Library Progress International
Vol.44 No.3, Jul-Dec 2024: P.19854-19862

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

Innovative Symmetric Encryption with Minimal Spanning Tree Approaches

[1*]MPR Murthy, [2]CH. Suneetha,[3] B.Nageswara Rao

[1]Research Scholar, GSS, GITAM University, Visakhapatnam, India
[2], Associate Professor, Department of Mathematics, GSS, GITAM University, Visakhapatnam, India
[3]Associate Professor, Department of Mathematics, School of Technology, The Apollo University, Chittooru Andhra Pradesh,India bendi151977@gmail.com.
nageswararao_b@apollouniversity.edu.in
*Author for Correspondence: mprmurthy75@gmail.com

How to cite this article: MPR Murthy, Suneetha, B.Nageswara Rao (2024) Innovative Symmetric Encryption with Minimal Spanning Tree Approaches. *Library Progress International,* 44(3), 19854-19862.

ABSTRACT
As digital communication becomes an integral part of daily life, the rise in cybercrime has become increasingly concerning. To safeguard messages from hacking and tampering, concealing original information is essential. Modern cryptography goes beyond simple message encryption, emphasizing the protection of encrypted data against cyber attacks. Besides enabling secure communication, modern cryptography also prioritizes message authentication and integrity. This paper presents a symmetric block cipher model that utilizes graph structures. The encryption process in this method occurs in two stages. In the first stage, the message is encrypted using the adjacency matrix of an undirected graph. The second stage then applies an exclusive OR (XOR) operation to the output from the first stage. Unique encryption and decryption keys are generated for each block and each round, derived through the Minimum Spanning Tree (MST) technique on an undirected graph.This approach enables multiple peers to exchange messages simultaneously. One of the participants selects random ASCII characters, equal in number to the block length, and uploads these to the cloud. All group members then generate block keys using a shared, confidential key generation technique.

Key Words: Encryption, Decryption, Minimum Spanning Tree (MST), Adjacency matrix

Introduction: Cryptography is the science of secret communication of sensitive information. Encryption protects the data from unauthorized entities. Cryptography is basically divided into two categories symmetric key (Private key) cryptography and Asymmetric key (Public key) cryptography. In symmetric key cryptography the encryption and decryption key is same, secret between the communicating parties. Extensive use of internet across the world leads to many cybercrimes like hacking or stealing the data, interrupting the data in its journey from one entity to the other. Simply communicating the encrypted data does not fulfill the aim of data protection. Modern cryptography not only encrypts the data but also provides authentication and integrity. It heavily depends on mathematical techniques like number theory, graph theory etc. Algebraic graph theory has several applications in cryptographyespecially in multivariate cryptography. Abundant research has been done in cryptography using Ramanujan graphs, Expander graphs etc. The present encryption scheme is based on adjacency matrix and Minimal Spanning Tree (MST) as a secret key.

Adjacency Matrix:

An indirected graph G(v,e) having V vertices, e edges is called cyclic if the edge from one vertex leads to the itself. The graph is said to be complete if there is an edge between all vertices. Adjacency matrix is square matrix whose entries are either zero or one. If there is connection between two vertices i and j then the corresponding element $a_{ij}=a_{ji}=1$. Otherwise the entry is zero. For an undirected graph adjacency matrix is symmetric denoting the connection between the vertices.

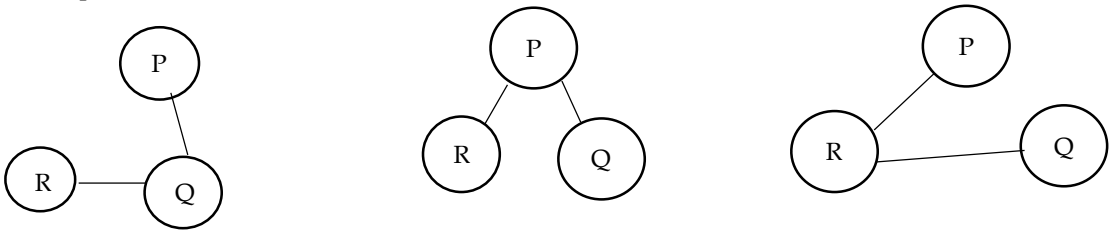
In general, an adjacency matrix of order l is represented as

$$\begin{bmatrix} 0 & K_{12} & K_{13} & \cdots & K_{11} \\ K_{21} & 0 & K_{23} & \cdots & K_{21} \\ K_{31} & K_{32} & 0 & \cdots & K_{31} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ K_{11} & K_{12} & K_{13} & \cdots & 0 \end{bmatrix}$$

