Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 6, pp. 2393–2408
DOI : 10.47974/JDMSC-2653

Secure communication via c-exponential mean-labeled graphs : A novel message encryption scheme

K. V. Madhumitha[@]
Anuradha Rao [†]
T. Prajwal Prabhu [§]
Swati Nayak [‡]
Sabitha D'Souza *
Manipal Institute of Technology
Manipal Academy of Higher Education
Manipal 576104
Karnataka
India

Abstract

Cryptography plays an important role in transferring information safely in the computerized world. It consists of two parts; one is encryption of the message so that the cipher text will not reveal the original message to anyone. The other part is decryption at the receiver's side which reverses the encryption process and restores the original message. There is numerous encryption algorithm to secure the information. Graph labeling is one of the tools to encrypt a message. This paper proposes a new encryption technique to encode and decode the message very securely using C-exponential mean graph labeling by considering various graphs.

Subject Classification: *94A60, 05C78, 05C90.*

Keywords: *Encryption, Decryption, Graph labeling, C-exponential mean labelling, Secure communication.*

Introduction

We rely on digital tools practically for almost all of our daily activities. So, safeguarding our information which are being shared is the utmost need of the

[@] *E-mail:* madhumithakv467@gmail.com
[†] *E-mail:* anuradha.rao@manipal.edu
[§] *E-mail:* prajwalprabhu.tellar@gmail.com
[‡] *E-mail:* swati.nayak@manipal.edu
[*] *E-mail:* sabitha.dsouza@manipal.edu (Corresponding Author)

day. The field of cryptology is divided into two parts, namely, cryptography and cryptanalysis. Cryptography is a method of protecting information and communications through the use of codes, so that only the intended can read and process it. Contrarily, cryptanalysis is a technique for dismantling a cryptosystem.

A cryptosystem is an application of cryptographic methods and the infrastructure that supports them that offers information security services. Hash functions, public key cryptography, and symmetric key cryptography are the three types of schemes used in modern cryptography [1]. In public key cryptography, one key is used for encryption and another for decryption, while symmetric key cryptography utilises a single key for both. A transformation is used by hash algorithms to irreversibly encrypt data.

Cryptography and graph theory are deeply interconnected, as many cryptographic protocols and algorithms leverage graph-based structures to ensure security and efficiency. Graph theory provides a mathematical framework to model relationships and structures that are crucial in cryptographic applications. For instance, key exchange mechanisms, such as those based on expander graphs, utilize the properties of highly connected but sparse graphs to enhance security while maintaining computational efficiency. Additionally, the complexity of certain graph problems, such as finding the shortest path in a weighted graph or solving the Hamiltonian path problem, is often exploited to create cryptographic schemes that are resistant to attacks.

One notable application of graph theory in cryptography is in public-key cryptosystems and digital signatures. Lattice-based cryptography, for example, relies on the hardness of graph-related problems, such as the shortest vector problem (SVP) and closest vector problem (CVP) in high-dimensional lattices, to provide security against quantum computing threats. Graph-based hash functions also use complex graph structures to generate collision-resistant cryptographic hashes, ensuring data integrity and authenticity. These approaches highlight how graph theory contributes to designing cryptographic protocols that are both secure and computationally feasible.

Emerging cryptographic techniques like secure multi-party computing and zero-knowledge proofs heavily rely on graph theory, which extends beyond classical encryption. A common basis for zero-knowledge proofs, which enable one person to demonstrate knowledge of a secret without disclosing it, is graph isomorphism problems. Similarly, secure network communication protocols leverage graph connectivity and network flow algorithms to optimize data transmission while preserving confidentiality and integrity. As cryptography continues to evolve, graph theory remains a powerful tool in developing innovative and robust security solutions.

A graph-theoretic problem can be applied to any real-world scenario. In recent years, there has been a boom in interest in using graphs as a tool to propose novel strategies in several domains of cryptography. Because there are so many graphs available online, it is difficult to tell whether or not the graph is used in the encryption process. Few attempts have been made in the past to

encrypt a message using a graph-theoretic concept. Using that as a foundation, we attempted to use the concept of C-exponential mean labeling of graphs to encrypt a secret message in this study.

A graph $G = (V, E)$ is made up of a prescribed set E of unordered pairings of distinct elements of E and a finite non-empty set V. The edges the vertices of G are elements of E and V, respectively. Let $|V| = a$ and $|E| = b$. For additional details on concepts related to graph theory, see [2].

