INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Cryptosystem Using the Generalized Petersen Graph $GP(2n+1, 2)$

C Beaula[1]*, P Venugopal[2]

1 Department of Mathematics, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, 603 110, Tamilnadu, India
2 Department of Mathematics, School of Science and Humanities, Shiv Nadar University Chennai, Kalavakkam, 603 110, Tamilnadu, India

OPEN ACCESS

Received: 06-04-2024

Accepted: 07-06-2024

Published: 02-08-2024

Citation: Beaula C, Venugopal P (2024) Cryptosystem Using the Generalized Petersen Graph $GP(2n+1, 2)$. Indian Journal of Science and Technology 17(30): 3125-3137. https://doi.org/10.17485/IJST/v17i30.1130

* **Corresponding author**.

beaulac@ssn.edu.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objective: In the digital era, data transfer in a network without external interference is one of the challenging problems. External interference can be minimized by creating a strong cryptosystem. For this purpose, different mathematical concepts are incorporated to construct a cryptosystem. Recently, techniques in graph theory, a branch of mathematics, are also employed in cryptography. The objective of this paper is to propose a new cryptosystem to encrypt and decrypt an alphabetical string of lengths less than equal to 16 using graph decomposition and edge labeling on a generalized Petersen graph. **Method**: The edges of the decomposed graphs of the union of the generalized Petersen graph are labelled using the technique of vertex strongly*-graph and these labels are used to encrypt and decrypt the alphabetical string. **Findings:** A graph $G(V, E)$ is a vertex strongly*-graph if there exists a bijection $f : E \rightarrow \{1, 2, \ldots, q\}$ such that $\sum f(uv_i) + \prod f(uv_i)$ are distinct for every vertex $u \in V$, where uv_i are the edges incident to a vertex u. The Generalized Petersen graph $GP(2n+1, 2)$ is proved to be a vertex strongly*-graph. Using this concept, a new cryptosystem is proposed. **Novelty:** The usage of the decomposition of the union of the generalized Petersen graph $GP(2n+1, 2)$ in the cryptosystem is the novelty of this paper. **Application:** To encrypt and decrypt an alphabetical string of size up to 16.

Keywords: Encryption; Decryption; Generalized Petersen Graph; Edge Labeling; The Union of Graphs; Graph Decomposition.

1 Introduction

In cryptography[(1)], *encryption* is a tool for converting a readable message to an unreadable one so the message can be transferred securely without unauthorized intrusion. *Decryption* is the reverse process of encryption. The receiver uses a decryption tool to convert unreadable messages to readable messages using a secret key. Based on these secret keys, cryptography is classified into three kinds. The first one is called *symmetric key cryptography*, which has a single key for encryption and decryption. In this case, the sender and receiver have the same secret key. The second one is *asymmetric key* or *public key cryptography*. In this case, the encryption key is

the public key, which can be shared with anyone, and the decryption key is the private key known only by the receiver. The third one is the *hash function*, which only has one key used for encryption. This function is used in the authentication. By way of usage, cryptography is of two kinds: one is a *stream cipher*, in which the encryption is applied to each bit, and the other one is a block cipher, in which the encryption is applied to a block of a fixed number of bits.

As technology advances at an unprecedented pace, ensuring the security and privacy of individuals has become a major challenge. To meet the challenges strong cryptosystems are proposed using various mathematical techniques. One such technique involves graph concepts, which have proven to be a promising approach. The flexibility of graphs in visualization and structure makes the graph theory more viable to use as a key for building a cryptosystem. The application of graph theory in cryptography adds robustness to cryptosystems. This motivated us to use the graph theory concepts in our proposed work.

Graph theory is a branch of mathematics that helps visualize complicated problems using dots and lines. The definitions of graph theory are taken from[2]. A *graph*G is a triplet $(V,\ E,\ \varphi)$, where V is the set of all vertices, E is a set of all edges and φ is a function from E to V such that $\varphi(e) = (u,\ v), \forall\ e \in E$ and $u,\ v \in V$. A *path* is a sequence of vertices and edges arranged alternatively. A path in which the initial and end vertices coincide is called a *cycle*. If there is a path between every pair of vertices in a graph, then the graph is *connected*, otherwise, it is a *disconnected* graph. A graph H is called a *subgraph* of a graph G; if the vertex set and edge set of H is subsets of a vertex set and edge set of G respectively. Let G be a graph $(V,\ E)$ and $V_1 \subset V$. The *induced subgraph* on V_1 is a subgraph of G, whose vertex set is V_1 and the edge set consists of all edges in G that have both endpoints in V_1. All the graphs considered in this paper are simple, connected, and undirected.

Definition 1.1. [3]*Labelling* in graph theory is a function of assigning numbers to edges or vertices or both. If the vertices are labelled, it is called *vertex labelling*; if the edges are labelled, it is called *edge labelling*.

Definition 1.2.[4] A graph $G(V,\ E)$ is *vertex strongly*-graph* if there exists a bijection $f : E \rightarrow \{1,\ 2,\ \ldots,\ q\}$ such that $\sum f(uv_i) + \prod f(uv_i)$ are distinct for every vertex $u \in V$, where uv_i are the edges incident to vertex u.

Definition 1.3:[2] The *union* of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is a simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. It is denoted by $G_1 \cup G_2$.

Definition 1.4.[5] If $G = \bigcup_{i=1}^{k} H_i$ and $\bigcap_{i=1}^{k} H_i = \phi$ then $H_1, H_2, H_3, \ldots, H_k$ are said to be the *decompositions* of G.

Definition 1.5.[6] The pair $(S,\ T)$ is said to be a *multi-decomposition* of a graph G; if G can be partitioned into copies of S and T with at least one copy of S and one copy of T. Brief survey of some of the graph decompositions are presented here.

Austin and Wagner[7] proved that all orientations of an oriented graph can be factored into triangles, with a large portion of the triangles being transitive, have an ascending subgraph decomposition. This result obtains an ascending subgraph decomposition for any orientation of complete multipartite graphs with $3n$ partite classes, each containing two or four vertices. Ilayaraja and Muthusamy[8] obtained necessary and sufficient conditions for decomposing complete bipartite graphs into cycles and stars with four edges. Sethuraman and Murugan[9] proposed a new conjecture which states that the complete graph K_{4m+1} can be decomposed into copies of two arbitrary trees, each of size m, $m \geq 1$, and gave a decomposition of K_{4cm+1}(c is any positive integer) into copies of a random tree with m edges and copies of either a path with m edges or a star with m edges. El-Mesady, Bazighifan, and Al-Mdallal[10] derived a generalised algorithm for constructing the decompositions of the circulant graphs $C_{2r,r}$, and the circulant graphs $C_{mr,(m-1)r}$ that have mr vertices with $(m-1)r$ degree into different graph classes. Rangasamy and Sangeetha[11] obtained necessary conditions for a $\{P_{k+1},\ C_l\}$-decomposition of a complete multigraph $K_n(\lambda)$ and proved that the necessary conditions are also sufficient when $k = 4$ and $l = 6$.

