Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 5, pp. 1645–1658
DOI : 10.47974/JIOS-1880

Encryption scheme of modified Caesar cipher using bipartite graph

Koushani Roy *
Supriyo Mazumder †
Department of Mathematics
Adamas University
Barasat - Barrackpore Road
Jagannathpur 700126
Kolkata
India

Abstract

In recent years, the demand for data security has exploded in various areas of daily life and society such as science, academia, Networks, industry, financial services, government, telecommunications, media, law and many more. The protection of data has become a major concern over time. Cryptography is the study of secure communication and the protection of data in various fields. However, the complexity and constant evolution of cryptographic algorithms require continuous research and development. In recent years, Graph Theory has emerged to be a valuable tool in the field of cryptography, providing new insights and techniques to ensure the security of communication in networks. The algorithm proposed in this research work develops a graph-based encryption scheme that protects the shared data from being accessed, altered, or interfered with by unwanted and unauthorized parties. The shared symmetric key ensures confidentiality and high security. Not just alphabets; with this algorithm, data including all 256 characters along with their ASCII values can be transmitted securely.

Subject Classification: *05C90, 05C99, 94A60.*

Keywords: *Cryptography, Bipartite graph, Caesar cipher, Shift cipher.*

1. Introduction

Data security is very crucial in the current era of digital communication and technology that is constantly evolving. The confidentiality, integrity,

* *E-mail:* koushani1.roy@stu.adamasuniversity.ac.in (Corresponding Author)
† *E-mail:* supriyo88@gmail.com;
supriyo.mazumder@adamasuniversity.ac.in

and authenticity of sensitive information can be safeguarded against unwanted access via cryptography. The need of a secure algorithm has broadened the research area of this field, and the characteristics of graph theory provides perfect justice to the research need. Graph Theory is a branch of Discrete Mathematics, which can be implemented to deal the amplified requirement of network and data security.

Throughout the times, graph theory has become one of the most important aspects in encryption and decryption[16][18][23][25]. In 2018, P. Amudha et al.[2] found an encryption scheme by finding the ASCII value of the texts, finding the corresponding binary values and applying XOR operation on them. This research got a boost when Joseph et al.[12] introduced a key in encryption and decryption. Kedia et al.[13] worked in this field with the hexadecimal and binary ASCII values of the plain text by converting them into a graph that includes consideration of graph connection. The studies of W. Etaiwi [9] and U. Dixit[8] show that a plain text can be converted to the vertices of a weighted graph with pre-declared values and then finding Complete graph and Minimum Spanning Tree with their adjacency matrices is a successful and secured algorithm. The idea of adjacency matrix can also be seen in some algorithms [20][19].

V. Ustimenko[24] developed a symmetrical Cryptographical algorithm with infinite families of directed graphs of large girth back in 2007. The matrix approach of graph theory has also emerged enormously in the field of network security [17]. The concept of complement of a graph to obtain the ciphergraph has been discussed in the study of M. H. R. et al.[11]. Agarwal[1] explained how an algorithm can function with prime weighted graph and also developed a Java program for the same. The amalgamation of double vertex graph with Cryptography can be seen in the research works on Beaula[4]. The author also demonstrated a robust cryptosystem applying 'Graph labeling' on Turan graphs[5]. Kuppan [14] encoded and decoded thirteen secret numbers by operating face antimagic labelling in a tree graph. Sharmila et al.[21] have carried out the idea of Fuzzy graph in Cryptography.

