Republic of Iraq
Ministry of Higher Education
and Scientific Research
University of Babylon
College of Education for Pure Sciences
Department of Mathematics

Secure Cryptographic Schemes Using the Graph Theory

A Research
Submitted to the Council of the College of Education for Pure Sciences in the University of Babylon Partial Fulfillment of the Requirements for the Degree of Higher Diploma Education /Mathematics

By
Marwah Ali Hussein Ali

Supervised by
Prof. Dr. Ahmed Abd Ali Omran

2021 A. D. 1442 A. H.

بِسْمِ اللَّـهِ الرَّحْمَـٰنِ الرَّحِيمِ

(وَلَقَدْ آتَيْنَا دَاوُدَ وَسُلَيْمَانَ عِلْمًا وَقَالَا الْحَمْدُ لِلَّهِ الَّذِي فَضَّلَنَا عَلَى كَثِيرٍ مِنْ عِبَادِهِ الْمُؤْمِنِينَ)

صدق اللَّهُ العظيم

سورة النمل أية(١٥)

Supervisor Certification

I certify that the thesis entitled the **“Secure Cryptographic Schemes Using the Graph Theory"** by **"Marwah Ali Hussein Ali”** has been prepared under my supervision in Babylon University/ College of Education for Pure Sciences as a partial requirement for the degree of Higher Diploma Education / Mathematics.

Signature:

Name: Dr**.** Ahmed Abd Ali Omran
Title: Professor
Date: / / 2021

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

Signature:

Name: Dr. Azal Jaafar Musa
Head of Mathematics Department
Title: Assistant Professor
Date: / / 2021

Examining Committee Certification

We certify that we have read the thesis entitled the **“Secure Cryptographic Schemes Using the Graph Theory"** by **"Marwah Ali Hussein Ali”** 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 Education / Mathematics.

Signature: Signature:

Name: Dr. Assad M. A. Alhossaini Name: Dr. Ruma Kareem K.Ajeena
Title: Professor Title: Assistant Professor
Date: / / 2021 Date: / / 2021
Chairman Member

Signature: Signature:

Name: Dr.Hussein Abd AL− wasi Hussein Name: Dr. Ahmed Abd Ali Omran
Title: Lecturer Title: Professor
Date: / / 2021 Date: / / 2021
Member / Member / Advisor

I hereby certify the decision of the examining committee. Signature:

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

Linguistic Supervisor's Certification

This is to certify that I have read this thesis entitled **" Secure Cryptographic Schemes Using the Graph Theory"** and I found that this thesis is qualified for debate.

Signature:

Name: Dr. Ali Hussein Mahmood AL− Obaidi

Title: : Lecturer

Address: Department of , College Mathematics of Education for pure Sciences, University of Babylon

Date: / / 2021

Scientific Supervisor's Certification

This is to certify that I have read this thesis entitled **"Secure Cryptographic Schemes Using the Graph Theory"** and I found that this thesis is qualified for debate.

Signature:

Name: Dr. Luay Abd Al−Haine Al−Swidi

Title: Professor

Address:

Date: / / 2021

Dedication

Give this humble effort

To my family

To everyone who supported me and

Gave me support

Acknowledgments

Praise be to Allah, the Lord of the Worlds, and peace and blessings be upon the noblest creation and messengers of our Prophet Muhammad and on the house of the good and pure and his companions.

The first and last thanks to the Almighty who surrounded me with his divine care and pleased me my command

I cannot finish the preparation of my research, but to thank you very much and a tribute to "my Supervisor" (Prof. Dr. Ahmed Abd Ali Omran) for his supervision on my research and the scientific sponsorship and guidance and sound opinions through the search process, God rewarded me all the best.

I would like to thank the faculty members for their good opinions that contributed to the development of the idea of research. The closing of my thanks and appreciation and respect for those who lend a helping hand to me and I missed them God grants success...

Abstract

This work proposed new versions of the symmetric encryption schemes based on the Cartesian product graph. These versions are Cartesian are product graphic **CPG** encryption scheme based on English alphabet values, **CPG** encryption scheme based on ASCII values and **CPG** polyalphabetic encryption scheme. The message is chosen as an English word or an English sentence .The ciphertexts of the original messages are considered as the **CPG** which are sent to the receiver by sender. Several experimental results of the proposed **CPG** encryption schemes are discussed. The security considerations of the proposed **CPG** encryption scheme are determined.

List of Content:

Chapter One

Introduction

1.1 General Introduction

Graph theory is a branch of applied mathematics, which deals the problem with the help of graph. It is very easy to deal a problem graphically, as compare to theoretically. In mathematics and computer science, graph theory is the study of graphs, which are mathematical structures used to model pair-wise relations between objects.

Cryptography was concerned totally with message encryption, i.e., the conversion of message from an intelligible form into unintelligible one and reverse again at the other end, rendering it unreadable by an unauthorized person without the knowledge of secret key (decryption key). In the modern age of technology cryptography is becoming a more and more central topic within in mathematics, computer science and others. As there is a need for more secure cryptographic schemes, the application of graph theory is going to increase for the development of secure encryption algorithms. Cryptography have proposed a selective encryption mechanism using message specific key and spanning tree concept of graph theory. The mechanism provides the protection of privacy in communication as it avoids the formation of self-loops and parallel edges and key is exchanged only among the authenticated persons only. Graph theory has a great contribution in the development of various encryption techniques. In this work we a scheme for secure communication using graph, is proposed.

Security mechanisms that rely on cryptography are an integral part of almost any computer system. Users rely on cryptography every time, they access a secured website. Cryptographic methods are used to enforce access control in multi-user operating systems, and to prevent thieves from extracting trade secrets from stolen laptops. Software protection methods which is employed using the encryption, authentication, and other tools to prevent copying. Because of this, breaking a cryptosystem is not restricted to break the underlying cryptographic algorithms; usually it is far easier to break the system as a whole.

Cryptanalysis refers to the art and science of analyzing information systems in order to study the hidden aspects of the systems. Cryptanalysis is used to breach cryptographic security systems and gain access to the contents of encrypted messages, even if the cryptographic key is unknown.

1.2 Previous Studies

In1976 , BONDY, et al [1], A table can be created by taking the Cartesian product of a set of rows and a set of columns. If the Cartesian product rows × columns is taken, the cells of the table contain ordered pairs of the form (row value, column value). In1990, Rivest, Ronald L [2], Cryptography is the practice and study of techniques for secure communication in the presence of adversarial behavior. In1990, Warner,S [3], terms of set-builder notation. In 2005, Bellare, Mihir; Rogaway, Phillip [4], More generally, cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages. In 2005, Ioannis,Sfyrakis ,et [5], Anonymous credential schemes are a cryptographic building block that enables the certification of data structures and prove properties over their

representations without disclosing the innards of their data structures in zero-knowledge. The graph signature (GRS) scheme enables the certification and proof methods to sign infrastructure topologies represented as graph data structures and use zero-knowledge to prove properties over their certificates. They represent a powerful privacy-preserving method that proves properties over a signed topology graph to another party without disclosing the blueprint of its topology. In 2012, Comtet, L [6],in mathematics, specifically set theory, the Cartesian product of two sets A and B, denoted $A \times B$, is the set of all ordered pairs (a, b) where a is in A and b is in B. $A \times B = \{ (a, b) \mid a \in A \text{ and } b \in B \}$. In 2012, Clark, W, E .and Suen, S [7], Cartesian plane is the main historical example in analytic geometry. In order to represent geometrical shapes in a numerical way, and extract numerical information from shapes' numerical representations, René Descartes assigned to each point in the plane a pair of real numbers, called its coordinates. Usually, such a pair's first and second components are called its x and y coordinates, respectively. The set of all such pairs (i.e., the Cartesian product $\mathbb{R}\times\mathbb{R}$, with $\mathbb{R}$ denoting the real numbers) is thus assigned to the set of all points in the plane. In 2013,Thea Peacock, Zhe Xia [8], Modern cryptography exists at the intersection of the disciplines of mathematics, computer science, electrical engineering, communication science, and physics. Applications of cryptography include electronic commerce, chip-based payment cards, digital currencies, computer passwords, and military communications Modern cryptography is heavily based on mathematical theory and computer science practice, cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in actual practice by any adversary. While it is theoretically possible to break into a well-designed system, it is infeasible in actual practice to do so. Such schemes, if well designed, are therefore termed "computationally secure"; theoretical advances, e.g., improvements in integer factorization algorithms, and faster computing technology require these designs

