RESEARCH PAPER

Hybrid Graph Theory and Cryptography: Edge Bimagic Mean Labeling for Secure Encryption and Decryption

P. Divya[1], Prof. Dr. R. Nagarathinam[2*]

[1]Assistant Professor, Dr. M.G.R Educational and Research Institute, Chennai -600095. Email - divya.math@drmgrdu.ac.in
ORCHID author :- 0009-0000-7592-2315
[2*]Dr. M.G.R Educational and Research Institute, Chennai -600 095, Chennai -95. Email - nagarathinam.hs@drmgrdu.ac.in

***Corresponding Author:** Prof. Dr. R. Nagarathinam,
*Dr. M.G.R Educational and Research Institute, Chennai -600 095, Chennai -95. Email - nagarathinam.hs@drmgrdu.ac.in

Abstract :
Data security still trips up a lot of standard cryptographic methods—they struggle to stay both user-friendly and complex enough to actually keep sensitive information safe. This paper introduces something different: a hybrid approach that weaves together graph theory and cryptography. The idea to make encrypting and decrypting private data faster, smarter, and more secure. We take personal data and represent it as a graph—a collection of nodes and edges, built specifically for each individual's information. Think of each person's confidential data like their own special graph, maybe a complete one, maybe Hamiltonian, with all the data mapped out as connections. Then we go a layer deeper and turn these graphs into algebraic matrices. By assigning weights to the edges or giving each graph a unique label, the information gets buried under several layers—much harder for outsiders to make sense of. Key generation is where this approach gets especially creative. We use bimagic mean labeling on the graph edges, along with a twist: "slanted" graph structures. Together, these elements produce unique keys designed specifically for encryption. Any text you want to encrypt first turns into a numeric value and gets woven into the graph, either as a weight or another numeric/algebraic tweak based on the generated key. After all this, the result is a graph and a matrix that mean absolutely nothing without the matching decryption key. If someone tries to peek at the data, they just end up with a mess—no way to trace things back to the original info. So, this study lays out a new hybrid model: blending the math-heavy precision of graph theory with classic cryptography, giving us a flexible way to protect confidential data and work around some of the old hurdles that still trip up traditional methods.

Keywords:
Graph theory, edge bimagic mean labeling, slantting graphs, ASCII value, binary value, cryptography, encryption, and decryption
How to cite this article: Divya P, Nagarathinam R., Hybrid Graph Theory and Cryptography: Edge Bimagic Mean Labeling For Secure Encryption and Decryption. Int J Drug Deliv Technol. 2026;16(43s): 1177-1186; Doi: 10.25258/Ijddt.16.43s.123.

Introduction:-
Graph theory is one of the central components of mathematics. Graph theory deals with graphs. Graphs are used as simple models. They indicate relationships between objects. A graph consists of points referred to as vertices. They are connected by lines referred to as edges. Vertices can be imagined as dots. Edges as lines connecting them. Graphs assist in charting relationships between things. Graphs are useful in numerous fields. Science applies them to analyze networks. Computer scientists construct programs out of them. Biologists monitor cell connections. Social scientists map friendships. Engineers plan circuits. All these disciplines use graphs to manage difficult configurations. You can use math terms to describe a graph. Let's call it G equals V comma E. V stands for a set of vertices. It should have at least one vertex. E enumerates the edges. Every edge connects two vertices. This organization makes it simple and precise. Graph theory has its origin long, long ago. In the 1700s, Leonhard Euler began it. He solved the Seven Bridges of Königsberg problem. That city possessed seven bridges along a river. Folks questioned whether you could cross each once and go back to the beginning. Euler demonstrated that it was not possible. His work established foundational concepts. Such as paths and loops. Those fundamentals still hold us today. Such as his demonstration of a proof using counts at every land location. That concept of degree—how many edges converge at a vertex—is still at the core. Graph theory today encompasses broad subject matter. It discusses various graph types. Undirected graphs possess edges which travel in both directions. Such as a highway that you can travel either direction. Directed have one direction. Consider one-way streets. Weighted graphs put numbers on edges. These depict weights or distances. Bipartite graphs divide vertices into two sets. Edges only connect between sets. Not within them. Fundamental characteristics include degree. That is the edge number for each vertex. Connectivity examines whether paths connect all segments. Otherwise, the graph divides. Graphs appear in numerous shapes. Adjacency matrices employ grids. Rows and columns are used to list vertices. A one indicates an edge. Zeros indicate none. Adjacency lists associate each vertex with its neighbours. These instruments accelerate computer work. All this gives rise to intelligent algorithms. They find actual solutions. Networks require shortest paths. Like routes found by GPS. Cycles assist in identifying loops in data. Paths organize deliveries. For example, in traffic apps, graphs represent roads as edges. Cities as vertices. Algorithms reduce travel time. In practice, graph theory assists large tasks. Routing finds

