Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Education of pure science
Department of Mathematics

The Graph Theory and its Applications in Cryptography

A Research

Submitted to the Council of College of Education for pure Sciences in the University of Babylon in partial Fulfillment of the Requirements for the Degree of Higher Diploma Education / Mathematics

By

Muna Haider Hashim Mohamed Hassan

Supervised by

Asst. Prof. Dr. Ruma Kareem K. Ajeena

2021 A.D. 1442 A. H.

بسم الله الرحمن الرحيم

وَيَرَى الَّذِينَ أُوتُوا الْعِلْمَ
الَّذِي أُنْزِلَ إِلَيْكَ مِنْ رَبِّكَ
هُوَ الْحَقَّ وَيَهْدِي إِلَى
صِرَاطِ الْعَزِيزِ الْحَمِيد

صدق الله العلي العظيم

سورة سبأ / الآية ٦

Supervisor Certificate

I certify that this research entitled ((**The Graph Theory and its Applications in Cryptography))** for the student **Muna Haider Hashim Mohamed HASSAN**, was prepared under my supervision in University of Babylon, 66 College of Education for pure Science as a partial requirement for the Degree of Higher Diploma Education /Mathematic.

Signature :
Name : Dr. Ruma Kareem K. Ajeena
Ttitle : Assistant Professor
Date : / / 2021

In view of available recommendation, I forward this thesis for debate by the examining committee.

Signature :
Name :Dr. Azal Jaafar Mera
Head of mathematics Department
Title : Assistant Professor
Date : / / 2021

Examination Committee Certification

We certify that we have read the thesis entitled the "**The Graph Theory and its Applications in Cryptography''** by **''Muna Haider Hashim Mohamed Hassan''** and as a committee examined the student in its contents and, according to our opinion, it is accepted as a thesis for the Degree of Higher Diploma in Education / Mathematics.

Signature:
Name :Dr. Ahmed AbedAli Omran
Title : Professor
Date: / / 2021
Chairman

Signature:
Name : Mustafa Hasan Hadi
Title : Assistant professor
Date : / /2021
Member

Signature:
Name :Dr.Hayder Kadhim Zghair
Title : Lecturer
Date: / / 2021
Member

Signature:
Name :Dr. Ruma Kareem K. Ajeena
Title : Assistant Professor
Date : / /2021
Member / supervisor

I hereby certify the decision of the examining committee.

Signature:
Name : Dr. Bahaa Hussien Salih Rabee
Title : Professor
Address : Dean of Collage of Education for Pure Sciences
Date: / / 2021

Linguistic Supervisor's Certification

This is to certify that I have read this thesis entitled **'' The Graph Theory and its Applications in Cryptography ''** And I found that this thesis is qualified for debate.

Signature :
Name : Dr. Tefool Hussein Omran
Title : Lecturer
Address : Department of English, College of Education for human Sciences, University of Babylon
Date : / / 2021

Scientific Supervisor's Certtification

This is to certify that I have read this thesis entitled **'' The Graph Theory and its Applications in Cryptography ''** And I found that this thesis is qualified for debate.

Signature :
Name : Dr. Zahir Abdul Haddi Hassan
Title : Professor
Address :
Date : / / 2021

الاهداء

الهي لا يطيب الليل الا بشكرك ....ولا يطيب النهار الا بطاعتك....

ولا تطيب اللحظات الا بذكرك.....ولا تطيب الاخرة الا بعفوك.....

ولا تطيب الجنة الا برؤيتك

((الله ﷻ))

أهدي ثمرة جهدي المتواضع الى من وهبوني الحياة

والامل وعلموني ان ارتقي سلم الحياة بحكمة وصبر

ابي وامي

الى من كاتفني وانا اشق طريق النجاح وملأ حياتي بالتحدي وتخطي الصعاب

الى رفيق دربي

زوجي الغالي

الى ثمرة فؤادي ورياحين حياتي

اولادي

Acknowledgement

I can after the completion of the research, I would like to extend my thanks and gratitude to

Asst. Prof. Dr. Ruma Karim K.Ajeena,

who kindly supervised this research, as she gave me all the bribes and bribes throughout the period of review, and her openness, her interventions and her distinctive style in following up the research had the greatest impact In helping to complete it .

Abstract

New versions of the symmetric encryption schemes are proposed in this work . These versions are used new definition of the Tensor and Strong Product Graph. These new proposed schemes depended on the English alphabet values, ASCII values and the poly alphabetic cipher respectively . The message is chosen as an English word or an English sentence . The cipher text of the original message is considered as the Tensor and Strong Product Graph which is sent to the receiver by sender several experimental results of the proposed encryption schemes are discussed . The security considerations of the proposed Tensor and Strong Product Graph encryption schemes is determined.

Publications

Muna Haider H. (2021) **The Tensor Product Bipartite Graph for Symmetric Encryption Scheme**, AL- Kadhum 2nd International Conference for Modern Applications of Information and Communication Technology, AIP journal, Scopas, 11-2021,(Submitted).

MAICT 2021

المؤتمر الدولي الثاني لكلية الامام الكاظم للتطبيقات الحديثة لتكنولوجيا المعلومات والاتصالات
MODERN APPLICATIONS OF INFORMATION AND COMMUNICATION TECHNOLOGY

Scopus

Date: August-17, 2021

Paper ID: MAICT- 19

Acceptance Notification

Paper Title: **The Tensor Product Bipartite Graph for Symmetric Encryption Scheme**

Authors: ***Muna Haider Hashem and Ruma Kareem K. Ajeena***

Congratulations, the review processes for your paper has been completed. Based on recommendations of our reviewers and Technical Program Committees Apporval, we are pleased to inform you that your paper identified above has been accepted for Possible publication in AIP Conference Proceedings (ISSN: 0094-243X, 1551-7616, SCOPUS, WOS, Indexed). You are cordially invited to present your paper at MAICT to be held in IKC college, Baghdad, Iraq in December 8-9, 2021.

*We ask you to Complete Registration fees which includes publishing Services and complete your copyright transform form to be able to finalize your Final accepted letter. Please visit the conference Website Before the conference date in order to get information for your presentation date.

Registration Plan (Please Choose One) || الرجاء اختيار خطة المشاركة في المؤتمر	
Virtual Attendance Including Publication in AIP || المشاركة الافتراضية شاملة النشر	**150$**
Attendance to Conference Including Publication in AIP || المشاركة الحضورية شاملة النشر	**175$**

Please do not hesitate to contact for any further clarification.

Sincerely Yours

Prof. Dr. Salih M. Al-Qaraawi
Chair of Scientific Committee

A.Prof. Dr. Ahmed J. Obaid
Chair of Secretariate Board

+964 0781 9166 679
maict.conf@gmail.com
www.2021.maict.net

Contents

Abbreviations

Abbreviation	Name
EEA	Extended Euclidean Algorithm
TPG	Tensor Product Graph
TPB	Tensor Product Bipartite
STPG	Security Tensor Product Graph
SPG	Strong Product Graph
SPB	Strong Product Bipartite
Eve	Attake

Chapter One

General Introduction

Chapter One

General Introduction

1.1 Introduction

Cryptography is the science of secret writing with the goal of hiding the meaning of a message . Cryptography has long been the art of spies and soldiers . Nowadays , it is used every day by billions of people for securing electronic mail and payment transactions . The science of cryptography touches on many other disciplines , both within mathematics and computer science and in engineering. In mathematics, cryptology uses, and touches on, algebra, number theory, graph and lattice theory, algebraic geometry and probability and statistics. Analysis of cryptographic security leads to using theoretical computer science especially complexity theory. The actual implementation of cryptosystems, and the hardwork of carrying out security analysis for specific cryptosystems falls into engineering and practical computer science and computing.

In this research we have discussed and proposed the Graph Theory and its Applications in Cryptography by the Tensor and Strong Product Graph.

1.2 Previous Works

In 2008, Tanush Shaska [13], families of simple graphs of high girth had been used for the development of algorithms in Cryptography and Turbocoding They discussed some explicit construction of simple and directed graphs which can be applicable to Turbocoding and Cryptography.

In 2013, S. Saha ray [11], Santanu Saha Ray intends to provide a course text for students in computer science, applied mathematics and operations research. Graph Theory with Algorithms and its Applications could serve as an excellent reference and contains some interesting applications.

In 2014, Wael Etaiwi [16], The proposed algorithm represents a new encryption algorithm to encrypt and decrypt data securely with the benefits of graph theory properties, the new symmetric encryption algorithm use the concepts of cycle graph, complete graph and minimum spanning tree to generate a complex cipher text using a shared key.

In 2015, V.Raja, H.P.Patil*[15], They obtain a necessary and sufficient condition for the tenser product of two or more graphs to be connected, bipartite or eulerian. Also, they present a characterization of the duplicate graph $G \oplus k_2$ to be unicyclic. The girth and the formula for computing the number of triangles in the tensor product of graphs are worked out.

In 2015, P.L.K. Priyadarsini [8], this paper a review of the works carried out in the field of Cryptography which use the concepts of Graph Theory, is given. Some of the Cryptographic Algorithms based on general graph theory concepts, External Graph Theory and Expander Graphs are analyzed. The cryptography is an algorithm which provides secure

communication. In this paper proposed a technique where each character of the data will be encrypted into an Euler Graph.

In 2018, A. C. Shantha Sheela and etc. [1], the need of secure communication of messages is nothing new. It has been present since ages. Security in today's world is one of the important challenges. Ciphers can be converted into graphs for secret communication. The field of Graph Theory plays a vital role in various fields. Especially Graph theory is widely used as a tool of encryption, due to its various properties and its easy representation in computers as a matrix. Various papers based on graph theory applications have been studied and we explore the usage of Graph theory in cryptography has been proposed here.

In 2020, Srilekha Chowdhury and etc, [12], Hamiltonian Circuit is used as key to secure the data. Thus, decryption is practically incomprehensible unless the Hamiltonian circuit and the encoding plan is known. In this cryptography technique, the decryption is very high as each graph represents a character of the message. This algorithm ensures the safety of the data.

In 2021, Baizhu Ni, and etc, [3], this paper proposes some new encryption algorithms for secure transmission of messages using some special bipartite graph along with some algebraic properties. These proposed encryption schemes will lead to more secure communication of secret messages.

1.3 The problem statement of this Research

This work proposes new symmetric encryption schemes. This proposition employed using the tensor and strong product graph to increase the security level of these schemes. The security here is determined based on encrypting the message using the tensor and strong product graph and sending it to the receiver.

1.4 The Structure of This Research

This research consists of five chapters :

Chapter 1 includes the general introduction.

Chapter 2 includes the mathematical background of the graph theory concepts.

Chapter 3 displays the tensor product graph for encryption schemes.

Chapter 4 includes the tensor and strong product graph for polyalphabetic encryption scheme.

Chapter 5 displays the conclusions and future works.

Chapter Two

Mathematical Background to Graph Theory

Chapter Two

Mathematical Background to Graph Theory

2.1 Introduction

This chapter discusses most important Graph Theory concepts that are related to this work in Chapter 3 and Chapter 4. The definitions of the graph and types of graphs are presented with some examples as follows.

2.2 Introduction to Graph Theory

Definition 2.2.1. A graph G = (V, E) consists of two finite sets. A set V is the vertex set of the graph, which is a non-empty set of elements called vertices and a set E is the edge set of the graph, which is a possibly empty set of elements called edges, such that each edge e in E is assigned as an unordered pair of vertices (u ,v).[12]

.For example, a graph G = (V,E) with vertex and edge sets

$$V = \{ 1,2,3,4,5\} \text{ and } E = \{12 , 13 , 24 , 34 , 45\}$$

is shown in Figure (2.1).

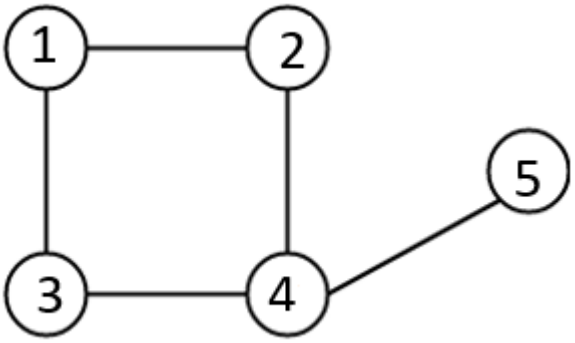


Figure 2.1. A graph G = (V, E).

Definition 2.2.2. Suppose G = (V, E) is a graph with n vertices and m edges. The order of G is |V|=n and the size of G is |E|= m.[12]

Definition 2.2.3. (Self-loop). An edge of a graph that joins a node to itself is called loop or a self-loop. That is, a loop is an edge uv, where u = v.[4]

Definition 2.2.4. (Parallel Edges). The edges connecting the same pair of vertices are called multiple edges or parallel edges.[12]

Definition 2. 2.5. (Simple Graph). A graph *G* which does not have loops or parallel edges is called a simple graph shown in Figure (2.1). A graph, which is not simple, is generally called a multigraph as shown in Figure (2.2).[4]

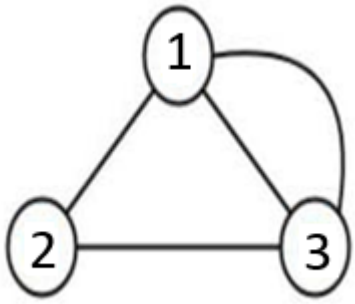


Figure 2.2. A multigraph.