In graph theory, graph labeling is an essential concept in which values are assigned to each vertex, edge, or both based on predetermined guidelines. These labels can represent various properties, such as weights, colors, or unique identifiers, and are used to solve practical problems in network design, coding theory, and cryptography. Graph labeling was first introduced in 1960 by Rosa [3]. He identified 4 types of labelings namely, $\alpha, \beta, \sigma,$ and ρ −labeling. β −labeling was later renamed as graceful labeling by S. Golomb. Mean labeling of graphs was first introduced by Somasundaram and Ponraj [4]. Later, it was elaborated to super mean, k-super mean, even vertex mean, odd vertex mean, even vertex odd mean, odd vertex even mean labelings and many more. Most of the papers related to graph labeling are found in Gallian survey of graph labeling [5].

Graph labeling has wide applications in computer science, engineering, and optimization. In wireless network frequency assignment, for example, vertex labeling helps minimize signal interference by assigning different frequencies to nearby nodes. Similarly, in X-ray crystallography and bio-informatics, graph labeling assists in modelling molecular structures and genetic sequences. Another key application is in the design of secure cryptographic protocols, where labeling techniques contribute to encoding information and ensuring data integrity. In the field of cryptography, the use of labeled graphs is becoming a valuable source for developing improved encryption techniques for safer data transfer between two parties because there are numerous types of graph labeling available in the literature. This makes it extremely safe to use graphs in cryptography because it is challenging or impossible to detect or know the type of graph used.

Related Works

Few works have been carried out in such a way that some graph labeling technique has been used to encrypt a message.

Authors in [6] explored a novel approach to secure communication using artistic graph creation. It introduces an algorithm that converts textual data into unique graphical representations, providing both an aesthetic and cryptographic method of communication. The proposed encryption method employs graph labeling techniques and generalized graph complements to transform plain text messages into cipher graphs.

The paper [7] explores a novel method for cryptographic key generation using radio frequency (RF) path characteristics. It discusses how unique environmental factors affecting radio wave propagation can be leveraged to

create secure and unpredictable cryptographic keys. The authors introduce a technique based on path coloring, where variations in RF signals due to multi path effects and interference contribute to randomness in key generation.

In [8] authors explored the use of total magic labeling in graph networks to improve computational efficiency and network security. The paper discusses the theoretical foundations of this labeling approach, its impact on network performance, and its practical applications in secure communication and efficient data transmission. Through mathematical modeling and experimental analysis, the study demonstrates how total magic labeling can optimize resource allocation, minimize computational complexity, and strengthen cryptographic protocols in network applications.

In [9] authors explored how expander graphs, known for their strong connectivity and rapid mixing properties, can be used to construct secure and efficient hash functions. These graphs ensure high randomness and diffusion, making the hash functions resistant to collision and preimage attacks. By leveraging expander graph structures, the proposed hash functions offer improved security while maintaining efficiency, particularly in lightweight cryptography and resource-constrained environments. Comparisons with traditional hash functions like SHA-2 and SHA-3 highlight their potential as a viable alternative for cryptographic applications.

In [10] authors proposed an enhanced encryption method that applies two layers of encryption to strengthen data security. The technique combines classical and modern cryptographic approaches, ensuring better resistance against brute-force and cryptanalysis attacks. The first encryption layer transforms plain text into an intermediate cipher, which is then re-encrypted using a different algorithm, adding complexity to unauthorized decryption attempts. The paper evaluates the efficiency and security of this method, highlighting its potential for secure data transmission in sensitive applications. Experimental results demonstrate that the double encryption approach significantly improves confidentiality while maintaining computational efficiency.

For more work related to encryption, one can refer to [11-16].

This manuscript has developed a novel method for encrypting and decrypting data using the graph labeling technique, inspired by all of these studies. The following is the structure of the manuscript. Some preliminary definitions utilised in this work are included in Section 1. The proposed algorithms are presented step-by-step in Section 2. An illustration of the suggested algorithm for each of the graphs under study is given in section 3. The analysis of the suggested algorithm for each of the graphs under consideration is done in section 4, which is followed by the conclusion of the work.