*Author for Correspondence: nagarathinam.hs@drmgrdu.ac.in

optimal routes for data packets in the internet. Scheduling allocates jobs without conflicts. Data structures hold information connected by relations. Cryptographic protocols encrypt messages based on graph patterns. An example is from social media. Sites utilize graphs to recommend friends. They identify shared connections. In biology, graphs represent protein bonds. It assists drug design. Graph theory combines solid math foundations with wide applications. It powers pure math proofs. It also powers applied solutions. That is why students and professionals learn it. This summary emphasizes its foundational concepts and worth in mathematics. Cryptography protects conversations and information. Cryptography converts ordinary information into encrypted code. Only people with the key can access it. One of the central deceptions in this area is the XOR trick. XOR operates on bits. Bits are small units of data. Zero or one. It requires two bits. Outputs one if they're different. Zero if they're identical. That's it. XOR excels in crypto for certain characteristics. It is reversible. XOR the same key twice. You return to the beginning. It rearranges order without modification. Include a third bit. Combine them in any manner. Result remains unchanged. These are benefits which make math quick and simple. In XOR encryption, begin with plaintext. That's your message in bits. Combine it with a key. The key is the same length. Bit by bit, perform XOR. Out comes ciphertext. It appears as noise. Random bits. To decrypt, take that ciphertext. XOR it again with the key. Self-reverse magic and plaintext appears. This arrangement employs a single key for both stages. Symmetric is shared secret. No slick pairs like other systems. Simple to code. Fast to execute. Suitable for small devices. Such as smart cards or sensors. They require light security without burdensome power. But strength binds to the key. Make it length. Match message size. Never repeat patterns. Keep it secret. Then, it repels attacks. Frequency checks fail. Attackers cannot guess common letters. Like E in English. XOR conceals that. Bad keys are bad news. Short ones are repeats. Patterns are predictable. Use same across files. Then, cryptanalysis breaks it. Software detects repeats. Crack the code quickly. In example, in older systems, short keys allow hackers to guess within hours. Long random ones last years. XOR constructs larger crypto instruments. Flips bits within blocks. Creates foundation for ciphers such as AES segments. Or stream codes in programs. Provides fast bit shifts. Critical to secure web conversations. Email protectors. Bank transfers. Essentially, XOR enables safe data flow everywhere.

a. Background on Graph Theory and Cryptography

• Brief overview of graph theory and its applications

Graph theory studies or creates a network using nodes (or vertices) that are linked together through edges. Each edge represents the link between a pair of nodes. The vast potential of this network makes it easy to apply to many fields. In Computer Science, for example, Graph Theory can be used to develop communications networks for transmitting data to perform tasks and create effective algorithms. In Biology, Graph Theory provides a foundation to model the molecular structures of cells, understand how diseases spread throughout an organism and the environment, and provide a model of the interactions within ecological networks. In the Social Sciences, Graph Theory can model the flow of information and the spread of influence from one person or group to another through social networks. Graph Theory provides a foundation for creating a model of the real-world challenges we face today using technology and nature to connect with other people through society.

