Malaysian Journal of Mathematical Sciences 18(2): 371–397 (2024)
https://doi.org/10.47836/mjms.18.2.10

Malaysian Journal of Mathematical Sciences

Journal homepage: https://mjms.upm.edu.my

Encryption and Decryption Using Decomposition of Complete Graph $K_{3(6n+1)}$

Beaula, C.*[1], Venugopal, P.[2], and Sujaudeen, N.[3]

[1]*Department of Mathematics, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam 603 110, Tamil Nadu, India*
[2]*Mathematics, School of Science and Humanities, Shiv Nadar University Chennai, Kalavakkam 603 110, Tamil Nadu, India*
[3]*Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam 603 110, Tamil Nadu, India*

beaulac@ssn.edu.in
**Corresponding author*

Received: 16 October 2023
Accepted: 25 February 2024

Abstract

Encryption and decryption are the two processes in the cryptosystem that ensure the safe transfer of data or sensitive information. Apart from classical mathematics, graph theory techniques are employed nowadays to construct a strong cryptosystem. This paper uses graph techniques such as decomposition and labelling to encrypt and decrypt an alphanumeric string of length 8. The novelty of this paper is the introduction of (S_9, C_3)-multi-decomposition and a new labelling technique - anti-magic decomposed labelling, which is applied in the cryptosystem.

Keywords: encryption; decryption; decomposition of a complete graph; anti-magic decomposed labelling.

1 Introduction

Encryption and decryption play a significant role in different domains to communicate secured data and authenticate financial transactions. Cryptography uses mathematical concepts like number theory, algebra, and, recently, graph theory techniques to construct cryptosystems for sharing confidential information and money transactions. Real-life problems can be modelled as graph theory problems. Graph theory is a collection of a non-empty set of points (vertices) and lines (edges) with incidence relationships. This helps to visualise the complicated problem and arrive at a solution faster than other methods. Graph theory has many applications in science and engineering [14]. A brief survey of the use of graph theory in constructing cryptosystems is presented here.

Etaiwi and Safaa [19] have proposed a symmetric encryption algorithm using graphs and matrices to convert plaintext into cipher text. Tokareva [29] mentioned the applications of different graph structures like sparse, expander, and random graphs in social networking, mobile networking, hash functions, and cryptography. Zemor [35] introduced a cryptographic hash function using Cayley graphs. Charles et al. [11] investigated Pizer and LPS families of optimal expander graphs for provable collision-resistant hash function constructions. Auparajita [22] applied the concept of graph labelling in encryption and decryption. Ustimenko and Romanczuk [31] used the girth of a graph in constructing Turing encryption machines. Tian et al. [28] used graph labelling to generate graphical honeywords.

Wang et al. [32] used odd-elegant labelled graphs to construct a new type of graphical password. Priyadarsini and Ayyagari [26] proposed binary encoding and decoding algorithms using Hadamard matrices of order 16×16 from strongly regular graphs with specific properties. Wardak et al. [33] proposed a cryptography mechanism for secure data transmission and retrieval using a signed graph, its adjacency matrix, and the RSA algorithm. Lavanya and Saravanakumar [23] studied graph-based encryption in a two-factor authentication mechanism to secure a one-time password. Mishra et al. [25] proposed encryption and decryption processes performed over the graphs using the operation defined in the group.

Decomposition is one of the research topics in graph theory and has many applications in science and technology. This paper uses the graph decomposition technique to construct a cryptosystem. A lot of work in graph decomposition has been carried out in the past years by many researchers, a few of which are listed. Truszczy ′ nski [30] proved that $K_{(m,n)}$ has a $P_{(k+1)}$-decomposition if and only if $m \geq \left\lceil \dfrac{(k+1)}{2} \right\rceil$; $n \geq \left\lceil \dfrac{k}{2} \right\rceil$; and $mn = 0 \mod k$, where m and n are even natural numbers with $m \geq n$. Chartrand and Lesniak [12] decomposed an even size nontrivial connected graph G into paths of size 3. Chou et al. [13] decomposed a graph G into p copies of C_4, q copies of C_6, and r copies of C_8 with the condition that $4p+6q+8r = |G|$, where (a) $G = K_{(m,n)}, m, n \geq 4$ are even numbers except $K_{(4,4)}$, (b) G is obtained from $K_{(n,n)}$ with n odd by deleting a perfect match. Laurent et al. [10] decomposed subcubic graphs into paths, claws, and triangles.

Jeevadoss and Muthusamy [21] decomposed the Cartesian product, tensor product, lexicographic of paths, complete graphs, and cycles. Abueida and Daven [1] decomposed the complete graph K_n into complete graphs of size k and stars of size $k+1$, where $n \equiv 0, 1 \mod k$. Alspah [3] proved that for all integers $m \geq 1$, there exists a Hamilton cycle decomposition of the graph $K_{(2m+1)}$ and a Hamilton path decomposition of the graph K_{2m}. Arumugam et al. [4] gave a path decomposition number π, obtained some bounds for π, and characterised graphs attaining the bounds. Bhat and Sudhakara [6] obtained a commuting decomposition of regular complete

k-partite graph $K_{(n_1,n_2,\ldots,n_k)}$ in terms of a Hamiltonian cycle and its k-complement, and also a commuting decomposition of a complete k-partite graph $K_{(n_1,n_2,\ldots,n_k)}$ in terms of a generalized wheel and its k-complement. Beaula et al. [5] decomposed the Turan graph into paths and stars and used these results to construct a block and decryption cryptosystem.

Hasni et al. [18] determined the exact value of edge irregularity strength of disjoint union of cycles and generalised prisms. Yoong et al. [34] proposed the edge irregular reflexive labelling on plane graphs and determined its reflexive edge strength. Gomathi and Venugopal [16] proposed radio antipodal numbers for honeycomb-derived networks. Alan et al. [7] used Rosa-type labelling to decompose complete graphs into unicyclic, disconnected, bipartite graphs on nine edges.

In this paper, a complete graph is considered for the cryptosystem. A complete graph K_n is a graph with n vertices $(n = 1, 2, \ldots)$ in which each vertex is adjacent to every other vertex. A complete graph is an example of a strongly connected network. This paper decomposes the complete graph $K_{3(6n+1)}$ into stars and triangles. We have also proved that the anti-magic decomposed labelling exists for this graph. Then, these results are applied in the construction of the cryptosystem. The novelty of this paper is that it combines two graph techniques- labelling and decomposition- in constructing the cryptosystem, which strengthens it.

The remaining paper is organised as follows. Section 2 discusses graph theory and cryptosystem concepts needed for our research. Section 3 deals with the main results, followed by the conclusion.

2 Preliminaries

This section deals with the basics of graph theory and cryptography. Graph theory definitions are taken from [8, 17]. All the graphs considered in this paper are simple, finite, connected, and undirected. The basics of cryptosystems can be referred from [27].

A graph G is a triplet (V, E, ϕ), 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 $\phi(e) = (u, v), \forall\, e \in E$ and $u, v \in V$. If there is more than one edge between a pair of vertices, these edges are called multiple edges, and if an edge has one vertex as its end vertex, then the edge is called a loop or self-loop. A graph having multiple edges is called a multi-graph. A graph with no multiple edges and loops is called a simple graph. A path is an alternative sequence of vertices and edges in a graph. 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 are subsets of the 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.

Labelling in graph theory is a function of assigning numbers to edges or vertices or both satisfying certain conditions. If the vertices are labelled, it is called vertex labelling; if the edges are labelled, it is called edge labelling [15]. The union of two simple graphs $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ is a simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. It is denoted by $H_1 \cup H_2$. If $\bigcup_{i=1}^{k} H_i = G$, 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 [9]. If G can be partitioned into copies of S and T, with at least one copy of S and one copy of T then the pair (S, T) is said to be a multi-decomposition of G [2]. A graph G with vertex partition V_1 and V_2 such that every edge of G has one end in V_1 and another end in V_2, is called a bipartite graph. A bipartite graph having edges from every vertex of V_1 to every vertex of V_2 is a complete bipartite graph. A complete bipartite graph with one vertex in V_1 and n vertices in V_2 is said to be a star of size n.

Cryptography [27] is the science of communicating confidential information and securing transactions without a third party′s knowledge. Cryptosystem is a part of cryptography with a two-way process called encryption and decryption. Encryption is a process of converting readable script to unreadable script. Decryption is the reverse process of encryption. Cryptography is classified into symmetric key cryptography and public key cryptography. In symmetric key cryptography, encryption and decryption have the same key, whereas in public key, cryptography, encryption and decryption have different keys. 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. This paper uses a stream cipher to authenticate a password of fixed-length 8.

3 Methodology

In this section, the essential graph concepts needed for constructing a cryptosystem are introduced and then applied to the encryption and decryption of an alphanumeric password of length 8.

3.1 Decomposition and edge labelling of $K_{(3(6n+1))}$

In this section, two new terminologies (S_9, C_3)-multi-decomposition and anti-magic decomposed labelling are introduced and studied for a complete graph of size $3(6n+1)$ where,
$n = 1, 2, 3, \ldots$.

Definition 3.1. (S_9, C_3)***-multi-decomposition***
A graph G is said to be (S_9, C_3)-multi-decomposition if it can be decomposed into copies of stars S_9 and triangles C_3, with at least one copy of S_9 and C_3.