to be continually reevaluated, and if necessary, adapted. There exist information-theoretically secure schemes that provably cannot be broken even with unlimited computing power, such as the one-time pad, but these schemes are much more difficult to use in practice than the best theoretically breakable but computationally secure schemes. In 2015, Pierre Lestringant [9], Softwares use cryptographic algorithms to secure their communications and to protect their internal data. However the algorithm choice, its implementation design and the generation methods of its input parameters may have dramatic consequences on the security of the data it was initially supposed to protect. Therefore to assess the security of a binary program involving cryptography, analysts need to check that none of these points will cause a system vulnerability. It implies, as a first step, to precisely identify and locate the cryptographic code in the binary program. Since binary analysis is a difficult and cumbersome task, it is interesting to devise a method to automatically retrieve cryptographic primitives and their parameters. In 2015, Shubham Agarwall [10], Cryptography is the study of techniques for ensuring the security and authentication of the information. Public-key encryption schemes are secure only if the authenticity of the public-key is assured. Graph theory plays an important role in the field of cryptography for developing security schemes. In 2018, Menezes, A. J. Alfred [11], Various aspects in information security such as data confidentiality, digit integrity, authentication, and non-repudiation. In 2018, P. Amudha [12], 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.

1.3 The Problem Statement

This work proposed new versions of the symmetric encryption schemes. This proposition employed using the ***CPG*** to increase the security level of these schemes. The security determined based on encrypted the message using ***CPG*** and sending it to the receiver.

1.4 The Aim of This Research

The research aims to develop the symmetric the encryption schemes, using the graph theory concepts, especially the Cartesian product graph to increase the security.

1.5 The Structure of this Research

Chapter 1. includes the general introduction.

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

Chapter 3. includes, the Cartesian Product Graph, for Encryption Schemes.

Chapter 4. includes the Cartesian Product Graph for Polyalphabetic Encryption Scheme.

Chapter 5. More examples.

Chapter 6. Conclusions and future works.

Chapter Two

Mathematical Background to the Research

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)[13, p.1].

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

$$V = \{ A , B , C , D , E \} \text{ and } E = \{AB , AC , BD , CD , DE\}$$

is shown in Figure (2.1).

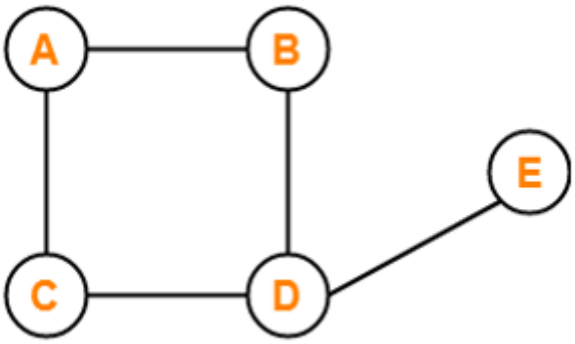

Figure 2.1. A graph G = (V, E).

Definition 2.2.2.(Order and Size of Graph). 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[13, p˙1].

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[13, p˙1].

Definition 2.2.4. (Parallel Edges). The edges connecting the same pair of vertices are called multiple edges or parallel edges[13, p˙1].

Definition 2.12.5. (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) [13, p˙1].

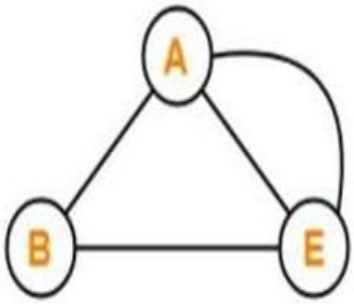

Figure 2.2. A multigraph.

Definition 2.2.6.(Null graph). 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[13, p: 5].

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[13, p: 4].

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 $\delta(G)$ and Δ (G), respectively

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

$d(v_1)+d(v_2)+\ldots+d(v_5)=14$= twice the number of edges[13, p: 4].

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[13, p:14].

Definition 2.1.8. (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[13, p:14].

Definition 2.2.10. (Trivial walk). A trivial walk is one containing no edge. Thus for any vertex v of G, W ≡ v gives a trivial walk. It has length 0[13, p:14].

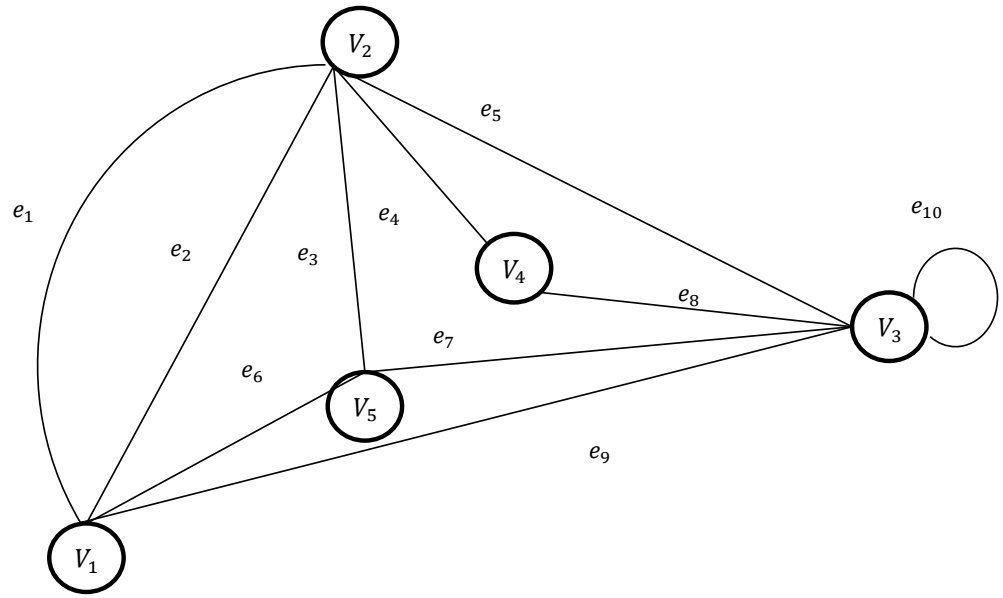

Figure 2.4. Defined Walks.

W_1= $v_1e_1v_2e_5v_3e_{10}v_3$

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 Figure (2.4). [13, p:14].

Definition 2.2.11. (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 [13, p:15].

Definition 2.2.12. (Path). If the vertices $v_0, v_1, \ldots, v_k$ of the walk

$$W\equiv v_0e_1v_1e_2v_2 \ldots\ldots e_kv_k$$

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,

$$W_4 = v_2v_4v_3v_5v_1$$

is a path in the graph G as shown in Figure (2.4). [13, p:15].

Definition 2.2.13. (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

[13, p:15].

Definition 2.2.14. (Connected graph). A graph G is called connected if every two of its vertices are connected as shown in Figure (2.5) [13, p:15].

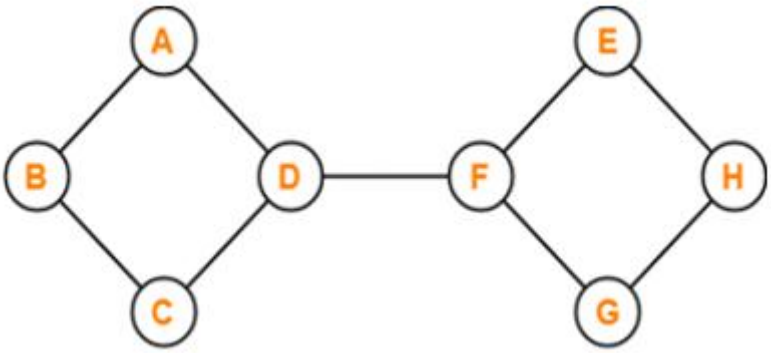

Figure 2.5. A connected graph.