In recent times, many studies have extensively incorporated the concept of Bipartite graph in data security[6][15][26]. Later in 2021, Gurjar et al.[10] designed an encryption scheme with graph labelling on bipartite trees. The concept of a complete graph is seen in the work of Anwar et al.[3] where all 26 alphabets with some special characters can be encrypted with the help of the matrix introduced by the authors. The completeness of the Bipartite graph along with connectivity has been elaborated in the works of Swadi et al. [28] and Reddy et al. [27] But a requirement of

encoding all the existing characters including the capotal and small alphabets, the special characters and also numbers, can be found in this research area. Keeping this in mind and also to introduce more confidentiality, this paper has been prepared with all the existing characters with their ASCII values. Our aim is to develop a new encryption algorithm based on the concept of Bipartite Graph. The proposed algorithm allows a sender to send a text message of any length including all the existing characters(alphabets, signs, numbers, punctuations etc.) to a receiver without facing any third party access or disturbance. The receiver gets a Bipartite graph as the protected text and the decryption part of this algorithm can be used to get back to the actual plain text. The whole algoithm has been developed with the concept of Symmetric key encryption.

In this paper, some basic needed definitions are recapitulated in Section 2. Section 3 includes the proposed encryption and decryption algorithm. A successful application of the proposed algorithm has been explained with example in section 4.

2. Preliminaries

We go through some concepts of Graph Theory[7] and Cryptography[22] which are used to construct the proposed algorithm.

Definition 2.1: A graph $G(V,E)$ is a set of two sets E of edges and V of vertices such that each edge is defined as a joining side between a pair of vertices. If two vertices are joined by an edge then those vertices are said to be connected.

Definition 2.2: A Bipartite Graph can be a directed or undirected graph where the vertices can be divided into two disjoint sets such that no two vertices of the same set are connected (Figure 1).

Definition 2.3: Cryptography is the study of encryption and decryption of messages. We find a code, known as a cipher text which is the encrypted form of a plain text and can be decrypted to find the plain text and this process may follow one or more algorithms.

Definition 2.4: Encryption is the procedure of getting a pseudo text from a given plain text to maintain the authenticity and the confidentiality of the plain text shared by the sender. Decryption is the procedure to get back to the shared plain text from the pseudo text sent by the receiver.

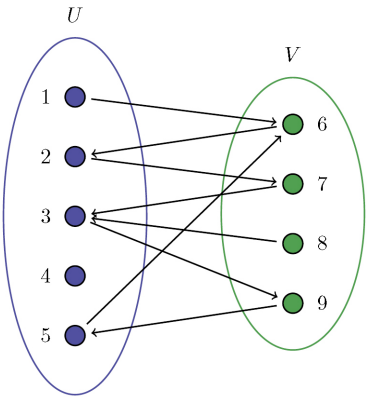

Figure 1

Example of a Directed Bipartite Graph

Definition 2.5: A Plain Text is the main text sent by the sender.

Definition 2.6: A Cipher text is the Encrypted text that is received by the receiver.

Definition 2.7: The American Standard Code for Information Interchange(ASCII) is a World wide used encoding system that gives the codes of alphabets, characters, numbers etc. to be used and accessed by the computer and other electronic devices.

Definition 2.8: A Caesar cipher is the shift cipher. It is a type of substitution cipher in which each character of the plain text is replaced by another alphabet some fixed number of positions down the alphabet.

Definition 2.9: A key is a string of characters shared only between the sender and the receiver and helps to encrypt and decrypt the data.

Definition 2.10: A symmetric key encryption is an encryption technique that only takes one single key for encryption and that single key is shared between the sender and the receiver.

3. Proposed algorithm of Encryption and Decryption using Bipartite Graph

In this section we have used the concepts and properties of the Graph Theory to find a feasible algorithm that can be used to encrypt and decrypt messages.