Preliminary Definitions

Definition 1: [17] A graph G is *exponential mean graph* if there exists an injective function $\chi: \mathrm{V}(\mathrm{G}) \rightarrow \{1, 2, 3, \ldots, \mathrm{b}+1\}$ and an induced bijective function $\chi*: E(G) \longrightarrow \{1,2,3, \ldots, b\}$, as

$$\chi*(st) = \left\lfloor \frac{1}{e}\left(\frac{\chi(t)^{\chi(t)}}{\chi(s)^{\chi(s)}}\right)^{\frac{1}{\chi(t)-\chi(s)}} \right\rfloor, \text{ for all } st \in E(G).$$

Definition 2: [18] A graph $\boldsymbol{G}$ is *power exponential mean graph*, if there exists a one to one correspondence $f: V \longrightarrow \{1,2,3, \ldots, b\}$ such that the induced function $f*: E(G) \longrightarrow N$ given by $f*(st) = \lceil (f(s)^{f(s)} * f(t)^{f(t)})^{\frac{1}{f(s)+f(t)}} \rceil$, for every $st \in E(G)$ are all distinct**.**

Definition 3: [19] A function $\boldsymbol{h}$ is *C-exponential mean labeling* of a graph $\boldsymbol{G}$ that possesses $\boldsymbol{a}$ vertices and $\boldsymbol{b}$ edges, if $h*: V(G) \longrightarrow \{1,2,3, \ldots, b+1\}$ is injective and the generated function $h*: E(G) \longrightarrow \{2,3,4, \ldots, b+1\}$ defined by $h*(st) = \left\lfloor \frac{1}{e}\left(\frac{h(t)^{h(t)}}{h(s)^{h(s)}}\right)^{\frac{1}{h(t)-h(s)}} \right\rfloor$, for all $st \in E(G)$ is bijective**.**

Definition 4: [19] Let $s_1, s_2, \ldots, s_n$ be the consecutive vertices of $\boldsymbol{P_n}$; *triangular tree* as shown in Figure 1 is calculated by amalgamating each $\boldsymbol{s_i}$ with a leaf of $\boldsymbol{P_i}$**.** We denote this tree by $\boldsymbol{T_n}$ and refer to $\boldsymbol{P_n}$ as the base of $\boldsymbol{T_n}$**.** Note that $\boldsymbol{T_n}$ has size $\frac{n(n+1)}{2}$, which means that its order is a triangular number. We say that the first vertex of $\boldsymbol{P_i}$ leaf is amalgamated with the vertex of $\boldsymbol{P_n}$.

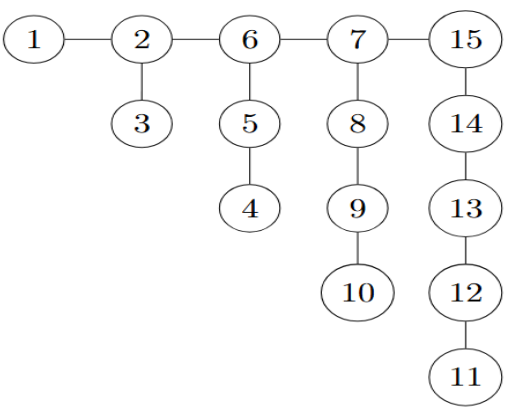

Figure 1
C-exponential mean labeling of T_5

[19] The *Cartesian product* $G \square H$ of graphs $\boldsymbol{G}$ and $\boldsymbol{H}$ is a graph with vertex set $V(G) \times V(H)$ in which any two vertices (s, s') and (t, t') are adjacent if and only if either

1. $s = t$ in G and $s' \sim t'$ in H or
2. $s' = t'$ in H and $s \sim t$ in G.

Cartesian product of P_5 □ P_5 is given in Figure 2.

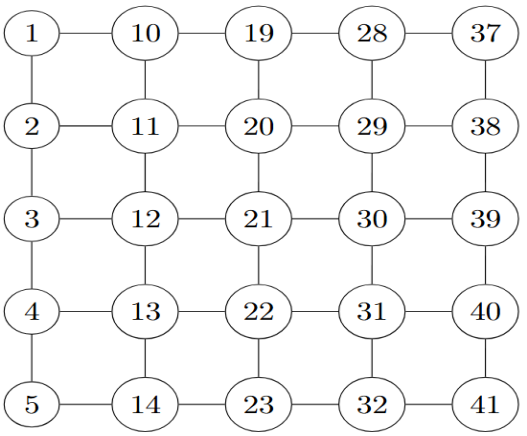

Figure 2
C-exponential mean labeling of Cartesian product of P_5

[19] Let P_n be a path on n vertices represented by $s_{1,1}, s_{1,2}, \ldots, s_{1,n}$ and with $n-1$ edges signified by $e_1, e_2, \ldots, e_{n-1}$, where e_i represents the edge connecting $s_{1,i}$ and $s_{1,i+1}$. On every edge e_i, erect a ladder that has $n-(n-1)$ steps counting the edge e_i, for $i = 1,2,3, \ldots, n-1$. The graph hence drawn is defined as a *one-sided step graph* as shown in Figure 3, and it is represented by ST_n.