Definition 2.2.6. A graph whose edge set is empty is called as a null graph. In other words, an empty (or trivial) graph is a graph with no edges.[10]

Figure 2.3. A null graph.

Definition 2.2.7. (Adjacent). Two nonparallel edges are said to be adjacent if they are incident on a common vertex. are adjacent. Similarly, two vertices are said to be adjacent if they are the end vertices of the same[10]

Definition 2.2.8. (Degree). Let v be a vertex of the graph G. The degree d (v) of v is the number of edges of G incident with v, counting each self-loop twice. The minimum degree and the maximum degree of a graph G are denoted by $\delta(G)$ andΔ (G), respectively.[12]

. For example, d(v_1)=3=d(v_3)=d(v_4),d(v_2)=4and d(v_5)=1

d(v_1)+d(v_2)+…….+d(v_5)=14= twice the number of edges:

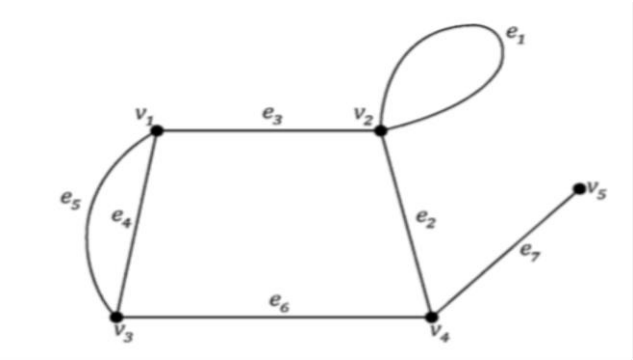


Figure 2.4 shows the degree of graph.

Definition 2.2.9. (Walk). A walk in a graph G is a finite sequence

$$W \equiv v_0 e_1 v_1 e_2 \ldots\ldots v_{k-1} e_k v_k$$

whose terms are alternately vertices and edges such that for $1 \leq i \leq k$; the edge e_i has ends v_{i-1} and v_i. Thus, each edge e_i is immediately preceded and succeeded by the two vertices with which it is incident. We say that W is a $v_0 - v_k$ walk or a walk from v_0 to v_k .[5]

Definition 2.2.10. (Origin and terminus). The vertex v_0 is the origin of the walk W, while v_k is called the terminus of W. v_0 and v_k need not be distinct.

The vertices$v_1, v_2, \ldots, v_{k-1}$ in the above walk W are called its internal vertices. The integer k, the number of edges in the walk, is called the length of W, denoted by |W|.

In a walk W, there may be repetition of vertices and edges.[12]

Definition 2.2.11. (Trivial walk). Awalk of length zero, i.e, with one vertex and no edges.

Thus for any vertex v of G, W ≡ v gives a trivial walk. It has length 0.

In Figure (2.5),

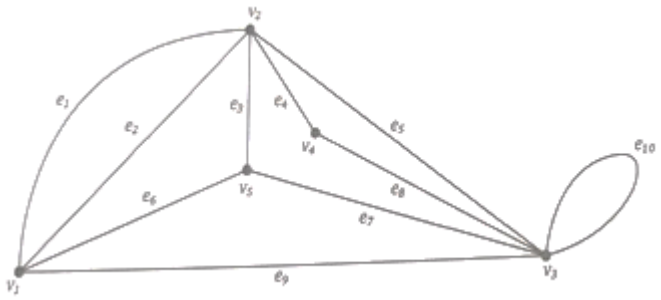


Figure 2.5. walks in a graph G.

$$W_1 = v_1e_1v_2e_5v_3e_{10}v_3e_5v_2e_3v_5 \text{ and } W_2 = v_1e_1v_2e_1v_1e_1v_2$$

are both walks of length 5 and 3, respectively and from v_1 to v_5 and from v_1 to v_2, respectively.

Given two vertices u and v of a graph G, a u–v walk is called closed or open, depending on whether u = v or u ≠ v.

Two walks W_1 and W_2 above are both open, while $W_3 = v_1v_5v_2v_4v_3v_1$ is closed in[4]

Definition 2.2.12. (Trail). If the edges $e_1, e_2, \ldots, e_k$ of the walk

$$W = v_0e_1v_1e_2v_2 \ldots\ldots e_kv_k$$

are distinct then W is called a trail. In other words, a trail is a walk in which no edge is repeated. W_1 and W_2 are not trails, since for example e_5 is repeated in W_1, while e_1 is repeated in W_1. However, W_3 is a trail.[4]

Definition 2.2.13. (Path). If the vertices $v_0, v_1, \ldots, v_k$ of the walk

$$W \equiv v_0 e_1 v_1 e_2 v_2 \ldots\ldots e_k v_k$$

are distinct then *W* is called a path. A path with *n* vertices will sometimes be denoted by P_n. Note that P_n has length *n* - 1.

In other words, a path is a walk in which no vertex is repeated. Thus, in a path no edge can be repeated either, so a every path is a trail. Not every trail is a path, though. For example, W_3 is not a path since v_1 is repeated. However,

$$W_4 = v_2 v_4 v_3 v_5 v_1$$

is a path in the graph G. [4]

Definition 2.2.14. (Connected vertices). A vertex u is said to be connected to a vertex v in a graph G if there is a path in G from u to v.[10]

Definition 2.2.15. (Connected graph). A graph G is called connected if every two of its vertices are connected as shown in Figure (2.6).[10]

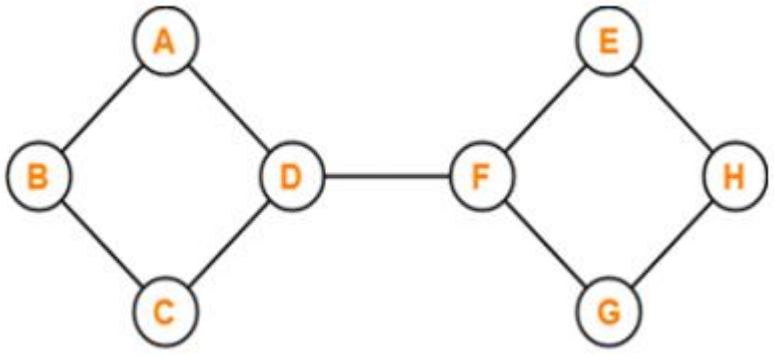


Figure 2.6. A connected graph.

A graph that is not connected is called disconnected as shown in Figure (2.7).

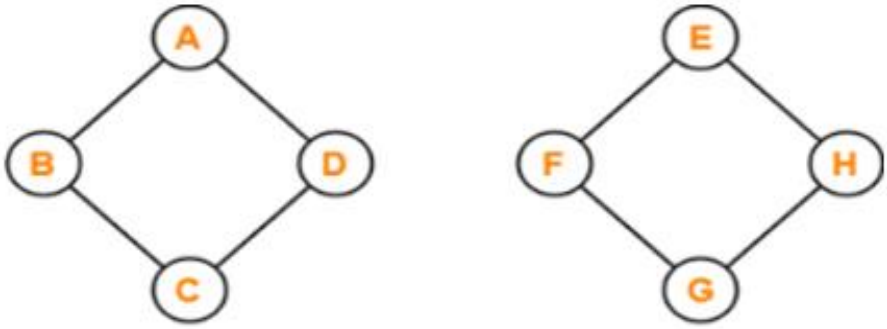


Figure 2.7. A disconnected graph.

Definition 2.2.16. (Bipartite graph). Let G be a graph. If the vertex set V of G can be partitioned into two non-empty subsets X and Y (i.e., X ∪ Y = V and X∩Y = Ø) in such a way that, each edge of G has one end in X and other end in Y, then G is called bipartite. The partition V = X ∪ Y is called a bipartition of G. Figure (2.8) shows an example of Bipartite graph.[5]

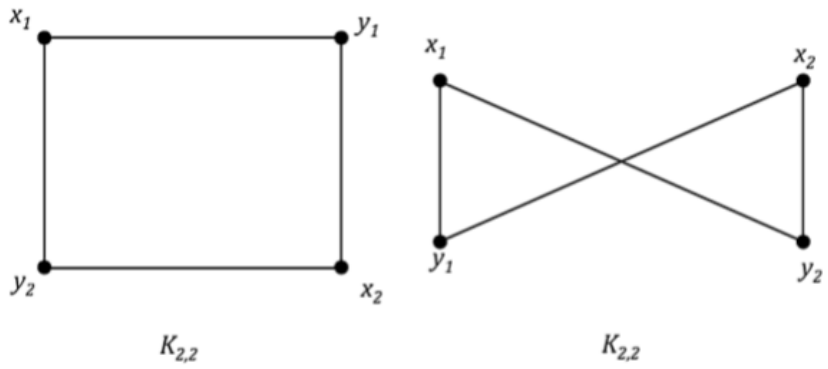


Figure 2.8. Bipartite graph.

2.3 The Diffie-Hellman key exchange

The Diffie- Hellman key exchange algorithm solves the following dilemma.

Alice and Bob want to share a secret key for use in a symmetric cipher, but their only means of communication is insecure. Every piece of information that they exchange is observed by their adversary Eve. The first step is for Alice and Bob to agree on a large prime p and a nonzero integer g modulo p. Alice and Bob make the values of p and public knowledge.

The next step is for Alice to pick a secret integer a that she does not reveal to anyone, while at the same time Bob picks an integer b that he keeps secret. Alice and Bob use their secret integers to compute:

$A \equiv g^a \pmod{p}$ and $B \equiv g^b \pmod{p}$

Respectively. They next exchange these computed values, Alice sends A to Bob and Bob sends Bto Alice. Note that Eve gets to see the values of A and B, since they are sent over the insecure communication channel. Finally, Alice and Bob again use their secret integers to compute:

$A' \equiv B^a \pmod{p}$ and $B' \equiv A^b \pmod{p}$

The values that they compute, A' and B' respectively, are actually the same, since $A' \equiv B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \equiv B' \pmod{p}$

This common value is their exchanged key. [6]

Example 2.3.1 : Let p= 47 be a prime number. The generator element g=3. Alice picks secret a=10, while at the same time Bob picks an integer b=6 that he keeps it as a secret. They use the secret integers to compute

$A \equiv g^a \pmod{p} \equiv 3^{10} \pmod{47} \equiv 17 \pmod{47}$ and

$B \equiv g^b \pmod{p} \equiv 3^6 \pmod{47} \equiv 24 \pmod{47}$

These computations are exchanged between Alice and Bob, namely Alice sends A=17 to Bob and Bob sends B=24 to Alice. Finally, Alice and Bob again use their secret integers to compute

$A' \equiv B^a \pmod{p} \equiv 24^{10} \pmod{47} \equiv 14 \pmod{47}$ and

$B' \equiv A^b \pmod{p} \equiv 17^6 \pmod{47} \equiv 14 \pmod{47}$

The value $A' \equiv B' \equiv 14 \pmod{47}$ is a shared secret key for Alice and Bob.

Chapter Three

The Tensor Product Graph for Symmetric Encryption Scheme

Chapter Three

The Tensor Product Graph for Symmetric Encryption Scheme

3.1 Introduction

In this chapter, the definition of the tensor product graph (TPG) has been presented. The tensor product bipartite graph is used for encryption schemes. Two types of symmetric encryption schemes have been proposed. First one based of the English alphabet values and second one used the ASCII values to represent the letters of the plaintexts. Some examples on these types of the proposed symmetric encryption schemes are discussed. The security considerations of the proposed schemes are determined.

3.2 The Tensor Product Graph

The concept of the tensor product graph (TPG) is defined as follows.

Definition 3.2.1. The tensor product graph $G_1 \times G_2$ of graphs G_1 and G_2 is a graph such that the vertex set of $G_1 \times G_2$ is the Cartesian product graph $V(G_1) \times V(G_2)$ and the vertices (g_1, g_2) and (g_1', g_2') are adjacent in $G_1 \times G_2$ if and only If g_1 is an adjacent to g_1' and g_2 is an adjacent to g_2'.[16]

For instance, if G_1 and G_2 are two graphs as shown in Figure (1), then the tensor product $G_1 \times G_2$ is a bipartite graph of graphs G_1 and G_2.

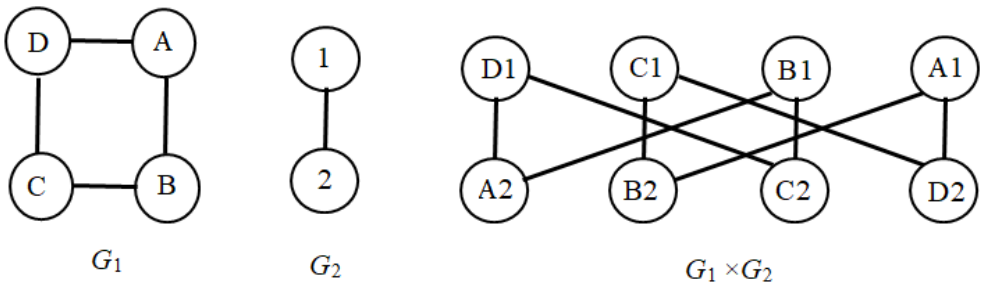


Figure 3.1. The tensor product bipartite (TPB) graph $G_1 \times G_2$.