A graph that is not connected is called disconnected as shown in Figure (2.6).

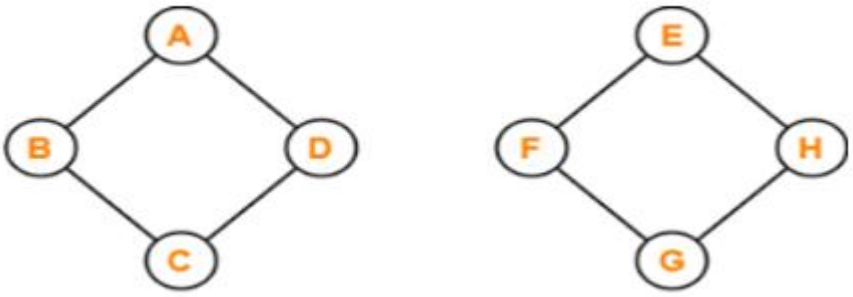

Figure2.6. A disconnected graph.

2.3 The Operations on Graphs

Definition 2.3.1. The union of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is another graph $G_3 = (V_3, E_3)$ denoted by $G_3 = G_1 \cup G_2$, where vertex set $V_3 = V_1 \cup V_2$ and the edge set $E_3 = E_1 \cup E_2$[13, p.12].

Definition 2.3.2. The intersection of two graphs G_1 and G_2 is denoted by $G_1 \cap G_2$ which is a graph G_4 consisting only of those vertices and edges that are in both G_1 and G_2[13, p.12].

. **Definition 2.3.3.** The ring sum of two graphs G_1 and G_2 denoted by $G_1 \oplus G_2$, is a graph consisting of the vertex set $V_1 \cup V_2$ of edges that are either in G_1 or G_2; but not in both[13, p.12].

2.4 Introduction to Cryptography

Definition 2.4.1.(Cryptography) is the design and analysis of mathematical techniques that enable secure communications in the presence of adversaries[14].

Definition 2.4.2.(Cryptosystem),A cryptographic system is specifically a set of methods (algorithms) for computing (implementing) the encryption and decryption[14].

Definition 2.4.3.(Cryptanalysis) is the study of analyzing cryptosystem in order to study the hidden aspects of the systems[14].

Definition 2.4.4. (Plaintext), the information which we want to the protect from other people (attackers) [14].

Definition 2.4.5. (Security). It mean that the difficulty to know the information which transferred over the channel easily[14].

Chapter Three

The Cartesian Product Graph for Encryption Schemes

3.1 Introduction

In this chapter, the definition of the Cartesian product graph (CPG) has been presented. The Cartesian product 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 Cartesian Product Graph

The concept of the Cartesian product graph (CPG) is defined as follows.

Definition 3.2.1. The Cartesian product of two simple graphs G and H is the graph $K = G \times H$ with

$$V(K) = V(G) \times V(H) = \{ (u, v) : u \in V(G) \text{ and } v \in V(H)\},$$

in which vertices (u, v) and (u', v') are adjacent if and only if either

i. $u = u'$ and v, v' are adjacent in H, or
ii. $v = v'$ and u, u' are adjacent in G.[13]

For instance, a simple graph G has four vertices A, B, C, D and four edges, whereas a null graph H has one vertex only. The Cartesian product graph G×H of the graphs G and H is computed by

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

Figure (3.1) shows as the CPG of graphs G and H.

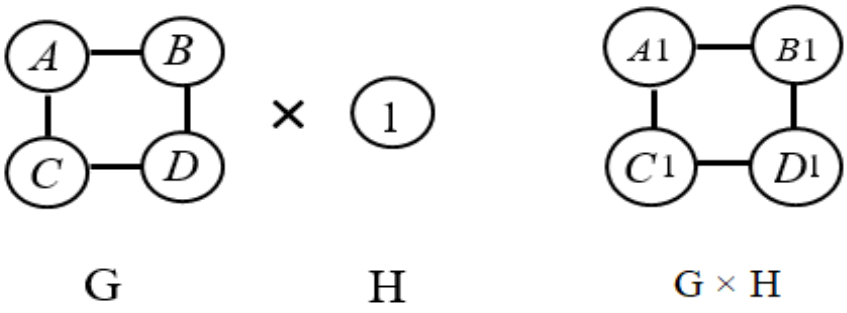

Figure 3.1. The Cartesian product graph G×H of a simple graph G and a null graph H.

Another example of the CPG can be formed based on two path graphs. Let G is a path graph consists of four vertices and four edges and H is also a path graph has two vertices and one edges. The G×H of the path graphs G and H, as shown in Figure (3.2), is computed by

G×H = { (A , 1) , (B , 1) , (C , 1) , (D , 1) , (A , 2) , (B , 2) , (C , 2) , (D , 2)}.

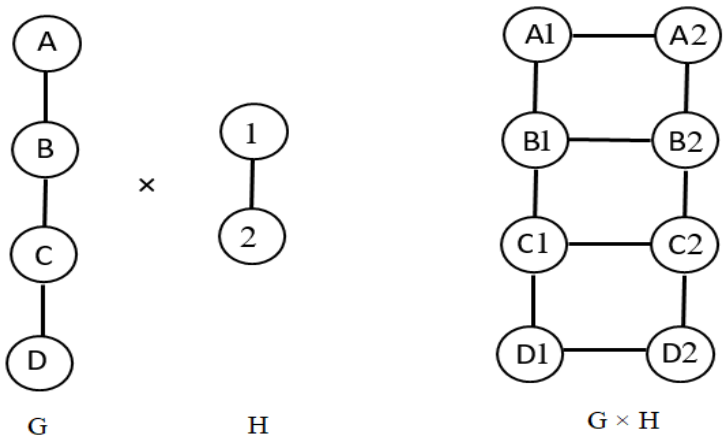

Figure 3.2. The Cartesian product graph G×H of path graphs G and H.

3.3 The Cartesian Product Graph For Encryption Schemes

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

3.3.1. Cartesian Product Graph for Encryption Schemes: Case I.

Let m be a plaintext can be given as an English word or an English sentence. This word or sentence has some English letters. Based on the English alphabet Table (3.1) of these letters, one can convert the letters in the plaintext m into numbers.

Table 3.1. English alphabet Table

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
0	1	2	3	4	5	6	7	8	9	10	11	12	13
O	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**		
14	15	16	17	18	19	20	21	22	23	24	25		

The length of m is equal to K. So, adding K to all of these numbers one by one has been done. For instance, if the letters of plaintext m is $m_1, m_2, \ldots, m_K$ then

$$\#m_1+K \pmod{26} \equiv a_1, \#m_2 + K \pmod{26} \equiv a_2, \ldots, \#m_K+ K \pmod{26} \equiv a_K,$$

where $\#m_i$ are numbers in Table (3.1). In other words, it is possible to write these numbers in two lists

$$\text{List 1: } \{ a_1, a_2, \ldots, a_l\} \quad \text{and} \quad \text{List 2: } \{ a_{l+1}, a_{l+2}, \ldots, a_K \}.$$

These lists can be represented by two graphs G and H. These graphs, namely G and H, are used to form the Cartesian product graph (CPG), $G \times H$. The ciphertext C of a message m is computed by

$$C = G \times H = \{(a_1, a_{l+1}), (a_1, a_{l+2}), \ldots,\{(a_1, a_K)\}$$

and considered as the CPG to send to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

$$\text{List 1: } \{ a_1, a_2, \ldots, a_l \} \quad \text{and} \quad \text{List 2: } \{ a_{l+1}, a_{l+2}, \ldots, a_K \}.$$

Since the length of m is K that is determined based on the elements (vertices) in the tow lists, so he/ she computes the following computations:

$$a_1 - K \pmod{26} \equiv \#m_1,\ a_2 - K \pmod{26} \equiv \#m_2,\ \ldots,\ a_K - K \pmod{26} \equiv \#m_K.$$

Based on the English alphabet Table (3.1), the previous numbers converted into

$$\#m_1 \rightarrow m_1, \#m_2 \rightarrow m_2, \ldots, \#m_l \rightarrow m_l \text{ and } \#m_{l+1} \rightarrow m_{l+1}, \ldots, \#m_K \rightarrow m_K.$$

Thus, the original plaintext is recovered by $m = m_1 m_2 \ldots m_K$.

