Oblong Slant For Encryption And Decryption Of Coupled Scheme To Unique Route In Associated Grid

Thiagarajan Kittappa
Professor, Department of Mathematics, Rajalakshmi Institute of Technology, Chennai, vidhyamannan@yahoo.com

Aasif J
Department of Electronics and communication Engineering, K.Ramakrishnan College of Technology, Tiruchirappalli, India, aasif.rj1308@gmail.com

Jacob Raa. S
Department of Electronics and communication Engineering, K.Ramakrishnan College of Technology, Tiruchirappalli, India, jacobsanthanaraj2001@gmail.com

Anish Anburaja
Information Sciences, University of Maryland, Maryland, USA, anishanbu@gmail.com

Suriya Prakash N
Aptean India Private Limited, Bangalore , India, prakashsurya@gmail.com

Abstract

A graph is, informally, a collection of nodes connected by a collection of edges. A node in a graph is a location where two or more branches covered. One branch may occasionally be the sole one to connect to the other node. A branch is a line segment that connects two nodes. Encryption in cryptography is the process of encoding messages so that only authorized parties may decode them to read. Every day, more digital information is being produced and shared online. The quantity a result of the weaknesses in the network and software, the number of security assaults/threats has also increased. The proposed research, a novel algorithm to capture the data encryption and decryption technique by graph theoretical manner using hexadecimal values is proposed. To perform substitution, first-level encryption uses binary values. As the second level of encryption, Hexadecimal dialogue is utilized to achieve permutation. A key is not explicitly used by the algorithm to encrypt the data. Through experimentation, the algorithm has been validated.

Keywords: *Binary, Decimal, Hexadecimal, Minimal spanning tree, Octal, System tree.*

I. INTRODUCTION

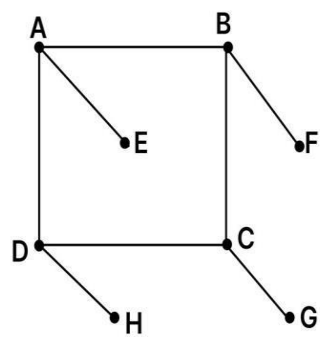
The study of points and lines is known as graph theory. It is concerned with how to link groupings of points called vertices using lines (or arcs) called edges. Graphs differ from the more familiar coordinate plots in this regard. A connected graph is one that is linked in the sense of a topological space, meaning that every point in the network has a path to every other point in the network.[1] A graph with no links is called a disconnected graph. A tree is made up of straight-line segments that are connected at their ends but do not have any closed loops (cycles). In technical terms, it is an acyclic network that is simple, undirected, and linked (or, equivalently, a connected forest). A graph can have numerous spanning trees in general, but a graph that is not linked will not have a spanning tree.[2-8]

A. Proposed procedures

Find adjacent matrix for the given connected network

Convert equivalent binary, octal and hexadecimal code for each row, or each columns of the constructed adjacent matrix which is explained in TABLE I for Case-1.

B. Case-1



Graph G=Corresponding is one of the tree of G is-

Let's take an example G! here vertex set, edge set are V= {a, b, c, d, e, f, g, h}and E= {ab, ad, ae, bc, bf, cd, cg, dh}

Let us consider:

G18t = [88,164,82,161,128,64,32,16]10 for each row there corresponding row matrix will be identified and generated uniquely.

TABLE I. CONVERTING TREE TO CODES

Name	binary Codes	Octal Codes	hexadecimal codes
G_{18t}	$[01011000,10100100, 01010010,10100001, 1000000,01000000, 00100000,00010000]_2$	$[130,244,122,241, 200,100,40,20]_8$	$[58,A4,52,A1,80\ 40,20,10]_{16}$

A minimal spanning tree is a graph subset with the same number of vertices as the graph and one less edge than the number of vertices.

A binary is converted to another number system, such as decimal, octal, or hexadecimal, changing numerical bases from binary to decimal, for example (decimal). This is referred to as base conversion. To convert a number in a binary number system (base-2) to a number in a decimal number system, use binary-to-decimal conversion (base-10). Understanding binary-to-decimal conversion is required for computer programming. Converting decimal to binary, To convert a base 10 number to a base 2 integer, we use simple methods. If 1210 is a decimal number, the binary equivalent is (10010110100)

The following python code will convert decimal number system to octal and hexadecimal number system.