Minimum Spanning Tree:

The subset of a graph G covering all vertices using minimum possible number of edges is called a spanning tree. So, a spanning Tree has no cycles and cannot be disconnected.

Example:



are three different spanning trees of a complete graph. For a complete un directed graph with p nodes has p^{p-2} spanning trees having same number of vertices and edges. An undirected graph with p nodes has (p-1) edges in spanning tree. A spanning tree having minimum weight is called Minimum Spanning Tree(MST) of a weighted graph. In practical the edge weight is the distance between two nodes [1]. Minimum Spanning Trees(MST) have several applications in telecommunication and transportation networks, cluster analysis ect. [2]. Several researchers used Minimal Span Tree(MST) for encryption algorithms.

Literature Survey: Graph theory one of the branches of mathematics has wide range of applications in cryptography both for encryption and key transmission. V.A. Ustimenko [3] suggested a multivariate cryptography using polynomial affine space in a finite commutative ring. NTokareva [4] found how graph theory is related to cryptography. There the authors used some graph structures and bent functions to develop mobile networks. Wael Mahmoud Al Etaiwi [5] suggested a new complex cipher basing on cyclic graphs, complete graphs and minimum spanning tree. P. Amudha et.al. [6] surveyed many graph theory applications of cryptography. That chapter proposes an encryption algorithm using Euler graphs. Yamuna et.al [7] suggested a double encryption technique using Hamilton path and complete graph. Steve Lu et.al. [8] applied graph theory to image cryptography by assigning nodes and edges to images. Charles et.al. [9] designed new collision resistant hash functions from expander graphs. AnmariaCostache et.al. [10] studied the security of a proposal for post quantum cryptography using super singular isogeny graphs. Later, Hyungrok Jo [11] introduced cryptographic hash functions based on Charles expander graph in post quantum cryptography. Considering the literature on graph theory applications in cryptography the present paper explains an innovative encryption technique using adjacency matrix of an undirected weighted graph. This algorithm uses Minimum Span Tree (MST) technique for key generation.

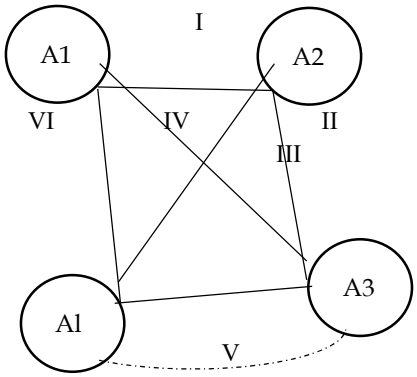
Proposed Method:

The proposed method describes a double encrypted symmetric block cipher. The first stage encryption is done by using adjacency matrix of an undirected graph. At the second stage all the first stage encrypted entries will undergo logical XOR encryption with Minimum Span value of the MST of the corresponding block. Each block is encrypted two times using different keys for different blocks.

Key Generation and Key Scheduling Algorithm:

In symmetric key cryptography (Private key cryptography) the strength of the cipher relies on the key. A key in symmetric key cryptography must be un guessable to the attacker. Key generation and key management

are most vital aspects in cryptography. Here the key generation technique is secret among the communicating parties. If a group of multiple peers want to communicate messages, one of the members randomly select some ASCII characters whose numbers is same as the length of the block and upload all the characters to the cloud as master key. All the peers of the group generate keys for first, second and sub sequent blocks. Here the block length is agreed upon the communicating parties. Suppose agreed upon block length is l (say) then one of the group members selects l random characters from ASCII table A1 A2 A3- - -Al, publish or upload to the cloud. All the members of the group generate secret keys as follows. All the characters are written as vertices of an undirected graphand joined with edges to complete the graph. The edge weights are the differences between ASCII decimal values of the characters A1 A2 A3- - -Al measured from left to right cyclically.

