string:

1101	0111	1001	1111	0010	1110	0011
⇩	⇩	⇩	⇩	⇩	⇩	⇩
13	7	9	15	2	14	3

Step 13: Now convert this binary string into equivalent ASCII character which is the actual information:

1101011	1100111	1100101	1100011
⇩	⇩	⇩	⇩
k	g	e	c

V. COMPLEXITY AND SECURITY ANALYSIS

we have analysed the proposed algorithm in three stages vide CONVERT_STRING_TO_BINARY, CREATE_GRAPH, CHANGE_GRAPH following are the explanation of the respective complexity analysis.

Case a) CONVERT_STRING_TO_BINARY Algorithm This algorithm divided into two sub parts:

A.1 convert to value

A.2 breaks into blocks.

Complexity to convert any decimal value to corresponding binary value is O (n $\log_2$ n). For n element the complexity becomes O (n log2 n).In our algorithms the complexity is O (n $\log_2$ K).where k is the maximum ASCII value of all character. And total number of binary element is also O (n $\log_2$ K).

For each 4 bit binary bit we convert into corresponding decimal value. So complexity is O (n log2 K)/4 ≈ O (n log2 K)

Total complexity is convert to value + break to block = **O (n**$\log_2$ K). + O (n $\log_2$ K) ≈ O (n $\log_2$ K).

Case b) CREATE_GRAPH Algorithm

From mathematical induction we can compute that for n bit character the maximum number of node in graph is (3n)

Then the complexity of create graph is O (3n * 3n) ≈ O(n^2).

Case c) CHANGE_GRAPH Algorithm:

complexity of change graph is also O (n^2).

(Complexity of CONVERT_STRING_TO_BINARY) + (Complexity of CHANGE_GRAPH) + (Complexity of ALGORITHM CHANGE_GRAPH) = = O (n log2K) + O (n2). + O (n2). ≈ O(n2).

VI. SECURITY ANALYSIS

We have tested the proposed algorithm for different combinations of input text, keeping most of the characters unaltered. In spite of keeping almost the same set of characters in the input message the encrypted messages are different as shown in the following table:

Input	Encrypted Message			Input	Encrypted Message		
k**a**ec	32767,81918,90110,94206, 96254,97278,65535,98046, 66047,66175,66239,98286, 98294,98298,66271,66273	65534,48411,56603,60699, 62747,63771,65022,64539, 64894,64830,64798,64779, 64787,64791,64796, 64795	3448	k**b**ec	32767,65535,90110,942 06,96254,97278,81919, 98046,82431,82559,826 23,98286,98294,98298, 82655,82657	65534,49150,40219,4 4315,46363,47387,48 638,48155,48510,484 46,48414,48395,4840 3,48407,48412,48411	3448

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 13, October 2016

kcec	97790,114174,122366,126462,128510,129534,65023,130302,65535,65663,65727,130542,130550,130554,65759 ,65761,129820,129819	97051,113435,121627,125723,127771,128795,130046,129563,129918,129854,129822,129803,129811,129815	3449	kdec	81918,49151,106494,110590,112638,113662,65535,114430,66047,66175,66239,114670,114678 ,114682,66271,66273	64795,98302,89371,93467,95515,96539,97790,97307,97662,97598,97566,97547,97555,97559,97564,97563	3449
keec	32767,81918,90110,94206,96254,97278,65535,98046,66047,66175,66239,98286,98294,98298,66271,66273	65534,48411,56603,606996 2747,63771,65022,6453964 894,64830,64798,64779647 87,64791,64796,64795	3449	kfec	32767,65535,90110,94206,96254,97278,81919,98046,82431,82559,82623,98286,98294,98298,82655,82657	65534,49150,40219,44315,46363,47387,48638,48155,48510,48446,48414,48395,48403,48407,48412,48411	3449
kgec	97790,114174,122366,126462,128510,129534,65023,130302,65535,65663,65727,130542,130550,130554,65759,65761	97051,113435,121627,125723,127771,128795,130046,129563,129918,129854,129822,129803,129811,129815 ,129820,129819	3450				

VI. CONCLUSION & FUTURE SCOPE

Here we have represented an idea of hiding message by using a directed graph. In this approach there is no direct relationship between the input message and the encrypted message. So Brute-Force attack is quite hard here. Here information is hiding into a directed graph but the adjacency matrix of the graph is not sent into the receiver side. Eavesdropper always tries to hacked information, but in this technique quite impossible to hacked information because here we maintain a in build database as a key although if the data is hacked we can realized it because in the receiver side check the received data that the data is correct or not.

We will try to increase the complexity of the algorithm so that eavesdropper cannot hack the information.

In this approach any special character, numeric character is not accepted; only characters accept. We will try further implementation for accept special character or any numeric value.

REFERENCES

[1]Data Communications & Networking by A. Behrouz Forouzan.
[2] Computer Networks by A Tanenbaum.
[3] Cryptography and Network Security, Fourth Edition, Pearson Education India, June 3, 2010.
[4] NETWORK SECURITY ESSENTIALS: APPLICATIONS AND STANDARDS FOURTH EDITION by William Stallings.
[5] Robert Krenn, “Steganography and steganalysis”, An Article, January 2004.
[6]"Steganography and Steganalysis: A Review by Vipul Singhal, Dhananjay Yadav, Devesh Kumar Bandil”. International Journal of Electronics and Computer Science Engineering. ISSN: 2277-1956
[7]”Data hiding in audio by using image steganography technique by Budda Lavanya, Yangala Smruthi, Srinivasa Rao Elisala”. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) ISSN 2278-6856.
[8] “Data Hiding in Audio by Reserving Room in Advance of Encryption by B.Kavitha, Mrs.Brindha Rajakumari”. B.Kavitha et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 2788-2792
[9] INFORMATION HIDING USING AUDIO STEGANOGRAPHY – A SURVEY by Jayaram P, Ranganatha H R, Anupama H S. The International Journal of Multimedia & Its Applications (IJMA) Vol.3, No.3, August 2011
[10] “Graphs as an Aid to Security by Debranjan Pal, Samar Sen sarma” International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) ISSN 2278-6856
[11] A Graph–Theoretic Approach to Steganography Stefan Hetzl and Petra Mutzel.
[12] A Graph–Theoretic Approach to Steganography By Stefan Hetzl, Petra Mutzel. Lecture Notes in Computer Science Volume 3677, 2005, pp 119-128.
[13] New Directions in Cryptography by WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-22, NO. 6, NOVEMBER 1976.
[14] “A New Steganographic Method For Grayscale Image Using Graph Coloring Problem. Sidi Mohamed Douiri, Mohamed Boy Ould Medeni, Souad Elbernoussi1, El Mamoun Souidi” Applied Mathematics & Information Sciences An International Journal Appl. Math. Inf. Sci. 7, No. 2, 521-527 (2013).