Table 1

ASCII Value table of 256 characters

ASCII control characters			ASCII printable characters						Extended ASCII characters							
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	á	192	└	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ü	161	í	193	┴	225	ß
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	┬	226	Ô
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ú	195	├	227	Ò
04	EOT	(End of Trans.)	36	$	68	D	100	d	132	ä	164	ñ	196	─	228	õ
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	┼	229	Õ
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	å	166	ª	198	ã	230	µ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	Ã	231	þ
08	BS	(Backspace)	40	(	72	H	104	h	136	ê	168	¿	200	╚	232	Þ
09	HT	(Horizontal Tab)	41	)	73	I	105	i	137	ë	169	®	201	╔	233	Ú
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	¬	202	╩	234	Û
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	ï	171	½	203	╦	235	Ù
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	¼	204	╠	236	ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ì	173	¡	205	═	237	Ý
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ä	174	«	206	╬	238	¯
15	SI	(Shift In)	47	/	79	O	111	o	143	Å	175	»	207	¤	239	´
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	░	208	ð	240	≡
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	▒	209	Ð	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	▓	210	Ê	242	‗
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ô	179	│	211	Ë	243	¾
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ö	180	┤	212	È	244	¶
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	ò	181	Á	213	ı	245	§
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	û	182	Â	214	Í	246	÷
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ù	183	À	215	Î	247	¸
24	CAN	(Cancel)	56	8	88	X	120	x	152	ÿ	184	©	216	Ï	248	°
25	EM	(End of medium)	57	9	89	Y	121	y	153	Ö	185	╣	217	┘	249	¨
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ü	186	║	218	┌	250	·
27	ESC	(Escape)	59	;	91	[	123	{	155	ø	187	╗	219	█	251	¹
28	FS	(File separator)	60	<	92	\	124	|	156	£	188	╝	220	▄	252	³
29	GS	(Group separator)	61	=	93	]	125	}	157	Ø	189	¢	221	¦	253	²
30	RS	(Record separator)	62	>	94	^	126	~	158	×	190	¥	222	Ì	254	■
31	US	(Unit separator)	63	?	95	_			159	ƒ	191	┐	223	▀	255	nbsp
127	DEL	(Delete)														

3.1 *Encryption algorithm : Caesar cipher with Bipartite Graph*

The encryption technique of the proposed algorithm follows the following steps:

1. Find the ASCII values of the plain text from the ASCII table (Table 1). Let these values be s_i where, $i = 1,...,n$ for $n \in \mathbb{N}$..
2. As there are 256 ASCII characters, we make a matrix of order $m * n$ such that $m * n \geq 256$. Although we can take any pair of factors of 256 such as (64×4), (32×8), (128×2) etc but we only take (16×16) as the sum of 16 and 16 is 32 which is less than the sum of the other pairs (like, 64 + 4 = 68, 32 + 8 = 40, 128 + 2 = 130 etc). Taking this will reduce the time and space complexity. Therefore, we form a (16×16) matrix and the elements of the matrix are the ASCII numbers of all the characters arranged in the ascending order.
3. We take the first 16 + 16 = 32 prime numbers which are {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97}. Next, we form a matrix of order 16×16 such that the first 16 prime numbers are leading the columns and next 16 prime numbers are

Table 2

The 16 × 16 matrix including all the ASCII values

	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53
59	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
61	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
67	2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
71	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
73	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
79	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
83	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
89	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
97	8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
101	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
103	10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
107	11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
109	12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
113	13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
127	14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
131	15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

leading the rows. The elements of this matrix are the ASCII values taken from 0 to 255 for all the characters. (Table 2)

4. We introduce a string of any length as the key which is to be shared between the sender and the receiver.
5. We find the corresponding ASCII values of the characters of the shared key string and consider them as k_i, s.
6. We locate the ASCII values of the plain text characters one by one and find the prime numbers leading the row and the column of the location of the values from the Table 2. Take the product of the row heading and the column heading as p_i.
7. We use the concept of Caesar Shift Cipher and shift the p_i's for k_i times to get the new values of a_i's as $a_i = p_i + k_i$ for $i = 1,...,n$ for $n \in \mathbb{N}$. The maximum length of an a_i can be 7198 which is of 4 digits as maximum possible value of p_i is 131 × 53 = 6943 and of k_i is 255. If the key is shorter than the plain text then repeat the key maintaining the order until the end of the plain text.
8. Take two sets v_1 and v_2 where, all the unit's place digits of the a_i's will be in v_2 and the remaining section of the numbers will be in set v_1. Here, all the elements of v_1 will be represented as $4-1=3$ digit numbers. If a section is containing less than three digits, then add 0's in front of the numbers to make them having the length of three digits.