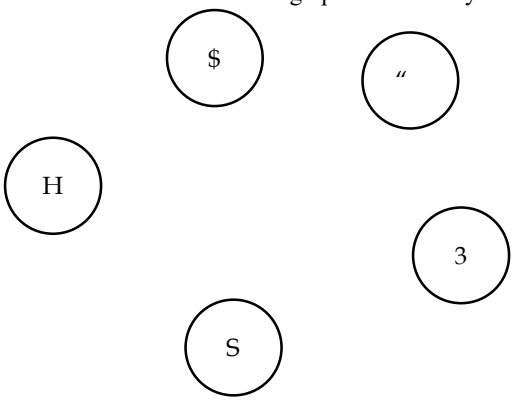


The edge weights I,II,III,- - - are calculated as the differences of ASCII decimals of A1 A2 A3- - Al from left to right.

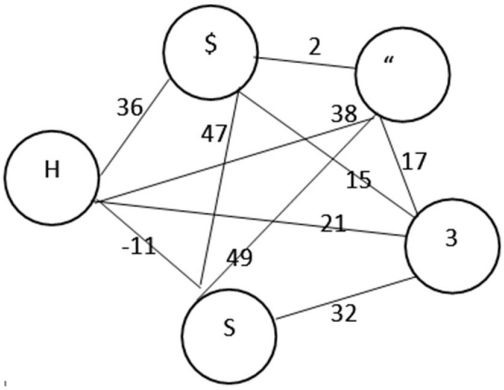
A1 A2 A3- - -Al

Edge weight I = ASCII dec [A2-A1]
Edge weight II = ASCII dec [A3-A2]
Edge weight III = ASCII dec [A3-A1]
- - - - - - - - - - -- - - - - - - - - - - - - -
Edge weight VI = ASCII dec [A1-Al]

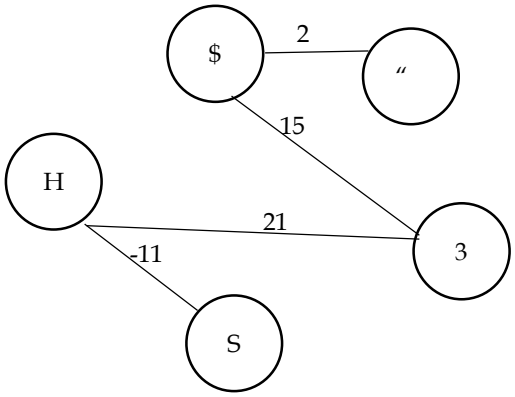
For this complete undirected weight graph minimum spanning tree is formed by using either Prims algorithm or Kruskal's algorithm. An adjacency matrix of the Minimum Spanning Tree is formulated. It is a symmetric matrix where all the principal diagonal elements are zeros. Replace the diagonal zeros with 1 for the first block key matrix, 2 for the second block key, 3 for the third block key and so on. For instance, if the agreed upon block length is 5, one member of the group selects randomly the ASCII characters $, ", 3, S, H. Then these characters are represented as adjacent vertices of an undirected graph successively from left to right



All the vertices are joined with edges to complete graph.



In the above example the minimum spanning tree is



With minimum spanning tree weight 27

The adjacency matrix of the Minimum Spanning Tree is

	$	“	3	S	H
$	0	2	15	0	0
“	2	0	0	0	0
3	15	0	0	0	21
S	0	0	0	0	-11
H	0	0	21	-11	0

The elements on the principal diagonal are replaced by 1 for the first block key K1