C. Converting Decimal To Octal and Hexadecimal

```
def dec_to_bin_oct_hex(numbers):
    result = []
    for number in numbers:
        binary = format(number, 'b')
        octal = format(number, 'o')
        hexadecimal = format(number,
'x').upper()
        result.append({"Decimal": number,
"Binary": binary, "Octal": octal,
"Hexadecimal": hexadecimal})
    return result

decimal_numbers = input("Enter a list of
decimal numbers separated by space:
").split()
numbers = [int(x) for x in
decimal_numbers]
converted_numbers =
dec_to_bin_oct_hex(numbers)
for item in converted_numbers:
    print("Binary:", item["Binary"],
"Decimal:", item["Decimal"], "Octal:",
item["Octal"], "Hexadecimal:",
item["Hexadecimal"])
```

Consider a tree, and the entries were zeros and ones based on the location of the nodes of the adjacent matrix is created for each tree, and the tree has a unique name based on the graph diagram (G1t,G21t, etc..). The binary code forms the tree was derived from the acquired matrix. This binary code is used quickly build a tree network. The binary code may be transformed into octal and hexadecimal to simplify and obtain unique codes for the respective tree.

By employing this conversion easily and efficiently form a tree or network and utilizing the position of the node and the data obtained or derived from each tree and its node. Also, it can encrypt the value based on the number of nodes given in the structured manner of collecting the edges and its value merging all the binary codes obtained from the adjacent matrix. And separately the binary code in the given corresponding graph in tree structure on the capacity of all the nodes. The proposed procedure combining the binary values encrypted from decimal or any other by the above procedure, it can be changed into octal or hexadecimal etc... These formats can be used for security purpose and for communication.

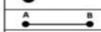
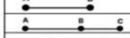
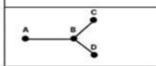
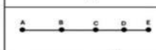
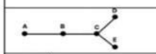
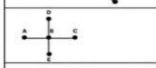
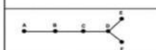
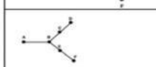
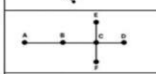
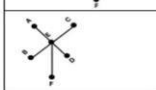
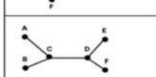
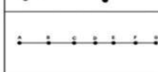
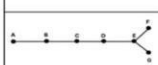
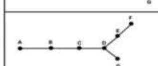
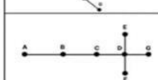
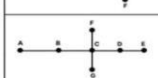
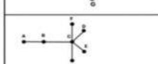
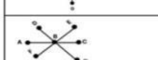
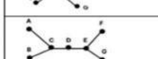
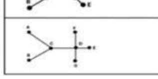
TABLE II. MST FOR HEXACODE

Name	Decimal Codes	Octal codes
G_{11t}	$[0,1]_1$	$[0,1]_8$
G_{12t}	$[1,2]_2$	$[1,2]_8$
G_{13t}	$[2,5,2]_3$	$[2,5,2]_8$
G_{41t}	$[4,10,5,2]_4$	$[4,12,5,2]_8$
G_{42t}	$[4,10,4,4]_4$	$[4,12,4,4]_8$
G_{51t}	$[8,20,10,5,2]_5$	$[10,24,12,5,2]_8$
G_{52t}	$[8,20,11,4,4]_5$	$[10,24,13,4,4]_8$
G_{53t}	$[8,23,8,8,8]_5$	$[10,27,10,10,10]_8$
G_{61t}	$[16,48,20,10,5,2]_6$	$[20,60,24,12,5,2]_8$
G_{62t}	$[16,48,20,11,4,4]_6$	$[20,60,24,13,4,4]_8$
G_{63t}	$[16,42,20,8,17,2]_6$	$[20,52,24,10,21,2]_8$
G_{64t}	$[16,42,23,8,8,8]_6$	$[20,52,27,10,10,10]_8$
G_{65t}	$[2,2,2,2,63,2]_6$	$[2,2,2,2,77,2]_8$
G_{66t}	$[8,8,52,11,4,4]_6$	$[10,10,64,13,4,4]_8$
G_{71t}	$[32,80,40,26,10,5,2]_7$	$[40,120,50,32,12,5,2]_8$
G_{72t}	$[32,80,40,20,15,4,4]_7$	$[40,120,50,24,17,4,4]_8$
G_{73t}	$[32,80,40,21,10,4,8]_7$	$[40,120,50,25,12,4,10]_8$
G_{74t}	$[32,80,42,20,8,17,2]_7$	$[40,120,52,24,10,21,2]_8$
G_{75t}	$[32,80,40,23,8,8,8]_7$	$[40,120,50,27,10,10,10]_8$
G_{76t}	$[32,80,47,16,16,16,16]_7$	$[40,120,57,20,20,20,20]_8$
G_{77t}	$[32,87,32,32,32,32,32]_7$	$[40,127,40,40,40,40,40]_8$
G_{78t}	$[16,16,104,20,8,9,2]_7$	$[20,20,150,24,10,11,2]_8$
G_{79t}	$[32,70,40,12,10,9,5]_7$	$[40,106,50,14,12,11,5]_8$
G_{710t}	$[32,32,104,20,11,4,4]_7$	$[40,40,150,24,13,4,4]_8$
G_{711t}	$[16,16,104,23,8,8,8]_7$	$[20,20,150,27,10,10,10]_8$