9. Construct a Bipartite Graph taking v_1 and v_2 as the two vertex sets and connect the pair of the vertices if they are making any of the numbers of a_i.
10. Assign some random values(also known as the weights) to the edges in an ascending order maintaining the order of the characters of the plain text one by one such that no two edges will have equal weights. This weighted Bipartite graph is the required Cipher Text.

3.2 Decryption algorithm : Caesar cipher with Bipartite Graph

The recipient receives the labeled bipartite graph as the cipher text and the key.

1. At first, list out all the weights of the edges of the cipher text bipartite graph and arrange them in an ascending order.
2. Find the corresponding edges of the weights and the vertices they are connecting. Consider the set of vertices having three digit numbers as v_1 and the set of vertices having one digit numbers as v_2.
3. Construct the numbers with connected edges taking the three digit numbers as the first section and the one digit numbers as the unit's place. Consider these constructed numbers as a_i's.
4. Find the ASCII values of the characters of the shared key string from Table 1. Consider these values as k_i's.
5. Use the Caesar Shift Cipher to find p_i's defined as $p_i = a_i - k_i$. These obtained p_i's are the product of the prime numbers leading the rows and the columns of the ASCII value matrix Table 2.
6. Find the prime factors of the p_i's. These prime factors are denoting the row and column leading numbers of the 16×16 ASCII value matrix Table 2.
7. Find the numbers located at the intersections of these prime numbered rows and columns from the matrix Table 2. Consider them as s_i's. These obtained numbers are the ASCII values of the required plain text.
9. Find the characters corresponding to these ASCII values from the ASCII table (Table 1). This character string is the final plain text sent by the sender.

4. Examples of the proposed algorithm of Caesar Cipher with Bipartite Graph

Example 4.1: Let us take the plain text "I know! I'll do @ 9:30 p.m." and the key is "Assignment to do." The following are the required steps for the Encryption.

Step 1: The following table (Table 3) finds the final a_i values for the given plain text with the shared key. In the given table,

s_i = ASCII values of the plain text

p_i =Product of the prime numbers leading the row and the column of the location of the ASCII values.

k_i =ASCII values of the characters of the key string.

Table 3

Table to find the a_i's corresponding to the plain text

Plain Text	s_i	Location of s_i 's	p_i	Key String	k_i	$a_i = p_i + k_i$
I	73	(101,11)	1111	A	65	1176
space	32	(59,5)	295	s	115	410
k	107	(107,17)	1819	s	115	1934
n	110	(127,17)	2159	i	105	2264
o	111	(131,17)	2227	g	103	2330
w	119	(89,19)	1691	n	110	1801
!	33	(61,5)	305	m	109	414
space	32	(59,5)	295	e	101	396
I	73	(101,11)	1111	n	110	1221
'	39	(89,5)	445	t	116	561
l	108	(109,17)	1853	*space*	32	1885
l	108	(109,17)	1853	t	116	1969
space	32	(59,5)	295	o	111	406
d	100	(73,17)	1241	*space*	32	1273
o	111	(131,17)	2227	d	100	2327
space	32	(59,5)	295	o	111	406
@	64	(59,11)	649	.	46	695
9	57	(101,7)	704	A	65	772
:	58	(103,7)	721	s	115	836
3	51	(71,7)	497	s	115	612

Contd...