K1=

1	2	15	0	0
2	1	0	0	0
15	0	1	0	21
0	0	0	1	-11
0	0	21	-11	1

Same Minimum Spanning Tree procedure is applied for finding the adjacency matrixK2 of second block considering the next ASCII characters to the first block key. Third and subsequent blocks the keys K3K4……are constructed by using the successive ASCII characters of previous block keys. For example if ASCII characters are $ " 3 S H for the first block key generation then for the second block the characters are % # 4 T I and so on. The key space here is 255. When the successive characters are considered for blocks key generation process, for a big data having many number of blocks the key for first block and 256th block coincides. Suppose there is same information in both the blocks the cipher is also same. So, there is possibility of cipher text attack. To overcome this, we choose the base as 65536 (256x256)

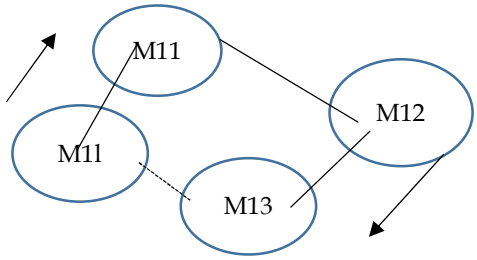
ENCRYPTION:

If two communicating parties start transmitting messages, first they agree upon to use block length and master key length as l (say), then both the uses generate the secret keys K1,K2…….Kn using the ASCII characters uploaded to the cloud. The whole message is split into blocks M1 M2……..Mn.Since the present algorithm is symmetric encryption algorithm both encryption and decryption start from the first block. All the characters of each block are written as vertices of an undirected graph.

M11 M12 M13…M1l
M21 M22 M23…M2l
……..
Mn1 Mn2Mn3…Mnl

Where n isnumber of blocks nϵN and l is the agreed upon block length.

All the characters are represented as the vertices of an undirected graph.



The edge weights of the complete graph of all the blocks are calculated using the same procedure as in key generation algorithm, i.e., the differences of ASCII decimal equivalents taken in the counter clockwise direction. Adjacency matrices of each block are foundas D1,D2,…… Dn, of order l ×l

First Stage Encryption:

At the first stage the adjacency matrices of the undirected graphs of all the data blocks are multiplied with the corresponding keys to get the matrices D1(I) D2(I)…….Dn(I)

Dn (I) = Dn∗Kn n is number of block n=1,2,3……..

Each element of the l × l square matrix Dn(I) is reduced to mod 256 to get the matrix Dn(II).

Dn(II) = Mod[Dn(I),256],n = 1,2……..

The elements of the matrix Dn(I) when reduced to mod 256 split into two parts the integer part and the residue parte.g., 521 when reduced to mod 256 the integer part is 2 and the residue is 9.

521=(2∗256)+9

So, the matrix Dn(I) when reduced to mod 256 divides into two parts the integer matrix In and Dn(II) where n=1,2,3……..

Second Stage Encryption:

All the elements of the first stage encrypted matrices D1(II),D2(II)……Dn(II)are converted to ASCII equivalent 8 bit binary numbers. Each binary element is undergone logical XOR operation with the ASCII binary equivalent of Minimum Span Value (total value of the edge weights) of Minimum Spanning Tree computed to generate the first block key resulting D1(III),D2(III)……Dn(III). For instance, if the minimum span value of the first block is 27 then each binary element of the first block first stage encrypted matrix is Xored with ASCII binary equivalent of 27. In case if the Minimum Spam Valueis outside ASCII range then mod256 value is considered.

Dn(III)=Dn(II) XOR(MSV Kn) n=1,2,3……..

All the two stage encrypted data blocks are converted to equivalent ASCII characters which constitutes the cipher text C.

The cipher text C is communicated to the receiver via public channel. Along with the cipher the elements of the integer matrices are also communicated as string of integers. Since the agreed block length is l a plain text block of l characters are encrypted to l^2 cipher characters and l^2 integers. To authenticate the message the sender inserts ASCII decimal values of first characters of each block at the positions l^2 +1,$2l^2$+1,$3l^2$+1,……. of both cipher and integer string.

Decryption:

The receiver after confirming the authentication of the sender splits the cipher text C into blocks of length l^2 characters each leaving the decimal values at l^2 +1,$2l^2$+1,$3l^2$+1,……. Now the cipher blocks are C1 C2……. Cn. Decryption takes place at two different stages, the first stage using logical XOR and the second stage using adjacency matrix.