Definition 3.2. ***Anti-magic decomposed labelling***
A graph G is said to be anti-magic decomposed labelling if the following conditions are satisfied

(*i*) *G is decomposed into subgraphs H_i in such a way that $\bigcup H_i = G$.*

(*ii*) *All the edges of G are uniquely labelled.*

(*iii*) *The sum of the edge label of each H_i is distinct.*

3.1.1 Condition for a complete graph to have (S_9, C_3) -multi-decomposition

The following theorem provides the condition for a complete graph to have (S_9, C_3)-multi-decomposition.

Theorem 3.1. *A complete graph of size $3(6n+1)$, $n \geq 1$ is (S_9, C_3)-multi-decomposition.*

Proof. Consider a complete graph $K_{(3(6n+1))}$. The number of vertices in $K_{(3(6n+1))}$ is $3(6n+1)$. The number of edges in $K_{3(6n+1)}$ is,

$$\frac{3(6n+1)(3(6n+1)-1)}{2} = 3(6n+1)(9n+1). \tag{1}$$

The complete graph $K_{3(6n+1)}$ is first decomposed into triangles. The remaining subgraph of the complete graph is then decomposed into stars. The vertex set of $K_{(3(6n+1))}$ is partitioned into $6n+1$ partitions of three vertices each. Let these partitions be $V_1, V_2, \ldots, V_{(6n+1)}$. The induced graph on each partition gives triangles $C_1, C_2, \ldots, C_{(6n+1)}$. Removing the edges of $C_1, C_2, \ldots, C_{(6n+1)}$ from $K_{(3(6n+1))}$ results in a complete $(6n+1)-$partite graph with $V_1, V_2, \ldots, V_{(6n+1)}$.

For $1 \leq i \leq 3n+1$, the stars are constructed by taking a vertex from V_i and connecting it with the vertices of each set $\{V_{(i+1)}, V_{(i+2)}, V_{(i+3)}\}, \{V_{(i+4)}, V_{(i+5)}, V_{(i+6)}\}, \ldots, \{V_{(i+3n-2)}, V_{(i+3n-1)}, V_{(i+3n)}\}$ separately.

For $i = 3n+2$, the stars are constructed by taking a vertex from $V_{(3n+2)}$ and joining it with the vertices of each set $\{V_{(i+3)}, V_{(i+4)}, V_{(i+5)}\}, \{V_{(i+6)}, V_{(i+7)}, V_{(i+8)}\}, \ldots, \{V_{(i+3n-2)}, V_{(i+3n-1)}, V_1\}$ separately.

For $i = 3n+3$, the stars are constructed by taking a vertex from $V_{(3n+3)}$ and joining it with the vertices of each set $\{V_{(i+1)}, V_{(i+2)}, V_{(i+3)}\}, \{V_{(i+4)}, V_{(i+5)}, V_{(i+6)}\}, \ldots, \{V_{(i+3n-1)}, V_1, V_2\}$ separately.

Proceeding like this,

For $i = 6n+1$, the stars are constructed by taking a vertex from V_i and joining it with the vertices of each set $\{V_1, V_2, V_3\}, \{V_4, V_5, V_6\}, \ldots, \{V_{(3n-2)}, V_{(3n-1)}, V_{3n}\}$ separately.
Therefore, there are $3n(6n+1)$ copies of stars with 9 edges.

By our construction, the number of edges in

$$\begin{aligned}
(S_9, C_3) - \text{multi-decomposition of } K_{3(6n+1)} = \ & (\text{Number of edges in the triangles}) \\
& + \ (\text{Number of edges in the stars}) \\
&= 3(6n+1) + 9(3n(6n+1)) \\
&= 3(6n+1)(1+9n) \\
&= \ \text{number of edges in } K_{(3(6n+1))} \text{ by } (1).
\end{aligned}$$

Hence, the proof. □

3.1.2 Edge labelling of stars and triangles

By Theorem 3.1, the complete graph $K_{(3(6n+1))}$ is decomposed into $3n(6n+1)$ copies of stars with 9 edges and $(6n+1)$ copies of triangles. Let $S_{(i,j)}$ be a star constructed from $V_i = \{v_{(i,j)}\}$, $1 \leq i \leq (6n+1)$, $1 \leq j \leq 3$ and C_i be the triangle induced by the vertices of V_i. The edges of stars and triangles are labelled by applying the following definition.

Definition 3.3. ***Congruence class of x modulo n* [24]**
Let $x, n \in Z$ with $n > 0$. The congruence class of $x \mod n$, denoted by $[x]_n$, is the set of all integers that are congruent to $x \mod n$; that is, $[x]_n = \{z \in Z \big| x - z = kn \text{ for some } k \in Z\}$.

Using Definition 3.3, the non-negative integers Z^+ are divided into $(6n+2)$ congruence classes. The edges of the complete graph on $3(6n+1)$ vertices are labelled using the congruent modulo $(6n+2)$. For $1 \leq i \leq (6n+1)$, the congruence class $[i]_{(6n+1)} = \{a|a > 0$ and $a = i \mod (6n+2)\}$ is used to label the edges of the stars $S_{(i,j)}$ and the congruence class $[0]_{(6n+1)} = \{b|b > 0$ and $b = 0 \mod (6n+2)\}$ is used to label the edges of the triangles C_i. For convenience, we denote $[i]_{(6n+1)}$ by $[i]$.

Theorem 3.2. *The anti-magic decomposed labelling exists for a complete graph* $K_{3(6n+1)}$.

Proof. Consider a complete graph $K_{3(6n+1)}$. By Theorem 3.1, a complete graph $K_{3(6n+1)}$ can be decomposed into $3n(6n+1)$ copies of stars with 9 edges and $6n+1$ copies of triangles. That is,

$$\bigcup_{i=1}^{6n+1}\left\{\bigcup_{j=1}^{3n} S_{(i,j)} \cup C_i\right\} = K_{3(6n+1)}.$$

Hence, the first condition of the anti-magic decomposed labelling is satisfied.

The edges of the copies of stars and triangles in $K_{(3(6n+1))}$ are labelled by an injective edge labelling $f : E \to \{1, 2, 3, \ldots, 3(6n+1)(9n+1)\}$ such that;

If $e \in S_{(i,j)}$ $1 \leq i \leq (6n+1)$, then $f(e) \in \{i, i+(6n+2), i+2(6n+2), \ldots, i+(3n-1)(6n+2)\}$. Hence, $\Sigma_{(e \in S_{(i,j)} 1 \leq i \leq (6n+1))} f(e) = i + i + (6n+2) + i + 2(6n+2) + \ldots + i + (3n-1)(6n+2)$.

If $e \in C_i, 1 \leq i \leq (6n+1)$, then $f(e) \in \{(3i-2)(6n+2), (3i-1)(6n+2), (3i)(6n+2)\}$. Hence, $\Sigma_{(e \in C_i)} f(e) = (6n+2) + 2(6n+2) + \ldots + (3(6n+1))(6n+2)$.

Hence, all the edges of $K_{3(6n+1)}$ are labelled distinctly.

To prove: The sum of the edge labels for each star $S_{(i,j)}$ and triangle C_i in $K_{3(6n+1)}$ are distinct.

Consider two stars $S_{(a,b)}$ and $S_{(c,d)}$, $a \neq c$, $1 \leq a, c \leq (6n+1)$, and $1 \leq b, d \leq 3$ in $K_{3(6n+1)}$. Suppose $\Sigma_{(e \in S_{(a,b)})} f(e) = \Sigma_{(e \in S_{(c,d)})} f(e)$;

$$\begin{aligned}
\implies a + a + (6n+2) + \ldots + a + (3n-1)(6n+1) &= c + c + (6n+1) + \ldots + c + (3n-1)(6n+1) \\
\implies \frac{(3n-1)}{2}\Big(a + a + (3n-1)(6n+1)\Big) &= \frac{(3n-1)}{2}\Big(c + c + (3n-1)(6n+1)\Big) \\
&\text{(Using Arithmetic progression)} \\
\implies 2b + (3n-1)(6n+1) &= 2c + (3n-1)(6n+1) \\
\implies 2a &= 2c \\
\implies a &= c,
\end{aligned}$$

which is a contradiction.

Now consider triangles C_g and $C_h, g \neq h$ and $1 \leq g, h \leq (6n+1)$ in $K_{(3(6n+1))}$. Suppose $\Sigma_{(e \in C_g)} f(e) = \Sigma_{(e \in C_h)} f(e)$;

$$\begin{aligned}
\implies (3g-2)(6n+2) + (3g-1)(6n+2) + (3g)(6n+2) &= (3h-2)(6n+2) + (3h-1)(6n+2) \\
&\quad + (3h)(6n+2) \\
\implies \frac{(3)}{2}\Big((3g-2)(6n+2) + (3g)(6n+2)\Big) &= \frac{(3)}{2}\Big((3h-2)(6n+2) + (3h)(6n+2)\Big) \\
&\text{(Using Arithmetic Progression)}
\end{aligned}$$

$$\begin{aligned} \implies (6g-2)(6n+2) &= (6h-2)(6n+2) \\ \implies 6g-2 &= 6h-2 \\ \implies \qquad g &= h, \end{aligned}$$

which is a contradiction.