TABLE III. BINARY FOR ENCRYPTED CODES

Name	binary Codes	encrypted hexadecimal codes
G_{11t}	$[0,1]_2$	$[0,1]_8$
G_{12t}	$[1,10]_2$	$[1,2]_8$
G_{13t}	$[10,101,10]_2$	$[1,2,6]_8$
G_{41t}	$[100,1010,101,10]_2$	$[9,5,6]_8$
G_{42t}	$[100,1010,100,100]_2$	$[1,2,2,4]_8$
G_{51t}	$[1000,10100,1010,101,10]_2$	$[4,A,A,16]_8$
G_{52t}	$[1000,10100,1011,100,100]_2$	$[8,14,17,4]_8$
G_{53t}	$[1000,10111,1000,1000,1000]_2$	$[1,2,1E,4,8]_8$
G_{61t}	$[10000,11000,10100,1010,101,10]_2$	$[21,C,21,12,28]_8$
G_{62t}	$[10000,1100,10100,1011,100,100]_2$	$[10,6,32,25,32]_8$
G_{63t}	$[10000,101010,10100,1000,10001,10]_2$	$[4,A,2A,11,6]_8$
G_{64t}	$[10000,101010,10111,1000,1000,1000]_2$	$[8,15,17,22,8]_8$
G_{65t}	$[10,10,10,10,111111,10]_2$	$[A,2B,3E]_8$
G_{66t}	$[1000,1000,110100,1011,100,100]_2$	$[22,D,B,24]_8$
G_{71t}	$[100000,1010000,101000,11010,1010,101,10]_2$	$[10,28,28,6A,4C]_8$
G_{72t}	$[100000, 1010000, 101000, 10100, 1111, 100, 100]_2$	$[20,50,51,27,64]_8$
G_{73t}	$[100000, 1010000, 101000,10101,1010,100,1000]_2$	$[41,21,22,5A,46]_8$
G_{74t}	$[100000, 1010000,101010,10100,1000,10001,10]_2$	$[41,21,2A,48,46]_8$
G_{75t}	$[100000,1010000,101000,10111,1000,1000,1000]_2$	$[41,21,22,78,44]_8$
G_{76t}	$[100000,1010000,101111,10000,10000,10000,10000]_2$	$[41,21,3E,8,16,20]_8$
G_{77t}	$[100000,1010111,100000,100000,100000,100000,100000]_2$	$[1,2,5A,4,8,A,20]_8$
G_{78t}	$[10000,10000,1101000,10100,1000,1001,10]_2$	$[8,21,51,24,26]_8$
G_{79t}	$[100000,1000110,101000,1100,1010,1001,101]_2$	$[20,46,51,49,4D]_8$
G_{710t}	$[100000,100000,1101000,10100,1011,100,100]_2$	$[20,41,51,23,64]_8$
G_{711t}	$[10000,10000,1101000,10111,1000,1000,1000]_2$	$[21,6,45,71,8]_8$

