DATA TRANSFER USING BIPARTITE GRAPHS

M. Yamuna[1], K. Karthika[2]

[1,2]*School of Advanced Sciences, VIT University, Vellore, (India).*

ABSTRACT

Transfer of data is and its safety is an issue in current world. Methods are developed and used for data encryption. Graph theory is growing as a promising field for this purpose. In this paper we propose a method of message encryption as a graph.

Keywords: Decryption, Encryption, Graph.

I INTRODUCTION

In cryptography, encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption does not of itself prevent interception, but denies the message content to the interceptor. In an encryption scheme, the message or information, referred to as plaintext, is encrypted using an encryption algorithm, generating cipher text that can only be read if decrypted. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is in principle possible to decrypt the message without possessing the key, but, for a well designed encryption scheme, large computational resources and skill are required. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients, but not to un authorized interceptors [1].

Graph theory is extensively used in encryption. In [2], M. Yamuna et al have provided a new genetic code for amino acids and by using this any details regarding amino acids can be encrypted. In [3], Wael Mahmoud Al Etaiwi has provided an encryption algorithm to encrypt and decrypt data securely with the benefits of graph theory properties. They have used the concepts of cycle graph, complete graph and minimum spanning tree to generate a complex cipher text using a shared key.

II PRELIMINARY NOTE

In this section we provide the basic results of graph theory which are required for proposed encryption scheme.

Graph

In the most common sense of the term, a graph is an ordered pair G = (V, E) comprising a set V of vertices or nodes together with a set E of edges or links, which are 2 – element subsets of V (that is an edge is related with two vertices, and the relation is represented as an unordered pair of the vertices with respect to the particular edge).

Weighted Graph

A graph is a weighted graph if a number (weight) is assigned to each edge. Such weights might represent, for example, costs, lengths or capacities, etc. depending on the problem at hand. Some authors call such a graph a network [4].

Multigraph

A multigraph is a graph which is permitted to have multiple edges (also called parallel edges), that is, edges that have the same end nodes. Thus two vertices may be connected by more than one edge [5].

Independent Set

An independent set is a set of vertices in a graph, no two of which are adjacent [6].

Bipartite Graph

A bipartite graph (or bigraph) is a graph whose vertices can be divided into two disjoint sets U and V (that is, U and V are each independent sets) such that every edge connects a vertex in U to one in V [7].

III RESULTS AND DISCUSSIONS

In this paper we proposed the encryption scheme for transfer the data into a graph. For that we have given a encryption table and graph construction in this section.

3.1. Construction of Encryption Table

First we decide the number of characters (S) required for the message encryption. We can randomly fix the number of rows and columns of the table, taking care that the number of cells available in the table is atleast of length of S. Assign numbers 1, 2, 3…. k, to the columns and numbers k + 1, k + 2, ... m, to the rows, where k = number of columns $k \leq 9$, m = number of rows. Distribute the characters in S randomly in the table.

For normal message we use the 26 alphabets and blank space. A model table for the same is seen is Table 1.

	1	2	3
4	A	B	C
5	D	E	F
6	G	H	I
7	J	K	L
8	M	N	O
9	P	Q	R
10	S	T	U
11	V	W	X
12	Y	Z	Space

Table 1

Now each character in the cell receives a number value. The first character represents the column number, remaining the row number. For example using Table 1 A receives value 14, U receives value 310.

3.2. Graph Construction from Number Sequence

Let M be message to be encrypted of length k. Convert each character in M into its corresponding number value using Table 1. Let the resulting sequence be M1. We know that each character will receive a two place value, one representing the row number and other the column number. So M1 will be a sequence of numbers. Let us represent them as c_1r_1, c_2r_2,…, c_kr_k. Note that c_1, c_2 ….c_k, r_1, r_2, …, r_k are numbers. We construct a graph G as follows

Vertices Set of G Number of vertices in G = number of distinct row numbers + column numbers used to generate M1.

Each vertex receives its corresponding row and column value as its label.

Edge Set of G Draw edges between the vertex pairs (c_1, r_1), (c_2, r_2),… (c_k, r_k). Let us label these edges as e_1, e_2, …, e_k.

Number of edges in G = length of M.

Note that c_1, c_2, …, c_k and r_1, r_2, …r_k are always independent sets. So, the graph G is always a bipartite graph.

Edge Weight Assign random numbers n_1, n_2, n_3,…, n_k as the edge weights to the edges e_1, e_2,…, e_k so that $n_1 > n_2 >…> n_k$.

3. 3. Encryption Algorithm

Let M**: GRAPH** be the message to be encrypted.

Step 1 Convert each character in M into its corresponding number value using Table 1 to generate M1.

For the message M, M1: 16 39 14 19 26

Step 2 Construct the graph corresponding to the sequence M1 as explained in Section 3. 2 to generate a graph G.

For M1

Vertex Set = { 1, 2, 3, 4, 6, 9 }

Edge Set = { (1 6), (3, 9), (1, 4), (1, 9), (2, 6) } = { e_1, e_2, e_3, e_4, e_5 }

Edge weights = { 24, 32, 42, 44, 86 } assigned to the edges e_1, e_2, e_3, e_4, e_5 respectively.

The resulting graph is as seen in Fig. 1

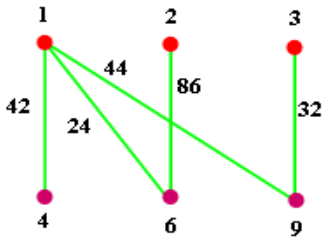

Fig. 1

Step 3 Send G to the receiver.

For decrypting the message we reverse the procedure.

Suppose the received graph is as seen in Fig. 2

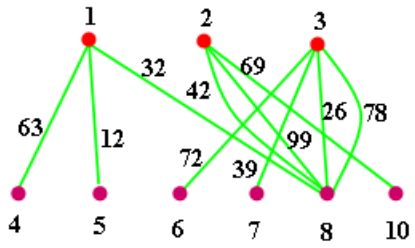

Fig. 2

Arranging the edge weights in increasing order we generate the sequence

12 26 32 39 42 63 69 72 78 99

Picking the corresponding vertex labels from the graph we generate the sequence

15 38 18 37 28 14 210 36 38 28

From Table 1 the message is decrypted as **DOMINATION**.

IV CONCLUSION

The number of columns can be decided as per. We can construct a graph with 1 or 2 or….or 9 columns. So the number of possible ways of constructing any table 1! + 2! + … + 9! = 409113 (this value is only for columns, similarly we can arrange for rows also)

For each of these 409113 ways we can arrange the characters in any message M of length k in k! ways. So we can construct atleast 409113 (k!) distinct tables .

Numerous weighted graphs are available in public domain for various reasons. It is difficult to find the difference between a fake graph and the encrypted one. So the proposed method is safe for encryption of any message.

REFERENCES

[1]. http://en.wikipedia.org/wiki/Encryption.

[2]. M. Yamuna, B. Joseph Sasikanth Reddy, Nithin Kumar Reddy, Paladugula Raghuram, Genetic Code for Amino Acids using Huffman Trees, International Journal of ChemTech Research, 6(1), 2014, 53 – 63.

[3]. Wael Mahmoud Al Etaiwi, Encryption Algorithm using Graph Theory, Journal of Scientific Research & Reports, 3(19), 2014, 2519 – 2527.

[4]. http://en.wikipedia.org/wiki/Graph_%28mathematics%29.

[5]. http://en.wikipedia.org/wiki/Multigraph.

[6]. http://en.wikipedia.org/wiki/Independent_set_%28graph_theory%29.

[7]. http://en.wikipedia.org/wiki/Bipartite_graph.