Graph theory has many applications in engineering, communication networks, artificial intelligence, social networks, data analytics, etc. Recently, graph theory techniques have been used in the construction of cryptosystems to make it unbreakable and keep it hard for unauthenticated intrusion.

Gupta and Selvakumar[12] used connected graphs to construct an innovative algorithm for the cryptosystem. Ni et al.[13] has given three cryptosystems with three graphs to encrypt and decrypt an alphabetical string. In the first cryptosystem, the encryption algorithm converts the alphabet string to a number string; the numbers are transformed using shift cipher $e_n + n(mod 26)$; these shifted numbers are disguised in the Corona graph $C_n \odot K_1$ as vertex labelling, the vertex labelled coronagraph is an encrypted message. The decryption is a reverse process. In the second cryptosystem, the alphabet string is converted to a number string using a defined table; these numbers are under through a shift cipher and then labelled in a bipartite graph. The labelled bipartite graph is an encrypted message. The decryption is a reverse process. In the third cryptosystem, the algorithm converts the alphabet string to a number string: the numbers under through a shift cipher $e_k(x) = x + k(mod\ 26)$, the resulting numbers are disguised with another transformation, In the final stage the numbers are labelled in a star graph $S_{n+1} = K_1 \odot \bar{K}_n$. The labelled star graph is an encrypted message. The decryption is a reverse process.

Hu et al.[14] proposed using a bipartite graph to overcome fraud detection in large advertising systems. Monika[15] applied graph techniques in coding theory and cryptography. Beaula et al.[16],[17] constructed cryptosystems using a double vertex graph, and decomposition of the Turan graph. Auparajita[18] has applied inner magic and inner anti-magic labelled graphs in

data transfer for greater security.

Prasad and Mahato [19] proposed a public key cryptography using Affine-Hill cipher with generalised Fibonacci matrix. This scheme exchange only a pair of number $(\lambda,\ k)$ instead of key matrix, which reduces the time and space complexity. Cusack and Chapman [20] reviewed two cryptographic methods, one constructed with a graph and another without a graph. Their study shows that graph-based cryptosystems may deliver comparable or improved security and performance in many required areas. To test the performance a graph-based cipher system was constructed, including visual cryptography, and tested it against RC4 and AES algorithms for performance and security.

Bekkaoui et al. [21] proposed an encryption method with block cipher using Hamiltonian circuits. The encryption keys are generated by a specific sub-key generator which has been set up according to the requirements of the proposed cryptosystem. El-Mesady et al. [22] have given (i) graph-transversal designs by mutually orthogonal graph squares. (ii) Construction of graph-authentication codes based on mutually orthogonal graph squares. (iii) Applications of graph-transversal designs in key pre-distribution in wireless sensor networks. Dunmore et al. [23] proposed a lightweight encryption algorithm called Matrix Encryption Walks, or MEW, based on existing cryptographic research into graph walks and literature regarding the use of matrices as encryption keys.

Wardak et al. [24] proposed a secure data transmission and retrieval cryptography mechanism using a signed graph, adjacency matrix, and the RSA algorithm. Lavanya and Saravanakumar [25] studied a graph-based data encryption strategy that performs encryption on data in a backend Redundant Array of Independent Disk (RAID) storage using crypto keys generated by the packing colouring process. Data is encrypted when placed on the RAID level storage for the first time; different blocks are encrypted using different keys during the encryption process, increasing security. Hence, this cryptosystem is strong and secure, and an attacker cannot dilute it. The stream cipher used in the system is established from a jump graph of a web graph using the packing coloring process from graph theory. The method generates key streams as invariable lengths according to the size of the graph. This attempt guarantees the existence of a strong cryptosystem.

A lot of study has been done on the generalized Petersen graph [26] [27] [28]. In this paper, we used a generalized Petersen graph to construct a cryptosystem. The generalized Petersen graph $GP(n,\ k)$ is a strong and beautiful structure aptly applicable in encryption. It is defined as follows.

Definition 1.6. [29] The *generalized Petersen graph*$GP(n,\ k)$, $n \geq 3$ and $1 \leq k < \frac{n}{2}$ is a connected and cubic graph with a vertex set $\{u_i, v_i \,|i = 1,\ 2,\ \ldots,\ n\}$ and an edge set $\{u_i u_{i+1},\ u_i v_i,\ v_i v_{i+k} \,|\, i = 1,\ 2,\ \ldots,\ n\}$. See Figure 1.

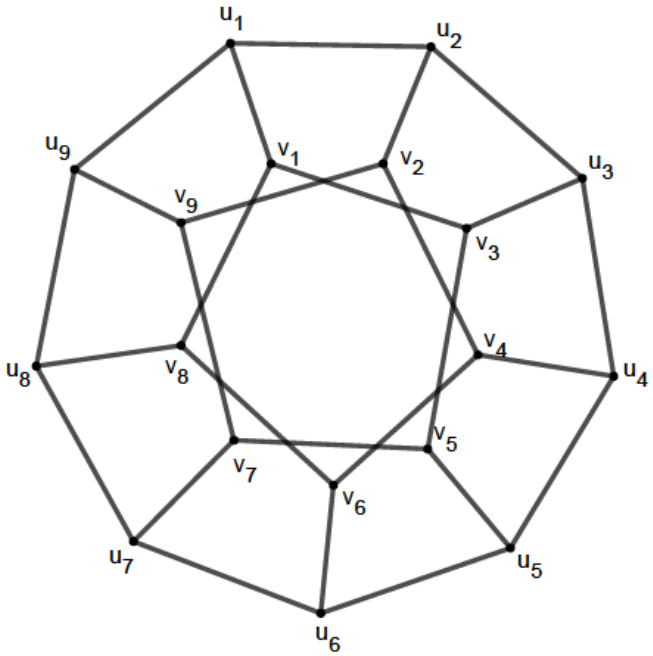

Fig 1. $GP(\mathbf{9},\ \mathbf{2})$

Remark 1.1 $GP(n,\ k)$ is a 3-regular graph consisting of two cycles of the same length and n edges joining the cycles. The vertices $v_1,\ v_2,\ \ldots, v_n$ form the inner cycle, and the vertices $u_1,\ u_2,\ \ldots, u_n$ form the outer cycle. These two cycles are linked

by n edges $(v_i, u_i),\ i = 1, 2, \ldots, n$.

The purpose of using the generalized Petersen graph in a cryptosystem is its structure. This structure is flexible, strongly connected, and complicated to understand without the knowledge of graph theory.