• Introduction to cryptography and its importance in modern communication

Cryptography is the science and practice of making communication secure by transforming information into a format that only authorized entities can read. This is crucial for modern communication since it protects the confidentiality, integrity, and authenticity of information sent across insecure channels, such as the internet. Cryptography encodes plain text messages with the help of encryption algorithms and keys, rendering them unreadable ciphertext. This keeps information private and secure from unauthorized access and interception. Information in this encoded manner offers security for sensitive information concerning personal messages, financial transactions, and government communications. In addition, ensuring integrity-that changes due to unauthorized sources must be detectable-is another important feature of security enabled by cryptography. It also can provide a mechanism for authentication that enables the people or entities involved in the communication to verify their identities. These properties-confidentiality, integrity, and authentication-operate in concert to engender an environment of trust and assurance between parties engaging in digital transactions. Today, cryptography plays an enabling role in fostering confidence and security between organizations and the people they interact with. Our lives are filled with data, and making that data private and secure is increasingly important in the interconnected world we find ourselves.

• Connection between graph theory and cryptography

Graph theory is significant in cryptography, as it introduces mathematical structures and hard computational problems essential for secure designs of cryptographic systems. In this case, messages or data can be modeled as vertices in a graph, with the relationships or transitions between the elements corresponding to edges. By modeling messages with graphs, possible encryption algorithms can be developed, including encoding messages by connecting nodes through the use of spanning trees which are cycles in the graph. This provides a way of ensuring secure data is communicated. Examples of additional graph-theory methodologies include hashing functions with expander graphs, which are sparse, highly connected graphs, zero-knowledge proofs, public-key cryptography, network security analysis through the use of attack graphs, and trust networks. Overall the combination of mathematical rigor and structures found in graph theory provides a versatile and powerful toolkit to advance the research in cryptography from advancing encryption and authentication to secure routing to detecting intrusions. Generally, graph theory supplies a toolkit to advance cryptography.

b. Objectives of the Study

• Significance of combining graph theory and cryptography

The inclusion of graph theory with cryptography is important because it provides a rich mathematical structure for designing secure communication systems, which are demonstrated to be better suited for modern digital environments. Graph theory offers the capability to model both entities and relationships using nodes and edges; therefore, network topologies, attack paths, and cryptographic structures may all be modelled in a consistent way employing graph theory. This modelling utilizes problems in graph theory that are algorithmically hard, for example, graph isomorphism and Hamiltonian paths, for the development of cryptographic systems that withstand classical and quantum attacks. Furthermore, the graph theory approach strengthens intrusion detection, secure routing, and authentication through the structural and visual inspection of vulnerabilities and trust relationships. The use of graph complements, adjacency matrices and optical properties of spectral graphs enhance encryption techniques to provide resilient deterrents against advanced cyber threats to enable scalability and effectiveness of security solutions to satisfy increasing requirements on data transmission and storage of connected systems.

Research Gap :-

The unexplored territory between graph theory and cryptography is grounded in the pressing demand for quantum-resistant cryptosystems that would be secure even during the time of quantum computing coming up. Despite graph theory's capacity for creating robust encryption and secure communication methods, many of the old ways encounter difficulties like high computational complexity, being not so efficient in large or dynamic network environments, and finally, being susceptible to quantum attacks. Quantum algorithms, particularly Shor's algorithm, threaten classical cryptography such as Elliptic Curve Cryptography (ECC), because these algorithms can solve the underlying mathematical problems of ECC in polynomial time. The situation created is that we are now forced to look for new graph theoretic ways that will not only assure security but also keep up with the speed of the current situation.

Besides, there is not much research that targets the designing of graph-based cryptographic systems in a manner that they will be easy to understand and at the same time adaptable to the ever-changing network conditions and evolving cyber threats. The present graph models are generally not equipped to work with dynamic temporal data, which is very important for the modern security environment. The shift to quantum-resistant cryptography also brings about challenges related to the compatibility with the existing digital infrastructure and the extra computational cost.

In this regard, lattice-based cryptography, which is based on hard mathematical problems related to the geometry of numbers and can be linked to graphs, is regarded as a very attractive option. On the other hand, more studies are necessary to thoroughly marry graph theory with these post-quantum schemes for the purpose of producing scalable, efficient, and clear encryption methods. It is of utmost importance to deal with these issues in order to pave the way for advanced cryptographic systems that will be able to assure users.