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

Suppose m is the plaintext that is given by the sentence **Hi Ali**. Based on the English alphabet Table (3.1) of the letters, one can convert the letters in the plaintext m into numbers. So,

$$H \rightarrow 7,\ i \rightarrow 8,\ A \rightarrow 0,\ \ l \rightarrow 11,\ i \rightarrow 8.$$

The length K of m is equal to 5. Adding K to all of these numbers one by one gives us

$$7 + 5 = 12,\ 8 + 5 = 13,\ \ 0 + 5 = 5,\ 11 + 5 = 16,\ 8 + 5 = 13.$$

In other words, it is possible to write these numbers in two lists

List 1: {12, 13} and List 2: {05, 16, 13}.

These lists can be represented by two graphs, say two path graphs as shown in Figure (3.3). The graph *G* and *H* are used to form the CPG, G × H, is computed by

$$G \times H = \{(12,05), (12,16), (12,13), (13,05), (13,16), (13,13)\}.$$

The $G \times H$ is shown in Figure (3.3).

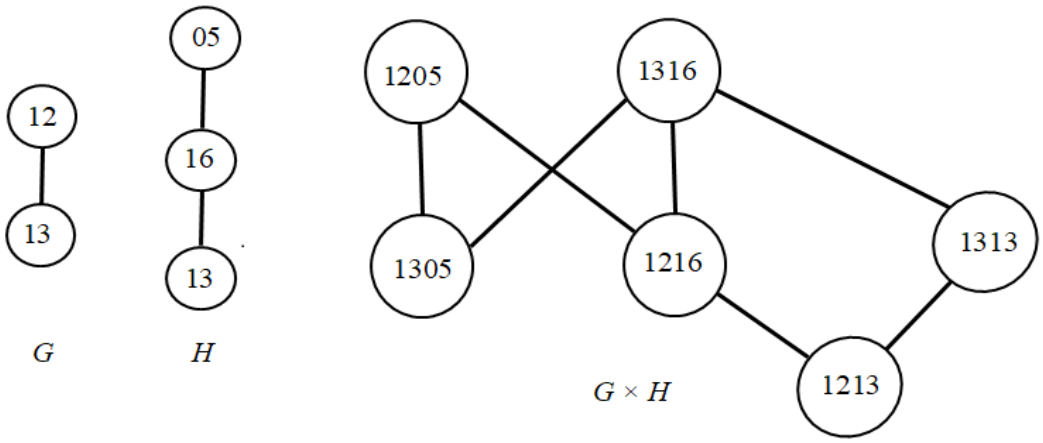

Figure 3.3. The ciphertext *C* as Cartesian product graph $G \times H$ of path graphs *G* and *H* with 2 and 3 vertices respectively.

The ciphertext *C* of a message *m* is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: {12, 13} and List 2 : {05,16,13}.

Since the length of m is $K = 5$, so he/ she computes the following computations:

$$12 - 5 = 7,\ 13 - 5 = 8,\ 05 - 5 = 0,\ 16 - 5 = 11,\ 13 - 5 = 8.$$

Based on the English alphabet Table (3.1), the previous numbers converted into

$$7 \rightarrow H,\ 8 \rightarrow i \text{ and } 0 \rightarrow A,\ 11 \rightarrow l,\ 8 \rightarrow i.$$

Thus, the plaintext m is **Hi Ali**.

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

Suppose m is the plaintext that is given by the word **security**. Based on the English alphabet Table (3.1) of the letters, one can convert the letters in the plaintext m into numbers. So,

$$S \rightarrow 18,\ E \rightarrow 4,\ C \rightarrow 2,\ U \rightarrow 20,\ R \rightarrow 17,\text{I} \rightarrow 8,\ T \rightarrow 19,\ Y \rightarrow 24.$$

The length K of m is equal to 8. Adding K to all of these numbers one by one gives us

$$S = 18 + 8 = 00,\ E = 4 + 8 = 12,\ C = 2 + 8 = 10,\ U = 20 + 8 = 02,\ R = 17 + 8 = 25,\ I = 8 + 8 = 16,\ T = 19 + 8 = 01,\ Y = 24 + 8 = 06.$$

In other words, it is possible to write these numbers in two lists

List 1: {00,12, 10} and List 2: {02, 25, 16, 01,06}.

These lists can be represented by two graphs, say two path graphs as shown in Figure (3.4). The graphs G and H are used to form the CPG, $G \times H$, that is computed by

$$G \times H = \{(00,02), (00,25), (00,16), (00,01), (00,06),\ldots,(10,02), (10,25), (10,16), (10,01), (10,06)\}.$$

The $G \times H$ is shown in Figure (3.4).

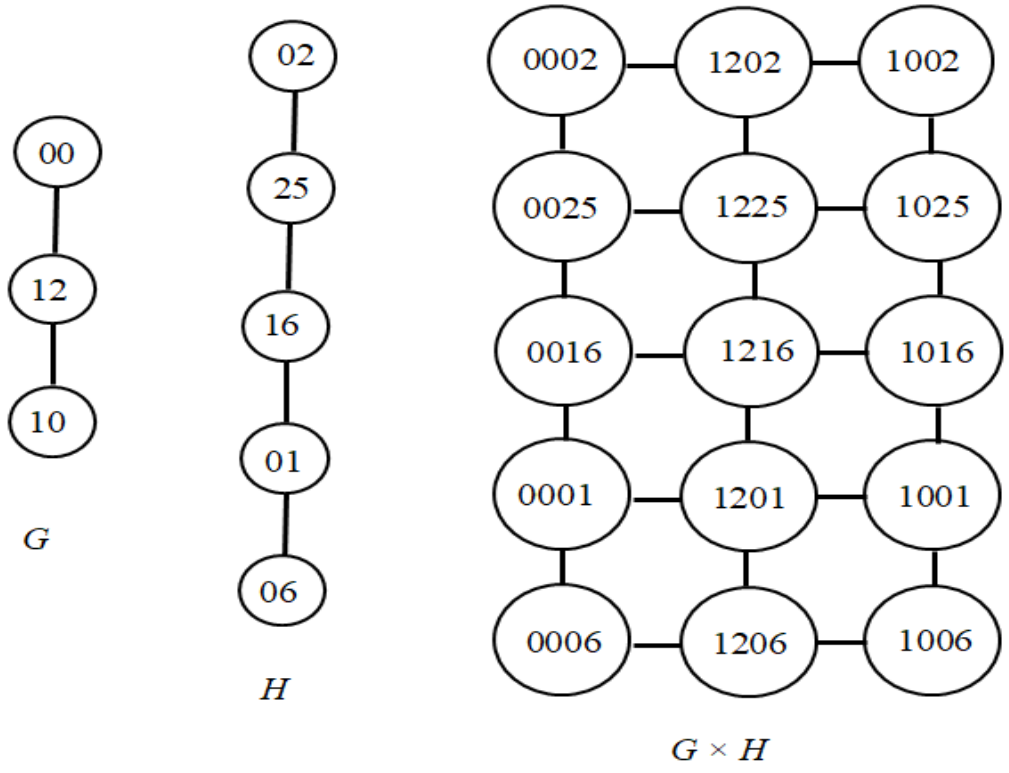

Figure 3.4. The ciphertext C as Cartesian product graph $G \times H$ of path graphs G and H with 4 and 3 vertices respectively.