0	48	(59,7)	413	i	105	518
space	32	(59,5)	295	g	103	398
p	112	(59,19)	1121	n	110	1231
.	46	(127,5)	635	m	109	744
m	109	(113,17)	1921	e	101	2022
.	46	(127,5)	635	n	110	745

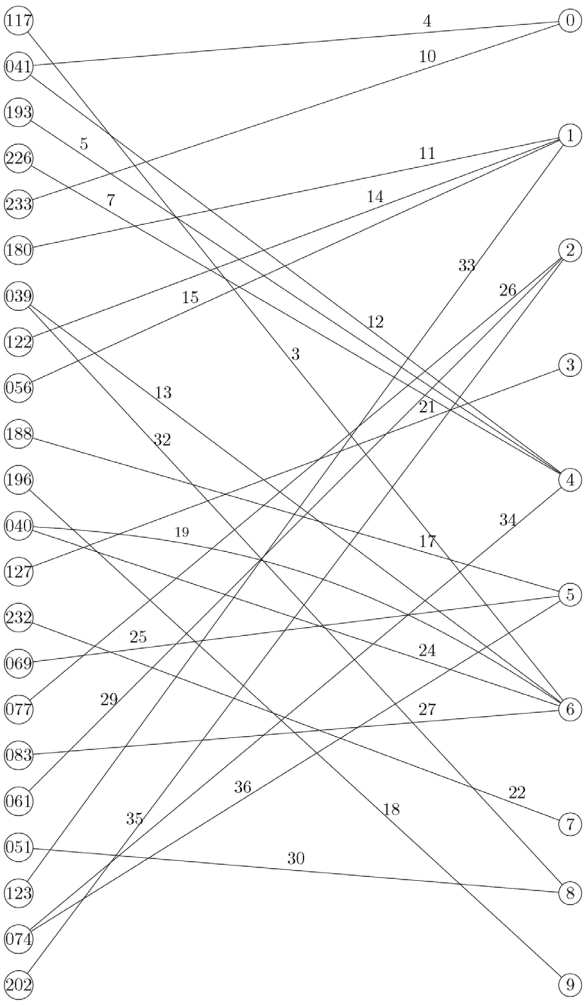

Figure 2

Bipartite Graph as the Cipher Text

Step 2: Form the set v_2 (unit's digits of each number) as v_2 = {6, 0, 4, 1, 5, 9, 3, 7, 2, 8} and the set v_1 taking the remaining section. Therefore, v_1 = {117, 041, 193, 226, 233, 180, 039, 122, 056, 188, 196, 040, 127, 232, 069, 077, 083, 061, 051, 123, 074, 202}

Step 3: Form the Bipartite graph with these vertex sets. Connect the vertices forming the numbers of a_i's and assign random weights to the edges in an ascending order. This weighted bipartite graph is the required cipher text (2).

For Decryption, we have the bipartite graph as the cipher text (Figure 2) and the shared key string: "Assignment to do.". The decryption of this cipher text can be done in the following steps.

Step 1: Arrange all the weights in an ascending order and we get,

3, 4, 5, 7, 10, 11, 12, 13, 14, 15, 17, 18, 19, 21, 22, 24, 25, 26, 27, 29, 30, 32, 33, 34, 35, 36.

Step 2: Finding the corresponding edges related to these weights, we get,

(117, 6), (041, 0), (193, 4), (226, 4), (233, 0), (180, 1), (041, 4), (039, 6), (122, 1), (056, 1), (188, 5), (196, 9), (040, 6), (127, 3)(232, 7), (040, 6), (069, 5), (077, 2), (083, 6), (061, 2), (051, 8), (039, 8), (123, 1), (074, 4), (202, 2), (074, 5)

Step 3: The a_i numbers are formed as:

1176, 410, 1934, 2264, 2330, 1801, 414, 396, 1221, 561, 1885, 1969, 406, 1273, 2327, 406, 695, 772, 836, 612, 518, 398, 1231, 744, 2022, 745

Step 4: We form a table (Table 4) to obtain the final plain text from these a_i values with the help of the ASCII values of the key string "Assignment to do.".