First Stage Decryption:

All the l^2 characters of each block are converted to ASCII binary equivalents, written as $l \times l$ matrix and each binary number is undergone logical XOR operation with Minimum Span Value (MSV) of the Minimum Spanning Tree(MST) obtained while generating the key K1 for first block encryption/decryption.

$D_n(II) = Cn \text{ XOR } MSV(Kn)$ n = 1,2,3,……

Second Stage Decryption:

Then the 8 bit binary numbers of each block are converted to ASCII decimals.The integer string received along with the cipher text C is split into l2 each leaving the decimal values at l^2 +1,$2l^2$+1,$3l^2$+1,……., represented as arrays In n = 1,2,3… of order lxl. Each array is multiplied with 256 and corresponding elements of Dn(II) are added to get the matrices Dn(I).

$Dn(I) = (256 * I_n) + Dn(II)$ n=1,2,3……

$Dn = Dn(I) * (Kn)^{-1}$ n = 1,2,3…….

D_n,n=1,2,3…….. are adjacency matrices of data blocks [adjacency matrices of weighted undirected graphs]. After getting adjacency matrices of all data blocks original message blocks are determined using the decimal equivalents of first characters of each block placed at

l^2 +1,$2l^2$+1,$3l^2$+1,…… positions in the cipher. Consider the adjacency matrix of first data block D_1 and the ASCII decimal of first character of first block. Suppose the first character of the first block is say '83'.

The first block entries are S M12 M13 …….M1l. Using adjacency matrix D_1, the first entry S, other characters M12 M13 ……M1l are determined.

$$D_1 = \begin{matrix} 83 \\ M_{12} \\ M_{13} \\ \vdots \\ M_{1l} \end{matrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1l} \\ a_{21} & a_{22} & a_{23} & & a_{2l} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3l} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{l1} & a_{l2} & a_{l3} & \cdots & a_{ll} \end{bmatrix} \begin{matrix} 83 & M_{12} & M_{13} & \cdots M_{1l} \end{matrix}$$

a12 = ASCII Decimal of [M12-83]

therefore, M12 = a12+83

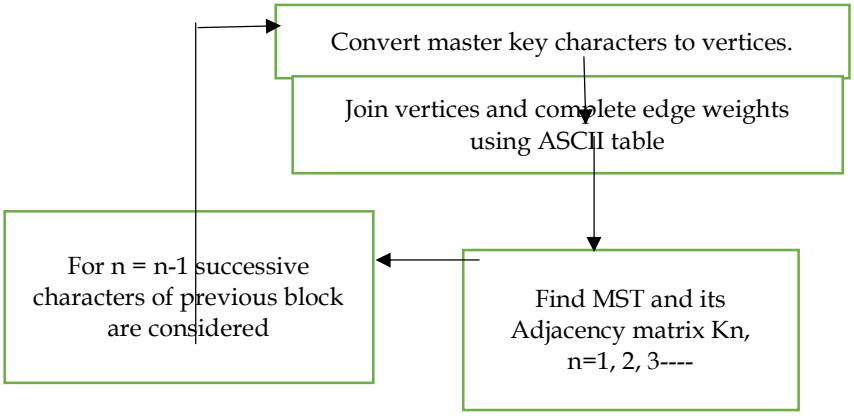
M13 = a13+83

⋮ ⋮

M1l = a1l+83

Same procedure is applied for all the blocks using the adjacency matrices D2,D3,-----,Dn using first characters of all the blocks sent by the communicator to get the original message.