The ciphertext C of a message m is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: {00,12, 10} and List 2: {02, 25, 16, 01,06}.

Since the length of m is $K = 8$, so he/ she the following computations:
Based on the English alphabet Table (3.1), the previous numbers converted into

$$00 - 8 = 18 \rightarrow S,\ 12 - 8 = 4 \rightarrow E,\ 10 - 8 = 2 \rightarrow C,\ 02 - 8 = 20 \rightarrow U,$$

$$25 - 8 = 17 \rightarrow R,\ 16 - 8 = 8 \rightarrow I,\ 01 - 8 = 19 \rightarrow T,\ 06 - 8 = 24 \rightarrow Y.$$

Thus, the plaintext m is security.

3.4 Cartesian product Graph for encryption schemes: Case II.

The same idea of case I can be applied to encrypt the plaintext m which has the English letters that are represented by numbers of ASCII Table (3.2). 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. So, the letters of the plaintext here have been converted into ASCII Table numbers.

The length of m is equal to K. So, adding K to all of these numbers one by one has been done. For instance, if the letters of plaintext m is $m_1, m_2, \ldots, m_K$ then

$$\#m_1 + K \pmod{127} \equiv a_1,\ \#m_2 + K \pmod{127} \equiv a_2,\ \ldots,\ \#m_K + K \pmod{127} \equiv a_K,$$

where $\#m_i$ are numbers in Table (3.2). In other words, it is possible to write these numbers in two lists as shown in case I. These lists can be represented by two graphs

G and *H*. These graphs, namely *G* and *H,* are used to form the Cartesian product graph (CPG), $G \times H$, and send to receiver by sender.

The second user (receiver) receivers the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

$$\text{List 1: } \{ a_1, a_2, \ldots, a_l \} \quad \text{and} \quad \text{List 2: } \{ a_{l+1}, a_{l+2}, \ldots, a_K \}.$$

Since the length of m is K that is determined based on the elements (vertices) in the first list, so he/ she computes the following computations:

$$a_1 - K \pmod{127} \equiv \#m_1,\ a_2 - K \pmod{127} \equiv \#m_2,\ \ldots,\ a_K - K \pmod{127} \equiv \#m_K.$$

Based on the ASSCI Table (3.2), the previous numbers converted into

$$\#m_1 \rightarrow m_1,\ \#m_2 \rightarrow m_2,\ \ldots,\ \#m_l \rightarrow m_l \text{ and } \#m_{l+1} \rightarrow m_{l+1},\ \ldots,\ \#m_K \rightarrow m_K.$$

Thus, the original plaintext is recovered by $m = m_1 m_2 \ldots m_K$.

Table 3.2. English ASCII Table[15].

	Char.	Dec.	Char.	Dec.	Char.	Dec.	Char.
0	Null	32	Space	64	@	96	`
1	Start of heading	33	!	65	A	97	a
2	start of text	34	"	66	B	98	b
3	end of text	35	#	67	C	99	c
4	end of transmission	36	$	68	D	100	d
5	Enquiry	37	%	69	E	101	e
6	Acknowledge	38	&	70	F	102	f
7	Bell	39	'	71	G	103	g
8	Backspace	40	(	72	H	104	h
9	horizontal tab	41	)	73	I	105	i
10	NL line feed, new line	42	*	74	J	106	j
11	vertical tab	43	+	75	K	107	k
12	NP form feed, new page	44	,	76	L	108	l
13	carriage return	45	-	77	M	109	m
14	shift out	46	.	78	N	110	n
15	shift in	47	/	79	O	111	o
16	data link escape	48	0	80	P	112	p
17	device control 1	49	1	81	Q	113	q
18	device control 2	50	2	82	R	114	r
19	device control 3	51	3	83	S	115	s
20	device control 4	52	4	84	T	116	t
21	negative acknowledge	53	5	85	U	117	u
22	synchronous idle	54	6	86	V	118	v
23	end of trans. Block	55	7	87	W	119	w
24	Cancel	56	8	88	X	120	x
25	end of medium	57	9	89	Y	121	y
26	Substitute	58	:	90	Z	122	z
27	Escape	59	;	91	[	123	{
28	file separator	60	<	92	\	124	|
29	group separator	61	=	93	]	125	}
30	record separator	62	>	94	^	126	~
31	unit separator	63	?	95	_	127	Del

Example 3.4.1.1. Study Case II: Encryption Scheme Based on the CPG

Suppose *m* is the plaintext is given by the word *"***ENCRYPTION***"*. Based on the ASCII Table (3.2) of the letters, one can converts these letters of the plaintext *m* into numbers. So,

"→ 34, *E* → 69, *N*→ 78, *C* →67, *R*→82, *Y*→89, *P*→80, *T*→84, I→73, O→79, N→78, "→34.

The length *K* of *m* is equal to 12. Adding *K* to all of these numbers one by one gives us

34 +12=046, 69+12=081, 78+12=090,67+12=079, 82+12=094, 89+12=101, 80+12=092, 84+12=096, 73+12=085, 79+12=091, 78+12=090, 34+12=046.

In other words, it is possible to write these numbers in two lists

List 1: {046, 081, 090, 079, 094, 101} and List 2: {092, 096, 085, 091, 090, 046}.

The ciphertext *C* of a message *m* is considered as the CPG as shown in Figure (3.5)

which is sent to receiver by sender.

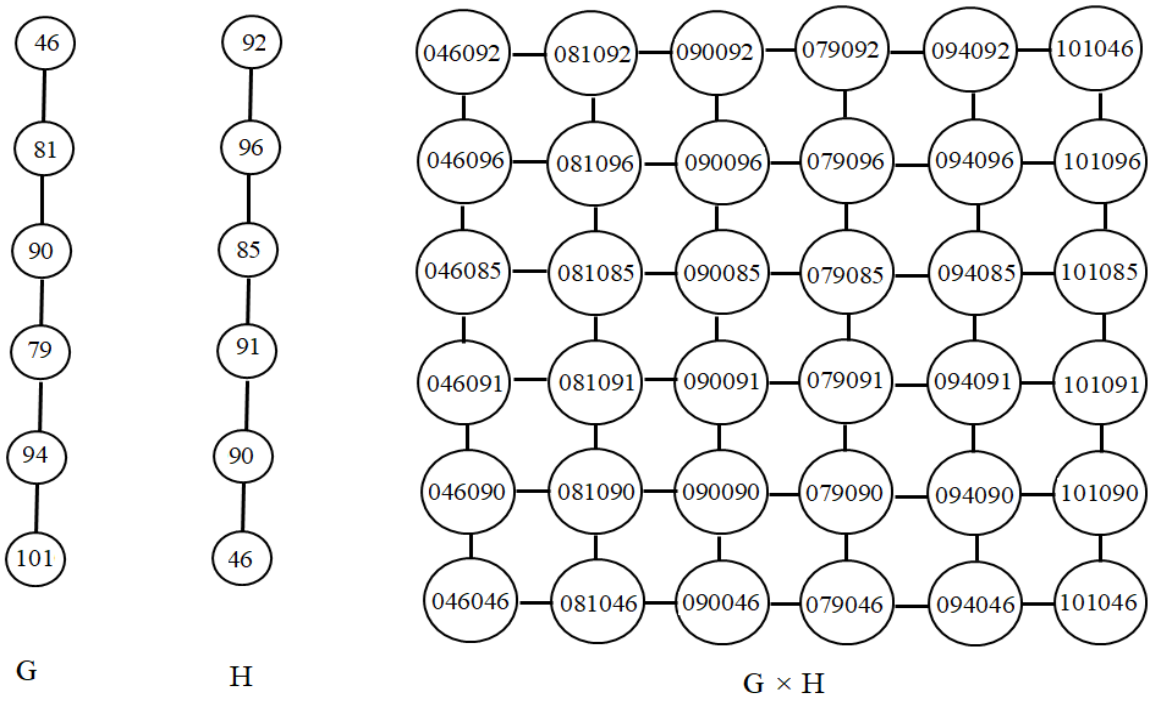

Figure 3.5. The ciphertext *C* as Cartesian product graph *G*×*H* of path graphs *G* and *H* with 6 vertices.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: {046, 081, 090, 079, 094, 101} and List 2: {092, 096, 085, 091, 090, 046}.