In this paper, we have proposed a mathematical model using graph theory for the cryptosystem, which can be programmed to use in any security-related field. In this paper, union of generalised Petersen graphs is used to encrypt and decrypt the alphabetical strings of lengths between 9 and 16. The structure of the generalised Petersen graph is flexible to use in cryptosystems and is complicated to hack without the knowledge of its structure. Also, we have proved that the generalised Petersen graph is a vertex strongly*-graph and used this way of labelling to label the edges of the generalised Petersen graph. Our model uses three levels of encryption with three symmetric keys to make the cryptosystem more secure. In the first level, the order of encryption is used; in the second level, the union of the generalized Petersen graph is used; and in the third level, the edge label list of the union of the generalized Petersen graph is used.

The paper is organized as follows: the first section is an Introduction- which gives an introduction to the work carried out in this paper, along with a list of basic definitions required for the paper. Section 2 proves the generalized Petersen graph as a vertex strongly*-graph. A union of two generalized Petersen graphs is used to construct the cryptosystem. It will be difficult for anyone to break the system without understanding the structure of the generalized Petersen graph. This section also includes an illustration of the encryption algorithm.

2 Methodology

In this section, we consider a generalized Petersen graph $GP(n,\ k)$ with odd values of n and $k = 2$. In Theorem 2.1, we prove that the $GP(n,\ k)$ is a vertex strongly*-graph. Using the concepts of vertex strongly*-graph and graph decomposition on the union of two copies of the generalized Petersen graphs, a cryptosystem is constructed for alphabetical strings of lengths up to 16. As this graph theory technique is a new approach to data transferring, there is not much work to be compared.

- **Theorem 2.1 .**

A Generalised Petersen graph $GP(n,k)$ is a vertex strongly*-graph for $n = 3,5,7,\ldots$ *and* $k = 2$.

Proof

Consider a generalised Petersen graph $GP(n,\ k)$ for $n = 3,5,7,\ldots$ and $k = 2$.

From definition 1.6, the vertex set and the edge set of $GP(n,\ 2),\ n = 3,5,7,\ldots$ are respectively $\{u_i, v_i \,|\, i = 1, 2, \ldots, n\}$ and $\{u_i u_{i+1},\ u_i v_i,\ v_i v_{i+2} \,|\, i = 1, 2, \ldots, n\}$, the subscripts being reduced to modulo n. It has two cycles of the same length and n edges joining the cycles. The vertices $v_1,\ v_2,\ \ldots, v_n$ form the inner cycle and the vertices $u_1,\ u_2,\ \ldots, u_n$ form the outer cycle. These two cycles are linked by n edges $(v_i\ u_i),\ i = 1, 2, \ldots, n$. The edges in the outer cycle, inner cycle, and between the outer cycle and internal cycle of $GP(n,\ 2)$ are respectively (u_i, u_{i+1}), (v_i, v_{i+2}) and (u_i, v_i), for $i = 1, 2, \ldots, n$.

The edges of $GP(n,\ 2)$ are labelled as follows.

$g(u_i,\ v_i) = 2n + i$,

$g(v_i,\ v_{i+2}) = 2i - 1$,

$g(u_i,\ u_{i+1}) = 2i$, for $i = 1, 2, \ldots, n$ the subscripts being reduced to modulo n.

The vertices of $GP(n,\ 2)$ are labelled using Definition 1.2 as shown below.

$f(v_1) = 4n - 1 + (2n-3)(2n+1)$

$f(v_2) = 4n + 4 + 3(2n-1)(2n+2)$

$f(v_i) = 2n + 5i - 6 + (2n+i)(2i-1)(2i-5)$ for $3 \le i \le n$.

$f(u_1) = 4n + 3 + 4n(2n+1)$

$f(u_i) = 2n + 6i + (2n-3)(2n+1)$ for $2 \le i \le n$.

Claim: The above labelling is distinct for each vertex of $GP(n,\ 2)$.

Let $u,\ v \in GP(n,\ 2)$.

Consider the following cases.

Case (i). Suppose $u,\ v$ are in the outer cycle.

Subcase (a). Let $u = u_i$ and $v = u_{i+1}$.

$f(u_i) = 2n + 6i + (2n-3)(2n+1)$

$f(u_{i+1}) = 2n + 6(i+1) + (2n-3)(2n+1)$

$\quad = 2n + 6i + (2n-3)(2n+1) + 6$

$\quad = f(u_i) + 6$

Therefore, $f(u_i) \neq f(u_{i+1})$

Subcase (b). Let $u = u_i$ and $v = u_{i+m} (m \neq 1,\ i-1)$.
$f(u_i) = 2n+6i+(2n-3)(2n+1)$
$f(u_{i+m}) = 2n+6(i+m)+(2n-3)(2n+1)$
$= 2n+6i+(2n-3)(2n+1)+6m$
$= f(u_i)+6m$
Therefore, $f(u_i) \neq f(u_{i+m})$
Case (ii). Suppose $u,\ v$ are in the inner cycle.
Subcase (a). Let $u = v_i$ and $v = v_{i+1}$.
$f(v_i) = 2n+5i-6+(2n+i)(2i-1)(2i-5)$
$f(v_{i+1}) = 2n+5(i+1)-6+(2n+i+1)(2(i+1)-1)(2(i+1)-5)$
$f(v_{i+1}) = f(v_i)+5+(2i+1)(2i-3)+8(i-1)(2n+i)$
Therefore, $f(v_i) \neq f(v_{i+1})$.
Subcase (b). Let $u = v_i$ and $v = v_{i+m} (m \neq 1,\ i-1)$.
$f(v_i) = 2n+5i-6+(2n+i)(2i-1)(2i-5)$
$f(v_{i+m}) = 2n+5(i+m)-6+(2n+i+m)(2(i+m)-1)(2(i+m)-5)$
$f(v_{i+m}) = f(v_i)+5m+4m\left((i+m)^2-6(i+m)+5\right)+2m(2n+i)(4i+2m-6)$
Therefore, $f(v_i) \neq f(v_{i+m})$.
Case (iii). Consider one vertex in the outer cycle and another vertex in the inner.
Subcase (a). Let $u = u_1$ and $v = v_1$.
$f(u_1) = 4n+3+4n(2n+1)$
$f(v_1) = 4n-1+(2n-3)(2n+1)$
$f(v_1) = f(u_1)-4-(2n+3)(2n+1)$
Therefore, $f(v_1) \neq f(u_1)$.
Subcase (b). Let $u = u_2$ and $v = v_2$.
The proof is similar to **Subcase (a).**
Subcase (c). Let $u = u_i$ and $v = v_i, 3 \leq i \leq n$
$f(v_i) = 2n+5i-6+(2n+i)(2i-1)(2i-5)$
$f(u_i) = 2n+6i+(2n-3)(2n+1)$
$f(u_i) = f(v_i)+6+i+(2n-3)(2n+1)-(2n+i)(2i-1)(2i-5)$
Therefore, $f(u_i) \neq f(v_i)$ for $3 \leq i \leq n$.
Subcase (d). Let $u = u_i$ and $v = v_{i+m} (m \neq 0,\ 1 \leq m \leq n-1)$, the subscripts being reduced to modulo n.
$f(u_i) = 2n+6i+(2n-3)(2n+1)$
$f(v_{i+m}) = 2n+5(i+m)-6+(2n+i+m)(2(i+m)-1)(2(i+m)-5)$
$f(v_{i+m}) = f(u_i)-(2n-3)(2n+1)+(2n+i+m)(2(i+m)-1)(2(i+m)-5)-i+5m-6$
Therefore $f(v_{i+m}) \neq f(u_i)$ for $(m \neq 0,\ 1 \leq i+m \leq n)$.
Therefore, every pair of vertices $u,\ v$ of $GP(n,\ 2)$ has distinct labelling.
Hence, $GP(n,2)$ is a vertex strongly*graph.