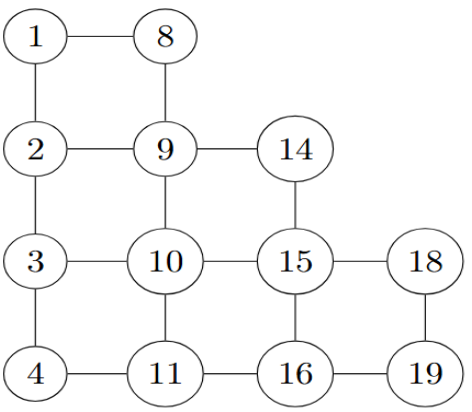

Figure 3
C-exponential mean labeling of ST_4

[19] *Subdivision graph* as shown in Figure 4 $\boldsymbol{S(G)}$ is obtained by inserting a vertex of degree 2 at each edge of graph $\boldsymbol{G}$.

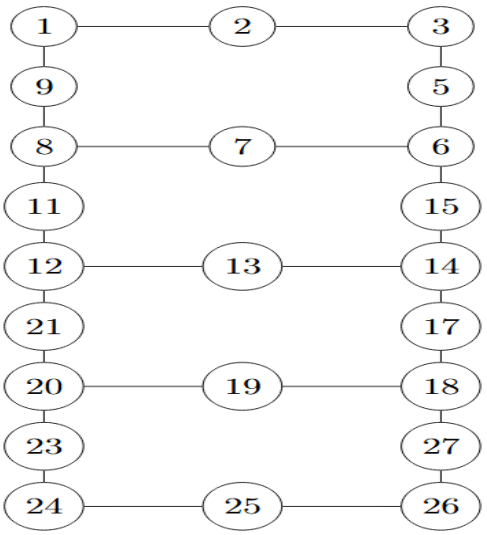

Figure 4
C-exponential mean labeling of $\boldsymbol{S(L_5)}$

Proposed Algorithm

The proposed algorithm considers the C-exponential mean labeling to develop the stream ciphers. The characters of the plaintext are considered as various vertices and C-exponential mean labeling is applied. The illustration considers triangular tree, one-sided step graph, Cartesian product of paths and subdivision of ladder graphs.

Steps followed to encrypt and decrypt the text:

1. The graph is generated using C- exponential mean labeling with the given parameters for the selected graph. These parameters can be generated from a key string using the ``polynomial rolling'' algorithm.
2. A hash map of this graph is created. It is done by converting the adjacency matrix of the graph to one-dimensional array and generate the map so that the key of the map becomes the index.
3. The example shown here considers only the characters, hence 65 can be added to the index of the generated 1-D array and the value will be the value in that index.
4. Now the given string (one to be encrypted) will be taken and iterated over its character-wise, searching for the ASCII value of that character in the hash-map. If there is a key of that ASCII then replace it with the value received from the hash-map and if the key does not exist, encrypt it as 0000000000.

5. To decrypt, steps 1-2 are repeated and a reverse hash map is created. Then by taking the encrypted string and iterating over it character wise, the ASCII value of the character in the hash map is searched and if there is a key of that ASCII then replace it with the value received from the hash map and if the key does not exist just decrypt it as it is.

Illustration

The illustration shows how encryption process can be performed using C-exponential mean labeling on triangular tree, one-sided step graph, Cartesian product of paths and subdivision of ladder graphs. To decrypt, steps 1-2 are repeated and a reverse hash map is created. Take the given encrypted string and iterate over it character wise to search for the ASCII value of the character in the hash map and if there is a key of that ASCII then replace it with the value received from the hash map and if the key does not exist just decrypt it as it is.

Triangular tree T_5

Let us consider a triangular tree graph T_5. The adjacency matrix of generated triangular tree graph is shown in Equation [0.1], which then converted into 1-D array Equation [0.2]. From this, a hash map is generated, after adding 65 to the index and to the value, as shown in Table [1]. The respective illustrations are given below.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 2 & 0 & 0 & 0 \\ 0 & 5 & 4 & 6 & 0 & 0 \\ 0 & 8 & 9 & 10 & 7 & 0 \\ 0 & 14 & 13 & 12 & 11 & 15 \end{bmatrix} \quad \text{Equation 0.1}$$

$$[1,3,2,5,4,6,8,9,10,7,14,13,12,11,15] \quad \text{Equation 0.2}$$

Table 1
Hash map table

A	B	C	D	E	F	G	H	I	J	K	L	M
B	D	C	G	F	I	J	K	H	P	O	N	M

Now, let us take a message "GRAPH THEORY". According to Table [1] $G \to J, R \to R, A \to B, P \to P, H \to K,$ <space> $\to < space >$ $, T \to T, H \to K, E \to F, O \to O, R \to R, Y \to Y$.