Since the length of *m* is *K* = 12, so he/ she the following computations:

46-12=34, 81-12=69, 90-12=78, 79-12=67, 94-12=82, 101-12=89,

92-12=80, 96-12=84, 85-12=73, 91-12=79, 90-12=78, 46-12=34.

Based on the ASCII Table (3.2), the previous numbers are converted into

$"\rightarrow 34$, $E \rightarrow 69$, $N \rightarrow 78$, $C \rightarrow 67$, $R \rightarrow 82$, $Y \rightarrow 89$, $P \rightarrow 80$, $T \rightarrow 84$, I$\rightarrow$73, O$\rightarrow$79, N$\rightarrow$78, "$\rightarrow$34.

Thus, the plaintext *m* is "**ENCRYPTION**".

Example 3.4.1.2. Study Case II: Encryption Scheme Based on the CPG

Suppose *m* is the plaintext that is given by the sentence **Thank you**. Based on the English ASCII Table (3.2) of the letters, one can convert the letters in the plaintext *m* into numbers. So,

$T \rightarrow 084$, $h \rightarrow 104$, $a \rightarrow 097$, $n \rightarrow 110$, $k \rightarrow 107$, $\rightarrow 032$, $y \rightarrow 121$, $o \rightarrow 111$,u$\rightarrow$117.

The length *K* of *m* is equal to 9. Adding *K* to all of these numbers one by one gives us

$T = 084 + 9 = 093$, $h = 104 + 9 = 113$, $a = 097 + 9 = 106$,$n = 110 + 9 = 119$, $k = 107 + 9 = 116$, $= 032 + 9 = 041$, $y = 121 + 9 = 003$,$o = 111 + 9 = 120$, $u = 117 + 9 = 126$.

In other words, it is possible to write these numbers in two lists

List 1: {093, 113,106, 119, 116, 041} and List 2: {003, 120, 126}.

These lists can be represented by two graphs, say two path graphs as shown in Figure (3.6). The graph *G* and *H* are used to form the Cartesian product graph (CPG), G × H, see Figure (3.6).

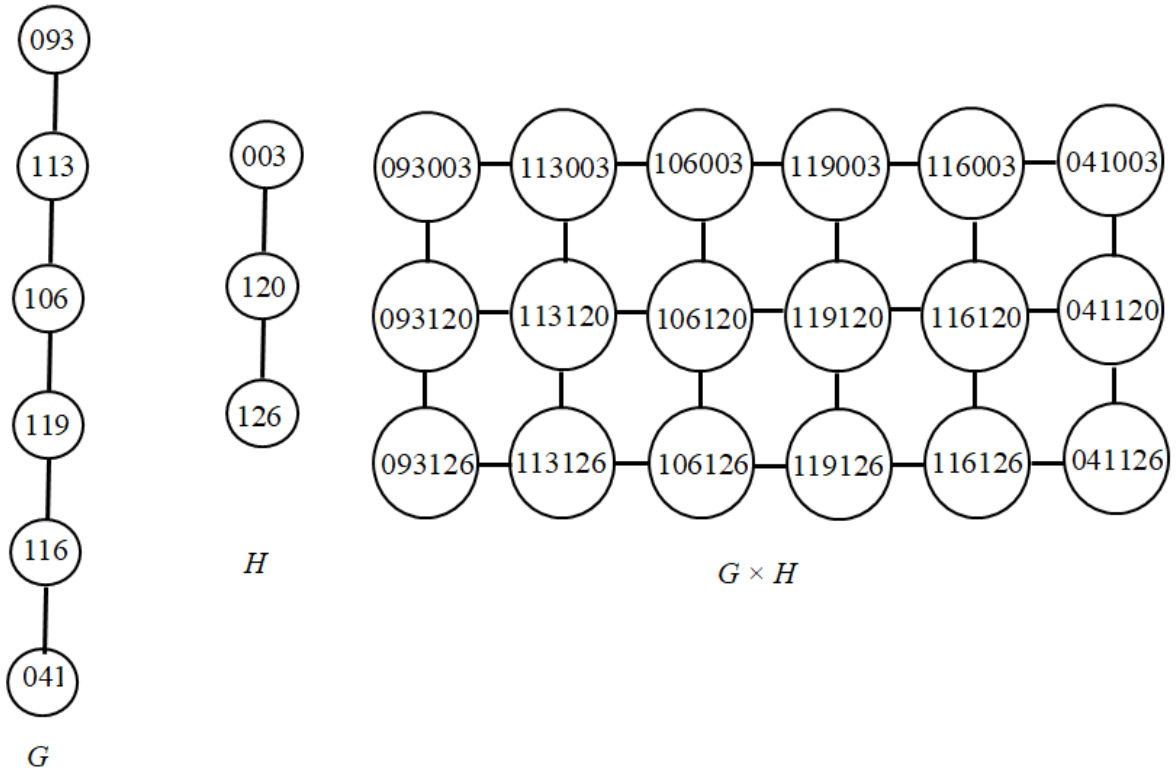

Figure 3.6.The ciphertext *C* as Cartesian product graph *G*×*H* of path graphs *G* and *H* with 6 and 3 vertices respectively.

The ciphertext C of a message *m* is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: {093, 113,106, 119, 116, 041} and List 2: {003, 120, 126}.

Since the length of m is K = 9, so he/ she the following computations: Based on the ASCII Table (3.2), the previous numbers converted into

$093 - 9 = 084 \rightarrow T$,$113 - 9 = 104 \rightarrow h$,$106 - 9 = 097 \rightarrow a$, $119 - 9 = 110 \rightarrow n$, $116 - 9 = 107 \rightarrow k$, $041 - 9 = 032 \rightarrow$space, $003 - 9 = 121 \rightarrow y$, $120 - 9 = 111 \rightarrow o$,$126 - 9 = 117 \rightarrow u$.

Thus, the plaintext m is **Thank you**.

3.4 The Security considerations of the SCPG Schemes

The proposed symmetric encryption schemes based on the CPG is a more secure in compare with other symmetric encryption schemes. With CPG, the ciphertext has been computed as the CPG by depending on two graphs G and H. These graphs are created based on the digit of the plaintext and a secret key K. The security considerations of new proposed SCPG schemes depended on random generating the graphs G and H that the attackers want to know them if they determine the ciphertext is computed as CPG. So, Eve needs to guess the vertices of graphs G and H. Therefore, in case I, there are 26 possible probabilities to form a graph G and 26 possible probabilities to create a graph H. Thus, the total probability of all cases is 676, two of them are correct. In other words, one correct case to create a graph G and one correct case to form a graph H.

Also, Eve needs to determine the probability of labeled number of each vertex in graphs G and H as well. So, in the study case I, first vertex in graph G takes 26 labeled possible probabilities. The probability of all possible labeled vertices is equal to 676. While, the graph H has three vertices, so the probability of all possible labeled vertices is equal to 17576. Hence, the total probability to choose correct graphs G and H is equal to 18252.

Whereas, on study case II, the probability of all possible labeled vertices of graph G is equal to 127^6 = 4195872914689 and the probability of all possible labeled vertices of graph H is equal to 4195872914689. The total probability to choose correct graphs G and H is equal to 8391745829378.

So, if the adversaries know a ciphertext of a plaintext m is computed by G × H and represented and sent as the CPG they need to guess more and more probability cases to generate the graphs G and H. Hence, it is more secure to recover the original message among all possible probabilities cases.