ASCII Value :-

The ASCII value is a numerical code assigned to each character in the American Standard Code for Information Interchange (ASCII) system. ASCII is a standard that allows computers and electronic devices to represent text by assigning a number to every letter (both capital and lowercase), number, punctuation mark, or control character. In the common 7-bit ASCII, the value of each character is a number from 0 to 127. This setup enables computers to store and handle text using binary code. For instance, the capital letter "A" has an ASCII value of 65, while the lowercase "a" has a value of 97. ASCII creates a standard for communication and information transfer between computer systems. It transforms character representations, which are simple for people to read, into numeric codes that machines can recognize. ASCII is a vital standard for programming, data transmission, and file storage because it establishes a common ground for text and binary processing in a computer.

Uppercase	Value		
A	65		
B	66		
C	67		
D	68		
E	69		
F	70		

Uppercase	Value		
G	71		
H	72		
I	73		
J	74		
K	75		
L	76		
M	77		
N	78		
O	79		
P	80		
Q	81		
R	82		
S	83		
T	84		
U	85		
V	86		
W	87		
X	88		
Y	89		
Z	90		

E. Theoretical Framework

a. Graph Theory Fundamentals

• Definition and basic concepts of graphs

A graph is a mathematical structure consisting of a set of objects, known as vertices or nodes, and links or edges, between pairs of vertices. Technically, a graph G = (V, E) is a pair, specifically where V is a non-empty collection of vertices and E are directed or undirected connections between those vertices. Graphs are used in mathematics and computer science to represent and model the relationships, or interactions, between distinct entities. In graphs, we begin by connecting points, which are called vertices, with lines, which are called edges. Graphs allow us to graphically represent data in an organized fashion with vertices representing distinct entities and edges representing connections among those entities. One can have undirected graph (meaning a symmetrical connection between 2 vertices) or directed graph (meaning a directed connection from one vertex to another). Graphs are a fundamental object in discrete mathematics and are of use in many different fields.

B. Basic Definitions:-

B1) Definition: Edge bimagic mean labelling

Let G be a finite, simple, undirected graph with vertex set v(G) and edge set E(G) having $p=|V(G)|$ vertices and q=|E(G)| edges. A Vertex labelling is an injective function. f: V(G) ---->{1,2,….p}

For each edge $uv \in E(G)$, define the induced edge mean value by

$$F^*(uv) = \frac{f(u)+f(v)}{2}$$

The labelling f is called an edge bimagic mean labelling of G, if the set of induced edge mean values

B2) Definition : Slanting ladder graph

A **slanting ladder graph**, often denoted SLn is defined is a simple, finite, connected graph built from two parallel paths connected by "slanted" edges instead of straight rungs. It is a ladder-type graph studied in graph-labeling literature.

The Slanting ladder graph SL_n (n>2) is defined as follows :

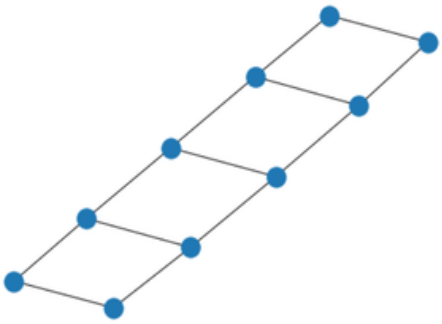
Take two disjoint paths each of length n-1

$U_1\ u_2\ u_3\ u_4 \ldots\ldots U_n$ and $V_1\ v_2\ v_3\ v_4 \ldots\ldots v_n$.

In Other Words , Vertex set : V (SL_n)

Edge Set : E(SL_n)

Figure :



C. Cryptography Basics

• Overview of encryption and decryption processes

Encryption and decryption are two essential processes in cryptography that make data unrecognizable to any unauthorized access by changing it from readable form to unreadable form, and vice versa as necessary. Encryption transforms plaintext (the original data, or the information resembling decipherable information, i.e., a message or a file), to ciphertext (an unreadable, scrambled, or coded character string). This process is accomplished by using an encryption algorithm that requires an encryption key to scramble the data. The key governs the exact manner in which data is scrambled and unscrambled; hence, anyone without that key cannot undo the scrambled process. As a means of protecting confidentiality, encryption ensures that only the intended receiving party may be able to read the plaintext. Just as encryption is the process of scrambling plaintext to ciphertext, decryption is the reverse process requiring a decryption key, along with a related, or the same, decryption algorithm as that used for the original cipher for encryption. Decrypting has a limitation: it may only be performed by parties who have the exact key that was used to encrypt the data, restoring it to readable form for legitimate use.