After encrypting character by character, we get an encrypted string "JRBPK TKFORY". While decrypting the encrypted string "JRBPK TKFORY", character by character, the mapping becomes $J \to G, R \to R, B \to A, P \to P, K \to H, \to$ $, T \to T, K \to H, F \to E, O \to O, R \to R, Y \to Y$. The reverse hash map is as shown in Table [2].

Table 2
Reverse hash map table

B	D	C	G	F	I	J	K	H	P	O	N	M
A	B	C	D	E	F	G	H	I	J	K	L	M

Cartesian product of $\boldsymbol{P_5 \square P_5}$

In case of Cartesian product of $P_5 \square P_5$, the generated adjacency matrix is shown in Equation [0.3] along with its 1-D array shown in Equation [0.4].

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 10 & 19 & 28 & 37 \\ 0 & 2 & 11 & 20 & 29 & 38 \\ 0 & 3 & 12 & 21 & 30 & 39 \\ 0 & 4 & 13 & 22 & 31 & 40 \\ 0 & 5 & 14 & 23 & 32 & 41 \end{bmatrix} \quad \text{Equation 0.3}$$

$$[1,10,19,28,37,2,11,20,29,38,3,12,21,30,39,4,13,22,31,40,5,14,23,32,41] \quad \text{Equation 0.4}$$

Table 3
Hash map table

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
V	-	h	E	N	W	‘	I	F	O	X	a	B	K	T	]	f	C	L	U	^	g	D	M	j

From Table [3] we have, $G \rightarrow ', R \rightarrow C, A \rightarrow V, P \rightarrow J, H \rightarrow I,$ <space> $\rightarrow < space >$ $, T \rightarrow U, H \rightarrow I, E \rightarrow N, O \rightarrow T, R \rightarrow C, Y \rightarrow j.$

“GRAPH THEORY” is encrypted as “’CVPI UINTCj”.

Subdivision of ladder graph $\boldsymbol{S(L_5)}$

In case of subdivision of ladder graph $S(L_5)$, the matrix is as shown in Equation [0.5] and its 1-D array is given in Equation [0.6].

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 3 \\ 8 & 7 & 6 \\ 12 & 13 & 14 \\ 20 & 19 & 18 \\ 24 & 25 & 26 \end{bmatrix} \quad \text{Equation 0.5}$$

$$[2,3,7,6,13,14,19,18,25,26] \quad \text{Equation 0.6}$$

The hash map is given in Table [4].

Table 4
Hash map table

A	B	C	D	E	F	G	H	I	J
C	D	H	G	N	O	T	S	Z	[

From the table, $G \to T, R \to R, A \to C, P \to P, H \to S$, <space> →< $space >$ $, T \to T, H \to S, O \to O, R \to R, Y \to Y$.

"GRAPH THEORY" is encrypted as "TRCPS TSORY".

Step graph $\boldsymbol{ST_7}$

In case of step graph ST_7, the generated matrix and its 1-D array are displayed in Equation [0.7] and Equation [0.8], respectively.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 14 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 15 & 26 & 0 & 0 & 0 & 0 \\ 0 & 3 & 16 & 27 & 36 & 0 & 0 & 0 \\ 0 & 4 & 17 & 28 & 37 & 44 & 0 & 0 \\ 0 & 5 & 18 & 29 & 38 & 45 & 50 & 0 \\ 0 & 6 & 19 & 30 & 39 & 46 & 51 & 54 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{Equation 0.7}$$

$$[1,14,2,15,26,3,16,27,36,4,17,28,37,44,5, 18,29,38,45,50,6,19,30,39,46,51,54] \quad \text{Equation 0.8}$$

Table 5
Hash map table

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[
B	O	C	P	[	DD	Q	\	e	E	R	]	f	m	F	S	^	g	n	s	G	T	-	h	o	t	w

From the Table [5],

$G \to Q, R \to g, A \to B, P \to S, H \to \backslash, < space > \to < space >, T \to s, H \to \backslash, O \to F, R \to g, Y \to o$

"GRAPH THEORY" is encrypted as "QgBS\ s\Fgo".