3.3 The Tensor Product of Bipartite Graph For Encryption Schemes

In this section, some encryption schemes have been proposed based on the tensor product bipartite graphs which are discussed as follows.

3.3.1. The Tensor Product Bipartite Graph for Encryption Schemes: Case I.

- Suppose a plaintext *m* is chosen as an English word or English sentence. This word or sentence consists of some English letters.
- These letters can be converted into numbers using the English alphabet Table (3.1).

A	B	C	D	E	F	G	H	I	G	K	L	M	**N**
1	2	3	4	5	6	7	8	9	10	11	12	13	14
O	P	Q	R	S	T	U	V	W	X	Y	Z		
15	16	17	18	19	20	21	22	23	24	25	26		

Table 3.1. English alphabet Table.

In other words,

$$m = \{m_1, m_2, \ldots, m_k\}.$$

where p is the near prime number greater than 26.

- The numbers $\#m_i$ that are corresponded to m_i, for $i = 1, 2, \ldots, k$. The length of the message is k.
- The first user chooses p, where p is the near prime number greater than 26 (26 is the number of the English alphabet letters) and p is a shared secret key that is computed by the Diffie – Hellman key exchange as explained in section (2-3).
- The computations of the inverse elements of the numbers $\#m_i$ have been done using the extended Euclidean algorithm (EEA) to get

$$(\#m_i)^{-1} \pmod{p} \equiv n_i, \text{ for } i = 1, 2, \ldots, k.$$

- The first user chooses the primes p_i such that the number of the primes $\# p_i$ is less than k-2, namely $p_1, p_2, \ldots, p_{k-3}$. Now, two graphs are formed based on the inverse elements and prime numbers.

- The ciphertext C of a message m is constructed as the tensor product bipartite graph of graphs G_1 and G_2. The ciphertext C of m is a tensor product bipartite (TPB) graph which is sent to second user.
- Upon second user receives the TPB graph, he / she will do the following steps:

  He / She takes one set of the vertices of the independent set of TPB graph

  List 1: $n_1p_1, n_2p_1, \ldots, n_lp_l$ or List 2: $n_{l+1}p_{l+1}, n_{l+2}p_{l+1}, \ldots, n_kp_{k-3}$.

- If he / she chooses first list, then the second user uses his/ her shared secret key p to compute the inverse elements of list 1 as follows.

  n_ip_i $n_i \rightarrow n_i^{-1} \pmod{p} \equiv s_i$ (which maybe equal to correct $\#m_i$ or not)

  $p_i \rightarrow p_i^{-1} \pmod{p} \equiv s_j$ (which maybe equal to correct $\#m_j$ or not).

- Since the length of the message k that are can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct message consist of k vertices to give us word or sentence with correct meaning.

Example 3.3.1.1. Study Case I: Encryption Scheme Based on the TPB graph

Suppose *m* is a message that is given by an English word **Graph**. Based on the alphabet Table (3.1), the letter of this word are converted into numbers as follows

$$G\rightarrow 7,\ R\rightarrow 18,\ A\rightarrow 1,\ P\rightarrow 16,\ H\rightarrow 8.$$

Now, the length of the message is $k = 5$. The first user chooses p, where p is the near prime number greater than 26 (26 is the number of the English alphabet letters) and p is a shared secret key that is computed by the Diffie – Hellman key exchange. Let $p = 31$. The computations of the inverse elements of the numbers 7, 18, 1, 16, and 8 modulo 31 have been done using the extended Euclidean algorithm (EEA) to get

$$7^{-1} \pmod{31} \equiv 9$$

$$18^{-1} \pmod{31} \equiv 19$$

$$1^{-1} \pmod{31} \equiv 1$$

$$16^{-1} \pmod{31} \equiv 2$$

$$8^{-1} \pmod{31} \equiv 4$$

After that, the first user chooses the primes p_i such that the number of the primes $\# p_i$ is less than k-2 , say p_1=5, p_2=7. Now, two graphs are formed based on the inverse elements and prime numbers as shown in Figure (3.1).

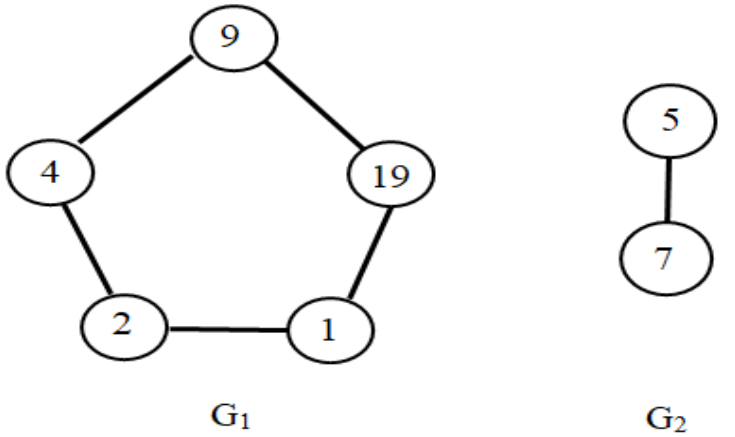


Figure 3.2. The graphs G_1 and G_2 with 5 and 2 vertices respectively.

The ciphertext *C* of a message *m* is constructed as the tensor product bipartite (TPB) graph of graphs G_1 and G_2 as shown in Figure (3.2), which is sent to second user.

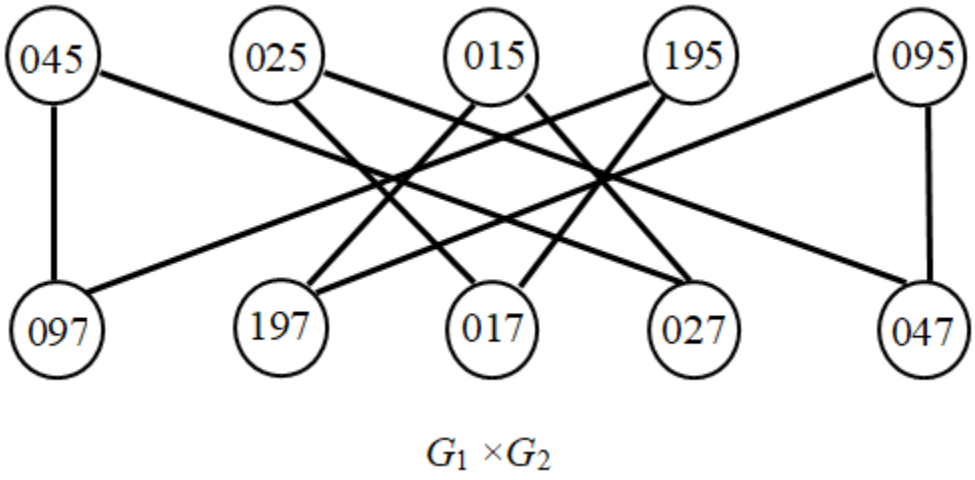


Figure 3.3. The TPB graph $G_1 \times G_2$ of graphs G_1 and G_2 that are shown in Figure (3.2).

Upon second user receives the TPB graph, he / she will do the following steps:

He / She takes one set of the vertices of the independent set of TPB graph

List 1: 095, 195, 015, 025, 045 or List 2: 047, 027, 017, 197, 097.

If he / she chooses first list, then the second user uses his/ her shared secret key p = 31 to compute the inverse elements of list 1 as follows.

95 $9 \rightarrow 9^{-1} \pmod{31} \equiv 7$

$5 \rightarrow 5^{-1} \pmod{31} \equiv 25$

195 $19 \rightarrow 19^{-1} \pmod{31} \equiv 18$

$5 \rightarrow 5^{-1} \pmod{31} \equiv 25$

15 $1 \rightarrow 1 \pmod{31} \equiv 1$

$5 \rightarrow 5^{-1} \pmod{31} \equiv 25$

25 $2 \rightarrow 2^{-1} \pmod{31} \equiv 16$

$5 \rightarrow 5^{-1} \pmod{31} \equiv 25$

45 $4 \rightarrow 4^{-1} \pmod{31} \equiv 8$

$5 \rightarrow 5^{-1} \pmod{31} \equiv 25$

Since the length of the message $k = 5$ that are can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct message consist of 5 vertices

$$7 \rightarrow 25 \rightarrow 18 \rightarrow 1 \rightarrow 16 \Rightarrow \text{GYRAP}$$

$$7 \rightarrow 25 \rightarrow 18 \rightarrow 1 \rightarrow 8 \Rightarrow \text{GYRAH}$$

$$\vdots$$

$$\mathbf{7 \rightarrow 18 \rightarrow 1 \rightarrow 16 \rightarrow 8 \Rightarrow GRAPH.}$$

The correct path is $7 \rightarrow 18 \rightarrow 1 \rightarrow 16 \rightarrow 8$ which gives the correct original plaintext **Graph**.

Example 3.3.1.2. Study Case I: Encryption Scheme Based on the TPB graph

Suppose *m* is a message that is given by an English word **March**. Based on the alphabet Table (3.1), the letter of this word are converted into numbers as follows

$$\text{M} \rightarrow 13,\ \text{a} \rightarrow 1,\ \text{r} \rightarrow 18,\ \text{C} \rightarrow 3,\ \text{h} \rightarrow 8.$$

Now, the length of the message is $k = 5$. The first user chooses *p*, where *p* is the near prime number greater than 26 (26 is the number of the English alphabet letters) and *p* is a shared secret key that is computed by the Diffie – Hellman key exchange. Let $p = 29$. The computations of the inverse elements of the numbers 13,

1, 18, 3, and 8 modulo 29 have been done using the extended Euclidean algorithm (EEA) to get

$$13^{-1} \pmod{29} \equiv 9$$

$$1^{-1} \pmod{29} \equiv 1$$

$$18^{-1} \pmod{29} \equiv 21$$

$$3^{-1} \pmod{29} \equiv 10$$

$$8^{-1} \pmod{29} \equiv 11$$

After that, the first user chooses the primes p_i such that the number of the primes # p_i is less than k-2 , say p_1=3, p_2=5. Now, two graphs are formed based on the inverse elements and prime numbers as shown in Figure (3.4).

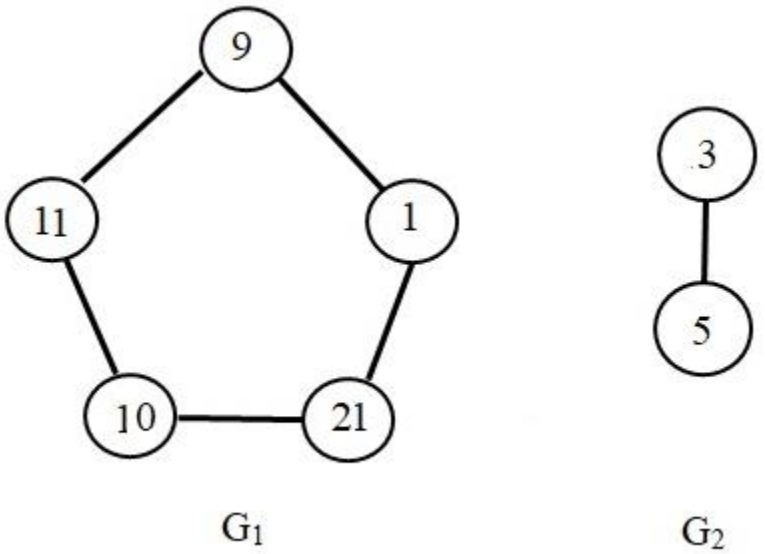


Figure 3.4. The graphs G_1 and G_2 with 5 and 2 vertices respectively.

The ciphertext *C* of a message *m* is constructed as the tensor product bipartite (TPB) graph of graphs G_1 and G_2 as shown in Figure (3.5), which is sent to second user.