Key Generation Algorithm



Encryption	Decryption
Step 1. Split message into blocks	Step1. Divide cipher into block matrices
Step 2. Denote characters as vertices	Step 2. Convert to ASCII binary
Step3. Complete the graph with edge weights	Step 3. Dn(II) = Cn XOR MSV [Kn] n = 1,2,3…
Step4. Compute adjacency matrix Dn(I) n = 1,2,3….	Step4. Change to ASCII decimals
Step 5. Dn(II) = Mod [Dn(I) , 256] n =1,2,3…	Step 5. Dn(I) = 256 * In + Dn(II) n = 1,2,3…
Step 6. Convert Dn(II) to ASCII binary	Step 6. Dn = Dn(I) * Inv[Kn] n = 1,2,3…
Step 7. Dn(III) = Dn(II) XOR MSV [Kn] n = 1,2,3…	Step 7 Get plain text from adjacency matrix
Step 8. Cipher text C, Integer string I	

Example:

If a group of communicators want to transmit messages, they agree upon to use the block length as 5 characters(say). One of the entities Alice selects randomly 5 ASCII characters “S A I !# ”and publish these characters in the cloud as public key. All the members of the group generate the keys for first, second and subsequent blocks using this master key and above-mentioned key generation, key scheduling algorithm.

If two legitimate users Alice and Bob want to communicate messages, both Alice and Bob establish the keys. If Alice wants to send the message GITAM/UNIVE/RSITY, she divides it into three blocks each having 5 characters since the master key length is 5.

She writes all the characters of each block as vertices of un directed graphs. The graphs are completed joining all vertices with edge weights differences of ASCII decimal values. Each block is encrypted twice using the keys K1, K2, K3 at first stage and using logical XOR operation with minimum span values of K1, K2, K3 at the second stage. The decimal equivalents of first characters of each block G →71, U → 85, R→ 82 are inserted at 26th, 51st, 76th positions of both cipher text and integer string to authenticate the communication. After two stages encryption the cipher becomes

Z”«äP„VGSŠòE=®Ußì®»rzøÒO.¶ 71 6$¾Tžò>DLEfJæÀSYNÌ•²î¼v:Ö¸®ˆp 85 ^ußbcá¸²‰>_ÒÀ§üÔéWÂì©Ì 82.

The integer string is

0,0,0,3,0,0,1,1,3,1,5,2,2,4,3,3,1,1,5,1,3,2,2,1,2,71,5,2,1,2,0,2,1,1,1,3,0,0,2,0,1,0,2,1,2,2,0,1,2,1,6,85,

1,0,1,1,0,1,1,1,0,0,1,1,2,1,0,1,1,0,2,1,3,2,1,4,1,82

The cipher and the integer string are sent to Bob via public channel

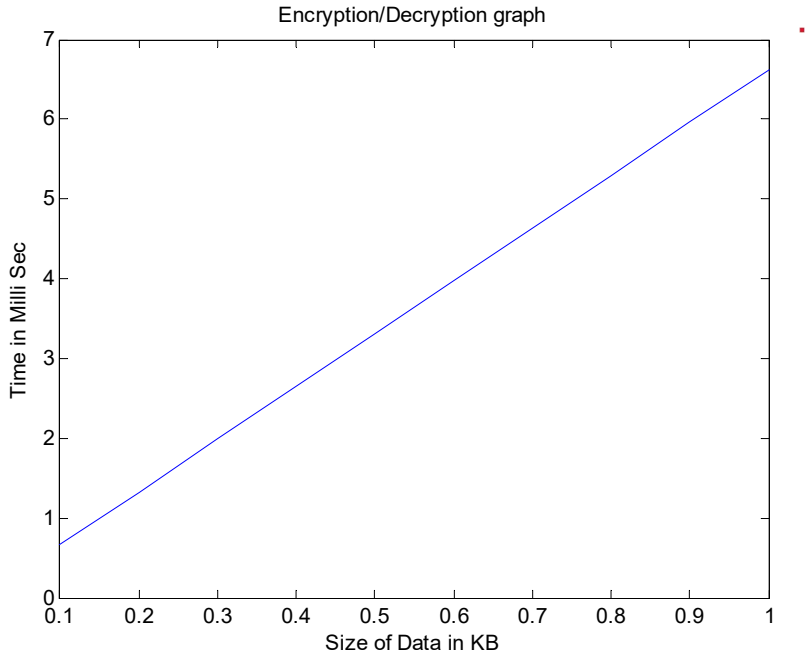
Security Analysis and conclusions

The Present block cipher uses Minimum Spanning Tree (MST) and Minimum Spanning Value(MSV) of an un directed weighted graph as secret key. Secret keys are not transmitted among the users rather they are generated by all the legitimate users of the group using key generation and scheduling algorithm. For symmetric ciphers key distribution and key safeguarding are most difficult aspects. But, in the present algorithm only the master key characters are open to all. Inner keys for all the blocks are generated by the users using the key