Hence, all the conditions of the anti-magic decomposed labelling are satisfied for $K_{3(6n+1)}$. Therefore, anti-magic decomposed labelling exists for $K_{3(6n+1)}$. □

3.2 Cryptosystem

In this section, the generalized results on the decomposition of a complete graph and edge labelling obtained under Subsection 3.1 are customised and incorporated in constructing a cryptosystem to encrypt and decrypt an alphanumeric string of length 8. For encryption and decryption, a complete graph K_{21} is considered. Its vertex set is partitioned into 7 partitions, namely $V_1, V_2, V_3, V_4, V_5, V_6$, and V_7, each consisting of three vertices. By Theorem 3.1, each of these partitions results in 3 stars of size nine and a triangle.

The proposed cryptosystem can be applied to encrypt and decrypt any alphanumeric string of length 8 in e $(e \geq 1)$ number of times.

3.2.1 Customized labelling of the edges of K_{21} for cryptosystem

The edges of the stars from V_i, $1 \leq i \leq 7$ are labelled with the elements of the congruence class of positive integers $[i]$, where $[i] = \{a | 0 < a \leq i + 29(8) \text{ and } a = i \mod 8)\}$. That is, $[i] = \{i, i+8, i+2(8), i+3(8), \ldots, i+29(8)\}$. The edges of the triangles are labelled with the elements of the congruence class $[0]$, where $[0] = \{b | 0 < b \leq 30(8) \text{ and } b = 0 \mod 8\}$. That is $[0] = \{8, 16, 24, 32, \ldots, 240\}$. We consider only 30 fixed numbers each of the $[i]$ and $[0]$, respectively, to label the stars and triangles of each partition. In each encryption, the numbers of the congruence class are used cyclically for labelling.

(a) Labelling of stars in e^{th} encryption

For every encryption, one of the stars from each partition V_i, $1 \leq i \leq 7$ is considered. The star in the e^{th} encryption $(e \geq 1)$ is $S_{(i,j)}$, where j takes the value 1, 2 or 3 depending upon the remainder l obtained by dividing the encryption order "e" by 3, and is expressed as,

$$A = \begin{cases} l, \quad e = l \mod 3, & \text{for} \quad l \neq 0, \\ 3, \quad e = l \mod 3, & \text{for} \quad l = 0. \end{cases}$$

By applying the techniques discussed in Theorem 3.2, the edges of the stars $S_{(i,j)}$ are labelled with $[i]$. In each encryption, the first edge of $S_{(i,j)}$ is labelled with $i + 8k \in [i]$, where k is the remainder when the encryption order "e" is divided by 30, and is expressed as $e = k \mod 30$.

The succeeding edges of $S_{(i,j)}$ are labelled by the procedure given in the algorithm 3.2.2.

(b) Labelling of triangles in e^{th} encryption

For every encryption, the first triangle C_i is chosen from one of the partitions $V_i, 1 \leq i \leq 7$. The triangle in the e^{th} encryption is C_i, where i takes the value $1, 2, \ldots, 7$ depends upon the remainder h obtained by dividing the encryption order "e" by 7, and is expressed as,

$$A = \begin{cases} h, & e = h \mod 7, \quad \text{for} \quad h \neq 0, \\ 7, & e = h \mod 7, \quad \text{for} \quad h = 0. \end{cases}$$

In each encryption, the first edge is labelled with $8 + 8k \in [0]$, where $e = k \mod 30$, and the succeeding edges are labelled as in the algorithm.

3.2.2 Encryption algorithm

This section proposes an algorithm to encrypt an alphanumeric string of length 8 to produce a ciphertext of size 210.

Input: Plaintext (an alphanumeric string) of length 8.
Output: Ciphertext (Encrypted message) of size 210 obtained from a labelled complete graph of size 21.
Symmetric key: e^{th} time (order of the password).

Step 1: Decompose K_{21} into stars $S_{(i,j)}$ of size 9 and triangles C_i for $1 \leq i \leq 7, 1 \leq j \leq 3$ using Theorem 3.1.

Step 2: Convert the plaintext into a binary string using ASCII [20] code.

Step 3: Apply encryption for every 8 bits of the binary string.

For i^{th}- 8 string $1 \leq i \leq 7$
Label the i^{th} partition star $S_{(i,j)}, 1 \leq j \leq 3$.

(i) Label the first edge of the $S_{(i,j)}$ using the customised labelling for the cryptosystem. The other edges of $S_{(i,j)}$ are labelled based on the corresponding binary sequence using (ii).

(ii) Suppose x is the labelling of an edge; the next edge is labelled as y using the following cases.
Case (a): Suppose the bit in the string is traced as 0. Take $y = x + 8$.
Case (b): Suppose the bit in the string is traced as 1. Take $y = x + 2(8)$.

(iii) The edges of the other two stars $S_{(i,j)}$ are labelled with the numbers not used in the encryption by applying the customised labelling for the cryptosystem, discussed in Subsection 3.2.1.

For 8^{th} - 8 string:
Label three consecutive triangles $C_i, C_{(i+1)}, C_{(i+2)}; 1 \leq i \leq 7$. For $i = 7$, the consecutive triangles used for encrypting the binary string are C_7, C_1, C_2.

(a) Using the customised labelling for the cryptosystem, label the first edge of the triangle C_i, $(1 \leq i \leq 7)$. the other edges of $C_i, C_{(i+1)}$, and $C_{(i+2)}$ are labelled based on the corresponding binary sequence using (b).

(b) Suppose x is a labelling of an edge; the next edge is labelled with y by using the following cases,
Case (i): If the bit in the string to be encrypted is 0. Take $y = x + 8$.
Case (ii): If the bit in the string to be encrypted is 1. Take $y = x + 2(8)$.

(c) Label the remaining triangles C_i with the numbers not used in encryption, with the customised labelling for the cryptosystem.

Step 4: Combine all the labelled stars and triangles to get an edge-labelled complete graph of size 21.

Encrypted message: Edge weight list of the complete graph of size 21.

Encryption architecture

The above encryption algorithm is explained through an architecture given in Figure 1. In this architecture, a password of length eight is encrypted. An ASCII [20] code converts the password with letters and numbers of length 8 into a binary string of length 64. The binary bits are encrypted by a stream cipher. A complete graph of size 21 is used in this encryption. The graph is decomposed into seven stars of size nine and seven triangles.

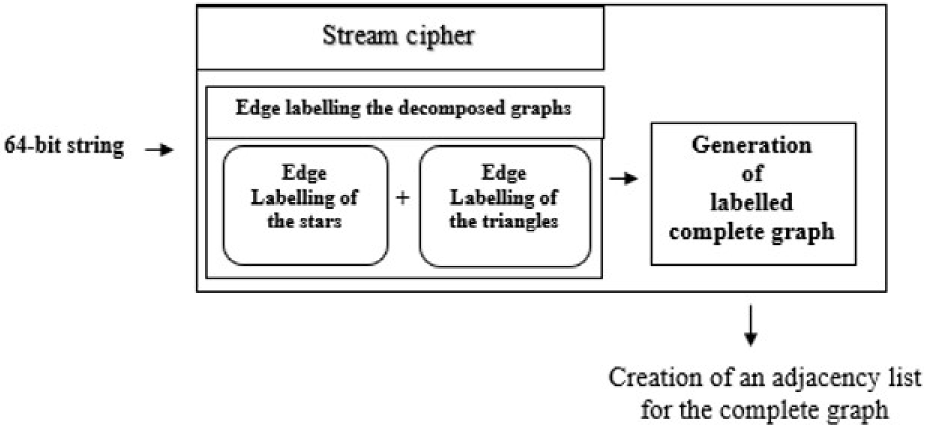

Stream cipher
Edge labelling the decomposed graphs
64-bit string
Edge Labelling of the stars
+
Edge Labelling of the triangles
Generation of labelled complete graph
Creation of an adjacency list for the complete graph

Figure 1: Encryption architecture for password.

Seven sets of 8 bits are labelled in the stars as programmed in the encryption algorithm, and the 8^{th} eight bits are labelled in the triangles as defined in the encryption algorithm. The labelled stars and triangles are combined into a complete graph K_{21}. The edge labels of K_{21} are extracted and listed as an adjacency list, which gives an encrypted message of size 210.

Detailed procedure of the Encryption algorithm

The encryption algorithm is explained in detail, taking the symmetric key $e = 50$.

Input: Password of length 8.
Output: Encrypted message of size 210.
Symmetric key: $e = 50$ $(50^{th}$ password$)$.

Step 1: Consider a complete graph of size 21. The vertex set is partitioned into seven parts, and the graph is decomposed into stars of size 9 and triangles. Three stars from each partition, and the partitions induce the triangles.

Step 2: Convert the plaintext of alphabets and numerals into a binary string using ASCII code.

Step 3: The binary string is divided into eight substrings of length 8. The first seven sets of 8 bits are encrypted by labelling the stars. The 8^{th} eight-bit is encrypted by labelling the triangles.

For i^{th} **-** 8 **string** $(1 \leq i \leq 7)$
$j = 50 \mod 3 = 2 \mod 3$
Label the i^{th} partition star $S_{(i,2)}$

i. The eight-bit strings are encrypted by labelling the second star $S_{(i,2)}$ out of each partition star, as defined in the algorithm.
ii. The stars other than $S_{(i,2)}$ are labelled with the remaining numbers of $[i]$ in each partition.