Table 4

Table to find the final plain text

a_i	**Key String**	k_i	p_i	**Location of s_i's**	s_i	**Plain Text**
1176	A	65	1111	(101,11)	73	I
410	s	115	295	(59,5)	32	*space*
1934	s	115	1819	(107,17)	107	k
2264	i	105	2159	(127,17)	110	n
2330	g	103	2227	(131,17)	111	o

Contd...

1801	n	110	1691	(89,19)	119	w
414	m	109	305	(61,5)	33	!
396	e	101	295	(59,5)	32	*space*
1221	n	110	1111	(101,11)	73	I
561	t	116	445	(89,5)	39	‘
1885	*space*	l	1853	(109,17)	108	l
1969	t	116	1853	(109,17)	108	l
406	o	111	295	(59,5)	32	*space*
1273	*space*	d	1241	(73,17)	100	d
2327	d	100	2227	(131,17)	111	o
406	o	111	295	(59,5)	32	*space*
695	.	46	649	(59,11)	64	@
772	A	65	704	(101,7)	57	9
836	s	115	721	(103,7)	58	:
612	s	115	497	(71,7)	51	3
518	i	105	413	(59,7)	48	0
398	g	103	295	(59,5)	32	*space*
1231	n	110	1121	(59,19)	112	p
744	m	109	635	(127,5)	46	.
2022	e	101	1921	(113,17)	109	m
745	n	110	635	(127,5)	46	.

Thus, the Plain text is obtained as “I know! I’ll do @9.30 p.m.”

5. Conclusion

Cryptography is the procedure to securely transfer a data such that the data is protected and accessible by no third party other than the sender and the receiver. Caesar Cipher, a type of shift cipher, is a classical cryptosystem that is executed with the shifts or keys. As this method is likely to be attacked through the brute-force attack, instead of a single numerical value, a string is used as the key to enhance the confidentiality. Implementation of the properties of the Bipartite Graphs from Graph Theory makes this algorithm more approachable. Not only the alphabets, a data including any of all the 256 characters with their ASCII values can be transferred securely with this algorithm. The amalgamation of Graph Theory and Caesar Cipher Encryption not only advances the field of

cryptography but also provides a versatile foundation for future research and development in secure communication systems. The ASCII value serves as a key component in encoding all existing characters, allowing them to be represented and transmitted securely. The purpose of this research is to develop a highly protected algorithm for the encryption procedure that allows confidential transmission of text data with authentication and security. This paper contributes to the flourishing intersection of mathematics and information security, offering a robust framework for Cryptographic schemes in diverse applications.

References

[1] S. Agarwal, "Prime weighted graph in cryptographic system for secure communication," *Int. J. Pure Appl. Math.*, vol. 105, pp. 325-338 (2015).

[2] P. Amudha, A. C. Charles Sagayaraj, and A. C. Shantha Sheela, "An application of graph theory in cryptography," *Int. J. Pure Appl. Math.,* vol. 119, no. 13, pp. 375-383 (2018).

[3] H. Anwar and Z. Shams, "Algorithms of encryption using graph theory," *Math. Sci. Appl.,* vol. 2, pp. 73-85 (2023).

[4] B. Charles, "Cryptosystem using double vertex graph," *Indian J. Sci. Technol.,* vol. 13, pp. 4483-4489 (2020).

[5] B. Charles, P. Venugopal, and B. Praba, "Block encryption and decryption of a sentence using decomposition of the Turan graph," *J. Math.,* vol. 2023, Article ID 7588535, 9 pages (2023).

[6] V. M. Chandrasekaran, B. Praba, A. Manimaran, and G. Kailash, "Data transfer using complete bipartite graph," in IOP Conf. Ser. Mater. Sci. Eng., vol. 263 (2017).

[7] N. Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice Hall (2010).

[8] U. Dixit, "Cryptography: A graph theory approach," *Int. J. Adv. Res. Sci. Eng.,* vol. 6, no. 1 (2017).