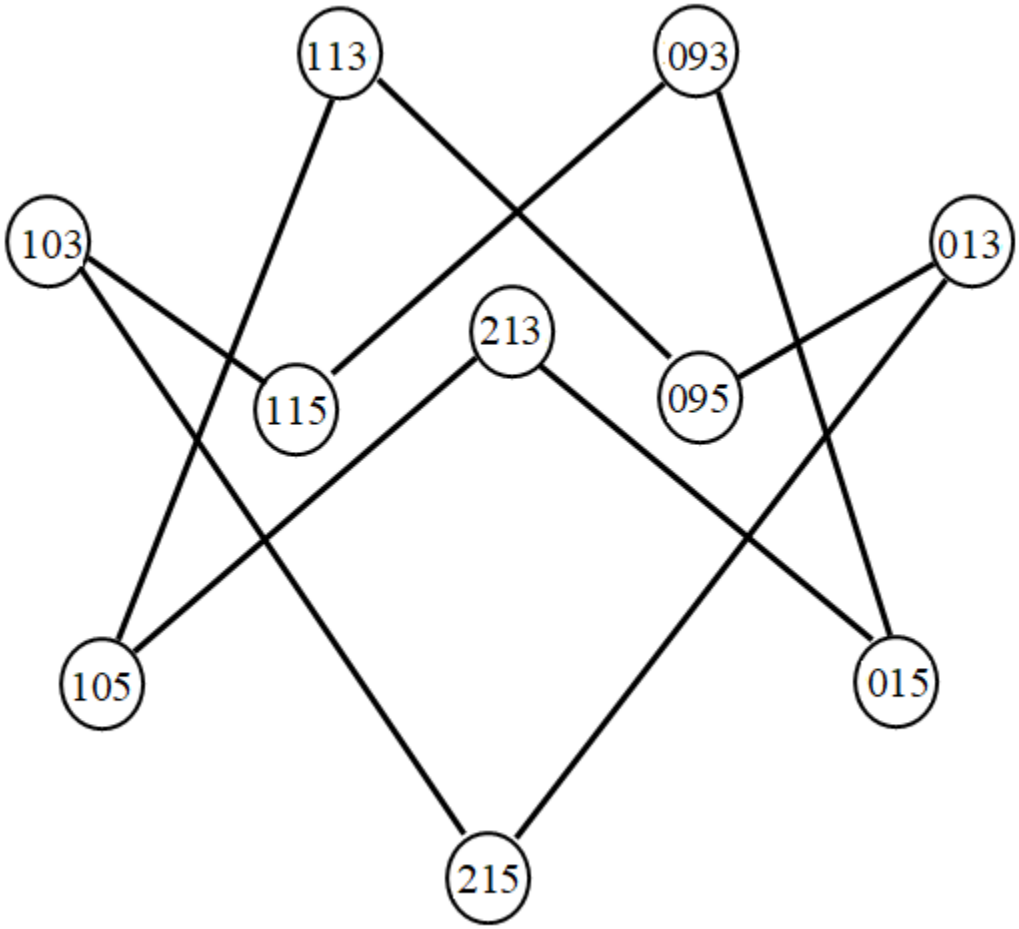


Figure 3.5. The TPB graph $G_1 \times G_2$ of graphs G_1 and G_2 that are shown in Figure (3.4).

Upon second user receives the TPB graph, he / she will do the following steps:

He / She takes one set of the vertices of the independent set of TPB graph

List 1: 093, 013, 213, 103, 113 or List 2: 095, 015, 215, 105, 115.

If he / she chooses second list, then the second user uses his/ her shared secret key $p = 29$ to compute the inverse elements of list 2 as follows.

095 $9 \rightarrow 9^{-1} \pmod{29} \equiv 13$

$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$

015 $1 \rightarrow 1^{-1} \pmod{29} \equiv 1$

$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$

215 $21 \rightarrow 21^{-1} \pmod{29} \equiv 18$

$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$

105 $10 \rightarrow 10^{-1} \pmod{29} \equiv 3$

$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$

115 $11 \rightarrow 11^{-1} \pmod{29} \equiv 8$

$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$

Since the length of the message $k = 5$ that are can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct message consist of 5 vertices

$$13\rightarrow 6 \rightarrow 1 \rightarrow 18 \rightarrow 3 \Rightarrow \text{Mfarc}$$

$$13\rightarrow 6 \rightarrow 1 \rightarrow 18 \rightarrow 8 \Rightarrow \text{Mfarh}$$

$$13\rightarrow 6 \rightarrow 1 \rightarrow 3 \rightarrow 8 \Rightarrow \text{Mfach}$$

$$\vdots$$

$$13\rightarrow 1 \rightarrow 18 \rightarrow 3 \rightarrow 8 \Rightarrow \text{March.}$$

The correct path is $13\rightarrow 1 \rightarrow 18 \rightarrow 3 \rightarrow 8$ which gives the correct original plaintext **March.**

3.4 The Tensor product Graph for encryption schemes: Case II.

The same idea of case (I) can be applied to encrypt the plaintext *m* has the English letters that are represented by numbers of ASCII Table (2.3). The possibility here to choose a plaintext as an English word or an English sentence consists of some words is more than 26 letters. The number of the allowed letters that can be chosen is 127.

Table 3.2. ASCII Table

Char	ASCII Code (Decimal)
a	97
b	98
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107
l	108
m	109
n	110
o	111
p	112
q	113
r	114
s	115
t	116
u	117
v	118
w	119
x	120
y	121
z	122

Char	ASCII Code (Decimal)
A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

Char	ASCII Code (Decimal)
space	32
!	33
"	34
#	35
$	36
%	37
&	38
'	39
(	40
)	41
*	42
+	43
,	44
-	45
.	46
/	47
:	58
;	59
<	60
=	61
>	62
?	63
@	64
[	91
\	92
]	93
^	94
_	95
`	96
{	123
|	124
}	125
~	126
'	145
'	146
"	147
"	148
•	149
-	152

Char	ASCII Code (Decimal)
0	48
1	49
2	50
3	51
4	52
5	53
6	54
7	55
8	56
9	57

Char	ASCII Code (Decimal)
€	128
£	163
¥	165
$	36
©	169
™	153
°	176
˜	152
¡	161
¿	191

So, the letters of the plaintext here have been ASCII Table numbers.

- Suppose

$$m = \{m_1, m_2, \ldots, m_k\}.$$

The numbers $\#m_i$ that are corresponded to m_i, for $i = 1, 2, \ldots, k$. The length of the message is k.

- The first user chooses p, where p is the near prime number greater than 127 (127 is the number of the English alphabet letters in ASCII Table) and p is a shared secret key that is computed by the Diffie – Hellman key exchange.
- The ciphertext C of a message m is computed in similar way as computed in Case (I) as the tensor product bipartite graph of graphs G_1 and G_2.
- Upon second user receives the TPB graph, he / she will do the following steps:

He / She takes one set of the vertices of the independent sets of TPB graph. If he/she chooses first list, then the second user uses his/ her shared secret key p to compute the inverse elements of list 1. Since the length of the message k that are can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct message consist of k vertices to give us word or sentence with correct meaning.

Example 3.4.1.1. (Study Case II: Encryption Scheme Based on the TPB Graph)

Suppose *m* is a plaintext that is given by an English word **Iraqi**. Based on the ASCII Table (3.2). The letters of this word are converted into numbers as follows.

$$I\rightarrow 73,\ r\rightarrow 114,\ a\rightarrow 97,\ q\rightarrow 113,\ i\rightarrow 105.$$

Now, the length of the plaintext is K =5. The first user chooses a prime number *p*, where *p* is the near prime number greater than 127 and *p* is a shared secret key. Let $p = 131$. The computations of the inverse elements of the numbers 73, 114, 97, 113, and 105 modulo 131 have been done using the extended Euclidean algorithm (EEA) to get

$$73^{-1} \pmod{131} \equiv 70$$

$$114^{-1} \pmod{131} \equiv 77$$

$$97^{-1} \pmod{131} \equiv 104$$

$$113^{-1} \pmod{131} \equiv 80$$

$$105^{-1} \pmod{131} \equiv 5$$

After that, the first user chooses the primes p_i such that the number of the primes # p_i is less than *k*-2 , say $p_1=3$, $p_2=5$. Now, two graphs are formed based on the inverse elements and prime numbers as shown in Figure (3.6).

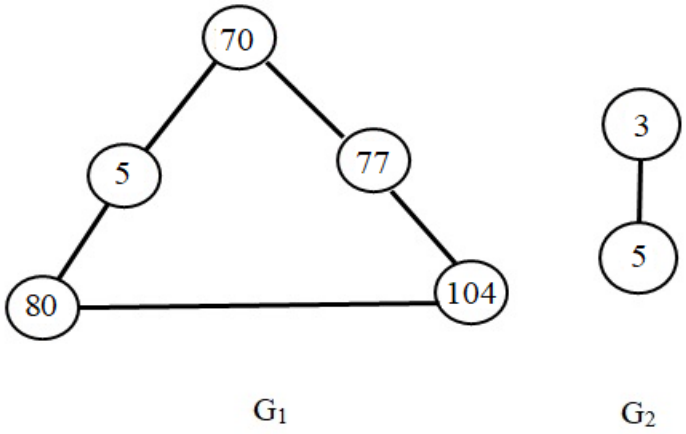
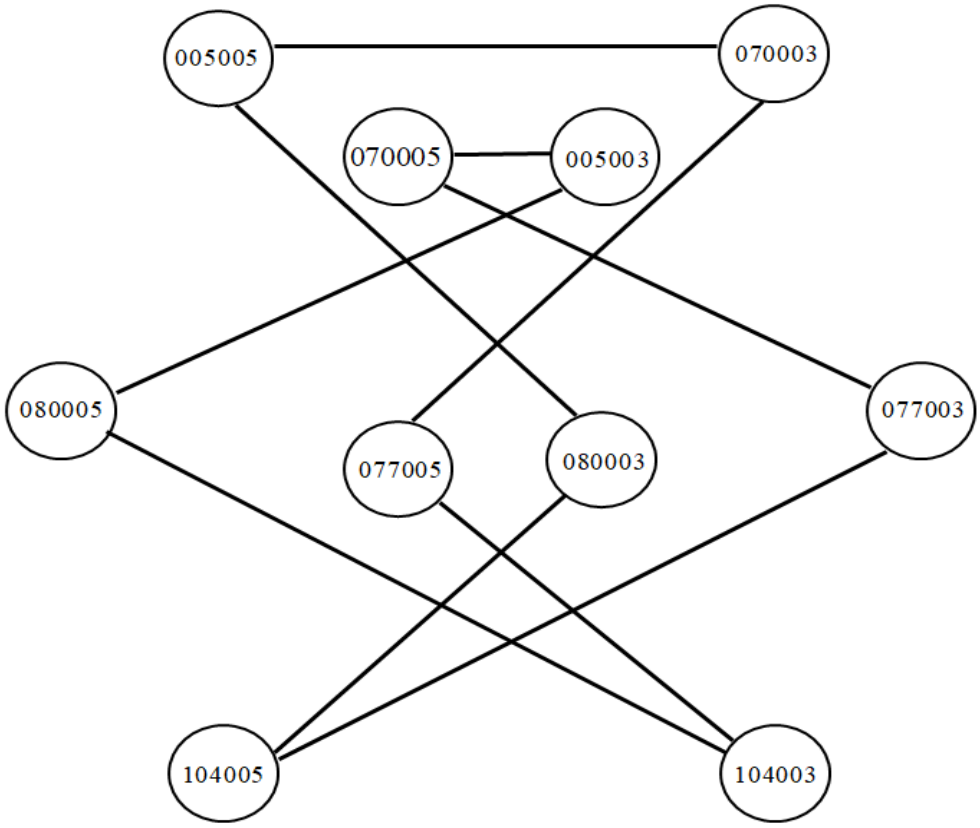


Figure 3.6. The cycle graph G_1 and path graph G_2 with 5 and 2 vertices respectively.

The ciphertext *C* of a message *m* is constructed as the tensor product bipartite (TPB) graph of graphs G_1 and G_2 by

$G_1 \times G_2$ = {(70,3), (77,3), (104,3), (80,3), (5,3), (70,5), (77,5), (104,5), (80,5), (5,5)}.

The TPB graph is shown in Figure (3.7) and which is sent to second user as a ciphertext.



$G_1 \times G_2$

Figure 3.7. The TPB graph $G_1 \times G_2$ of graphs G_1 and G_2 that are shown in Figure (3.6).

Upon second user receives the TPB graph, he / she will do the following steps:

He / She takes one set of the vertices of the independent set of TPB graph

List 1: 070003, 077003, 104003, 080003, 005003 or

List 2: 005005, 080005, 104005, 077005, 070005.

If he / she chooses first list, then the second user uses his/ her shared secret key p = 131 to compute the inverse elements of list 1 as follows.

703 $70 \rightarrow 70^{-1} \pmod{131} \equiv 73$

$3 \rightarrow 3^{-1} \pmod{131} \equiv 44$

773 $77 \rightarrow 77^{-1} \pmod{131} \equiv 114$

$3 \rightarrow 3^{-1} \pmod{131} \equiv 44$

1043 $104 \rightarrow 104^{-1} \pmod{131} \equiv 97$

$3 \rightarrow 3^{-1} \pmod{131} \equiv 44$

803 $80 \rightarrow 80^{-1} \pmod{131} \equiv 113$

$3 \rightarrow 3^{-1} \pmod{131} \equiv 44$

53 $5 \rightarrow 5^{-1} \pmod{131} \equiv 105$

$3 \rightarrow 3^{-1} \pmod{131} \equiv 44$

Since the length of the message k = 5 that are can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct message consist of 5 vertices

$$73 \rightarrow 44 \rightarrow 114 \rightarrow 44 \rightarrow 97 \Rightarrow \text{I,r,a}$$

$$73 \rightarrow 44 \rightarrow 114 \rightarrow 97 \rightarrow 44 \Rightarrow \text{I,ra,}$$

$$73 \rightarrow 44 \rightarrow 114 \rightarrow 97 \rightarrow 113 \Rightarrow \text{I,raq}$$

$$73 \rightarrow 44 \rightarrow 114 \rightarrow 97 \rightarrow 105 \Rightarrow \text{I,rai}$$

$$73 \rightarrow 44 \rightarrow 114 \rightarrow 113 \rightarrow 97 \Rightarrow \text{I,rqa}$$

$$73 \rightarrow 44 \rightarrow 114 \rightarrow 105 \rightarrow 97 \Rightarrow \text{I,ria}$$

$$73 \rightarrow 44 \rightarrow 114 \rightarrow 113 \rightarrow 105 \Rightarrow \text{I,rqi}$$

$$73 \rightarrow 44 \rightarrow 114 \rightarrow 105 \rightarrow 113 \Rightarrow \text{I,riq}$$

$$\vdots$$

$$\mathbf{73 \rightarrow 114 \rightarrow 97 \rightarrow 113 \rightarrow 105 \Rightarrow Iraqi}$$

The correct path is $73 \rightarrow 114 \rightarrow 97 \rightarrow 113 \rightarrow 105$ which gives the correct original plaintext **Iraqi**.