Chapter Four

The Cartesian Product Graph for Polyalphabetic Encryption Scheme

4.1 Introduction

In this chapter, the Cartesian product graph is used to give alternative modified polyalphabetic encryption schemes. Two types of symmetric polyalphabetic encryption schemes have been proposed. First one based of the

English alphabet values, whereas, 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 as same as of the CPG schemes that are proposed in Chapter (3).

4.2 The CPG 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 a 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.

Let m be a plaintext can be given as an English word or an English sentence. This word or sentence has some English letters. Based on the English alphabet Table (3.1) of these letters, one can convert the letters in the plaintext m into numbers.

It can work with alphabet table and some rules are putting on the key. So, putting $K(r_1, r_2$ or $r_k)$ to all of these numbers one by one has been done. For instance, if the letters of plaintext m is $m_1, m_2, \ldots, m_K$ then

$\#m_1$ it move according to the $r_1 \pmod{26} \equiv a_1$, $\#m_2$ it move according to the $r_2 \pmod{26} \equiv a_2, \ldots, \#m_K$ it move according to the $m_i \pmod{26} \equiv a_K$,

where $\#m_i$ are numbers in Table (3.1). In other words, it is possible to write these numbers in two lists

$$\text{List 1: } \{ a_1, a_2, \ldots, a_l\} \quad \text{and} \quad \text{List 2: } \{ a_{l+1}, a_{l+2}, \ldots, a_K \}.$$

These lists can be represented by two graphs *G* and *H*. These graphs, namely *G* and *H*, are used to form the Cartesian product graph (CPG), $G \times H$. The ciphertext C of a message *m* is computed by

$$C = G \times H = \{(a_1, a_{l+1}), (a_1, a_{l+2}), \ldots,\{(a_1, a_K)\}$$

and considered as the CPG to send to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

$$\text{List 1: } \{ a_1, a_2, \ldots, a_l\} \quad \text{and} \quad \text{List 2: } \{ a_{l+1}, a_{l+2}, \ldots, a_K \}.$$

Since some rules of *m* is *K* that is determined based on the elements (vertices) in the first list, so he/ she computes the following computations:

a_1 it moves in the opposite direction according to the $r_1 \pmod{26} \equiv \#m_1$, a_2 it moves in the opposite direction according to the $r_2 \pmod{26} \equiv \#m_2$, ..., a_K it moves in the opposite direction according to the $r_i \pmod{26} \equiv \#m_K$.

Based on the English alphabet Table (3.1), the previous numbers converted into

$$\#m_1 \rightarrow m_1, \#m_2 \rightarrow m_2, \ldots, \#m_l \rightarrow m_l \text{ and } \#m_{l+1} \rightarrow m_{l+1}, \ldots, \#m_K \rightarrow m_K.$$

Thus, the original plaintext is recovered by $m = m_1 m_2 \ldots m_K$.

Example 4.2.1. The CPG for Polyalphabetic Encryption Scheme Based on English Alphabet Values

Suppose m is the plaintext that is given by the word **"Security"**. Based on the English alphabet Table (3.1) of the letters, one can convert the letters in the plaintext m into numbers. So,

$$S \rightarrow 18, E \rightarrow 4, C \rightarrow 2, U \rightarrow 20, R \rightarrow 17, \mathrm{I} \rightarrow 8, T \rightarrow 19, Y \rightarrow 24.$$

Some rules on the key K are determined by

1- Shift first letters three positions to its right.

2- Shift the second letters four positions to its right.

3- Shift the third letters seven positions to its right.

In more details, the letter s moves into three positions to its right to become v, e letter moves to four positions to its right to become i, the letter c moves to seven positions to its right to become j. Repeating the key process for all letters of the word as follows.

$$sec \quad uri \quad ty$$

$$vij \quad xvp \quad wc$$

The letters of the encoded word "*vijxvpwc"* are divided into two lists

$$\text{List 1: } \{v, i, j\} \quad \text{and} \quad \text{List 2: } \{x, v, p, w, c\}.$$

These lists are represented by two graphs G and H which are path graphs as shown in Figure (4.1). The graphs G and H are used to form the Cartesian product graph (CPG) which is computed by

$$G \times H = \{(v, x),(v, v),(v, p),(v, w),(v, c),\ldots,(j, c)\},$$

see Figure (4.1).

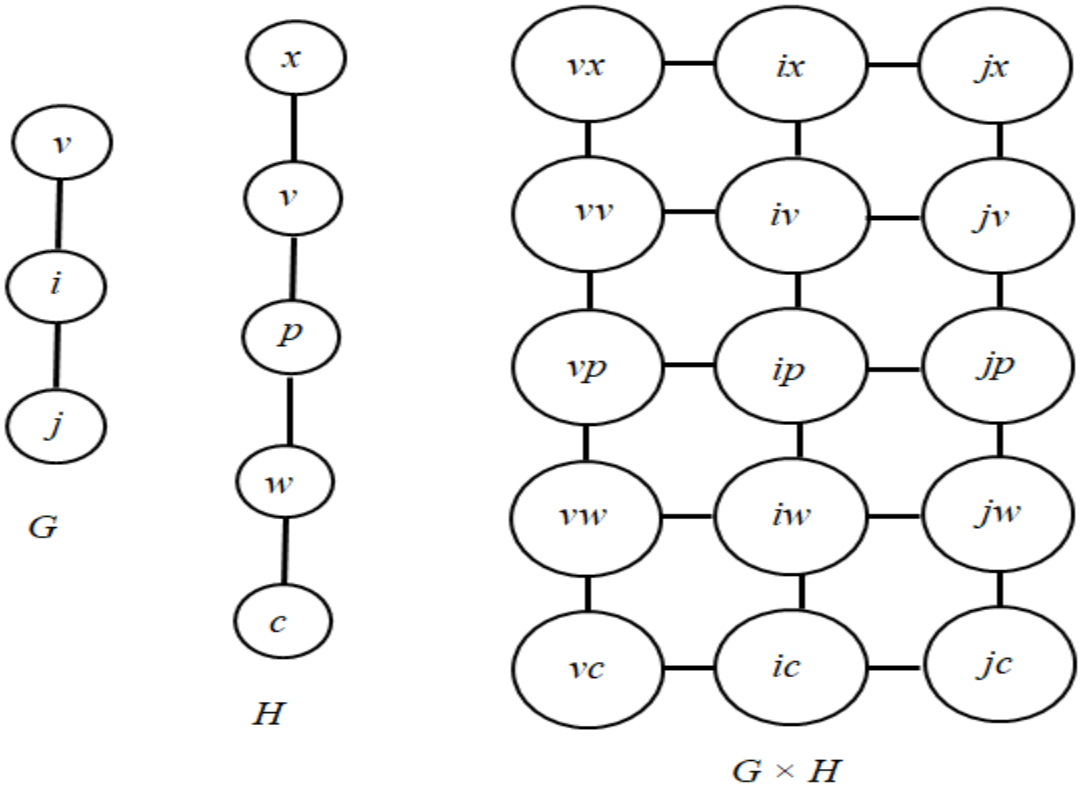

Figure 4.1. The CPG of path graphs G and H have 3 and 5 vertices respectively.

The ciphertext C of a message m is considered as the CPG which is sent to receiver by sender. The second user (receiver) receivers the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: $\{v, i, j\}$ and List 2: $\{x, v, p, w, c\}$.

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

1. Shift first letters three positions to its left.
2. Shift the second letters four positions to its left.
3. Shift the third letters seven positions to its left.

With more details, the letter v moves to three positions to its left to become s, theletter i moves to four positions to its left e, the letter j moves to seven

positions to its left *c*. Repeating the key process for all letters of the word and based on the English alphabet Table (3.1), the encoded word

vijxvpwc

becomes

security

Thus, the plaintext *m* is security.