Analysis of the Proposed Scheme

Table 6
Time taken (in milliseconds) for encryption and decryption of GRAPH THEORY using 4 graphs

Graph	Encryption time	Decryption time
Triangular tree	5.5	3.1
Cartesian product of P_n	2.9	3.7
One sided step graph	3.2	3.8
Subdivision of ladder graph	1.5	1.8

The time taken for encryption and decryption of **GRAPH THEORY** using C-exponential mean labeling of 4 different graphs are given in Table [6].

The algorithm is tested for the plain text containing varied number of characters and as well as decryption time are tabulated as shown in Table [7] and Table [8], respectively. Results show that the time increases linearly as number of characters increases. There is not much time difference in triangular tree and

subdivision of ladder graph. The overall comparison of encryption and decryption time for various graph labeling is as shown in Figure 5 and Figure 6. The trend shows there is an increase in encryption and decryption time with respect to number of characters encrypted in all 4 labeling cases. Out of four graphs considered, Cartesian product of P_n graph shows large time of encryption and decryption and hence can be considered for the applications where high security is required. Subdivision of a ladder graph, on the other hand, requires minimal time for both encryption and decryption and can thus be employed in communication network.

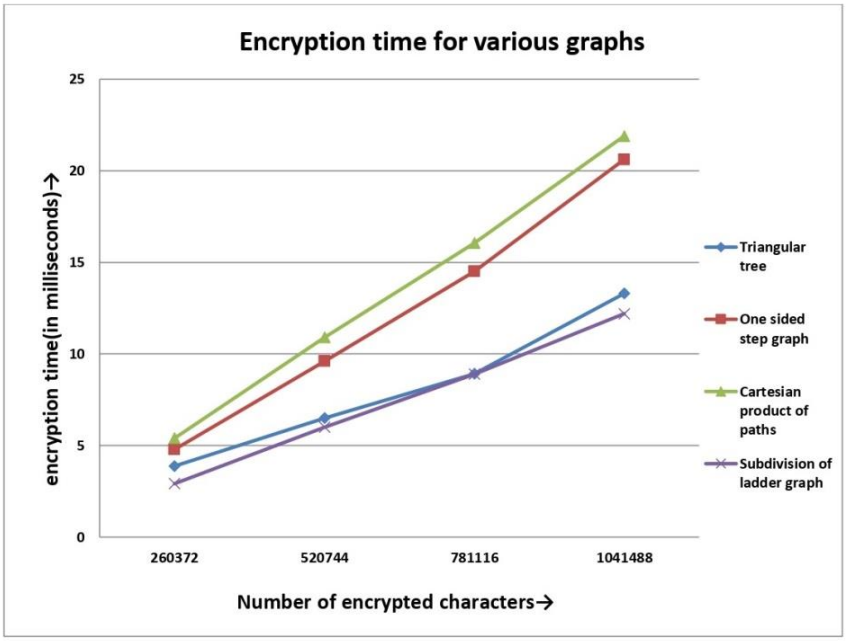

Figure 5
Encryption time for various graphs

Comparison of encryption time

The triangular tree has the highest encryption time at 5.5 milliseconds, indicating that its encryption algorithm is computationally heavier compared to the others. The subdivision of the ladder graph achieves the fastest encryption at 1.5 milliseconds, suggesting a more efficient structure for encryption. The Cartesian product of $P_n \square P_n$ and one-sided step graph have moderate encryption times at 2.9 milliseconds and 3.2 milliseconds, respectively.