- **Illustration for $GP(\mathbf{n},\ \mathbf{2})$as a vertex strongly*graph**

Consider $GP(9,\ 2)$. Its edges and vertices are labelled using Theorem 2.1., the edge labelling of $GP(9,\ 2)$ is shown in Figure 2.
The calculation $f(uv_i) + f(uv_i)$ for each vertex of $GP(9,\ 2)$ is shown in Table 1.

Table 1. Calculation for vertex labeling

v_i	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
$f(\mathbf{v_i})$	320	1060	132	494	1072	1890	2972	4342	6024
u_i	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
$f(\mathbf{u_i})$	723	186	535	1092	1881	2926	4251	5880	7837

Table 1 shows the vertices of GP(9,2) yields a distinct label.
Hence, the generalized Petersen graph $GP(9,2)$ is a vertex strongly*graph.
Remark 2.1 [30] For the cryptosystem, we consider two copies of generalized Petersen graphs. The first generalized Petersen graph is named $GP_{uv}(n,2)$, and its copy is taken as $GP_{vu}(n,2)$.

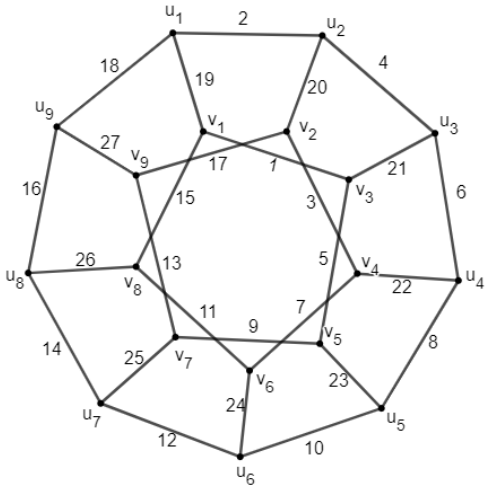

Fig 2. Edge labeling of *GP(9,2)*

The outer and inner cycles of $GP_{uv}(n,2)$ are denoted as C_{21} and C_{11} with edges joining them as J are listed below.

$C_{21}: u_1, u_2, \ldots, u_i, u_{i+1}, \ldots, u_n, u_1$,

$C_{11}: v_1, v_3, \ldots, v_i, v_{i+2}, \ldots, v_{n-1}, v_1$, and

$J: (u_1, v_1), (u_2, v_2), \ldots, (u_i, v_i), (u_{i+1}, v_{i+1}), \ldots, (u_n, v_n)$, the subscripts being reduced to modulo n. For example, $GP_{uv}(n,2)$ for $n = 9$ is shown in Figure 3.

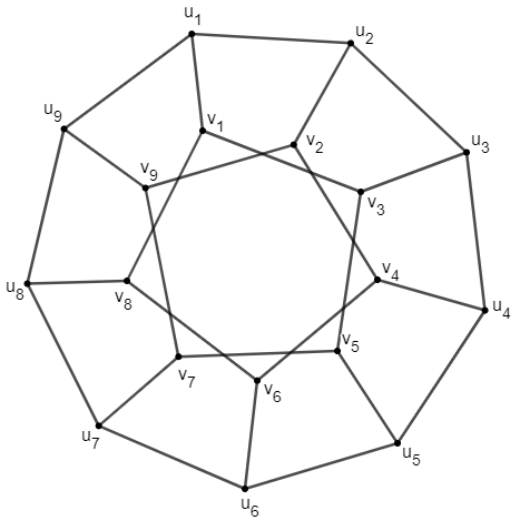

Fig 3. $GP_{\mathbf{uv}}(\mathbf{9},\mathbf{2})$

Similarly, the outer and inner cycles of $GP_{vu}(n,2)$ are denoted as C_{22} and C_{12} with edges joining them as J are listed below.

$C_{22}: u_1, u_3, \ldots, u_i, u_{i+2}, \ldots, u_{n-1}, u_1$,

$C_{12}: v_1, v_2, \ldots, v_i, v_{i+1}, \ldots, v_n, v_1$, and

$J: (u_1, v_1), (u_2, v_2), \ldots, (u_i, v_i), (u_{i+1}, v_{i+1}), \ldots, (u_n, v_n)$, the subscripts being reduced to modulo n. For example, $GP_{vu}(n,2)$ for $n = 9$ is shown in Figure 4.

The union of the above two generalized Petersen graphs denoted as $GP_{uv}(n,2) \cup GP_{vu}(n,2)$ with a vertex set $\{u_i, v_i \,|i = 1, 2, \ldots, n\}$ and the edge set $\{u_i u_{i+1}, v_i v_{i+1}, u_i v_i, v_i v_{i+2}, u_i u_{i+2} \,|1 \leq i < n\}$, the subscripts being reduced to modulo n. For example, $GP_{uv}(n,2) \cup GP_{vu}(n,2)$ for $n = 9$ is shown in Figure 5.

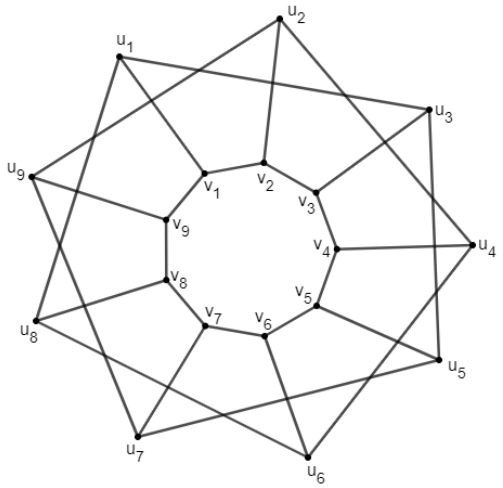

Fig 4. $GP_{\mathbf{vu}}(\mathbf{9},\mathbf{2})$

Remark 2.2.[30] $GP_{uv}(n,2)$ has an outer cycle C_{21} and inner cycle C_{11} with n edges joining them. Similarly, $GP_{vu}(n,2)$ has an outer cycle C_{22} and inner cycle C_{12} with n edges joining them. Therefore, the union of these two graphs $GP_{uv}(n,2) \cup GP_{vu}(n,2)$ has four cycles C_{21}, C_{11}, C_{22}, C_{12} and n edges. The graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ is decomposed into four cycles $C_{11}, C_{12}, C_{21}, C_{22}$ and nine edges.