TABLE IV. MST FOR HEXADECIMAL CODES

Graph	Name	Decimal Codes	Hexadecimal codes
	G_{11t}	$[0,1]_1$	$[0,1]_{16}$
	G_{12t}	$[1,2]_2$	$[1,2]_{16}$
	G_{13t}	$[2,5,2]_3$	$[2,5,2]_{16}$
	G_{41t}	$[4,10,5,2]_4$	$[4,A,5,2]_{16}$
	G_{42t}	$[4,10,4,4]_4$	$[4,A,4,4]_{16}$
	G_{51t}	$[8,20,10,5,2]_5$	$[8,14,A,5,2]_{16}$
	G_{52t}	$[8,20,11,4,4]_5$	$[8,14,B,4,4]_{16}$
	G_{53t}	$[8,23,8,8,8]_5$	$[8,17,8,8,8]_{16}$
	G_{61t}	$[16,48,20,10,5,2]_6$	$[10,30,14,A,5,2]_{16}$
	G_{62t}	$[16,48,20,11,4,4]_6$	$[10,30,14,B,4,4]_{16}$
	G_{63t}	$[16,42,20,8,17,2]_6$	$[10,2A,14,8,11,2]_{16}$
	G_{64t}	$[16,42,23,8,8,8]_6$	$[10,2A,17,8,8,8]_{16}$
	G_{65t}	$[2,2,2,2,63,2]_6$	$[2,2,2,2,3F,2]_{16}$
	G_{66t}	$[8,8,52,11,4,4]_6$	$[8,8,34,B,4,4]_{16}$
	G_{71t}	$[32,80,40,26,10,5,2]_7$	$[32,50,28,1A,A,5,2]_{16}$
	G_{72t}	$[32,80,40,20,15,4,4]_7$	$[20,50,28,14,F,4,4]_{16}$
	G_{73t}	$[32,80,40,21,10,4,8]_7$	$[20,50,28,15,A,4,8]_{16}$
	G_{74t}	$[32,80,42,20,8,17,2]_7$	$[20,50,2A,14,8,11,2]_{16}$
	G_{75t}	$[32,80,40,23,8,8,8]_7$	$[20,50,28,17,8,8,8]_{16}$
	G_{76t}	$[32,80,47,16,16,16,16]_7$	$[20,50,2F,10,10,10,10]_{16}$
	G_{77t}	$[32,87,32,32,32,32,32]_7$	$[20,57,20,20,20,20,20]_{16}$
	G_{78t}	$[16,16,104,20,8,9,2]_7$	$[10,10,68,14,8,9,2]_{16}$
	G_{79t}	$[32,70,40,12,10,9,5]_7$	$[20,46,28,C,A,9,5]_{16}$
	G_{710t}	$[32,32,104,20,11,4,4]_7$	$[20,20,68,14,B,4,4]_{16}$
	G_{711t}	$[16,16,104,23,8,8,8]_7$	$[10,10,68,17,8,8,8]_{16}$

II. CONCLUSION

Proposed procedure helps to secure transaction from one cloud networking to another. These methods engender unique code of anti-piracy in television broadcasting. Also the proposed technique of octal or hexadecimal code format is used to identify colours in image processing. Discussed procedures are providing unique authentication for secure backups in data analytics science area.

III. ACKNOWLEDGMENT

The Authors would like to thank Prof. Ponnammal Natarajan, Former Director-Research, Anna University-Chennai, her intuitive ideas and fruitful discussions with respect to the paper's contribution.

REFERENCES

[1] Bondy, J.A. Murty, U.S.R. Graph Theory with Application, vol. 290. Macmillan, London(1976).

[2] Gallian JA (2015) A dynamic survey of network labeling Electron. J. Comb.

[3] K. Thiagarajan, R. Satheesh Babu, K. Saranya "Construction of network and (I, J)-distance graph" in J---Appl. Sci Res ISSN 1819.

[4] Rajeswari V, Thiagarajan K (2018) "Graceful labeling of wheel graph and middle graph of wheel graph under IBEDE and SIBEDE approach". J Phys. 012078.

[5] Rajeswari V, Thiagarajan K (2017) SIBEDE approach for total graph of path and cycle graph in Middle-East J.Sci. Res. 25(7):15531558ISSN1999233©IDOSIPublication2017.https://doi/org/10.5829/idosi.mejsr.2017.1553.1558.

[6] Rajeswari V and K Thiagarajan (2018) Study on strong binary equivalent decimal edge graceful labeling. In Int. J. Pure Appl. Math. 199(10):1021-1030 ISSN: 1311-8080(Printed version); ISSN: 1314-3395(On-line version), special Issue.

[7] Rajeswari.V, Thiagarajan.K, Study on strong binary equivalent decimal edge graceful labeling. Int. J.Pure Appl. Math. 119(102018), 1021-1030. ISSN: 1311-8080.

[8] V. Rajeswari and K.Thiagarajan (2016) 'Study on Binary Equivalent Decimal Edge Graceful labeling' in Indian J.Sci Technol.9(S1).https://doi.org/10.17485/ijst/2016/v9iS1/1056 December, ISSN(Print): 0974-6846,ISSN(online):0974-5645.