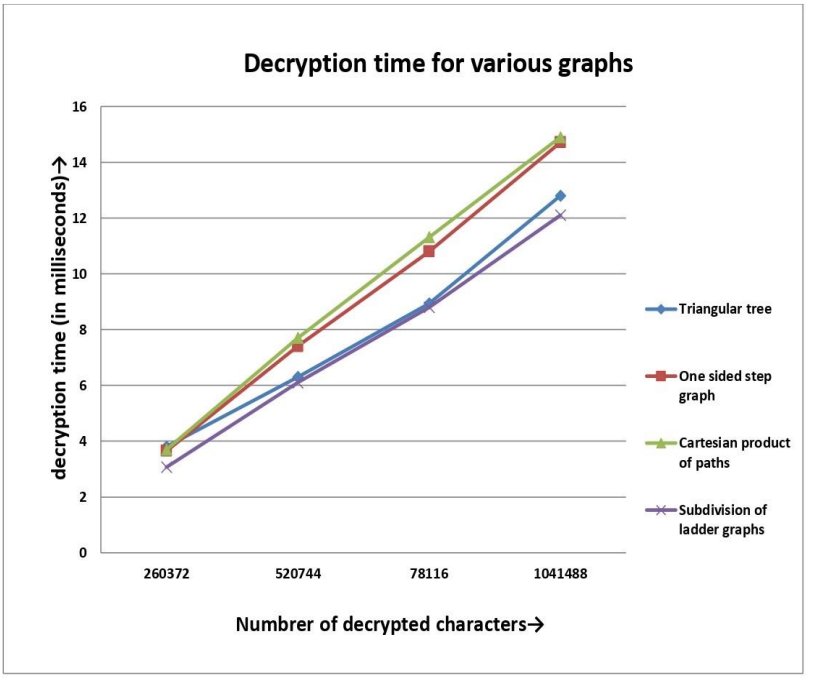

Figure 6
Decryption time for various graphs

Table 7
Time taken for encryption of varied number of characters (in milliseconds)

Graph	Characters			
	260372	**520744**	**781116**	**1041488**
Triangular tree	3.88	6.5	8.92	13.3
Cartesian product of P_n	5.4	10.9	16.05	21.9
One-sided step graph	4.79	9.6	14.5	20.62
Subdivision of ladder graph	2.92	6.0	8.9	12.2

Table 8
Time taken for decryption of varied number of characters (in milliseconds)

Graph	Characters			
	260372	**52.744**	**781116**	**1041488**
Triangular tree	3.81	6.43	8.94	12.8
Cartesian product of P_n	3.7	7.7	11.31	14.9
One-sided step graph	3.65	7.4	10.8	14.711
Subdivision of ladder graph	3.07	6.1	8.8	ssss12.1

Comparison of decryption time

The subdivision of the ladder graph also has the fastest decryption time at 1.8 milliseconds, making it the most efficient for both encryption and decryption. The triangular tree has the second fastest decryption time at 3.1 milliseconds, despite having the slowest encryption. The Cartesian product of P_n and one-sided step graph have slightly higher decryption times at 3.7 milliseconds and 3.8 milliseconds, respectively, indicating a small variation among them.

Complexity implications

Each graph structure introduces different levels of computational complexity for encryption and decryption, which can be inferred based on its performance:

Triangular tree has high encryption time, moderate decryption time. The high encryption time (5.5 milliseconds) suggests that the encryption process involves more computational steps, likely due to increased node or edge traversal. However, the decryption process (3.1 milliseconds) is relatively efficient, implying that once the structure is set, retrieval of the original message is not as costly. This may indicate an encryption scheme where key expansion or complex transformations are involved but are easily reversible.

Cartesian product of P_n and one-sided step graph (Balanced Performance)

Both have relatively low encryption times (2.9 milliseconds and 3.2 milliseconds, respectively) but slightly higher decryption times (3.7 milliseconds and 3.8 milliseconds). This suggests that encryption is computationally simpler, possibly due to straightforward graph traversal methods, but decryption might require additional steps to reconstruct the original data. These graphs may be suitable for applications where encryption needs to be fast but decryption overhead is acceptable.

Subdivision of ladder graph with an encryption time of just 1.5 milliseconds and a decryption time of 1.8 milliseconds, this graph appears to offer the best trade-off. The efficiency suggests that the structure allows rapid encoding and decoding, potentially due to a predictable and well-defined vertex/edge transformation. This makes it ideal for real-time applications or lightweight cryptographic protocols where speed is critical.

Key insights

The subdivision of the ladder graph is the most efficient graph for both encryption and decryption, making it the best choice for performance. The triangular tree, while slowest in encryption, has a relatively quick decryption time, which may be useful in applications where encryption speed is less critical. The Cartesian product of $P_n \square P_n$ and one-sided step graph show similar performance, making them interchangeable depending on specific cryptographic needs.

Security vs. speed trade-off

A more complex graph structure (like the triangular tree) likely provides better security at the cost of higher encryption time. Conversely, a simpler graph structure (like the subdivision of ladder graph) offers fast encryption and decryption but may have weaker resistance to cryptographic attacks.

Computational resources trade-off

If a system has limited processing power, choosing a graph with lower encryption and decryption times (e.g., the subdivision of ladder graph) is preferable. High-security applications may opt for computationally intensive graphs despite the longer processing times.

Application-specific needs

Real-time secure messaging may prefer the subdivision of ladder graph due to its quick processing. Secure storage systems (where encryption speed is less critical) may use more complex graphs like the triangular tree for enhanced security. Balanced encryption/decryption systems (such as authentication protocols) may benefit from the Cartesian product of P_n or one-sided step graph.