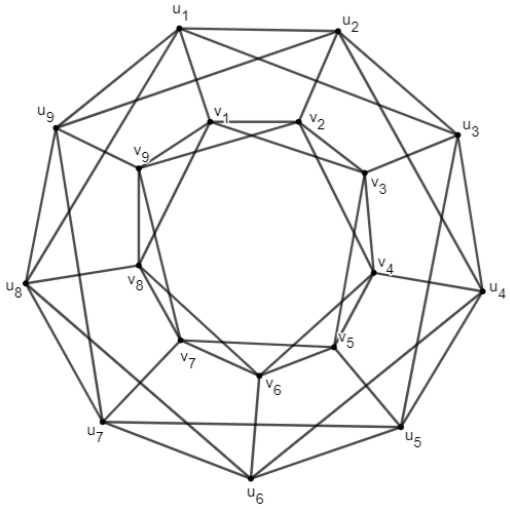

Fig 5. $GP_{\mathbf{uv}}(\mathbf{9},\mathbf{2}) \bigcup GP_{\mathbf{vu}}(\mathbf{9},\mathbf{2})$

2.1 Cryptosystem

A cryptosystem is proposed using the union of generalized Petersen graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$. The edges of $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ are labelled with the numbers from the list $C_E = \{1, 2, 3, 4, 5, 6, 7, \ldots, 150\}$ using the edge-labelling concept of vertex strongly*-graph to encrypt and decrypt an alphabetical string of length less than or equal to 16. Each encryption gives a ciphertext, which is a jumble from the list C_E.

In the proposed cryptosystem, to encode the alphabets the following two tables are considered - one with odd numbers Table 1 and the other with even numbers Table 2.

In these tables, the numbers 1 and 2 are not listed as they are reserved to label the extra edge in the encryption.

Table 2. Encoding Table with odd numbers

Alphabet	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
Coding Number	3	5	7	9	11	13	15	17	19	21	23	25	27
Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Coding Number	29	31	33	35	37	39	41	43	45	47	49	51	53
Alphabet	P_1	P_2	P_3	P_4	P_5	P_6	P_7	R_1	R_2	R_3	R_4	R_5	R_6
Coding Number	55	57	59	61	63	65	67	70	72	74	76	78	80

Table 3. Encoding Table with even numbers

Alphabet	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
Coding Number	4	6	8	10	12	14	16	18	20	22	24	26	28
Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Coding Number	30	32	34	36	38	40	42	44	46	48	50	52	54
Alphabet	P_1	P_2	P_3	P_4	P_5	P_6	P_7	R_1	R_2	R_3	R_4	R_5	R_6
Coding Number	56	58	60	62	64	66	68	70	72	74	76	78	80

In the subsequent subsections, we discuss the order of encryption, the method of encrypting the strings on the cycles C_{11}, C_{21} of $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ based on the order of encryption and finally the edge-labels of $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ are listed as the encrypted message.

2.1.1 Order of Encryption

The proposed cryptosystem encrypts and decrypts plaintext of length from 9 to 16. When the first plaintext is encrypted, then the encryption is said to be *first order of encryption*. When the encryption is used for the second plaintext, then the encryption is said to be of *second order of encryption*. Therefore, the n^{th} order of encryption indicates the n^{th} plaintext encryption.

In the proposed encryption, the plaintext is parted into two strings. The first eight letters of the plaintext are considered as the first string, and the remaining letters are taken as the second string. In this encryption, a plaintext is converted into a number string. The conversion of plaintext into a number string depends on the order of encryption in which a plaintext is encrypted. This order of encryption is of two cases depending on the odd or even order of encryption.

Case (i): Odd order of encryption

(a) Convert the first eight-letter string into a number string using Table 2 of odd numbers.

(b) Convert the remaining letters into a number string using Table 3 of even numbers. If the number of letters in the second string is less than 8, place the numbers P_1, P_2, P_3, P_4, P_5, P_6, and P_7 to pad the remaining places from Table 3 to make it of length 8.

Case (ii): Even order of encryption

(a) Convert the first eight-letter string into a number string using Table 3 of even numbers.

(b) Convert the remaining letters into a number string using Table 2 of odd numbers. If the number of letters in the second string is less than 8, place the numbers P_1, P_2, P_3, P_4, P_5, P_6, and P_7 to pad the remaining places from Table 2 to make it of length 8.

Repetition of letters in the plaintext: If a letter L is repeated in a word, then the repeated letters are labelled as follows.

First L : corresponding number from Encoding table.

Second L : corresponding number from Encoding table + R_1

Third L : corresponding number from Encoding table + R_2

And, so on as per the requirement.

2.1.2. Encryption of number string on GP_{uv}(9,2) U GP_{vu} (9,2)

For encryption of number string on the union of two generalized Petersen graphs, only two cycles C_{11}, C_{21} are considered as defined below.

Case (i): Odd order of encryption

(a) The first number string is encrypted on C_{11}. The first four edges of the cycle C_{11} are labelled with the first four numbers of the number string, the fifth edge is labelled 1, and the last four edges are labelled with the remaining numbers.

(b) The second number string is encrypted on C_{21}. The first four edges of the cycle C_{21} are labelled with the first four numbers of the number string, the fifth edge is labelled 2, and the last four edges are labelled with the remaining numbers.

Case (ii): Even order of encryption

(a) The first number string is encrypted on C_{21}. The first four edges of the cycle C_{21} are labelled with the first four numbers of the number string, the fifth edge is labelled 2 and the last four edges are labelled with the remaining numbers.

(b) The second number string is encrypted on C_{11}. The first four edges of the cycle C_{11} are labelled with the first four numbers of the number string, the fifth edge is labelled 1, and the last four edges are labelled with the remaining numbers.

2.1.3. The encrypted message

The encryption algorithm converts a plaintext into a ciphertext or encrypted message. The encrypted message is the edge weight list of $GP_{uv}(9,2) \cup GP_{vu}(9,2)$. The encrypted message depends on the order of encryption. Based on the order of encryption, the ciphertext is listed in two different ways. If the order of encryption is odd, the edge weights of the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ generate the ciphertext as defined in the odd listing. If the order of encryption is even, the edge weights of the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ generate the ciphertext as defined in the even listing.

Case (i): Odd listing

The first edge of C_{11}, the first edge of C_{12}, joining edge J, the first edge of C_{21}, the first edge of C_{22}, the second edge of C_{11}, the second edge of C_{12}, joining edge J, the second edge of C_{21}, the second edge of C_{22}, …, ninth edge of C_{11}, ninth edge of C_{12}, joining edge J, ninth edge of C_{21}, ninth edge of C_{22}.