For 8^{th} **- 8 string:**
$i = 50 \mod 7 = 1 \mod 7$

i. The last eight-bit string is encrypted by labelling the first triangle followed by the next two triangles.
ii. The remaining numbers of $[0]$ are used in labelling the remaining four triangles.

Step 4: Combine all the labelled stars and triangles to get an edge-labelled complete graph of size 21.

Encrypted message: Edge weight list of the complete graph of size 21.

The above illustration gives an encrypted 50^{th} alphanumeric password of length 8. In this process, a complete graph K_{21} of size 21 is used. The vertex set of K_{21} is partitioned into seven partitions of three vertices each. The complete graph is decomposed into 21 stars of size 9 and seven triangles. The stars are taken from each partition vertices separately in encryption. Thus, the seven sets of three stars encrypt the first seven sets of eight-bit binary strings. Edge labelling is incorporated in the encryption of binary bits.

Encryption in stars

Symmetric key $e = 50$.
$j = 50 \mod 3 = 2 \mod 3$.
In the decomposed graph, the second star $S_{(i,2)}$ of all the partitions encrypts the first seven sets of eight-bit binary sequences.

Symmetric key $e = 50$.
$k = 50 \mod 30 = 20 \mod 30$.
The first edge of the selected star is labelled with $i + 8(k) = i + 8(20)$ for each partition.
The binary bits are labelled in the stars $S_{(i,2)}$, $1 \leq i \leq 7$ as follows:

Suppose x is the labelling of an edge; the next edge is labelled as y by using the following cases;
Case (a): Suppose the bit in the string is traced as 0. Take $y = x + 8$.
Case (b): Suppose the bit in the string is traced as 1. Take $y = x + 2(8)$.

The stars $S_{(i,3)}$ and $S_{(i,1)}$, $1 \leq i \leq 7$ are labelled with the numbers of congruence class $[i]$, that are not used for encryption in $S_{(i,2)}$.

Encryption in triangles
Symmetric key $e = 50$.
$i = 50 \mod 7 = 1 \mod 7$.

To encrypt the 8^{th} eight-bit, the labelling starts from the first triangle ($i = 1 \mod 7$) followed by the next two triangles. The binary bits are labelled in the triangles C_1, C_2, and C_3 as follows:

> Suppose x is the labelling of the edge; the next edge is labelled with y by using the following cases;
> Case (i): If the bit in the string to be encrypted is 0. Take $y = x + 8$.
> Case (ii): If the bit in the string to be encrypted is 1. Take $y = x + 2(8)$.

The remaining triangles C_4, C_5, C_6 and C_7 are labelled with congruence class 8 numbers - that are not used for encryption in C_1, C_2 and C_3.

Now that all the decomposed stars and triangles are labelled. These labelled graphs are combined to get a complete graph. The resulting graph is the labelled complete graph K_{21}. The edge weights are extracted and listed as an encrypted message.

3.2.3 Illustration for the encryption algorithm

The encryption algorithm proposed in Subsection 3.2.2 is illustrated by considering an alphanumeric string of length 8.

Input: $23ag3k9o$.
Output: Encrypted message of length 210.
Symmetric key: $e = 421$,

Step 1: Decompose K_{21} into stars $S_{(i,j)}$ of size 9 and triangles C_i, $1 \leq i \leq 7, 1 \leq j \leq 3$ using Theorem 3.1. See Figure 2.

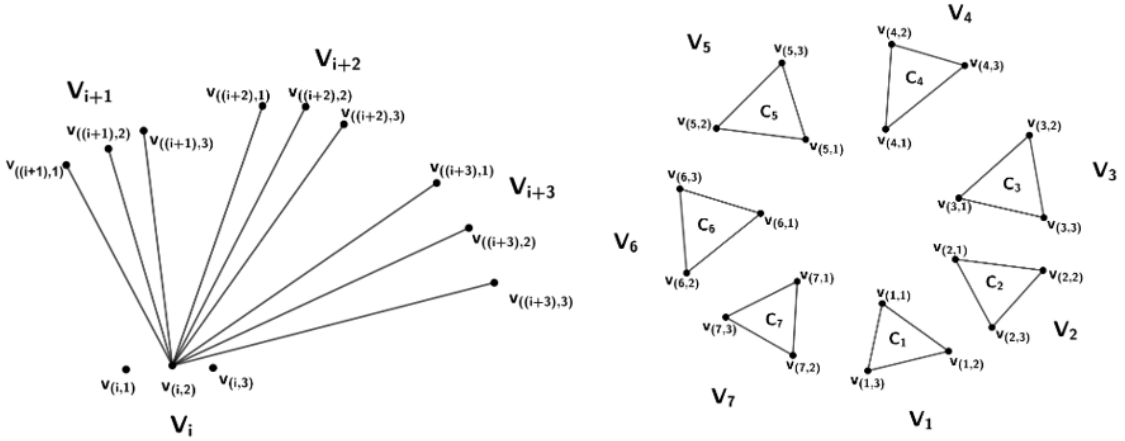


Figure 2: Star $S_{(i,1)}$ and triangles C_i, $(1 \leq i \leq 7)$ decomposed from K_{21}.

Step 2: ASCII code of $23ag3k9o$ is,
00110010 00110011 01100001 01100111 00110011 01101011 00111001 01101111.

Step 3: Encrypt each of eight bits binary strings in stars and triangles. The encrypted graph is given for the first eight-bit and the last eight-bit.

Encryption in stars

Symmetric key $e = 421$.
$j = 421 \mod 3 = 1 \mod 3$.

In the decomposed graph, the first star $S_{(i,1)}$ of all the partitions encrypts the first seven sets of eight-bit binary sequences.
Symmetric key $e = 421$.
$k = 421 \mod 30 = 1 \mod 30$.
The first edge of the selected star is labelled with $i + 8(k) = i + 8(1)$ for each partition. The binary bits are labelled in the stars $S_{(i,1)}$, $1 \leq i \leq 7$ as follows;

(i) **Encryption of first eight-bit** 00110010 **in** $S_{(1,1)}$**.**
The first edge is labelled with 1, and the remaining edges of $S_{(1,1)}$ are labelled depending on the 8-bit. See Table 1. The encrypted graph is given in Figure 3.

Table 1: Encrypting 00110010 in $S_{(1,1)}$.

Bit	Tracing edge	Label for the tracing edge
-	$(v_{(1,1)}, v_{(2,1)})$	1
0	$(v_{(1,1)}, v_{(2,2)})$	9
0	$(v_{(1,1)}, v_{(2,3)})$	17
1	$(v_{(1,1)}, v_{(3,1)})$	33
1	$(v_{(1,1)}, v_{(3,2)})$	49
0	$(v_{(1,1)}, v_{(3,3)})$	57
0	$(v_{(1,1)}, v_{(4,1)})$	65
1	$(v_{(1,1)}, v_{(4,2)})$	81
0	$(v_{(1,1)}, v_{(4,3)})$	89

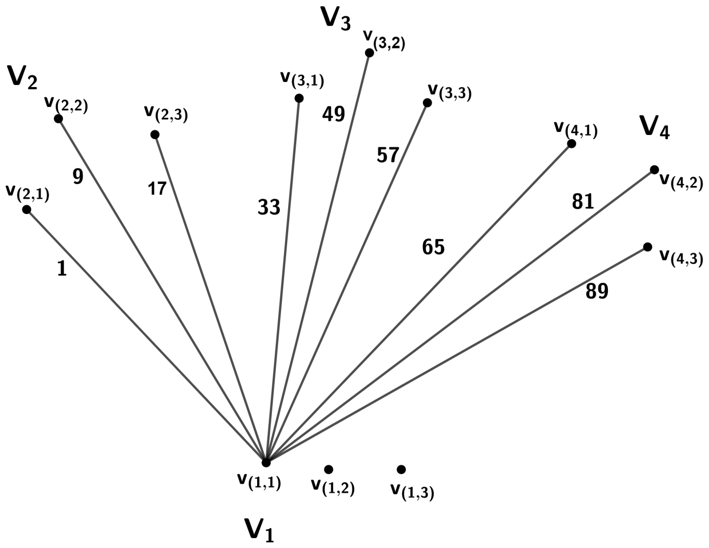


Figure 3: Encryption of 00110010 on star $S_{(1,1)}$.

Labels of the stars $S_{(1,j)}$ $(1 \leq j \leq 3)$ from the first partition vertices are listed in Table 2.

Table 2: Labels of the stars $S_(1, j)$, $(1 \leq j \leq 3)$.

	$v_{(2,1)}$	$v_{(2,2)}$	$v_{(2,3)}$	$v_{(3,1)}$	$v_{(3,2)}$	$v_{(3,3)}$	$v_{(4,1)}$	$v_{(4,2)}$	$v_{(4,3)}$
$v_{(1,1)}$	1	9	17	33	49	57	65	81	89
$v_{(1,2)}$	25	41	73	97	105	113	121	129	137
$v_{(1,3)}$	145	153	161	169	177	185	193	201	209

Similarly, the next 6 eight-bits are encrypted in the stars and are listed in tables as follows;

(ii) **Encryption of the second eight-bit** 00110011 **in** $S_{(2,1)}$.
The first edge is labelled with 2, and the remaining edges of $S_{(2,1)}$ are labelled depending on the 8-bit. See Table 3.