If the priority is speed, the subdivision of ladder graph is the best choice. If a balance between security and performance is needed, the Cartesian product of P_n and one-sided step graph offer a reasonable middle ground. If security is the top concern, the triangular tree might be preferable despite its high encryption time.

Conclusion

This paper presents the performance evaluation of encryption algorithm based on C-exponential mean labeling technique considering triangular tree, one sided step graph, Cartesian product of paths and subdivision of ladder graphs. The matrix generated from graph labeling is fully dependent on the polynomial rolling function, and thus its size cannot be predicted. As a result, the message is tightly guarded while being conveyed across the communication channel. The time to encrypt and decrypt is seen linearly increasing as the number of characters considered are increased. The result shows that the time required to encrypt and decrypt is large in Cartesian product of path graphs compared to other three. Hence, security is high in comparison with other three considered for illustration. This proves that this mode of encryption and decryption may be used in applications where high security is required as comparable to the standard algorithms.

References

[1] J. Katz and Y. Lindell, Introduction to modern cryptography: principles and protocols., Chapman and hall/CRC. (2007).

[2] D. B. West, Introduction to graph theory (Vol. 2, pp. 1-512)., Upper Saddle River: Prentice hall. (2001).

[3] A. Rosa, "On certain valuations of the vertices of a graph.," in In Theory of Graphs (Internat. Symposium, Rome (pp. 349-355)., Rome (1966).

[4] S. Somasundaram and R. Ponraj, "Mean labelings of graphs.," *National academy Science letters*, vol. 26, no. 7, pp. 210-213 (2003).

[5] J. A. Gallian, "A dynamic survey of graph labeling," *Electronic Journal of Combinatorics*, p. DS6 (2018).

[6] Medini, H. R., Sabitha D'Souza, C. Devadas Nayak, and Pradeep G. Bhat "Exploring secure communication through artistic graph creation.," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 28, no. 1, pp. 143-159 (2025).

[7] K. Meera and S. Baskar, "Key Generation in Cryptography Using Radio Path Coloring," *IEEE Access,* vol. 12, pp. 60475-60481 (2024).

[8] A. Meenakshi, A. Kannan, R. Cep, and M. Elangovan, "Efficient graph network using total magic labeling and its applications," *Mathematics*, vol. 11, no. 19, p. 4132 (2023).

[9] D. X. Charles, K. E. Lauter, and E. Z. Goren "Cryptographic hash functions from expander graphs," *Journal of Cryptology,* vol. 22, no. 1, pp. 93-113 (2009).

[10] Y. Rajput, D. Naik, and C. Mane "An improved cryptographic technique to encrypt text using double encryption," *International Journal of Computer Applications,* vol. 86, no. 6, pp. 24-28 (2014).

[11] A. B. Srinivasan and S. Hemalatha "A table-based end to end encryption technique without key exchange," *Engineered Science*, vol. 19, pp. 279-284 (2022).

[12] C. Beaula, P. Venugopal, and B. Praba "Block encryption and decryption of a sentence using decomposition of the Turan graph," *Journal of Mathematics*, vol. 2023, no. 1, p. 7588535 (2023).

[13] A. Krishnaa, "Inner magic and inner antimagic graphs in cryptography," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 22, no. 6, pp. 1057-1066 (2019).

[14] P. L. K. Priyadarsini "A survey on some applications of graph theory in cryptography," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 18, no. 3, pp. 209-217 (2015).

[15] G. Zémor, "Hash functions and Cayley graphs," *Designs, Codes and Cryptography*, vol. 4, no. 3, pp. 381-394 (1994).

[16] B. Ni, R. Qazi, S. U. Rehman, and G. Farid " Some Graph-Based Encryption Schemes," *Journal of Mathematics*, vol. 1, no. 6614172, p. 2021 (2021).

[17] A. R. Kannan, R. R. Jeyalakshmi, P. Manivannan, and A. D. Baskar "Exponential meanness of some ladder related graphs," *InJournal of Physics: Conference Series*, vol. 1, no. 012034, p. 1597 (2020).

[18] K. M. Nagaraja, S. Ramachandraiah, and V. B. Siddappa. "Power exponential mean labeling of graphs," *Montes Taurus Journal of Pure and Applied Mathematics,* vol. 3, no. 2, pp. 70-79 (2021).

[19] T. Baskaran, and R. Ganapathy. "A Study on C-Exponential Mean Labeling of Graphs," *Journal of Mathematics*, vol. 2022, no. 1, p. 2865573 (2022).

Received August, 2025