Example 3.4.1.2. (Study Case II: Encryption Scheme Based on the TPB Graph)

Suppose *m* is a plaintext that is given by an English word **(MATH)**. Based on the ASCII Table (3.2). The letters of this word are converted into numbers as follows.

$$(\rightarrow 40, \text{M} \rightarrow 77, \text{A} \rightarrow 65, \text{T} \rightarrow 84, \text{H} \rightarrow 72,) \rightarrow 41.$$

Now, the length of the plaintext is K = 6. The first user chooses a prime number *p*, where *p* is the near prime number greater than 127 and *p* is a shared secret key.

Let p = 137. The computations of the inverse elements of the numbers 40, 77, 65, 84, 72 and 41 modulo 137 have been done using the extended Euclidean algorithm (EEA) to get

$$24^{-1} \pmod{137} \equiv 24$$

$$77^{-1} \pmod{137} \equiv 121$$

$$65^{-1} \pmod{137} \equiv 78$$

$$84^{-1} \pmod{137} \equiv 31$$

$$72^{-1} \pmod{137} \equiv 59$$

$$41^{-1} \pmod{137} \equiv 127$$

After that, the first user chooses the primes p_i such that the number of the primes # p_i is less than k-2, say p_1=5, p_2=7. Now, two graphs are formed based on the inverse elements and prime numbers as shown in Figure (3.8).

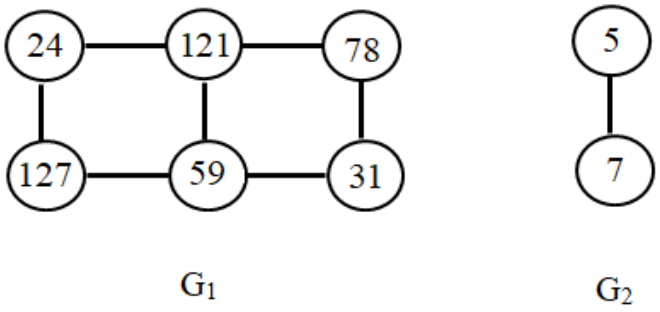


Figure 3.8. The bipartite graph G_1 and G_2 with 6 and 2 vertices respectively.

The ciphertext *C* of a message *m* is constructed as the tensor product bipartite (TPB) graph of graphs G_1 and G_2 by

$G_1 \times G_2$ = {(24,5), (121,5), (78,5), (31,5), (59,5), (127,5), {(24,7), (121,7), (78,7), (31,7), (59,7), (127,7)}.

The TPB graph is shown in Figure (3.9) and which is sent to second user as a ciphertext.

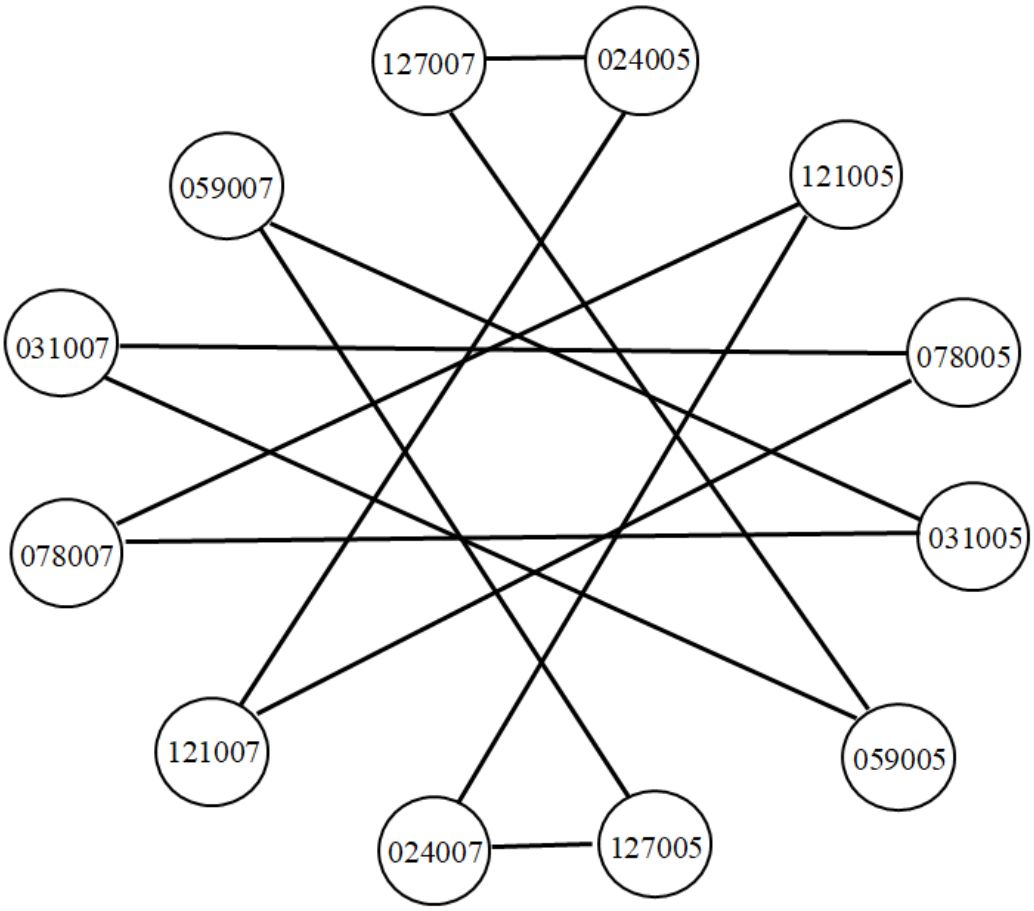


Figure 3.9 the TPB graph $G_1 \times G_2$ of graphs G_1 and G_2 that are shown in figure 3.8

Upon second user receives the TPB graph, he / she will do the following steps:

He / She takes one set of the vertices of the independent set of TPB graph

List 1: 024005, 121005, 078005, 031005, 059005, 127005 or

List 2: 024007, 121007, 078007, 031007, 059007, 127007.

If he / she chooses first list, then the second user uses his/ her shared secret key p = 137 to compute the inverse elements of list 2 as follows.

247 $24 \rightarrow 24^{-1} \pmod{137} \equiv 40$

$7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

1217 $121 \rightarrow 121^{-1} \pmod{137} \equiv 77$

$7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

787 $78 \rightarrow 78^{-1} \pmod{137} \equiv 65$

$7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

317 $31 \rightarrow 31^{-1} \pmod{137} \equiv 84$

$7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

$$597 \qquad 59 \rightarrow 59^{-1} \pmod{137} \equiv 72$$

$$7 \rightarrow \ 7^{-1} \qquad \pmod{137} \equiv 98$$

$$1277 \qquad 127 \rightarrow 127^{-1} \pmod{137} \equiv 41$$

$$7 \rightarrow \ 7^{-1} \qquad \pmod{137} \equiv 98$$

Since the length of the message $k = 5$ that are can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct message consist of 6 vertices

$$40 \rightarrow 98 \rightarrow 77 \rightarrow 65 \rightarrow 84 \rightarrow 72 \Rightarrow (\text{bMATH}$$

$$40 \rightarrow 98 \rightarrow 77 \rightarrow 65 \rightarrow 84 \rightarrow 41 \Rightarrow (\text{bMAT})$$

$$40 \rightarrow \ \rightarrow 77 \rightarrow 65 \rightarrow 72 \rightarrow 41 \Rightarrow (\text{bMAH})$$

$$40 \rightarrow 98 \rightarrow 77 \rightarrow 65 \rightarrow 41 \rightarrow 98 \Rightarrow (\text{bMA})($$

$$40 \rightarrow 98 \rightarrow 77 \rightarrow 65 \rightarrow 98 \rightarrow 41 \Rightarrow (\text{bMAb})$$

$$\vdots$$

$$\mathbf{40 \rightarrow 77 \rightarrow 65 \rightarrow 84 \rightarrow 72 \rightarrow 41 \Rightarrow (MATH)}$$

The correct path is $40 \rightarrow 77 \rightarrow 65 \rightarrow 84 \rightarrow 72 \rightarrow 41$ which gives the correct original plaintext **(MATH).**

3.5. The Security Considerations of STPG Encryption Schemes

The proposed symmetric encryption schemes based on the TPB graphs is a more secure in compare with other symmetric encryption schemes. With TPB, the ciphertext has been computed as the TPB by depending on two graphs G_1 and G_2. These graphs are created based on the inverses elements of the plaintext and a secret choice of the primes p_i. The security considerations of new proposed STPG encryption schemes depended on random generating the graphs G_1 and G_2 that the attackers want to know them if they determine the ciphertext is computed as TPB graph. So, Eve needs to guess the vertices of graphs G_1 and G_2. Therefore, in case I, there are 26 possible probabilities to form the correct path graph. Thus, the total probability can be determined by

$$C_k^n = \binom{n}{k} = \frac{n!}{k!\,(n-k)!}.$$

Based on the result of Case (I), with n = 26 and K =5, then

$$C_5^{26} = \frac{26!}{5!\,(26-5)!} = 65780.$$

Therefore, there are 65780 paths are corresponded to the plaintext; one of them is correct one.

Whereas, on study case (II), the probability of all possible to the path graphs is computed by

$$C_5^{127} = \frac{127!}{5!\,(127-5)!} = 254231775.$$

Thus, one of the 254231775 paths is the correct path graph that gives the correct original plaintext. So, if the adversaries know a ciphertext of a plaintext m is computed by $G_1 \times G_2$ and represented and sent as the TPB graph, they need to guess more and more probability cases to generate the graphs G_1 and G_2. Hence, it is more secure to recover the original message among all possible probabilities cases.

Chapter Four

The Tensor and Strong Product Graphs for Polyalphabetic Encryption Scheme

Chapter Four

The Tensor and Strong Product Graphs for Polyalphabetic Encryption Scheme

4.1 Introduction

In this chapter, the tensor and strong product graphs are used to give alternative modified polyalphabetic encryption schemes. Four types of symmetric polyalphabetic encryption schemes have been proposed. First and third ones based of the English alphabet values, whereas, second and fourth ones used the ASCII values to represent the letters of the plaintexts. Some examples on these types of the proposed symmetric encryption schemes are discussed.

And also the strong product graph concept is defined as follows.

4.2 The Strong Product Graph for Encryption Schemes

In this section, a strong product graph has been explained with some examples as follows.

Definition 4.2.1. A strong product graph $G \boxtimes H$ of graphs G and H is a graph such that

- The vertex set of $G \times H$ is the Cartesian product $V(G) \times V(H)$ and
- Distinct vertices (u,u') and (v,v') are adjacent in $G \boxtimes H$ if and only if:
  1. $u = v$ and u' is an adjacent to v', or
  2. $u' = v'$ and u is an adjacent to v, or
  3. u is an adjacent to v and u' is an adjacent to v'.[2]

Example 4.2.1. Suppose G is a graph has vertices {A, B, C} and H is a graph has vertices {1,2}. Then, a strong product graph $G \boxtimes H$ is computed by

$$G \boxtimes H = \{(A,1), (B,1), (C,1), (A,2), (B,2), (C,2)\}.$$

A strong product graph $G \boxtimes H$ is shown in Figure (4.1).

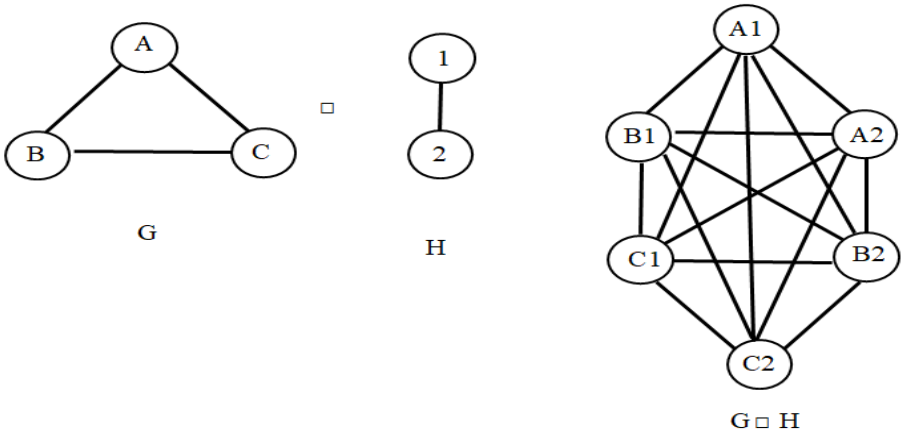


Figure 4.1. A strong product graph $G \boxtimes H$ of graphs G and H.