generation algorithm. This provides the key at most security unless it is compromised. Message to be sent is encrypted at two different stages using generated keys. For each block key is different. In first example "GITAMUNIVERSITY" the alphabet I is repeated thrice, T is repeated twice. But in cipher repetition of the characters is very less. In second example same message blocks 'GITAMGITAMGITAM' are mapped to Z"«äP„VGSŠòE=®Ußi®»rzøÒO.¶ 2BLb*z6à&ò.xÆiFzF¬ÖÒLF^„Nâ öp-RSd(êgHŽ¡áL!USpb±,SYN|–õ ². For a plain text with n characters, the resulted cipher text has n^2 characters. The decimal equivalents of first characters are inserted at different selected positions of both cipher and integer string. If the decimals are same at those positions of both cipher and integer string, then then receiver confirms that the cipher is not interrupted in between its journey from sender to receiver. This provides the authentication and integrity of the message. Since the original message and cipher are of different sizes, linear cryptanalysis and differential cryptanalysis are tough to implement in the present cipher. Though the cipher text appears lengthy the computational overhead is same as encryption.

Performance Analysis: The execution time for different size data is calculated on Intel i5,64 bit processor with 16 GB ram using MATLAB 14. The following table and give the encryption/decryption time for different size data.

Size of data in KB	Time in Milli Sec
0.1	0.662
0.2	1.256
0.3	1.991
0.4	2.449
0.5	3.457
0.6	3.972
0.7	4.634
0.8	5.289
0.9	5.893
1	6.62



The encryption/decryption graph is linear. For 1MB data the execution time is roughly 6 milli seconds which is very less when compared with conventional block ciphers.

Further Scope of the work: Preferably, the cipher and the original message are supposed to have almost same size to manage the storage space and communication time. Improvements can be done to minimize the cipher size. One of the ideas in this direction is to use the cipher block chaining mode after the first block encryption. This work will be extended in future.

References:

1. Tutorial point.com/data structures _algorithm / spanning _tree.htm.
2. personal.utdallas.edu/~ besp / teaching/ mst.applications.pdf
3. V.A. Ustimenko, "On algebraic graph theory and non-bijective multivariate maps in cryptography", **DOI:** http://dx.doi.org/10.15439/2018F204
4. Natalia Tokareva, "Connection between graph theory and cryptography G2C2:Graphs and groups, Cyclic and Coverings, Sep 2014, Novosibirsk Russia.
5. Wael Mahmoud Al Etaiwi, "Encryption algorithm using graph theory", Journal of scientific research and reports, 3(19) 2519-2527, 2014.
6. P. Amudha, A.C. Charles Sagayaraj, A.C. shantha Sheela, "An application of graph theory in cryptography" , International journal of pure and applied mathematics IJPAM, 119(3), 2018, pp 375-383.
7. Yamuna M, MeenalGogia, Ashish sikka,Md.Jazib Hayat Khan, "Encryption using graph theory and linear algebra", International journal of computer applications IJCA, 2012.
8. S Lu, Rafael Ostrovsky, Daniel Manchala, "Visual cryptography on graphs", Cite Seeix COCOON, 2008, 225-234.
9. Denis X Charles, Eyal Z. Goren, Kristic E. Lauter, "Cryptographic hash functions from expander graphs", Journal of Cryptology, 22, 93-113, 2009.
10. AnmariaCostache, Brooke Feigon, Kristin Lauter, MaikeMasierer, Anna Puskar, "Ramanujan graphs in cryptography", arXiv:1806. 05709V2 [math.NT]18th December 2018.
11. Hyungrok Jo, "Ramanujan graphs for post quantum cryptography", http://www.researchgate.net/publication/337150777.