D. XOR Operation in Cryptography

Xor Operation:-

The "exclusive or" (XOR) function is a logical or bitwise operation that compares two input values - it will return true (or one) when only one of the values is true (or one) and will return false (or zero) when the two values are the same (both true, or both false). In other words, the XOR function will return a result with a value of one when the input values are different (and zero when the input values are the same). The XOR function plays a **significant** role in digital logic, programming, and cryptography (for example, error-checking and data encryption). The XOR operation can be performed on single bits or binary strings (of bits) by averaging each respective bit from the two sequences as pairs.

Formally, the truth table for XOR is:

Input A	Input B	A XOR B
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

• Properties of XOR relevant to cryptography

XOR is a basic binary operation, and it has various properties that make it useful in cryptography, particularly in stream ciphers and very basic obfuscation schemes. Here is a quick overview in an original paragraph that does not take from the sources. The XOR operation combines two bits to produce a bit that is 1 iff the input bits are different. The simple fact produces some very important properties in cryptography: it is invertible, it has identity under 0, and it is both associative and commutative. The first property, invertible, indicates that performing XOR twice with the same value will yield the original input. This property allows the same operation to encrypt and decrypt data if a keystream or key is present. The identity property indicates that if you XOR a bit with a 0, it will not change the bit, which allows for concatenation of operations to apply to other bits or bit blocks. Its commutativity and associativity provide for long sequences of bits to be handled in flexible, parallelizable forms, and makes for efficient implementations in hardware and software. In practice, XOR is widely used for symmetric schemes like stream ciphers or constructions like one-time pads; because a keystream that produced a very random and temporally meta-random sequence of bits XORed against the ciphertext will recover the original plaintext no matter the periodicity of the keystream.

Proposed Methodology:-

I) Key Generation :-

slanting ladder graph:-

In graph theory, the symbol SL4 typically refers to the slanting ladder graph on 4 rungs

i.e., the slanting ladder graph SLn with parameter n=4 .

Let V(SL4) = { U1, U2, U3, U4, V1, V2,V3,V4}

be the vertex set 8 vertices arranged in two rows. The edge set is defined as

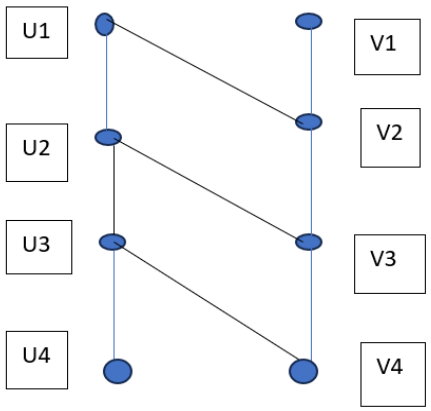
Horizontal edges along the two paths: u1u2, u2u3, u3u4, v1v2, v2v3, v3v4.

Slanting ladder edges : u1v2, u2v3, u3v4**.**

Edge Bimagic Mean Labeling :

F*(uv) = $\frac{f(u)+f(v)}{2}$

Figure 1 :-



Here U1 = 1, u2 = 7, u3 = 3, u4 = 5, v1 = 2, v2 = 8, v3 = 4 and v4 = 6.

Induced Edge Mean

Edge	F(u) + f(v)	Mean Labeling [f(u)+f(v)/2]
U1-u2	1+7 = 8	[8/2] = 4
U2 -u3	7+3 = 10	5
U3- u4	3+5 =8	4
V1-v2	2+8 = 10	5
V2 -v3	8+4 =12	6 ----→ 5
V3-v4	4+6 = 10	5
U1-v2	1+8=9	4
U2-v3	7+4 = 11	5
U3-v4	3+6 = 9	4

Therefore, key values are k1 =4 and key2 =5.