Table 3: Encrypting 00110011 in $S_{(2,1)}$.

Bit	Tracing edge	Label for the tracing edge
-	$(v_{(2,1)}, v_{(3,1)})$	2
0	$(v_{(2,1)}, v_{(3,2)})$	10
0	$(v_{(2,1)}, v_{(3,3)})$	18
1	$(v_{(2,1)}, v_{(4,1)})$	34
1	$(v_{(2,1)}, v_{(4,2)})$	50
0	$(v_{(2,1)}, v_{(4,3)})$	58
0	$(v_{(2,1)}, v_{(5,1)})$	66
1	$(v_{(2,1)}, v_{(5,2)})$	82
1	$(v_{(2,1)}, v_{(5,3)})$	98

(iii) **Encryption of third eight-bit** 01100001 **in** $S_{(3,1)}$.
The first edge is labelled with 3, and the remaining edges of $S_{(3,1)}$ are labelled depending on the 8-bit. See Table 4.

Table 4: Encrypting 01100001 in $S_{(3,1)}$.

Bit	Tracing edge	Label for the tracing edge
-	$(v_{(3,1)}, v_{(4,1)})$	3
0	$(v_{(3,1)}, v_{(4,2)})$	11
1	$(v_{(3,1)}, v_{(4,3)})$	27
1	$(v_{(3,1)}, v_{(5,1)})$	43
0	$(v_{(3,1)}, v_{(5,2)})$	51
0	$(v_{(3,1)}, v_{(5,3)})$	59
0	$(v_{(3,1)}, v_{(6,1)})$	67
0	$(v_{(3,1)}, v_{(6,2)})$	75
1	$(v_{(3,1)}, v_{(6,3)})$	91

(iv) **Encryption of fourth eight-bit** 01100111 **in** $S_{(4,1)}$.
The first edge is labelled with 4, and the remaining edges of $S_{(4,1)}$ are labelled depending on the 8-bit. See Table 5.

Table 5: Encrypting 01100111 in $S_{(4,1)}$.

Bit	Tracing edge	Label for the tracing edge
-	$(v_{(4,1)}, v_{(5,1)})$	4
0	$(v_{(4,1)}, v_{(5,2)})$	12
1	$(v_{(4,1)}, v_{(5,3)})$	28
1	$(v_{(4,1)}, v_{(6,1)})$	44
0	$(v_{(4,1)}, v_{(6,2)})$	52
0	$(v_{(4,1)}, v_{(6,3)})$	60
1	$(v_{(4,1)}, v_{(7,1)})$	76
1	$(v_{(4,1)}, v_{(7,2)})$	92
1	$(v_{(4,1)}, v_{(7,3)})$	108

(v) **Encryption of fifth eight-bit** 00110011 **in** $S_{(5,1)}$.
The first edge is labelled with 5, and the remaining edges of $S_{(5,1)}$ are labelled depending on the 8-bit. See Table 6.

Table 6: Encrypting 00110011 in $S_{(5,1)}$.

Bit	Tracing edge	Label for the tracing edge
-	$(v_{(5,1)}, v_{(6,1)})$	5
0	$(v_{(5,1)}, v_{(6,2)})$	13
0	$(v_{(5,1)}, v_{(6,3)})$	21
1	$(v_{(5,1)}, v_{(7,1)})$	37
1	$(v_{(5,1)}, v_{(7,2)})$	53
0	$(v_{(5,1)}, v_{(7,3)})$	61
0	$(v_{(5,1)}, v_{(1,1)})$	69
1	$(v_{(5,1)}, v_{(1,2)})$	85
1	$(v_{(5,1)}, v_{(1,3)})$	101

(vi) **Encryption of sixth eight-bit** 01101011 **in** $S_{(6,1)}$.
The first edge is labelled with 6, and the remaining edges of $S_{6,1}$. are labelled depending on the 8-bit. See Table 7.

Table 7: Encrypting 01101011 in $S_{(6,1)}$.

Bit	Tracing edge	Label for the tracing edge
-	$(v_{(6,1)}, v_{(7,1)})$	6
0	$(v_{(6,1)}, v_{(7,2)})$	14
1	$(v_{(6,1)}, v_{(7,3)})$	30
1	$(v_{(6,1)}, v_{(1,1)})$	46
0	$(v_{(6,1)}, v_{(1,2)})$	54
1	$(v_{(6,1)}, v_{(1,3)})$	70
0	$(v_{(6,1)}, v_{(2,1)})$	78
1	$(v_{(6,1)}, v_{(2,2)})$	94
1	$(v_{(6,1)}, v_{(2,3)})$	110

(vii) **Encryption of seventh eight-bit** 00111001 **in** $S_{(7,1)}$.
The first edge is labelled with 7, and the remaining edges of $S_{(7,1)}$ are labelled depending on the 8-bit. See Table 8.

Table 8: Encrypting 00111001 in $S_{(7,1)}$.

Bit	Tracing edge	Label for the tracing edge
-	$(v_{(7,1)}, v_{(1,1)})$	7
0	$(v_{(7,1)}, v_{(1,2)})$	15
0	$(v_{(7,1)}, v_{(1,3)})$	23
1	$(v_{(7,1)}, v_{(2,1)})$	39
1	$(v_{(7,1)}, v_{(2,2)})$	55
1	$(v_{(7,1)}, v_{(2,3)})$	71
0	$(v_{(7,1)}, v_{(3,1)})$	79
0	$(v_{(7,1)}, v_{(3,2)})$	87
1	$(v_{(7,1)}, v_{(3,3)})$	103

(viii) **Encryption in triangles**
Symmetric key $e = 421$.
$i = 421 \mod 7$.
$i = 1 \mod 7$.

To encrypt the 8^{th} eight-bit, the labelling starts from the first triangle ($i = 1 \mod 7$) followed by the next two triangles. The binary bits are labelled in the triangles C_1, C_2, and C_3 as follows:

Encrypting the 8^{th} 8-bit 01101111 **in** C_1, C_2**, and** C_3.

The first edge of C_1 is labelled with 8, and the remaining edges of C_1, C_2, and C_3 are labelled depending on the 8-bit. See Table 9. The encrypted graph is given in Figure 4.

Table 9: Encrypting 01101111 in C_1, C_2, and C_3.

Bit	Tracing edge	Label for the tracing edge
-	$(v_{(1,1)}, v_{(1,2)})$	8
0	$(v_{(1,2)}, v_{(1,3)})$	16
1	$(v_{(1,3)}, v_{(1,1)})$	32
1	$(v_{(2,1)}, v_{(2,2)})$	48
0	$(v_{(2,2)}, v_{(2,3)})$	56
1	$(v_{(2,3)}, v_{(2,1)})$	72
1	$(v_{(3,1)}, v_{(3,2)})$	88
1	$(v_{(3,2)}, v_{(3,3)})$	104
1	$(v_{(3,3)}, v_{(3,1)})$	120

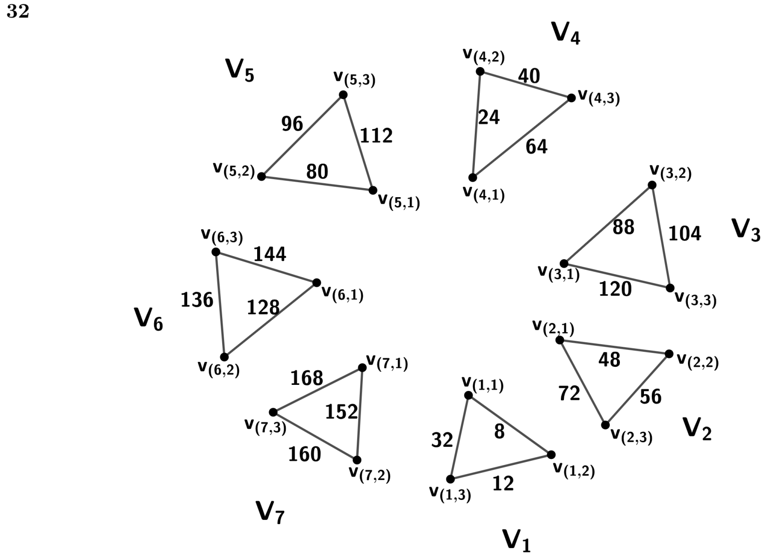
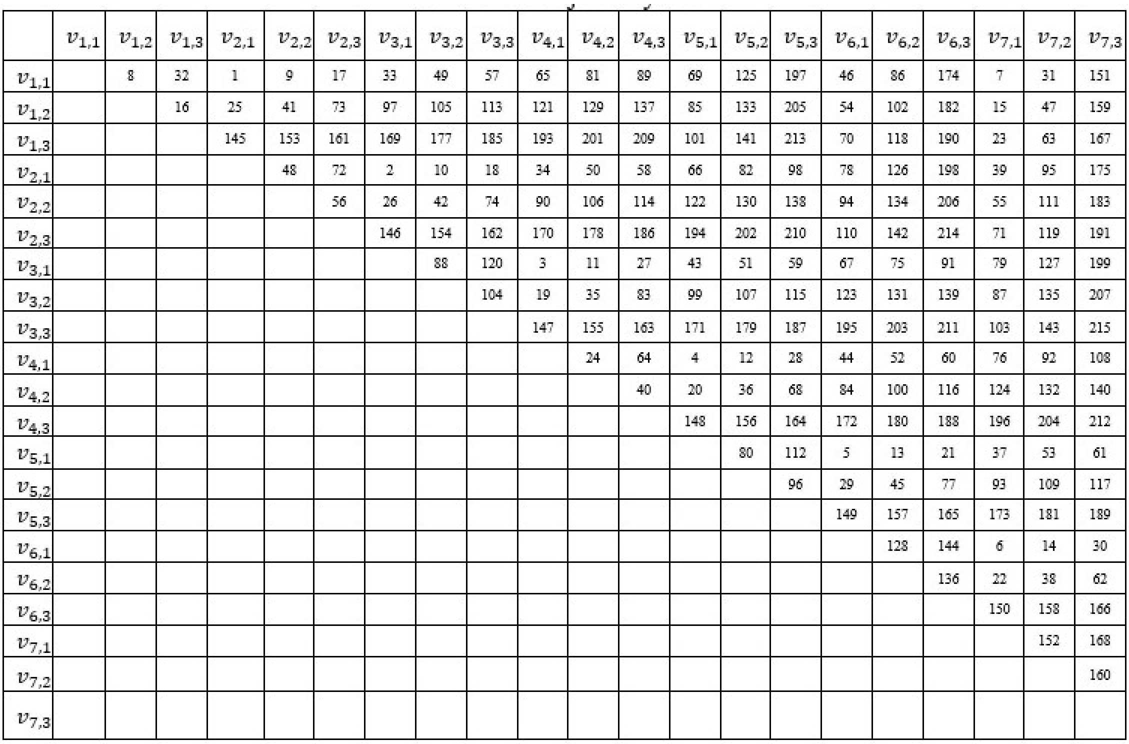