Case (ii): Even listing

The first edge of C_{21}, the first edge of C_{22}, the joining edge J, the first edge of C_{11}, the first edge of C_{12}, the second edge of C_{21}, the second edge of C_{22}, joining edge J, the second edge of C_{11}, the second edge of C_{12}, …, ninth edge of C_{21}, ninth edge of C_{22}, joining edge J, ninth edge of C_{11}, ninth edge of C_{12}.

2.2. Encryption Algorithm

In this section, an algorithm is proposed to encrypt an alphabetical string of length 9 to 16 into an encrypted message of fixed length 45.

Input: Alphabetical string (plaintext) of length 9 to 16

Output: Encrypted message of length 45

Symmetric key: Order of encryption

Step 1: Consider the first eight alphabets of the plain text as the first string, and the remaining alphabets as the second string. Convert the alphabet strings into number strings as defined in Section 2.1.1

Step 2: Decompose the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ into four cycles C_{11}, C_{12}, C_{21}, C_{22} and nine edges J : $(u_1,v_1),(u_2, v_2),(u_3, v_3), (u_4, v_4), (u_5, v_5),(u_6, v_6), (u_7, v_7), (u_8, v_8), (u_9,v_9)$.

Step 3: Label the number strings in the cycles C_{11} and C_{21} as instructed in Section 2.1.2.

Step 4: The remaining cycles C_{12}, C_{22}, and the edges of J are labelled as defined in Theorem 2.1 with the unused numbers of edge label list C_E.

Step 5: The labelled cycles and edges in Step 3 and Step 4 together give the edge labelled graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$. The edge weight list of the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ is the cipher text – which depends on the order of encryption as defined in Section 2.1.3

2.3. Decryption Algorithm

In this section, an algorithm is proposed to decrypt an encrypted message (ciphertext) to an alphabetical string.

Input: Cipher text of length 45

Output: Plain text of length 9 to 16

Symmetric key: Order of encryption

Step 1: Label the edges of the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ using the input as follows.

Case (i): Odd order of encryption

If the order of encryption is odd, use the procedure given in odd listing to label the edges of the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$.

Case (ii): Even order of encryption

If the order of encryption is even, use the procedure given in even listing to label the edges of the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$.

Step 2: From the labelled graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$, consider only the edge labels of the cycles C_{11} and C_{21} and convert them to alphabets as follows.

Case (i): Odd order of encryption

If the order of encryption is odd, convert the edge labels of the cycles C_{11} and C_{21} to alphabets using Table 2 and Table 3, respectively, after removing the edge labels 1 and 2. The alphabets decrypted from the cycles C_{11} and C_{21} respectively gives the first and second strings of the plaintext.

Case (ii): Even order of encryption

If the order of encryption is even, convert the edge labels of the cycles C_{21} and C_{11} to alphabets using Table 3 and Table 2, respectively, after removing the edge labels 1 and 2. The alphabets decrypted from the cycles C_{21} and C_{11} respectively gives the first and second strings of the plaintext.

Step 3: The decrypted message is the output.

Output: Plaintext of length 9 to 16.

2.4. Illustration for Encryption Algorithm

In this section, an illustration of the above encryption algorithm is presented to encrypt a plaintext with an order of encryption 101.

Input: GRAPHTHEORY

Output: Cipher text of length 45

Symmetric key: 101 (odd order)

Step 1: Split the plain text as follows.

First string: **GRAPHTHE**

Second string: **ORY**

Step 2: Decompose the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ into four cycles C_{11}, C_{12}, C_{21}, C_{22} and nine edges J : $(u_1,v_1),(u_2, v_2),(u_3, v_3), (u_4, v_4), (u_5, v_5),(u_6, v_6), (u_7, v_7), (u_8, v_8), (u_9,v_9)$.

Step 2: Convert the first string into a number string using Table 2. The corresponding number string is 15, 37, 3, 33, 17, 41, 87, 11. Consider the cycle C_{11} of the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$. Label the first four edges of the cycle C_{11} with the first four numbers of the number string. Label the fifth edge with 1. Then label the remaining four edges of the cycle C_{11} with the last four numbers of the number string. See Figure 6.

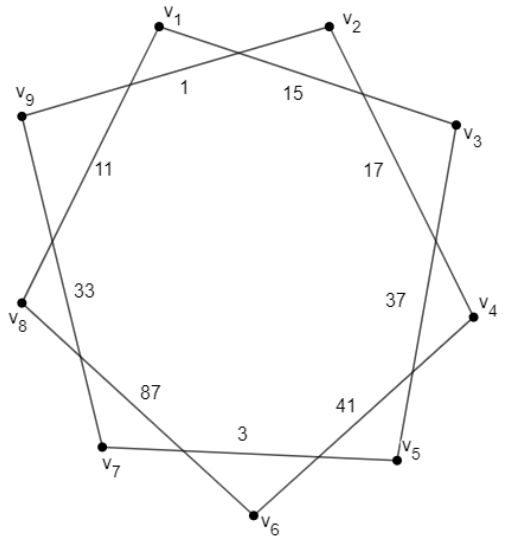

Fig 6. Cycle C_{11}

Convert the second string into a number string using Table 3. As the length of the number string is less than 8, the padding numbers from Table 3 are used to make it of length 8. Therefore, the number string of the second string is 32, 38, 52, 56, 58, 60, 62, 64. Consider the cycle C_{21} of the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$. Label the first four edges of the cycle C_{21} with the

first four numbers of the number string. Label the fifth edge with 2. Then the remaining four edges of the cycle C_{21} are labelled with the remaining four numbers of the number string. See Figure 7.

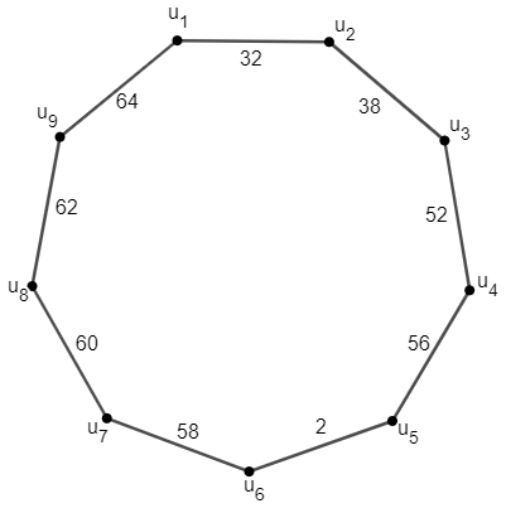

Fig 7. Cycle C_{21}

Step 3: Label the remaining cycles C_{12}, C_{22}, and edge set J edges of the graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ with the numbers of edge labelling lists C_E not used in C_{11} and C_{21}, without repeating the numbers. See Figure 8.