Example 4.2.2. Suppose G is a graph has vertices {A, B} and H is a graph has vertices {1,2}. Then, a strong product graph $G \boxtimes H$ is computed by

$$G \times H = \{(A,1), (B,1), (A,2), (B,2)\}.$$

A strong product graph $G \boxtimes H$ is shown in Figure (4.2).

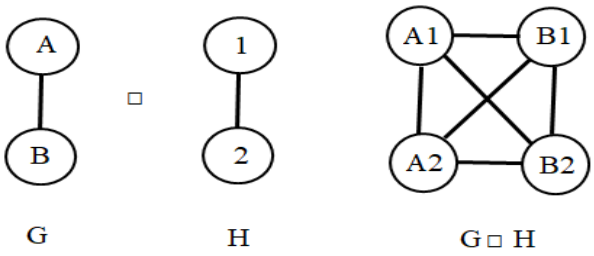


Figure 4.2. A strong product graph $G \boxtimes H$ of two path graphs G and H.

Example 4.2.3. Suppose G is a graph has vertices {A, B, C, D} and H is a graph has vertices {1,2}. Then, a strong product graph $G \boxtimes H$ is computed by

$$G \times H = \{(A,1), (B,1), (C,1), (D,1), (A,2), (B,2), (C,2), (D,2)\}.$$

A strong product graph $G \boxtimes H$ is shown in Figure (4.3).

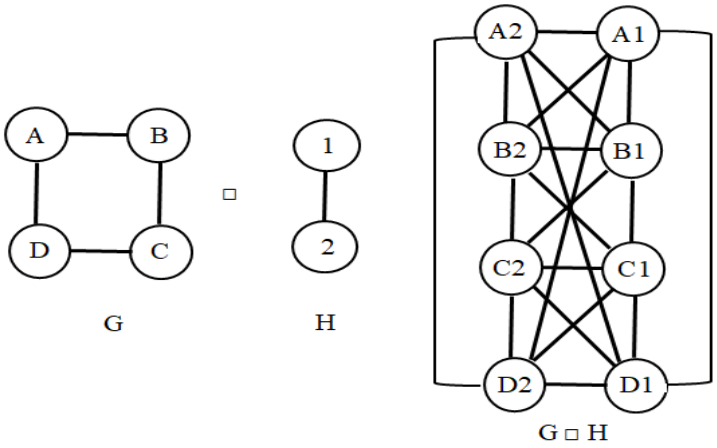


Figure 4.3. A strong product graph $G \boxtimes H$ of simple graph G and path H.

4.3 The TPG for Polyalphabetic Encryption Scheme Based on English Alphabet Values

Before starting with the proposed encryption schemes, it is important to explain the polyalphabetic cipher. This cipher based on the substitution using the multiple substitution English alphabets. It is considered as symmetric encryption scheme, since it depended on the shared secret key. On this key, some rules are determined to make it more secure and difficult to recover.

Suppose m is a message given by an English word or the English sentence. In other words,

$$m = \{m_1, m_2, \ldots, m_k\}.$$

The length of the message is k. Some rules on the key are determined. The letters m_i of message are converted using the rules of key into encoded letters e_i for i =1, 2, …, k, namely

$$m_1, m_2, \ldots, m_k \rightarrow e_1, e_2, \ldots, e_k.$$

The encoded letters of the message "$e_1e_2\ldots e_k$" are converted into numbers using the English alphabet values in Table (3.1).

$$e_1 \rightarrow \#e_1, e_2 \rightarrow \#e_2, \ldots, e_k \rightarrow \#e_k.$$

The first user chooses p, where p is the near prime number greater than 26 (26 is the number of the English alphabet letters) and p is a shared secret key that is computed by the Diffie – Hellman key exchange. The computations of the inverse elements of the numbers $\#e_1, \#e_2, \ldots, \#e_k$ modulo p have been done using the extended Euclidean algorithm (EEA) to get

$$(\#e_i)^{-1} \pmod{p} \equiv a_i, \text{ for } i = 1, 2, \ldots, k.$$

After that, the first user chooses the primes p_i such that the number of the primes $\#p_i$ is less than k-2, namely $p_1, p_2, \ldots, p_{k-3}$. Now, two graphs G_1 and G_2 are formed based on the inverse elements and prime numbers. The ciphertext C of a message m is constructed as the tensor product graph (TPG) of graphs G_1 and G_2 which is sent to second user.

Upon second user receives the TPG, he / she will do the following steps:

He/ She takes one set of the vertices of the independent set of TPG

List 1: $a_1p_1, a_2p_1, \ldots, a_lp_l$ or List 2: $a_{l+1}p_{l+1}, a_{l+2}p_{l+1}, \ldots, a_kp_{k-3}$.

If he / she chooses first list, then the second user uses his/ her shared secret key p to compute the inverse elements of list 1 as follows.

a_ip_i $\quad a_i \rightarrow a_i^{-1} \pmod{p} \equiv s_i$ (which maybe equal to correct $\#e_i$ or not)

$\quad p_i \rightarrow p_i^{-1} \pmod{p} \equiv s_j$ (which maybe equal to correct $\#e_j$ or not).

Since the length of the message k, which can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct encoded word consist of k vertices. The correct path is

$$e_1 \rightarrow \#e_1, e_2 \rightarrow \#e_2, \ldots, e_k \rightarrow \#e_k,$$

which gives the correct encoded word "$e_1e_2\ldots e_k$".

For decryption process, second user using inverse rules of a shared secret key to recover the original plaintext.

So, the encoded word

$$e_1e_2\ldots e_k,$$

becomes

$$m_1m_2\ldots m_k$$

which is the original plaintext "$m_1m_2\ldots m_k$".

In the similar way, one can apply the strong product graph (SPG) with the same idea to give another version of **the SPG for polyalphabetic encryption scheme**, see Example (4.4.2).

Example 4.3.1. The TPG for Polyalphabetic Encryption Scheme

Suppose m is a plaintext that is given by an English word **Cipher**. Based on the alphabet Table (3.1), some rules on the key are determined by

1. Shift first letter three positions to its right.
2. Shift second letter two positions to its left.
3. Shift third letter four positions to its right.

The message CIPHER converted using the rules of key into

CIP HER

FGT KCV

The letters of the word **"FGTKCV"** are converted into numbers as follows:

$$F \rightarrow 6, G \rightarrow 7, T \rightarrow 20, K \rightarrow 11, C \rightarrow 3, V \rightarrow 22.$$

Now, the length of the message is K = 6. The first user chooses p, where p is the near prime number greater than 26 (26 is the number of the English alphabet letters) and p is a shared secret key that is computed by the Diffie – Hellman key exchange. Let p = 29. The computations of the inverse elements of the numbers 6, 7, 20, 11, 3 and 22 modulo 29 have been done using the extended Euclidean algorithm (EEA) to get

$$6^{-1} \pmod{29} \equiv 5$$

$$7^{-1} \pmod{29} \equiv 25$$

$$20^{-1} \pmod{29} \equiv 16$$

$$11^{-1} \pmod{29} \equiv 8$$

$$3^{-1} \pmod{29} \equiv 10$$

$$22^{-1} \pmod{29} \equiv 4$$

After that, the first user chooses the primes p_i such that the number of the primes # p_i is less than k-2 , say p_1=5, p_2=7. Now, two graphs are formed based on the inverse elements and prime numbers as shown in Figure (4.4).

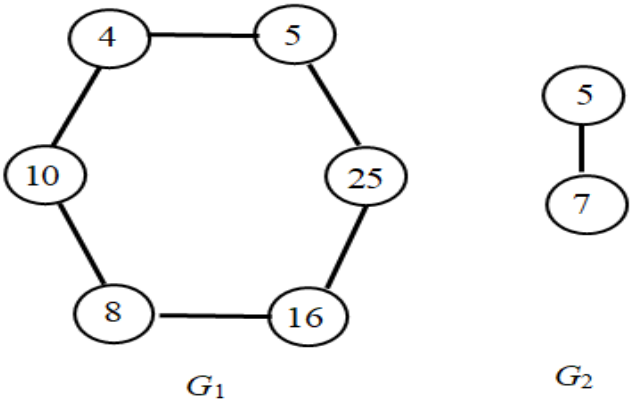


Figure 4.4. The graphs G_1 and G_2 with 6 and 2 vertices respectively.

The ciphertext *C* of a message *m* is constructed as the tensor product graph (TPG) of graphs G_1 and G_2 as shown in Figure (4.2), which is sent to second user.

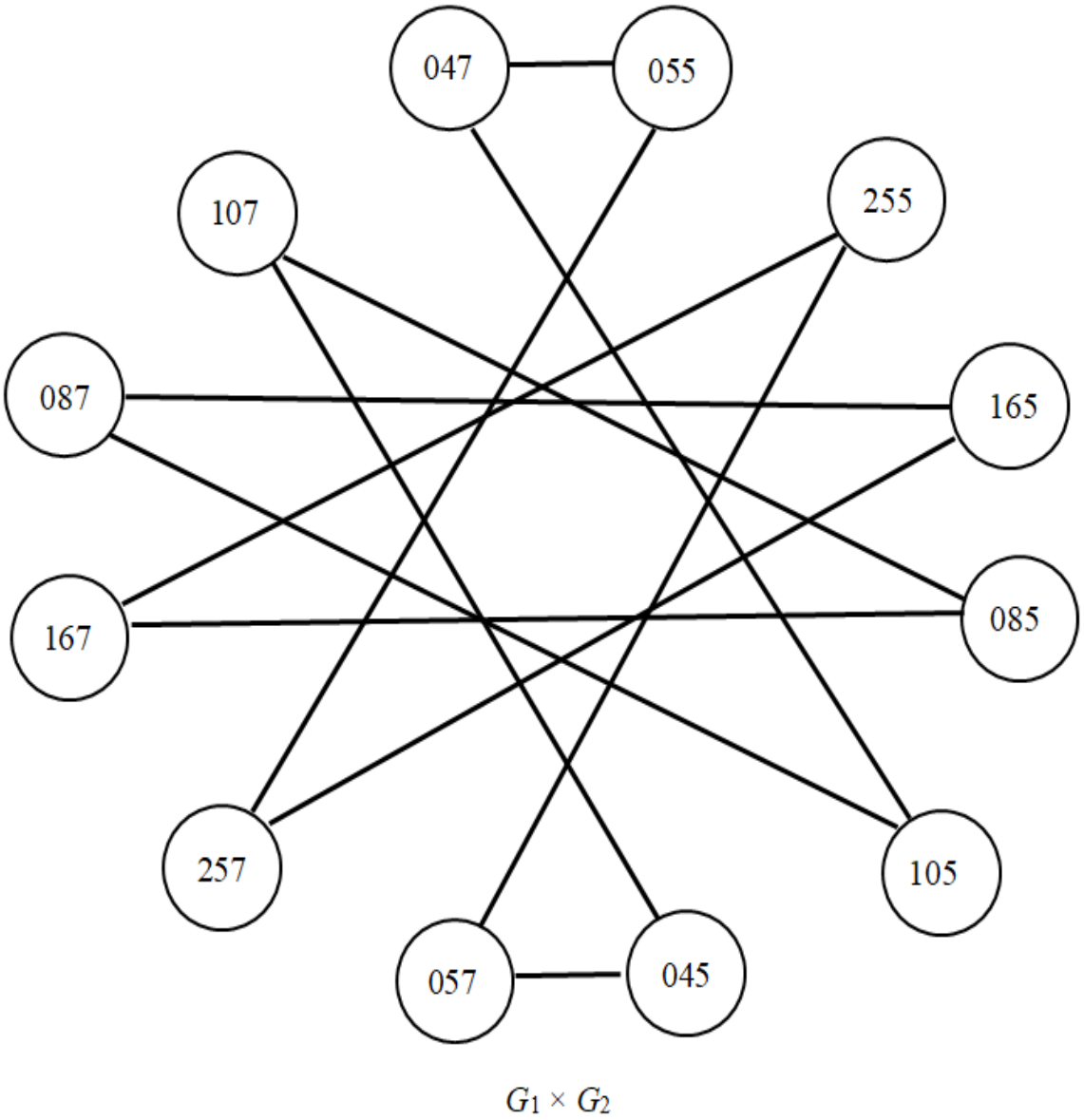


Figure 4.5. The TPG $G_1 \times G_2$ of graphs G_1 and G_2 that are shown in Figure (4.4).

Upon second user receives the TPB graph, he / she will do the following steps:

He / She takes one set of the vertices of the independent set of TPB graph

List 1: 055, 255, 165, 085, 105, 045 or List 2: 057, 257, 167, 087, 107, 047.