Figure 4: Encryption of 01101111 in $C_i, 1 \leq i \leq 7$.

Encrypted Message: $\{8, 32, 1, 9, 17, 33, 49, 57, 65, 81, 89, 69, 125, 197, 46, 86, 174, 7, 31, 151, 16, 25, 41, 73, 97, 105, 113, 121, 129, 137, 85, 133, 205, 54, 102, 182, 15, 47, 159, 145, 153, 161, 169, 177, 185, 193, 201, 209, 101, 141, 213, 70, 118, 190, 23, 63, 167, 48, 72, 2, 10, 18, 34, 50, 58, 66, 82, 98, 78, 126, 198, 39, 95, 175, 56, 26, 42, 74, 90, 106, 114, 122, 130, 138, 94, 134, 206, 55, 111, 183, 146, 154, 162, 170, 178, 186, 194, 202, 210, 110, 142, 214, 71, 119, 191, 88, 120, 3, 11, 27, 43, 51, 59, 67, 75, 91, 79, 127, 199, 104, 19, 35, 83, 99, 107, 115, 123, 131, 139, 87, 135, 207, 147, 155, 163, 171, 179, 187, 195, 203, 211, 103, 143, 215, 24, 64, 4, 12, 28, 44, 52, 60, 76, 92, 108, 40, 20, 36, 68, 84, 100, 116, 124, 132, 140, 148, 156, 164, 172, 180, 188, 196, 204, 212, 80, 112, 5, 13, 21, 37, 53, 61, 96, 29, 45, 77, 93, 109, 117, 149, 157, 165, 173, 181, 189, 128, 144, 6, 14, 30, 136, 22, 38, 62, 150, 158, 166, 152, 168, 160\}$.

The adjacency list of the encrypted complete graph $K_{3(6n+1)}$ is given in Figure 5.



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

Figure 5: Adjacency list.

The above illustration explains the encryption of an alphanumeric password $23ag3k9o$ using a complete graph on 21 vertices. The order of the plaintext $e = 421$ is taken as the symmetric key.

3.2.4 Decryption algorithm

The following algorithm provides a reverse encryption process. In this process, cipher text of length 210 is decrypted to an alphanumeric string of length 8.

Input: Cipher text of length 210.
Output: Plaintext of length 8 (an alphanumeric string).

Step 1: Use the received cipher texts as the edge weight list of the complete graph of size 21 and construct the graph K_{21}.

Step 2: Decompose the graph into triangles and stars using Theorem 3.1. If e^{th} password is to be decrypted, with the customised labelling for cryptosystem, only the binary sequence traced stars $S_{(i,j)}, 1 < i < 7$, and the three triangles $C_i, C_{(i+1)}, C_{(i+2)}; 1 \leq i \leq 7$, when $i = 7$, the consecutive triangles, C_7, C_1, C_2 are taken into consideration. With the edge labels of these stars and triangles, decrypt the binary sequence using the following steps;

For $i = 1$ **to** 7,
List the edge weights of the star $S_{(i,j)}$.
Case (a): If the difference of edge weights is 8, then take the binary code as 0.
Case (b): If the difference of edge weights is $2(8)$, then take the binary code as 1.

For $i = 8$,
List the edge weights of the three consecutive triangles $C_i, C_{(i+1)}, C_{(i+2)}; 1 \leq i \leq 7$, when $i = 7$, the consecutive triangles are C_7, C_1, C_2.
Case (a): If the difference of edge weights is 8, then take the binary code as 0.
Case (b): If the difference of edge weights is $2(8)$, then take the binary code as 1.

Step 3: Use ASCII code to retrieve the password from the binary sequences.

3.2.5 Illustration for the decryption algorithm

Input: $\{16, 40, 17, 41, 49, 57, 65, 73, 81, 89, 97, 77, 13, 149, 62, 222, 134, 15, 191, 111, 24, 185, 193, 209, 225, 233, 1, 9, 25, 33, 85, 29, 157, 70, 238, 142, 47, 199, 119, 105, 113, 121, 129.137, 145, 153, 161, 169, 93, 45, 165, 78, 6, 150, 55, 207, 127, 32, 56, 18, 34, 50, 58, 66, 74, 82, 90, 98, 86, 22, 158, 63, 223, 135, 48, 186, 194, 202, 218, 234, 2, 10, 26, 42, 94, 30, 166, 71, 239, 143, 106, 114, 122, 130, 138, 146, 154, 162, 170, 102, 46, 174, 79, 7, 151, 64, 80, 11, 43, 51, 59, 67, 75, 83, 91, 99, 87, 23, 159, 72, 187, 195, 203, 219, 235, 3, 19, 27, 35, 95, 31, 167, 107, 115, 123, 131, 139, 147, 155, 163, 171, 103, 39, 175, 88, 104, 4, 28, 44, 60, 68, 76, 84, 92, 100, 96, 188, 196, 212, 228, 236, 12, 20, 36, 52, 108, 116, 124, 132, 140, 148, 156, 164, 172, 112, 128, 21, 37, 45, 53, 61, 69, 120, 189, 197, 213, 229, 237, 5, 101, 109, 117, 125, 133, 141, 192, 208, 14, 38, 54, 200, 190, 198, 206, 110, 118, 126, 224, 8, 240\}$.
Output: Alphanumeric string of length 8.
Symmetric key: $e = 1343$,

Step 1: Use the received cipher texts as the edge weight list of the complete graph of size 21 and construct the graph K_{21}.

Step 2: Decompose the graph into triangles and stars using Theorem 3.1.
$j = e \mod 3$.
$j = 1343 \mod 3 = 2 \mod 3$.
Therefore, the stars selected for encryption are $S_{(i,2)}, 1 \leq i \leq 7$. See Figure 6.

(i) Consider the star $S_{(1,2)}$ from the decomposition for the first eight-bit.

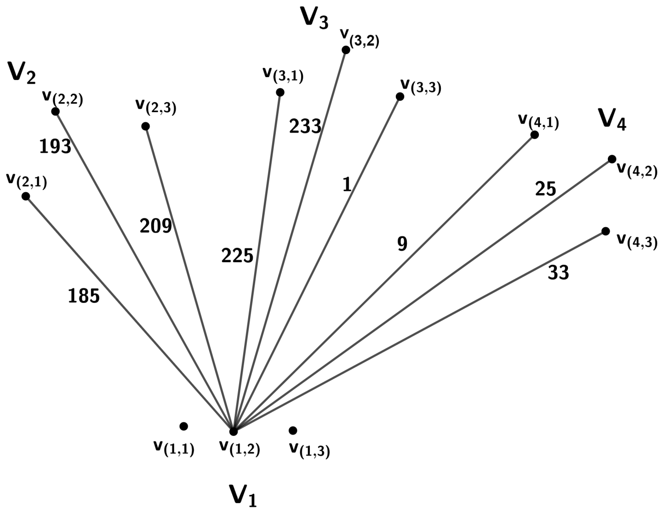


Figure 6: Encrypted star $S_{(1,2)}$ with first eight-bit string.

Table 10: Conversion of edge labels of $S_{(1,2)}$ to binary string.

Adjacent edge labels	Difference	Bits
(185,193)	193-185 = 8	0
(193,209)	209-193 = 16	1
(209,225)	225-209 = 16	1
(225,233)	233-225 = 8	0
(233,1)	240+1-233 = 8	0
(1,9)	9-1=8	0
(9,25)	25-9 = 16	1
(25,33)	33-25 = 8	0

The first eight-bit binary string is 01100010. Calculation shown in Table 10.