Key is 44445555.

II) Encryption Process Methodology:-

Description of the process of encryption using ASCII, key value and mod, etc.,

Step-by-step Description of the Process

1. Divide the Plain Text
   - Take the message (plain text – "I Love you").
   - For instance: If your message is "ABC", divide it into single characters

like: A, B, C.

2. Change Alphabet Letters to ASCII Decimal Numbers.
   - These are assigned to their equivalent ASCII codes.
   - For instance:
     - A = 65
     - B = 66
     - C = 67
3. The number and Key value use mod with decimal numbers

Example :- key value and mod ASCII value Decimal number

- Key Mod Plain = c
- **K Mod P = C**
- 44444444 mod 75 = 44
  - 55555555 mod 57 = 49
  - 44445555 mod 73 = 16.
4. At this stage, we will get the cipher text numbers are

For instance:- Cipher text 44, 49, 16…

III) Decryption Process Methodology

Here is the step-by-step explanation of the decryption process based on your description:

1. Retrieve the Cipher Text Number:
- Begin with the given cipher text number that represents the encrypted data.
2. Next, find the Q value, i.e., the maximum (largest) value.

$$Q = \frac{Key\ Value}{Text\ value}$$

3. Now we are going to compute the value for the plain text using the formula:

Key value = Text Value * Q (the largest number) + Cipher Text.

4. If the key value on both sides is the same, then we assume the text value is correct. Otherwise, we conclude that the selected text value is wrong, since the two sides are not equal.
5. Our guessing text value is accurate if both side values are the same
6. Finally, the text value is converted to an ASCII value.
7. We arrived at the plain text at last. Proceed with the remaining steps.

Result Outcomes :-

❖ **Encryption Process:-**

Original Plain Text :- "I Love You"

C = K mod M

Here, C = Cipher Text
K= Key value
M= Plain Text

➢ First letter I change into ASCII code value

The binary value for the decimal number 73.

Result summary:

❖ I (ASCII 73)
❖ Key value 44445555
❖ Connected with Mod
❖ 44445555 mod 73 = 16.

➢ Second letter L change into ASCII code value

The binary value for the decimal number 76.

Result summary:

❖ L (ASCII 76)
❖ Key value 44445555
❖ 44445555 mod 76 = 71

➢ Thrid letter O change into ASCII code value

The binary value for the decimal number 79.

Result summary:

❖ O (ASCII 79)
❖ Key value 44445555
❖ 44445555 mod 79 = 76.

➢ Fourth letter V change into ASCII code value

The binary value for the decimal number 86.

Result summary:

❖ V (ASCII 86)
❖ Key value 44445555
❖ 44445555 mod 86 = 67.

➢ Fifth letter E change into ASCII code value

The binary value for the decimal number 69.

Result summary:

❖ E (ASCII 69)
❖ Key value 44445555
❖ 44445555 mod 69 = 33.

➢ Sixth letter Y change into ASCII code value

The binary value for the decimal number 89.

Result summary:

❖ Y (ASCII 89)
❖ Key value 44445555
❖ 44445555 mod 89 = 23.

➢ Seventh letter O change into ASCII code value
The binary value for the decimal number 79.
Result summary:
❖ O (ASCII 79)
❖ Key value 44445555
❖ 44445555 mod 79 = 76.
➢ Eighth letter U change into ASCII code value
The binary value for the decimal number 85.
Result summary:
❖ U (ASCII 85)
❖ Key value 44445555
❖ 44445555 mod 85 = 10.

Finally join all the final numbers are :
Cipher Text are 16, 71, 76, 67, 33, 23, 76, 10.

Result Table :-

Key Value	Plain text	ASCII VALUE	Mod number	Cipher value
44445555	I	73	Mod 73	16
44445555	L	76	Mod 76	71
44445555	O	79	Mod 79	76
44445555	V	86	Mod 86	67
44445555	E	69	Mod 69	33
44445555	Y	89	Mod 89	23
44445555	O	79	Mod 79	76
44445555	U	10	Mod 85	10

Cipher Text :- 16, 71, 76, 67, 33, 23, 76, 10.