If he / she chooses first list, then the second user uses his/ her shared secret key p = 29 to compute the inverse elements of List 1 as follows.

$$55 \qquad 5 \rightarrow 5^{-1} \pmod{29} \equiv 6$$

$$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$$

$$255 \qquad 25 \rightarrow 25^{-1} \pmod{29} \equiv 7$$

$$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$$

$$165 \qquad 16 \rightarrow 16^{-1} \pmod{29} \equiv 20$$

$$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$$

$$85 \qquad 8 \rightarrow 8^{-1} \pmod{29} \equiv 11$$

$$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$$

$$105 \qquad 10 \rightarrow 10^{-1} \pmod{29} \equiv 3$$

$$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$$

$$45 \qquad 4 \rightarrow 4^{-1} \pmod{29} \equiv 22$$

$$5 \rightarrow 5^{-1} \pmod{29} \equiv 6$$

Since the length of the message k = 6 which can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct word consist of 6 vertices. The correct path is

$$6 \rightarrow \text{F},\ 7 \rightarrow \text{G},\ 20 \rightarrow \text{T},\ 11 \rightarrow \text{K},\ 3 \rightarrow \text{C},\ 22 \rightarrow \text{V}.$$

which gives the correct word **“FGTKCV”**.

For decryption process, second user using the following rules of the key to recover the original plaintext. The rules are

1. Shift first letter three positions to its left.
2. Shift second letter two positions to its right.
3. Shift third letter four positions to its right.

So, the word

FGT KCV,

becomes

CIP HER

which is the original plaintext **"Cipher"**.

Example 4.3.2. The SPG for Polyalphabetic Encryption Scheme

Suppose *m* is a plaintext that is given by an English word **Class**. Based on the alphabet Table (3.1), some rules on the key are determined by

1. Shift first letter two positions to its right.
2. Shift second letter three positions to its left.

The message Class converted using the rules of key

Cl as s

into

Ei cp u

The letter of the word **"Eicpu"** are converted into numbers as follows:

$$E \rightarrow 5, I \rightarrow 9, C \rightarrow 3, P \rightarrow 16, U \rightarrow 21.$$

Now, the length of the message is K = 5. The first user chooses *p*, where *p* is the near prime number greater than 26 and *p* is a shared secret key. Let *p* = 31. The computations of the inverse elements of the numbers 5, 9, 3, 16 and 21 modulo 31 have been done using the extended Euclidean algorithm (EEA) to get

$$5^{-1} \pmod{31} \equiv 25$$

$$9^{-1} \pmod{31} \equiv 7$$

$$3^{-1} \pmod{31} \equiv 21$$

$$16^{-1} \pmod{31} \equiv 2$$

$$21^{-1} \pmod{31} \equiv 3$$

After that, the first user chooses the primes p_i such that the number of the primes # p_i is less than *k*-2 , say p_1=7, p_2=11. Now, two graphs are formed based on the inverse elements and prime numbers as shown in Figure (4.6).

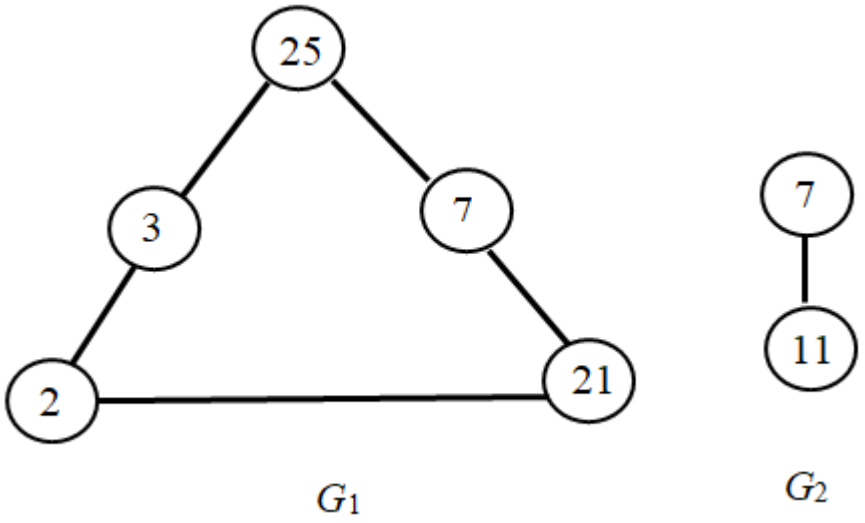


Figure 4.6. The graphs G_1 and G_2 with 5 and 2 vertices respectively.

The ciphertext *C* of a message *m* is constructed as the strong product graph (TPG) of graphs G_1 and G_2 as shown in Figure (4.7), which is sent to second user.

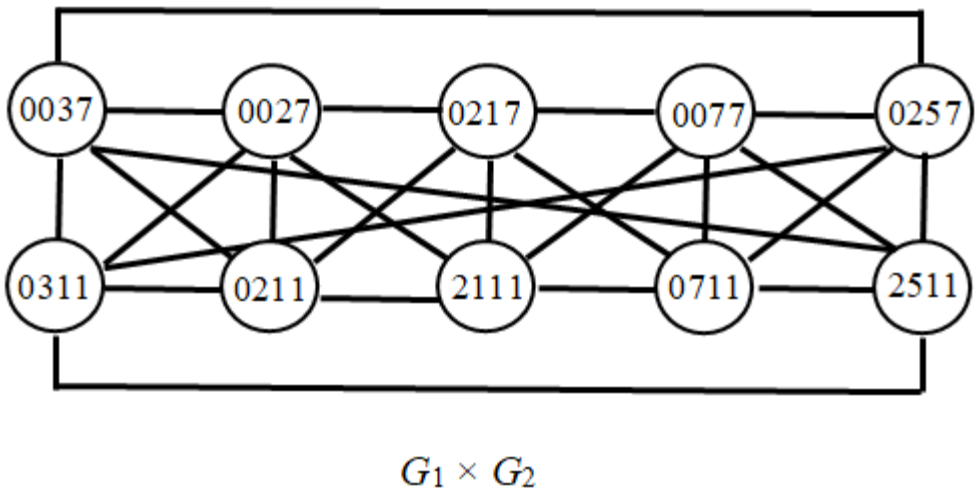


Figure 4.7. The SPG $G_1 \times G_2$ of graphs G_1 and G_2 that are shown in Figure (4.6).

Upon second user receives the SPB graph, he / she will do the following steps:

He / She takes one set of the vertices of the independent set of SPB graph

List 1: 0257, 0077, 0217, 0027, 0037 or List 2: 2511, 0711, 2111, 0211, 0311.

If he / she chooses second list, then the second user uses his/ her shared secret key $p = 29$ to compute the inverse elements of List 2 as follows.

2511 $25 \rightarrow 25^{-1} \pmod{31} \equiv 5$

$11 \rightarrow 11^{-1} \pmod{31} \equiv 17$

711 $7 \rightarrow 7^{-1} \pmod{31} \equiv 9$

$11 \rightarrow 11^{-1} \pmod{31} \equiv 17$

2111 $21 \rightarrow 21^{-1} \pmod{31} \equiv 3$

$11 \rightarrow 11^{-1} \pmod{31} \equiv 17$

211 $2 \rightarrow 2^{-1} \pmod{31} \equiv 16$

$11 \rightarrow 11^{-1} \pmod{31} \equiv 17$

311 $3 \rightarrow 3^{-1} \pmod{31} \equiv 21$

$11 \rightarrow 11^{-1} \pmod{31} \equiv 17$

Since the length of the message $k = 5$ which can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct word consist of 5 vertices. The correct path is

$$5 \rightarrow \text{E}, 9 \rightarrow \text{I}, 3 \rightarrow \text{C}, 16 \rightarrow \text{P}, 21 \rightarrow \text{U}.$$

which gives the correct encoded word **"EICPU"**.

For decryption process, second user using the following rules of the key to recover the original plaintext. The rules are

1. Shift first letter two positions to its left.
2. Shift second letter three positions to its right.

So, the word

EI CP U,

becomes

CL AS S

which is the original plaintext **"Class"**.

4.4 The TPG for Polyalphabetic Encryption Scheme Based on ASCII Values

Same idea that is applied with the English alphabet values can be implemented with the ASCII values as show in the following examples using the TPG and SPG.

Example 4.4.1. The TPG for Polyalphabetic Encryption Scheme Based on ASCII Values

Suppose *m* is a plaintext that is given by an English word **"(MATH)"**. Based on the ASCII Table (3.2), some rules on the key are determined by

1. Shift first letter two positions down.
2. Shift second letter three positions to up.

The message **(MATH)** converted using the rules of key into

(M AT H)
*J CQ J&

The letters of the encoded word **"***JCQJ&**"** are converted into numbers as follows:

$$* \rightarrow 42,\ J \rightarrow 74,\ C \rightarrow 67,\ Q \rightarrow 81,\ J \rightarrow 74,\ \& \rightarrow 38.$$

Now, the length of the message is K = 6. The first user chooses *p*, where *p* is the near prime number greater than 127 and *p* is a shared secret key that is computed by the Diffie – Hellman key exchange. Let *p* = 131. The computations of the inverse elements of the numbers 42, 74, 67, 81, 74 and 38 modulo 131 have been done using the extended Euclidean algorithm (EEA) to get

$$42^{-1} \pmod{131} \equiv 78$$

$$74^{-1} \pmod{131} \equiv 108$$

$$67^{-1} \pmod{131} \equiv 88$$

$$81^{-1} \pmod{131} \equiv 55$$

$$74^{-1} \pmod{131} \equiv 108$$

$$38^{-1} \pmod{131} \equiv 100$$

After that, the first user chooses the primes p_i such that the number of the primes # p_i is less than k-2 , say p_1=5, p_2=7. Now, two graphs are formed based on the inverse elements and prime numbers as shown in Figure (4.8).

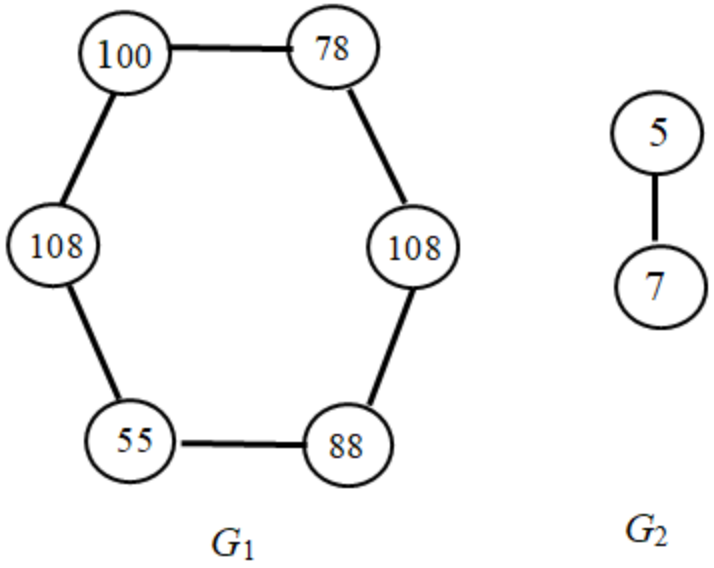


Figure 4.8. The graphs G_1 and G_2 with 6 and 2 vertices respectively.

The ciphertext C of a message m is constructed as the tensor product graph (TPG) of graphs G_1 and G_2 as shown in Figure (5.6), which is sent to second user.

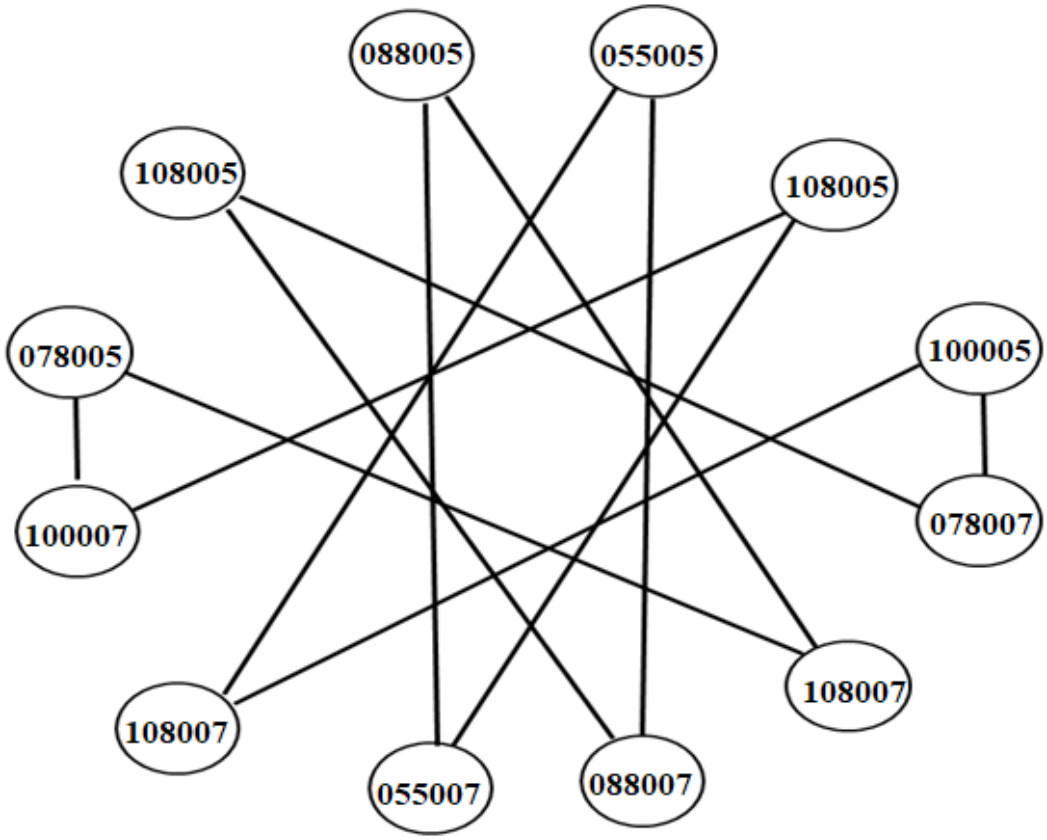


Figure 4.9. The TPG $G_1 \times G_2$ of graphs G_1 and G_2 that are shown in Figure (4.8).