(ii) Consider the star $S_{(2,2)}$ from the decomposition for the second eight-bit.

Table 11: Conversion of edge labels of $S_{(2,2)}$ to binary string.

Adjacent edge labels	Difference	Bits
(186, 194)	194-186 = 8	0
(194, 202)	202-194 = 8	0
(202, 218)	218-202 = 16	1
(218, 234)	234-218 = 16	1
(234,2)	240+2-234 = 8	0
(2, 10)	10-2=8	0
(10, 26)	26-10 = 16	1
(26, 42)	42-26 = 16	1

The second eight-bit binary string is 00110011. Calculation shown in Table 11.

(iii) Consider the star $S_{(3,2)}$ from the decomposition for the third eight-bit.

Table 12: Conversion of edge labels of $S_{(3,2)}$ to binary string.

Adjacent edge labels	Difference	Bits
(187, 195)	195-187 = 8	0
(195, 203)	203-195 = 8	0
(203, 219)	219-203 = 16	1
(219, 235)	235-219 = 16	1
(235,3)	240+3-235 = 8	0
(3, 19)	19-3=16	1
(19, 27)	27-19 = 8	0
(27, 35)	35-27 = 8	0

The third eight-bit binary string is 00110100. Calculation shown in Table 12.

(iv) Consider the star $S_{(4,2)}$ from the decomposition for the third eight-bit.

Table 13: Conversion of edge labels of $S_{(4,2)}$ to binary string.

Adjacent edge labels	Difference	Bits
(188, 196)	196-188 = 8	0
(196, 212)	212-196 = 16	1
(212, 228)	228-212 = 16	1
(228, 236)	236-228 = 8	0
(236,12)	240+12-236 = 16	1
(12, 20)	20-12=8	0
(20, 36)	36-20 = 16	1
(36, 52)	52-36 = 16	1

The fourth eight-bit binary string is 01101011. Calculation shown in Table 13.

(v) Consider the star $S_{(5,2)}$ from the decomposition for the fifth eight-bit.

Table 14: Conversion of edge labels of $S_{(5,2)}$ to binary string.

Adjacent edge labels	Difference	Bits
(189, 197)	197-189 = 8	0
(197, 213)	213-197 = 16	1
(213, 229)	229-213 = 16	1
(229, 237)	237-229 = 8	0
(237,5)	240+5-237 = 8	0
(5, 13)	13-5=8	0
(13, 29)	29-13 = 16	1
(29, 45)	45-29 = 16	1

The fifth eight-bit binary string is 01100011. Calculation shown in Table 14.

(vi) Consider the star $S_{(6,2)}$ from the decomposition for the fifth eight-bit.

Table 15: Conversion of edge labels of $S_{(6,2)}$ to binary string.

Adjacent edge labels	Difference	Bits
(190, 198)	190-198 = 8	0
(198, 206)	206-198 = 16	0
(206, 222)	222-206 = 16	1
(222, 238)	238-222 = 16	1
(238,6)	240+6-238 = 8	0
(6, 22)	22-6=16	1
(22, 30)	30-22 = 8	0
(30, 46)	46-30 = 16	1

The sixth eight-bit binary string is 00110101. Calculation shown in Table 15.

(vii) Consider the star $S_{(7,2)}$ from the decomposition for the fifth eight-bit.

Table 16: Conversion of edge labels of $S_{(7,2)}$ to binary string.

Adjacent edge labels	Difference	Bits
(191, 199)	199-191 = 8	0
(199, 207)	207-199 = 8	0
(207, 223)	223-207 = 16	1
(223, 239)	239-223 = 16	1
(239,7)	240+7-239 = 8	0
(7, 23)	23-7=16	1
(23, 31)	31-23 = 8	0
(31, 39)	39-31 = 8	0

The seventh eight-bit binary string is 00110100. Calculation shown in Table 16.

(viii) Consider the triangles from the decomposition for the eighth eight-bit,

32

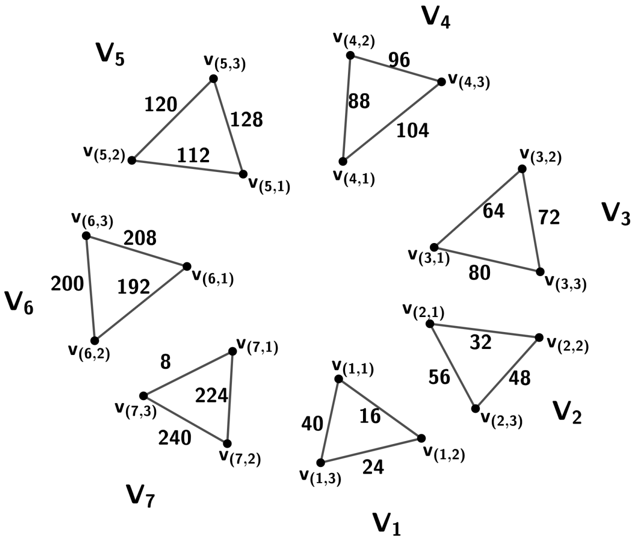


Figure 7: Encrypted triangles

$i = e \mod 7$.
$i = 1343 \mod 7 = 6 \mod 3$.
Therefore, the three triangles used for encryption are C_6, C_7, C_1. See Figure 7.

Table 17: Conversion of edge labels of C_6, C_7, C_1 to binary string.

Adjacent edge labels	Difference	Bits
$(192, 200)$	$200 - 192 = 8$	0
$(200, 208)$	$208 - 200 = 8$	0
$(208, 224)$	$224 - 208 = 16$	1
$(224, 240)$	$240 - 224 = 16$	1
$(240, 8)$	$240 + 8 - 240 = 8$	0
$(8, 16)$	$16 - 8 = 8$	0
$(16, 24)$	$24 - 16 = 8$	0
$(24, 40)$	$40 - 24 = 16$	1

The eighth eight-bit binary string is 00110001. Calculation shown in Table 17.

Therefore, the 64-bit binary string is 01100010 00110011 00110100 01101011 01100011 00110101 00110100 00110001.

Step 3: Use ASCII code to convert the binary string to an alphanumeric string.

Plaintext: $b34kc541$.

In this paper, a complete graph on $3(6n + 1)$ vertices is considered. First, we decomposed this graph into stars and triangles and proved that the anti-magic decomposed labelling exists for this graph. Finally, these techniques are used in the construction of the cryptosystem. This novel approach strengthens and increases the efficiency of the cryptosystem. The different graph structures incorporated in cryptography give the most manipulated cipher texts. This makes it difficult for any analyser to break the cryptosystem. Using numbers in this cryptosystem is elaborated in "customising cryptosystem". It is tough to break the system without knowing the concepts of graph theory.

3.3 Cryptosystem analysis

This section discusses the proposed cryptosystem's performance, security, and comparative analysis.

3.3.1 Performance analysis

To validate the performance of the proposed cryptosystem's encryption algorithm, the execution time was observed for the encryption process by varying input sizes with password lengths 10, 100, 1000, and 10000 over 10 iterations. The average execution time obtained for completing the encryption process is displayed in Table 18.

Table 18: Average execution time for encryption process over ten iterations.

Encryption Algorithm	
Password Length (Character)	Average Runtime (ms)
10	0.01266
100	0.01354
1000	0.01397
5000	0.01563
10000	0.02068

Table 18 shows that the average time to complete the encryption process ranges from 0.01266 to 0.02068 mille seconds for varying passwords between 10 and 10000. It is clear from the table that the average execution time of the encryption algorithm steadily increases as the length of the password increases. Hence, the cryptosystem's performance for higher password lengths takes more time in the proposed cryptosystem, and the variation time is significantly less.

The decryption algorithm inputs the cipher text as an adjacency matrix and produces the password as alphanumeric characters. The system is tested for varying input sizes as an adjacency matrix, and the average execution time is calculated for 10 iterations each, as shown in Table 19.

Table 19: Average execution time for the decryption process over 10 iterations.

Decryption Algorithm	
Adjacency Matrix (Numeric Values)	Average Runtime (ms)
10×10	0.0014
100×100	0.0013
1000×1000	0.0026
5000×5000	0.005
10000×10000	0.0027

Table 19 shows that the average time taken to complete the decryption process is much less than in fractions of milliseconds, making the proposed method more effective and efficient with high performance. As the input matrix is large, there is a slight significant variation in the execution time. However, it becomes tedious for the hacker to break the cryptosystem and determine the password without the actual values, such as numbers.

Hence, the proposed system can be quickly built as it takes significantly less execution time and cannot be broken easily as it involves numbers based on the graphs used. Unless the numbers are known, it is highly impossible to crack the password, making the system more efficient and effective. The efficiency of the cryptosystem is studied by running the encryption and decryption algorithm using C++, which shows nearly 99% accuracy.

3.3.2 Security analysis

The cryptosystem used in this manuscript is compared with the Caesar cipher [27] in Table 20.

Table 20: Comparative analysis of proposed cryptosystem with Caesar cipher.

S. No.	Caesar cipher	Cryptosystem of this paper
1	Encrypts and decrypts only alphabets.	Encrypts and decrypts alphabets and numbers.
2	One level of encryption.	Three levels of encryption.
3	Only one key k is used.	Two keys, 8 and 16, are used.
4	Encryption and decryption are done, without any graph structure.	Encryption and decryption are done using a complete graph to strengthen the system.