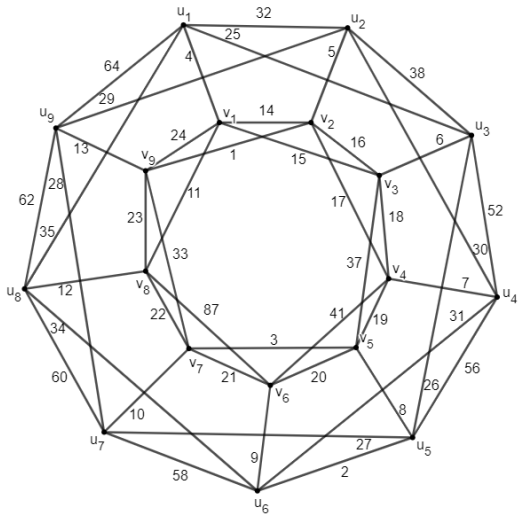

Fig 8. Encrypted $GP_{\mathbf{u}v}$(9,2)$\cup$ $GP_{v\mathbf{u}}$(9,2)

Step 4: Extract the edge labels and list them by odd listing, which gives the following cipher text as output.

Ciphertext: {15, 14, 4, 32, 25, 37, 16, 5, 38, 26, 3, 18, 6, 52, 27, 33, 19, 7, 56, 28, 1, 20, 8, 2, 29, 17, 21, 9, 58, 30, 41, 22, 10, 60, 31, 87, 23, 12, 62, 34, 11, 24, 13, 64, 35}.

3 Results and Discussion

The encryption algorithm presented under Section 2 converts any alphabetical string of length between 9 and 16 to a cyphertext of length 45 using the concept of graph decomposition on the union of generalized Petersen graph. By decryption algorithm, the original text can be retrieved.

The graph used in this paper for encryption and decryption is the union of two Generalised Petersen graphs $GP_{uv}(9,2) \cup GP_{vu}(9,2)$. This graph is decomposed into four cycles C_{11}, C_{12}, C_{21},$C_{22,}$ and nine distinct edges J. For variable lengths of the plaintext, different combinations of the decompositions of generalised Petersen graph and their union can be used to construct cryptosystems; a few of the possibilities are discussed below.

(a) Encrypting a plaintext of length 1 to 8

Plain text of lengths 1 to 8 is primarily used in the case of password authentication. To encrypt this plain text, one cycle of length 9 is enough.

Case (i): Encryption using a generalized Petersen graph $GP_{\mathbf{uv}}(\mathbf{9},\mathbf{2})$.

For cryptosystems, one of C_{11}, C_{12}, or J of $GP_{uv}(9,2)$ can be used. The choice of C_{11}, C_{12}, or J gives three ways of constructing the cryptosystem with ciphertext of length 27.

Case (ii): Encryption using $GP_{\mathbf{uv}}(\mathbf{9},\mathbf{2}) \bigcup GP_{\mathbf{vu}}(\mathbf{9},\mathbf{2})$.

For cryptosystem one of C_{11}, C_{12}, C_{21}, C_{22} or J of $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ can be used. The choice of $C_{11}, C_{12}, C_{21}, C_{22}$ or J gives five ways of constructing the cryptosystem with the ciphertext of length 45.

(b) Encrypting a plaintext of length 9 to 16

The graph $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ is decomposed into four cycles C_{11}, C_{12}, C_{21}, C_{22}, and nine edges J. For plaintext of length 9 to 16, any two of the decomposed graphs from $5C_2$ combinations $\{\{C_{11}, C_{12}\}, \{C_{11}, C_{21}\}, \{C_{11}, C_{22}\}, \{C_{11}, J\}, \ldots, \{C_{22}, J\}\}$ can be considered for constructing the cryptosystem. In the proposed cryptosystem of this chapter, only $\{C_{11}, C_{21}\}$ combination of two cycles is considered. Similarly, the cryptosystems can be constructed with any other two combinations.

(c) Encrypting a plaintext of length 17 to 24

For encrypting of plaintext of length 17 to 24, two combinations of the decomposed graphs of $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ are not sufficient. Increasing the plaintext length will require more edges for encrypting the numbers. In this case, three of the decomposed graphs of $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ from $5C_3$ combinations $\{\{C_{11}, C_{12}, C_{21}\}, \{C_{11}, C_{12}, C_{22}\}, \{C_{11}, C_{12}, J\}, \{C_{11}, C_{21}, C_{22}\}, \ldots, \{C_{21}, C_{22}, J\}\}$ can be considered for constructing the cryptosystem.

(d) Encrypting a plaintext of length 25 to 32

For encrypting the plaintext of lengths 25 to 32, four of the five decomposed graphs of $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ from $5C_4$ combinations $\{\{C_{11}, C_{12}, C_{21}, C_{22}\}, \{C_{11}, C_{12}, C_{21}, J\}, \{C_{11}, C_{12}, C_{22}, J\}, \{C_{11}, C_{21}, C_{22}, J\}, \{C_{12}, C_{21}, C_{22}, J\}\}$ can be considered for constructing the cryptosystem.

(e) Encrypting a plaintext of length 33 to 40

To encrypt a plaintext of length 32 to 40, all the five decomposed graphs C_{11}, C_{12}, C_{21}, C_{22}, J of $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ can be considered.

(f) Encrypting a plaintext of length more than 40

To encrypt a plaintext of size more than 40 needs a graph $GP_{uv}(n,2) \cup GP_{vu}(n,2)$ with $n > 9$ as the union of generalized Petersen graphs $GP_{uv}(9,2) \cup GP_{vu}(9,2)$ can encrypt a plaintext of maximum length 40 with its 45 edges distributed in four cycles and nine single edges.

4 Conclusion

The proposed cryptosystem can encrypt an alphabetical string of length varying from 9 to 16. For plain texts greater than 16, the cases discussed in the previous section can be taken as an open problem, and different crypto graphical techniques can be incorporated to construct cryptosystems. Only the alphabet strings are considered here as plain text. To convert the alphabet to numbers, we used two tables, Table 2 and Table 3. These tables can be modified, and numerals can be inserted so the plain text is alphanumeric or only numeric in different sizes. This same model can be executed differently to encrypt longer or shorter plain texts or authenticate an identity. This paper gives the graph theoretical approach to cryptography. The generalized Petersen graph is a vertex strongly*-graph, which is proved in this paper. The union of two generalized Petersen graphs was constructed and applied in the cryptosystem, which will be difficult for anyone who tries to break it because of its structure.