Upon second user receives the TPB graph, he / she will do the following steps:

He / She takes one set of the vertices of the independent set of TPB graph

List 1: 078005, 108005, 088005, 055005, 108005, 100005 or

List 2: 078007, 108007, 088007, 055007, 108007, 100007.

If he / she chooses first list, then the second user uses his/ her shared secret key $p = 131$ to compute the inverse elements of List 1 as follows.

$$078005 \quad 78 \rightarrow 78^{-1} \pmod{131} \equiv 42$$

$$5 \rightarrow 5^{-1} \pmod{131} \equiv 105$$

$$108005 \quad 108 \rightarrow 108^{-1} \pmod{131} \equiv 74$$

$$5 \rightarrow 5^{-1} \pmod{131} \equiv 105$$

$$088005 \quad 88 \rightarrow 88^{-1} \pmod{131} \equiv 67$$

$$5 \rightarrow 5^{-1} \pmod{131} \equiv 105$$

$$055005 \quad 55 \rightarrow 55^{-1} \pmod{131} \equiv 81$$

$$5 \rightarrow 5^{-1} \pmod{131} \equiv 105$$

$$108005 \quad 108 \rightarrow 108^{-1} \pmod{131} \equiv 74$$

$$5 \rightarrow 5^{-1} \pmod{131} \equiv 105$$

$$100005 \quad 100 \rightarrow 100^{-1} \pmod{131} \equiv 38$$

$$5 \rightarrow 5^{-1} \pmod{131} \equiv 105$$

Since the length of the message $k = 6$ which can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct word consist of 6 vertices. The correct path is

$$42 \rightarrow *,\ 74 \rightarrow J,\ 67 \rightarrow C,\ 81 \rightarrow Q,\ 74 \rightarrow J,\ 38 \rightarrow \&.$$

which gives the correct encoded word **"***JCQJ&**"**.

For decryption process, second user using the inverse rules of the key to recover the original plaintext. The rules are

1. Shift first letter two positions to up.
2. Shift second letter three positions to down.

So, the encoded word

*J CQ J&

becomes

(M AT H)

which is the original plaintext **"**(MATH)**"**.

Example 4.4.2. The SPG for Polyalphabetic Encryption Scheme Based on ASCII Values

Suppose *m* is a plaintext that is given by an English word **"Plan32"**. Based on the ASCII Table (3.2), some rules on the key are determined by

1. Shift first letter three positions into up.
2. Shift second letter two positions into up.
3. Shift third letter one position into down.

The message **"Plan32"** converted using the rules of key into

Pla n32

Mjb k13

The letters of the encoded word **"Mjbk13"** are converted into numbers as follows:

$$M \rightarrow 77, j \rightarrow 106, b \rightarrow 98, k \rightarrow 107, 1 \rightarrow 49, 3 \rightarrow 51.$$

Now, the length of the message is k = 6. The first user chooses p, where p is the near prime number greater than 127 and p is a shared secret key. Let p = 137. The computations of the inverse elements of the numbers 77, 106, 98, 107, 49 and 51 modulo 137 have been done using the extended Euclidean algorithm (EEA) to get

$$77^{-1} \pmod{137} \equiv 121$$

$$106^{-1} \pmod{137} \equiv 53$$

$$98^{-1} \pmod{137} \equiv 7$$

$$107^{-1} \pmod{137} \equiv 105$$

$$49^{-1} \pmod{137} \equiv 14$$

$$51^{-1} \pmod{137} \equiv 43$$

After that, the first user chooses the primes p_i such that the number of the primes # p_i is less than k-2 , say p_1=5, p_2=7. Now, two graphs are formed based on the inverse elements and prime numbers as shown in Figure (4.10).

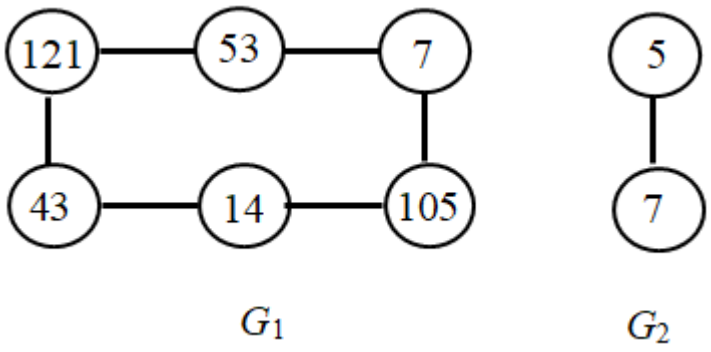
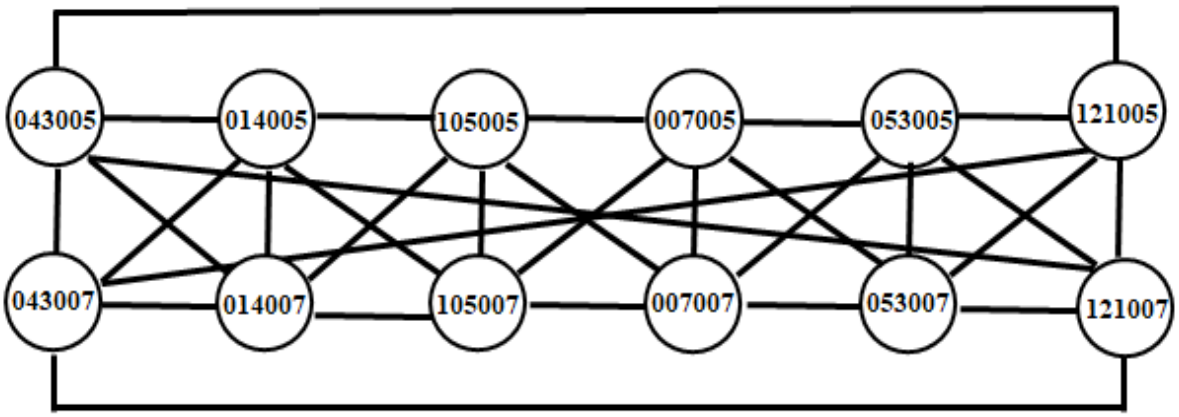


Figure 4.10. The graphs G_1 and G_2 with 5 and 2 vertices respectively.

The ciphertext C of a message m is constructed as the strong product graph (SPG) of graphs G_1 and G_2 as shown in Figure (5.8), which is sent to second user.



$G_1 \times G_2$

Figure 4.11. The SPG $G_1 \times G_2$ of graphs G_1 and G_2 that are shown in Figure (4.10).

Upon second user receives the SPB graph, he / she will do the following steps:

He / She takes one set of the vertices of the independent set of SPB graph

List 1: 121005, 053005, 007005, 105005, 014005, 043005 or

List 2: 121007, 053007, 007007, 105007, 014007, 043007.

If he / she chooses second list, then the second user uses his/ her shared secret key $p = 137$ to compute the inverse elements of List 2 as follows.

121007 $121 \rightarrow 121^{-1} \pmod{137} \equiv 77$

$7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

053007 $53 \rightarrow 53^{-1} \pmod{137} \equiv 106$

$7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

007007 $7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

$7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

105007 $105 \rightarrow 105^{-1} \pmod{137} \equiv 107$

$7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

014007 $14 \rightarrow 14^{-1} \pmod{137} \equiv 49$

$7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

043007 $43 \rightarrow 43^{-1} \pmod{137} \equiv 51$

$7 \rightarrow 7^{-1} \pmod{137} \equiv 98$

Since the length of the message $k = 6$ which can be known from the number of the vertices of the independent set, so the path graphs that correspond to the correct word consist of 6 vertices. The correct path is

$$77 \rightarrow M,\ 106 \rightarrow j,\ 98 \rightarrow b,\ 107 \rightarrow k,\ 49 \rightarrow 1,\ 51 \rightarrow 3.$$

which gives the correct encoded word **"Mjbk13".**

For decryption process, second user using the inverse rules of the key to recover the original plaintext. The rules are

1. Shift first letter three positions into down.
2. Shift second letter two positions into down.
3. Shift third letter one position into up.

So, the encoded word

Mjb k13

becomes **Pla n32**

which is the original plaintext **"Plan32"**.

Chapter Five

Conclusions and Future Works

Chapter Five

Conclusions and Future Works

5.1 Conclusions

In this work, one can conclude that the concepts of graph theory have been used to give new sights for proposing new versions of symmetric encryption schemes. This application used the TPG and SPG to design these versions with more secure level to create the ciphertext of the original message. These versions are TPG encryption scheme based on English alphabet values and SPG encryption scheme based on ASCII values. On the other hand, these graphs are applied to modify the polyalphabetic substitution cipher.

5.2 Future Works

- It is possible to apply the same idea of the proposed encryption scheme with other kinds of symmetric and asymmetric encryption schemes.
- Also it can use other types of the graphs.

References

[1] A. C. Shantha Sheela, P.Amudha, A.C.Charles Sagayaraj (2018) " An Application of Graph Theory in Cryptography ". Internationl journal of pure and Applied Mathematics 119 (13), 375-383.

[2] B. Bresar, S.Spacapan (2007) "Edge connectivity of strong products of graphs".Discussiones Mathematicae Graph Theory 27(2), 333-343,.

[3] BNi, R Qazi, SU Rehman and G Farid. (2021) "Some-Based Encryption\ Schemes". Journal of mathematics.

[4] Jonathan L. Gross, Tay Yellen, Ping Zhang (2013) "Hand book of graph Theory" Second Edition,books.google.com..

[5] Jonathan L. Gross, Jay Yellen, Mark Anderson (2018) "Graph Theory and Its Applications". 3rd Edition,

[6] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, (2014), An Introduction to Mathematical Cryptography, Springer-Verlag– Undergraduate Texts in Mathematics, ISBN: 978-1-4939-1710-5 – 2nd ed.

[7] M. Tavakoli, F. Rahbarnia Ali Reza Ashrafi (2013) "Note on strong Product of graphs ". Kragujevac Journal of Mathematics 37(1), 187-193.

[8] Natalia Tokareva (2014) "Connections between graph theory and cry ptography" G2C2: Graphs and Groupts, Cycles and Covering, September, 24-26,.

[9] P.L.K.Priyadarsini, (2015) "Asurvey onsome applications of graph theory in cryptography ". Journal of Discrete Mathematical Science and Cryptograph 18(3), 209-217,.

[10] Robin J. Wilson (2015) "Introduction to Graph Theory " Fourth edition.

[11] Sirous Moradi (2012) "A Note on Tensor Product of Graphs ".Iranian Journal of Mathematical,Sciences and In formatics 7(1), 73-81.

[12] S. Saha Ray. (2013) "Graph Theory with Algorithms its Applications" springer:Berlin, Germany.

[13] Srilekha Chowdhury, Promita Ghosh, Mayurakshi Jana, (2020)" An Approach of Graph Theory for Solving Cryptographic Problem".

[14] Tanush Shaska, V.Ustimenko, (2008) "On some application S of graphs to Cryptography and turbocoding". Albanian J.Math. 2 (3), 249-255,.

[15] U. P. A charga and H.S.Mehta (2014) "Tensor Product of Graphs". International Journal of Mathematics and Scientific Compoting 4(1),.

[16] V. Raja, H. P. Patil* (2015), " On Tensor Product of Graphs, Girth and Triangles" Department of Mathematics Pondicherry University, Pondicherry, India..

[17] Wael Etaiwi, (2014) "Encryption Algorithm Using Graph Theory". Journal of Scientific- Research and Reports.

ملخص البحث :

تم اقتراح اصدارات جديدة من انظمة التشفير المتماثل في هذا العمل . تستخدم هذه الاصدارات تعريفا جديدا لمؤشر Tensor و Strong Product Graph .

اعتمدت هذه المخططات المقترحة الجديدة على قيم الابجدية الانجليزية وقيم ASCII والتشفير الابجدي المتعدد على التوالي . يتم اختيار الرسالة ككلمة انجليزية أو جملة انجليزية . يعتبر نص التشفير للرسالة الاصلية بمثابة Tensor وStrong Product Graph الذي يتم ارساله الى جهاز الاستقبال عن طريق ارسال العديد من النتائج التجريبية لمخططات التشفير المقترحة . يتم تحديد اعتبارات الامان لأنظمة تشفير Propose Tensor و Strong Product Graph

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية التربية للعلوم الصرفة
قسم الرياضيات

نظرية البيان وتطبيقاتها في التشفير

بحث مقدم

الى مجلس كلية التربية للعلوم الصرفة / جامعة بابل كجزء
من متطلبات نيل درجة الدبلوم العالي تربية / رياضيات

من قبل

منى حيدر هاشم محمد حسن

بأشراف

د. رومى كريم خضر عجينة

١٤٤٣هـ **٢٠٢١ م**