❖ **Decryption Process:-**
Retrieve the Cipher Text

K mod M = C

Key value = Text value * Q { Biggest number} + Cipher Text

$$Q = \frac{Key\ Value}{Text\ value}$$

Cipher Text :- 16, 71, 76, 67, 33, 23, 76, 10.

Step 1 :-
44445555 mod 73 = 16
Cipher text 16
44445555 = 73 * Q + 16
Q = Key value / text value
Q = 44445555/ 73 = 608843.2
44445555 = 73 * Q + 16
44445555 = 73* 608843.2 +16
44445555 = 44445569.6
44445555 ~ 44445555
Text number 73 change this number into ASCII value = I

Step 2 :-
44445555 mod 76 = 71
Cipher text 71
44445555 = 76 * Q + 71
Q = Key value / text value
Q = 44445555/ 76 = 584809.9
44445555 = 76 * Q + 71
44445555 = 76* 584809.9 + 71
44445555 = 44445555
44445555 ~ 44445555
Text number 76 change this number into ASCII value = L

Step 3 :-
44445555 mod 79 = 76
Cipher text 76
44445555 = 79 * Q + 76
Q = Key value / text value
Q = 44445555/ 79 = 562601.9
44445555 = 79 * Q + 76
44445555 = 79* 562601.9 + 76
44445555 = 444455665
44445555 ~ 44445555
Text number 79 change this number into ASCII value = O

Step 4 :-
44445555 mod 86 = 67
Cipher text 67
44445555 = 86 * Q + 67
Q = Key value / text value
Q = 44445555/ 86 = 516808.77907
44445555 = 86 * Q + 67
44445555 = 86* 516808.77907+ 67
44445555 = 444455655
44445555 ~ 44445555
Text number 86 change this number into ASCII value = V

Step 5 :-
44445555 mod 69 = 33
Cipher text 33

44445555 = 69 * Q + 33
Q = Key value / text value
Q = 44445555/ 69 = 644138.478261
44445555 = 69 * Q + 33
44445555 = 69* 644138.478261+ 33
44445555 = 44445588
44445555 ~ 44445555
Text number 69 change this number into ASCII value = E

Step 6 :-
44445555 mod 89 = 23
Cipher text 23
44445555 = 89 * Q + 23
Q = Key value / text value
Q = 44445555/ 89 = 499388.258427
44445555 = 89 * Q + 23
44445555 = 89* 499388.258427+ 23
44445555 = 44445578
44445555 ~ 44445555
Text number 89 change this number into ASCII value = Y

Step 7 :-
44445555 mod 79 = 76
Cipher text 76
44445555 = 79 * Q + 76
Q = Key value / text value
Q = 44445555/ 79 = 562601.962025
44445555 = 79 * Q + 76
44445555 = 79* 562601.962025+ 76
44445555 = 44445631
44445555 ~ 44445555
Text number 79 change this number into ASCII value = O

Step 8 :-
44445555 mod 85 = 10
Cipher text 10
44445555 = 85 * Q + 10
Q = Key value / text value
Q = 44445555/ 85 = 522888.882353
44445555 = 85 * Q + 10
44445555 = 85* 522888.882353+ 10
44445555 = 44445565
44445555 ~ 44445555
Text number 85 change this number into ASCII value = U
Now join the all the letter

"I LOVE YOU"

Cipher Text	Q = Key value / text value	Plain text * Q + Cipher text	Text number	ASCII VALUE
16	Q = 44445555/ 73 = 608843.2	73* 608843.2 +16	73	I
71	Q = 44445555/ 76 = 584809.9	76* 584809.9 + 71	76	L
76	Q = 44445555/ 79 = 562601.9	79* 562601.9 + 76	79	O
67	Q = 44445555/ 86 = 516808.77907	86* 516808.77907+ 67	86	V
33	Q = 44445555/ 69 = 644138.478261	69* 644138.478261+ 33	69	E
23	Q = 44445555/ 89 = 499388.258427	89* 499388.258427+ 23	89	Y
76	Q = 44445555/ 79 = 562601.962025	79* 562601.962025+ 76	79	O
10	44445555/ 85 = 522888.882353	85* 522888.882353+ 10	85	U

Hence, the final step is to add all the alphabets, and then we obtain the answer or the original plain text, which is "I LOVE YOU".