[9] W. Etaiwi, "Encryption algorithm using graph theory," *J. Sci. Res. Rep.,* vol. 3, pp. 2519-2527 (2014).

[10] D. Gurjar and A. Krishnaa, "Labeled paths in cryptography," *Adv. Math. Res.* (2021).

11] H. R. Medini, S. Dsouza, D. C. Nayak, and P. Bhat, "Encoding and decoding of messages using graph labeling and complement of a graph," *Global Stoch. Anal.,* vol. 11, no. 1 (2024).

[12] B. Joseph and B. Thomas, "Malaya Journal of Matematik," *Malaya J. Matematik,* vol. 9, no. 1, pp. 470-473 (2021).

[13] P. Kedia and S. Agarwal, "Encryption using Venn diagrams and graph," *Int. J. Adv. Comput. Technol.,* vol. 4, pp. 94-99 (2015).

[14] R. Kuppan, L. Shobana, and I. N. Cangl, "Encrypting and decrypting algorithms using strong face graph of a tree," *Int. J. Comput. Math. Comput. Syst. Theory*, vol. 5, pp. 225-233 (2020).

[15] B. Ni, R. Qazi, S. Rehman, and G. Farid, "Some graph-based encryption schemes," *J. Math.*, vol. 2021, pp. 1-8 (2021).

[16] A. Paszkiewicz, A. Górska, K. Górski, Z. Kotulski, K. Kulesza, and J. Szczepański, "*Proposals of graph-based ciphers, theory and implementations*," (2001).

[17] S. Perera and S. Wijesiri, "Encryption and decryption algorithms in symmetric key cryptography using graph theory," Psychol. Educ., vol. 58, no. 1, pp. 3420-3427 (2021).

[18] P. L. K. Priyadarsini, "A survey on some applications of graph theory in cryptography," *J. Discrete Math. Sci. Cryptogr.,* vol. 18, no. 3, pp. 209-217 (2015).

[19] A. Razaq, G. Alhamzi, S. Abbas, M. Ahmad, and A. Razzaque, "Secure communication through reliable S-box design: A proposed approach using coset graphs and matrix operations," Heliyon, vol. 9, no. 5, May 2 (2023).

[20] D. Sensarma and S. Sarma, "Application of graphs in security," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, pp. 2273-2279 (2019).

[21] C. R. Sharmila and S. Meenakshi, "A fuzzy graph theory approach to symmetric key cryptography," *J. Propuls. Technol.,* vol. 45, no. 1, pp. 467-477 (2024).

[22] W. Stallings, Cryptography and Network Security, 7th ed., Pearson Education Pvt. Ltd., Global Edition (2017).

[23] N. Tokareva, "Connections between graph theory and cryptography," G2C2: Graphs and Groups, Cycles and Coverings, Novosibirsk, Russia, Sept. 24-26 (2014).

[24] V. Ustimenko, "On graph-based cryptography and symbolic computations," *Serdica J. Comput.,* vol. 1 (2007).

[25] M. Yamuna, M. Gogia, A. Sikka, and Md. J. H. Khan, "Encryption using graph theory and linear algebra," *Int. J. Comput. Appl.,* vol. 5, no. 2, Oct. (2012).

[26] M. Yamuna and K. Karthika, "Data transfer using bipartite graphs," *Int. J. Adv. Res. Sci. Eng.,* vol. 4, no. 2, Feb. (2015).

[27] P. S. K. Reddy and P. S. Hemavathi, "Generalization of bipartite graphs," *J. Discrete Math. Sci. Cryptogr.,* vol. 23, no. 3, pp. 787-793 (2020).

[28] S. A. Swadi and A. A. Najim, "The generalized k-connectivity of equally complete bipartite graphs and their line graphs," *J. Discrete Math. Sci. Cryptogr.,* vol. 27, no. 5, pp. 1567-1573 (2024).

Received June, 2024