References

1) Stallings W. Cryptography and Network Security. Pearson Education Inc. 2020. 2020. Available from: https://doi.org/10.1201/b11517-4.
2) Bondy JA, Murty USR. Graph Theory with Applications. Macmillan Press. 1976. Available from: https://doi.org/10.37236/11668.
3) Gallian JA. A Dynamic Survey of Graph Labeling. *The Electronic Journal of Combinatorics*;1000:2020–2020. Available from: https://doi.org/10.37236/11668.

4) Beaula C, Baskar JB. On Vertex Strongly*-graph. *Proceedings of International Conference Mathematics and Computer Science*. 2008;p. 25–26. Available from: https://doi.org/10.37236/11668.
5) Bosak J. Decomposition of Graphs. Dordrecht. Kluwer Academic Publishers. 1990.
6) Abueida AA, Daven M. Multi-decompositions of the complete graph. *ARS Combinatoria*. 2004;72:17–22. Available from: https://doi.org/10.61091/ars.
7) Andrea DA, Brian CW. Ascending Subgraph Decompositions of Oriented Graphs that Factor into Triangles. *Discussiones Mathematicae Graph Theory*. 2022;42(3):811–822. Available from: https://doi.org/10.7151/dmgt.2306.
8) Ilayaraja M, Muthusamy A. Decomposition of Complete Bipartite Graphs into Cycles and Stars with Four Edges. *AKCE International Journal of Graphs and Combinatorics*. 2020;17(3):697–702. Available from: https://doi.org/10.1016/j.akcej.2019.12.006.
9) Sethuraman G, Murugan V. Decomposition of complete graphs into arbitrary trees. *Graphs and Combinatorics*. 2021;37:1191–1203. Available from: https://doi.org/10.1007/s00373-021-02299-5.
10) El-Mesady A, Omar B, Qasem A. On infinite circular-balanced complete graphs decompositions based on generalised algorithmic approaches. *Alexandria Engineering Journal 2022*;61(12):11267–11275. Available from: https://doi.org/10.1016/j.aej.2022.04.022.
11) Rangasamy C, Sangeetha R. Kn (λ) is fully {P5, C6}-decomposable. *Contributions to Discrete Mathematics*. 2022;17(1):1–12. Available from: https://cdm.ucalgary.ca/article/view/69695.
12) Selvakumar R, Gupta N. Fundamental circuits and cutsets used in cryptography. *Journal of Discrete Mathematical Sciences and Cryptography*. 2012;15:287–301. Available from: https://doi.org/10.1080/09720529.2012.10698381.
13) Baizhu N, Rabiha Q, Shafiq UR, Ghulam F. Some Graph-based encryption schemes. *Journal of Mathematics*;2021:1–8. Available from: https://doi.org/10.1155/2021/6614172606.
14) Hu J, Liang J, S D. A bipartite graph propagation approach for mobile advertising 607 fraud detection. *Mobile Information Systems*. 2017;p. 6412521–6412521. Available from: https://doi.org/10.1155/2017/6412521.
15) Monika P, Urszula R, Vasyl U, Aneta W. One the application of extremal graph theory to coding theory and cryptography. *Electronic Notes in Discrete Mathematics*. 2013;43:329–342. Available from: https://doi.org/10.1016/j.endm.2013.07.051.
16) Beaula C, Venugopal P. Cryptosystem using double vertex graph. *Indian Journal of Science and Technology*. 2020;13(44):4483–4489. Available from: https://doi.org/10.1745/IJST/v3i44.1699.
17) Beaula C, Venugopal P, Praba B. Block encryption and decryption of a sentence using decomposition of the Turan graph. *Journal of Mathematics*;2023:7588535–7588535. Available from: https://doi.org/10.1155/2023/7588535.
18) Auparajita K. Inner magic and inner anti-magic graphs in cryptography. *Journal of Discrete Mathematical Sciences and Cryptography*. 2019;(6):1057–1066. Available from: https://doi.org/10.1080/09720529.2019.1675298.
19) Prasad K, Mahato H. Cryptography using generalized Fibonacci matrices with Affine-Hill cipher. *Journal of Discrete Mathematical Sciences and Cryptography*. 2022;25(8):2341–2352. Available from: https://doi.org/10.1080/09720529.2020.1838744.
20) Cusack B, Chapman E. Using graphic methods to challenge cryptographic performance. In: Proceedings of 14th Australian Information Security Management Conference. 2016;p. 30–36. Available from: https://doi.org/10.4225/75/58a6991e71023.
21) Bekkaoui K, Ziti S, Omary F. Data Security: A New Symmetric Cryptosystem. *International Journal of Advanced Computer Science and Applications*. 2021;12(9):742–750. Available from: https://doi.org/10.14569/IJACSA.2021.0120982.
22) El-Mesady A, Bazighifan O, Shabana HM. On Graph-Transversal Designs and Graph-Authentication Codes Based on Mutually Orthogonal Graph Squares. *Journal of Mathematics*. 2022;2022:1–10. Available from: https://dx.doi.org/10.1155/2022/8992934.
23) Dunmore A, Samandari J, Jang-Jaccard J. Matrix Encryption Walks for Lightweight Cryptography. *Cryptography*. 2023;7(3):41–41. Available from: https://dx.doi.org/10.3390/cryptography7030041.
24) Wardak O, Sinha D, Sethi A. Encryption and decryption of signed graph matrices through RSA algorithm. *Indian Journal of Pure and Applied Mathematics*. 2023;p. 1–8. Available from: https://dx.doi.org/10.1007/s13226-023-00452-9. doi:10.1007/s13226-023-00452-9.
25) Lavanya S, Saravanakumar NM. Secured two factor authentication, graph based replication and encryption strategy in cloud computing. *Multimedia Tools and Applications*. 2023;82(11):16105–16125. Available from: https://dx.doi.org/10.1007/s11042-022-13838-4.
26) John BG, Cheryl ZX. A note on isomorphic generalized Petersen graphs with an application to the crossing number of GP. *Discrete Mathematical Letters*. 2019;2:44–46. Available from: https://www.dmlett.com/archive/DML19_v2_p.44_46.pdf.
27) Ma G, Wang J, Klavžar S. On Distance-Balanced Generalized Petersen Graphs. *Annals of Combinatorics*. 2024;28(1):329–349. Available from: https://dx.doi.org/10.1007/s00026-023-00660-4.
28) Iqbal T, Bokhary SAUH, Hilali SO, Alhagyan M, Gargouri A, Azhar MN. Edge Resolvability in Generalized Petersen Graphs. *Symmetry*. 2023;15(9):1633–1633. Available from: https://dx.doi.org/10.3390/sym15091633.
29) Nasir S, Idrees N, Sadiq A, Farooq FB, Kanwal S, Imran M. Strongly Multiplicative Labeling of Diamond Graph, Generalized Petersen Graph, and Some Other Graphs. *Journal of Mathematics*. 2022;2022:1–5. Available from: https://dx.doi.org/10.1155/2022/3203108.
30) Beaula C, Venugopal P. Decomposition of the union of generalized Peterson graph. .