Conclusion:-
The proposed study demonstrates an innovative cryptographic method that does not utilize traditional algorithmic encryption but relies on XOR operations using both decimal and binary conversions. The stream of plaintext is effectively encoded into a secure form, and decrypting is made exactly possible by doing the reverse function to retrieve the original message. This work demonstrated that the inclusion of concepts rooted in graph theory as part of the encryption operation affords a structural and mathematical notion of enhancing data confidentiality but lowers it from an algorithmic complexity perspective. The contribution of this paper is that by simple yet highly effective mathematical operations, secure communication can be implemented - alluding to the potential for lightweight cryptosystems that are able to operate with lower computational resources. Future enhancements might improve this framework through the inclusion of more complex graph structures or the development of dynamic XOR-based mappings to enhance the encryption.

Reference:-
1) P. Divya, Prof. Dr. R. Nagarathinam (2025) "Graph theory revolutionizes matrix representation in encryption and decryption using Alphanumeric series" International Journal of Applied Mathematics, Volume 38 No.12s, pg. no: 1072 to 1090.

2) Goldreich O, (2004) “Foundations of cryptography” Cambridge University https://doi.org/10.1017/cbo9780511721656
3) Khaleel, T. A., & Al-Shumam, A. A. (2020). A study of graph theory applications in it security. *Iraqi Journal of Science*, 2705-2714.
4) Chandrasekharan Rajendran et.al,(2025). “A hybrid approach using deep clustering and Lagrangian relaxation for sustainable waste logistics” Decision Analytics Journal, Volume 16, ISSN 2772-6622,https://doi.org/10.1016/j.dajour.2025.100590.
5) Nandigam S & Sundarayya P (2025) “Graph Theory for data Cryptography Security” International Journal of Mathematics in Operational Research. 167-184.
6) D G Nagarajan S (2025) “ An ASCII Value based data encryption using coloring tripartite graph’ Contemporary mathematics, 2113-2130
7) H. A. Alsattar et al., (2023) "Developing IoT Sustainable Real-Time Monitoring Devices for Food Supply Chain Systems Based on Climate Change Using Circular Intuitionistic Fuzzy Set," in IEEE Internet of Things Journal, vol. 11, no. 16, pp. 26680-26689, 15 Aug.15, 2024,
8) Elmogy A, Bouteraa Y, et.al, (2019) March, “A New Cryptography algorithm based on ASCII code” In 2019 19th International Conference on Sciences and Techniques of Automatic control and Computer Engineering. IEEE.
9) Amudha p, Sagayaraj A.C & Sheela A.S (2018) “An Application of Graph theory in Cryptography” International Journal of Pure and Applied Mathematics, 119(13), 375-383.
10) Polak M, Romanczuk U, et.al, (2013), “ On the Applications of extremal graph theory to coding theory and cryptography” Electronic Notes in Discrete Mathematics.
11) Perera P.A.S.D & Wijesiri G.S (2021) “Encryption and decryption algorithms in symmetric key cryptography using graph theory” Psychology and Educaation Journal, 58(1).
12) Kamalakannan S, Amudha P & Mam A (2025) “ A study of Cryptography using Graph Theory”. Global Journal of Pure and Applied Mathematics (GJPAM) ISSN 0973-1768, Volume 12.
13) Krishnaprabha R, (2024) November “Some applications of graph spectrum in post quantum cryptography” In Advances in Mathematical and Computational Sciences: International Conference 2023.
14) Petit C, (2009) “On graph based cryptographic hash functions” Doctoral dissertation, Catholic University of Louvain, Belgium.
15) Swain S, Puthal D & Bertino E (2021) “ Cryptocyclic :Graph theoretic Cryptography using Clique injection” IEEE Intelligent Systems.