From Table 20, it is clear that the proposed cryptosystem is more secure as it involves three levels of encryption. This method is also used to encrypt alphanumeric strings.

4 Conclusion

The cryptosystem for passwords constructed in this paper using anti-magic decomposed labelling is a novel approach. A complete graph of 21 vertices is decomposed into 21 stars and 7 triangles. In the encryption process, the anti-magic decomposed labelling is applied to give a three-level encryption, which is listed as follows. The vertex set of the considered graph is partitioned into seven partitions to encrypt an alphanumeric string of length eight. One star from each partition encrypts one character of the password. If two stars and two sets of three triangles are considered in encryption, then the password length can be increased to 16. Therefore, the same graph can encrypt passwords of sizes varying from 8 to 16. The generalised graph results on decomposition and edge labelling of a graph with $3(6n + 1)$ vertices are discussed in this paper. The same results can be extended for longer passwords or messages of increased length. The proposed cryptosystem is strengthened by using more than one graph theory technique. The technical aspects of this model are still under study. This research can also be extended with different graph structures to construct cryptosystems.

Acknowledgement The authors thank the management of Shiv Nadar Foundation for their continuous support and encouragement for the research activities. We also sincerely thank the reviewers for their valuable comments and suggestions for improving the paper.

Conflicts of Interest No conflicts of interest exist in submitting this manuscript, and all authors approve the manuscript for publication.

References

[1] A. Abueida & M. Daven (2004). Multidecompositions of the complete graph. *Ars Combinatoria*, *72*, 17–22.

[2] A. Abueida & M. Daven (2013). Multi-decompositions of several graph products. *Graphs & Combinatorics*, *29*(3), 315–326. https://doi.org/10.1007/s00373-011-1127-x.

[3] B. Alspach, J. C. Bermond & D. Sotteau (1990). Decomposition into cycles I: Hamilton decompositions. In *Cycles and Rays*, pp. 9–18. Academic Publisher, Dordrecht. https://doi.org/10.1007/978-94-009-0517-7_2.

[4] S. Arumugam, I. S. Hamid & V. M. Abraham (2013). Decomposition of graphs into paths and cycles. *Journal of Discrete Mathematics*, *2013*(1), Article ID: 721051. https://doi.org/10.1155/2013/721051.

[5] C. Beaula, P. Venugopal & B. Praba (2023). Block encryption and decryption of a sentence using decomposition of the turan graph. *Journal of Mathematics*, *2023*(https://doi.org/10.1155/2023/7588535), Article ID: 7588535.

[6] K. A. Bhat & G. Sudhakara (2018). Commuting graphs and their generalised complements. *Malaysian Journal of Mathematical Sciences*, *12*(1), 63–84. https://mjms.upm.edu.my/lihatmakalah.php?kod=2018/January/12/1/63-84.

[7] A. Bohnert, L. Branson & P. Otto (2023). On decompositions of complete graphs into unicyclic disconnected bipartite graphs on nine edges. *Electronic Journal of Graph Theory & Applications*, *11*(1), 329–341. https://doi.org/10.5614/ejgta.2023.11.1.24.

[8] J. A. Bondy & U. S. R. Murty (1976). *Graph Theory with Applications* volume 290. Macmillan Press, London.

[9] J. Bosák (1990). *Decomposition of Graphs*. Mathematics and Its Applications. Kluwer Academic Publishers, Dordrecht.

[10] L. Bulteau, G. Fertin, A. Labarre, R. Rizzi & I. Rusu (2021). Decomposing subcubic graphs into claws, paths or triangles. *Journal of Graph Theory*, *98*(4), 557–588. https://hal.science/hal-03388424.

[11] D. X. Charles, K. E. Lauter & E. Z. Goren (2007). Cryptographic hash functions from expander graphs. *Journal of Cryptology*, *22*(1), 93–113. https://doi.org/10.1007/s00145-007-9002-x.

[12] G. Chartrand, L. Lesniak & P. Zhang (1996). *Graphs and Digraphs* volume 22. Chapman and Hall, London 3rd edition.

[13] C. C. Chou, C. M. Fu & W. C. Huang (1999). Decomposition of $k_{m,n}$ into short cycles. *Discrete Mathematics*, *197–198*, 195–203. https://doi.org/10.1016/S0012-365X(99)90063-8.

[14] N. Deo (1974). *Graph Theory with Applications to Engineering and Computer Science*. Prentice-hall, India.

[15] J. A. Gallian (2023). A dynamic survey of graph labellings. *The Electronic Journal of Combinatorics*, (Dynamic Surveys), DS6–Dec. https://doi.org/10.37236/27.

[16] S. Gomathi & P. Venugopal (2022). Radio antipodal number of honeycomb derived networks. *Scientific Reports*, *12*(1), Article ID: 18993. https://doi.org/10.1038/s41598-022-23618-7.

[17] F. Harary (1988). *Graph Theory*. Narosa Publishing House, India.

[18] R. Hasni, I. Tarawneh, M. K. Siddiqui, A. Raheem, M. A. Asim & A. Mall (2021). Edge irregular k-labeling for disjoint union of cycles and generalized prisms. *Malaysian Journal of Mathematical Sciences*, *15*(1), 79–90. https://mjms.upm.edu.my/lihatmakalah.php?kod=2021/January/15/1/79-90.

[19] S. Hraiz & W. Etaiwi (2017). Symmetric encryption algorithm using graph representation. In *8th International Conference on Information Technology (ICIT)*, pp. 501–506. IEEE,. https://doi.org/10.1109/ICITECH.2017.8080049.

[20] A. B. Injosoft. ASCII table - Table of ASCII codes, characters and symbols — ascii-code.com. https://www.ascii-code.com. [Accessed 06-06-2024].

[21] S. Jeevadoss & A. Muthusamy (2021). Decomposition of product graphs into paths and cycles of length four. *Graphs and Combinatorics*, *32*, 199–223. https://doi.org/10.1007/s00373-015-1564-z.

[22] A. Krishnaa (2019). Inner magic and inner antimagic graphs in cryptography. *Journal of Discrete Mathematical Sciences and Cryptography*, *22*(6), 1057–1066. https://doi.org/10.1080/09720529.2019.1675298.

[23] S. Lavanya & N. M. Saravanakumar (2023). Secured two-factor authentication, graph-based replication and encryption strategy in cloud computing. *Multimedia Tools and Applications*, *82*(11), 16105–16125. https://doi.org/10.1007/s11042-022-13838-4.

[24] B. Maarten (2009). Modular arithmetic before C.F. Gauss: Systematizations and discussions on remainder problems in 18th-century Germany. *Historia Mathematica*, *36*(1), 48–72. https://doi.org/10.1016/j.hm.2008.08.009.

[25] A. K. Mishra, M. S. Obaidat, D. Puthal, A. K. Tripathy & K. K. R. Choo (2018). Graph-based symmetric crypto-system for data confidentiality. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. Abu Dhabi, United Arab Emirates. IEEE. https://doi.org/10.1109/GLOCOM.2018.8647844.

[26] P. L. K. Priyadarsini & R. Ayyagari (2013). Ciphers based on special graphs. In *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 460–465. IEEE. https://doi.org/10.1109/ICACCI.2013.6637215.

[27] W. Stallings (2014). *Cryptography and Network Security*. Pearson, London 6th edition.

[28] Y. Tian, L. Li, H. Peng & Y. Yang (2021). Achieving flatness: Graph labelling can generate graphical honeywords. *Computers & Security*, *104*, Article ID: 102212. https://doi.org/10.1016/j.cose.2021.102212.

[29] N. Tokareva (2014). Connections between graph theory and cryptography. In *Graphs and Groups, Cycles, and Coverings*, pp. 24–26. Novosibirsk, Russia.

[30] M. Truszczyński (1985). Note on the decomposition of λkm, n (λkm, n*) into paths. *Discrete Mathematics*, *55*(1), 89–96. https://doi.org/10.1016/S0012-365X(85)80023-6.

[31] V. Ustimenko & U. Romańczuk (2013). On extremal graph theory, explicit algebraic constructions of extremal graphs and corresponding Turing encryption machines. In *Artificial Intelligence, Evolutionary Computing and Metaheuristics: In the Footsteps of Alan Turing*, volume 427 pp. 257–285. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-29694-911.

[32] H. Wang, J. Xu, M. Ma & H. Zhang (2018). A new type of graphical passwords based on Odd-Elegant labelled graphs. *Security and Communication Networks*, *2018*, Article ID: 9482345. https://doi.org/10.1155/2018/9482345.

[33] O. Wardak, D. Sinha & A. Sethi (2023). Encryption and decryption of signed graph matrices through RSA algorithm. *Indian Journal of Pure and Applied Mathematics*, *2023*, 1–8. https://doi.org/10.1007/s13226-023-00452-9.

[34] K. K. Yoong, R. Hasni, G. C. Lau & M. Irfan (2022). Edge irregular reflexive labeling for some classes of plane graphs. *Malaysian Journal of Mathematical Sciences*, *16*(1). https://doi.org/10.47836/mjms.16.1.03.

[35] G. Zémor (1994). Hash functions and Cayley graphs. *Designs Codes and Cryptography*, *4*(3), 381–394. https://doi.org/10.1007/BF01388652.