Example 4.2.2. The CPG for Polyalphabetic Encryption Scheme Based on English Alphabet Values

Suppose *m* is the plaintext that is given by the word **school**. Based on the English alphabet Table (3.1) of the letters, one can convert the letters in the plaintext *m* into numbers. So,

$$S \rightarrow 18,\ C \rightarrow 2,\ h \rightarrow 7,\ o \rightarrow 14, o \rightarrow 14,\ l \rightarrow 11.$$

K of *m* is equal to (sky). Adding *K* to all of these numbers one by one gives us

s= 18, k=10, y=24

$$S = 18 + 18 = 10,\ C = 2 + 10 = 12,\ h = 7 + 24 = 05,$$

$$o = 14 + 18 = 06,\ \ o = 14 + 10 = 24,\ l = 11 + 24 = 09.$$

In other words, it is possible to write these numbers in two lists

List 1: {10, 12, 05} and List 2: {06, 24, 09}.

These lists can be represented by two graphs, say two path graphs as shown in Figure (4.2). The graph *G* and *H* are used to form the Cartesian product graph (CPG), G × H, see Figure (4.2).

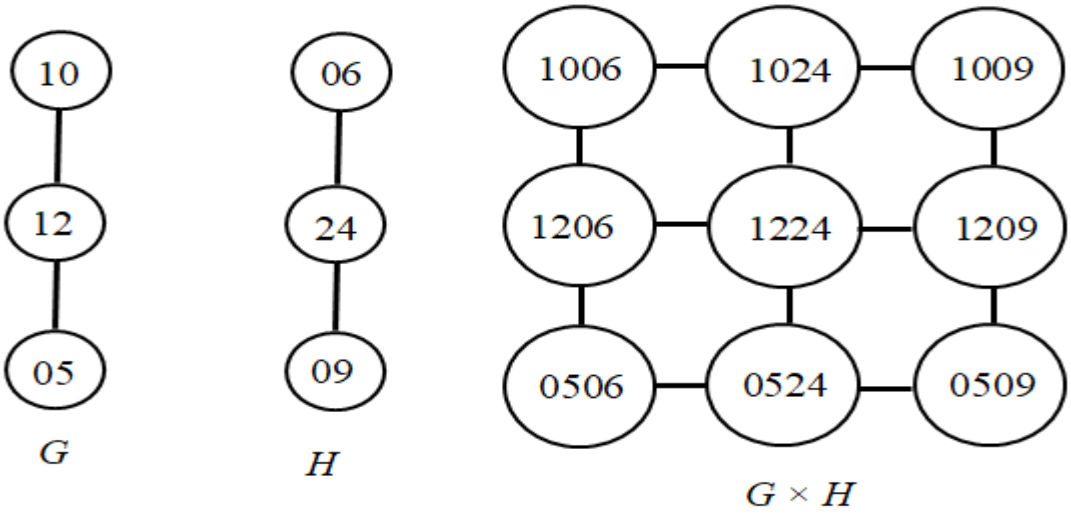

Figure 4.2. The CPG of path graphs *G* and *H* have 3 vertices respectively.

The ciphertext C of a message *m* is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: { 10,12, 05} and List 2: {06, 24, 09}.

Since the length of *m* is *K* =(18, 10, 24) with respectively , so he/ she the following computations :

Based on the English alphabet Table (3.1), the previous numbers converted into

$$10–18 = 18 \rightarrow S,\ 12 – 10 = 02 \rightarrow c,\ 05 – 24 = 7 \rightarrow h,$$

$$06 – 18 = 14 \rightarrow o,\ 24 – 10 = 14 \rightarrow o,\ 09\text{- } 24=11 \rightarrow l.$$

Thus, the plaintext *m* is **school**.

Example 4.2.3. The CPG for Polyalphabetic Encryption Scheme Based on ASCII Values

Suppose *m* is the plaintext that is given by the word **math**. Based on the ASCII Table (3.2) of the letters, one can convert the letters in the plaintext *m* into numbers. So,

$$m \rightarrow 109,\ a \rightarrow 097,\ \mathrm{t} \rightarrow 116,\ h \rightarrow 104.$$

Some rules on the key *K* are determined by

1. Shift first letters one positions into up.
2. Shift the second letters four positions into dawn.

In more details, the letter *m* moves one position into up to become *l*, *a* letter moves four positions into dawn to become *e*, the letter *t* moves one position into up to become *s*. *h* letter moves four positions into dawn to become *l*.

$$ma \rightarrow le \rightarrow 108,\ 101 \text{and} th \rightarrow sl \rightarrow 115,\ 108.$$

In other words, it is possible to write these numbers in two lists

$$\text{List 1: } \{108, 101\} \quad \text{and} \quad \text{List 2: } \{115, 108\}.$$

These lists can be represented by two graphs, say two path graphs as shown in Figure (4.3). The graph *G* and *H* are used to form the Cartesian product graph (CPG), G × H, see Figure (4.3).

$$C = G \times H = \{(108, 115), (108, 108), (101, 115), (101, 108)\}.$$

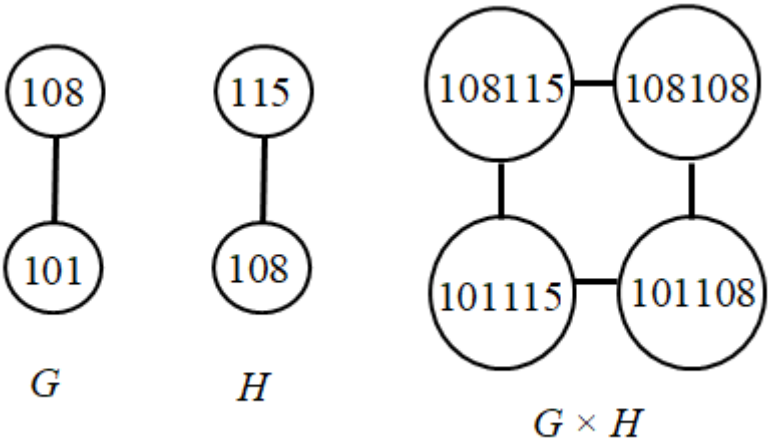

Figure 4.3. The CPG of path graphs *G* and *H* have 2 vertices respectively.

The ciphertext C of a message *m* is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: {108, 101} and List 2: {115, 108}.

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

1. Shift first letters one positions into dawn.
2. Shift the second letters four positions into up

Based on the key and ASCII values in Table (3.2), the second user performs the following computations:

107, 097 116, 104

ma th

Thus, the plaintext *m* is **math**.

Example 4.2.4. The CPG for Polyalphabetic Encryption Scheme Based on ASCII Values

Suppose *m* is the plaintext that is given by the word **mathematical**. Based on the ASCII Table (3.2) of the letters, one can convert the letters in the plaintext *m* into numbers. So,

$m \rightarrow 109$, $a \rightarrow 097$, t $\rightarrow 116$, $h \rightarrow 104$, $e \rightarrow 101$, $m \rightarrow 109$, a $\rightarrow 097$,

$t \rightarrow 116$, $i \rightarrow 105$, $c \rightarrow 099$, a $\rightarrow 097$, $l \rightarrow 108$.

The shared secret key *K* is determined as the word **now**. Adding *K* to all of these numbers one by one gives us

mat hem ati cal

m+n→109+110=092, a+o→097+111=081, t+w →116+119=108,

h+n →104+110=087, e+o →101+111=085, m +w →109+119=101,

a+n →097+110=080, t+o →116+111=100, i+w →105+119=097,

c+n →099+110=082, a+o →097+111=081, l+w →108+119=100.

In other words, it is possible to write these numbers in two lists

List 1:{092, 081, 108, 087, 085, 101, 080, 100} and List 2: {097, 082, 081, 100 }.

These lists can be represented by two graphs, say two path graphs as shown in Figure (4.4). The graph *G* and *H* are used to form the Cartesian product graph (CPG), G × H, see Figure (4.4).

$$C = G \times H = \{(092, 097), \ldots, (100, 100)\}.$$

The ciphertext C of a message *m* is considered as the CPG which is sent to receiver by sender.

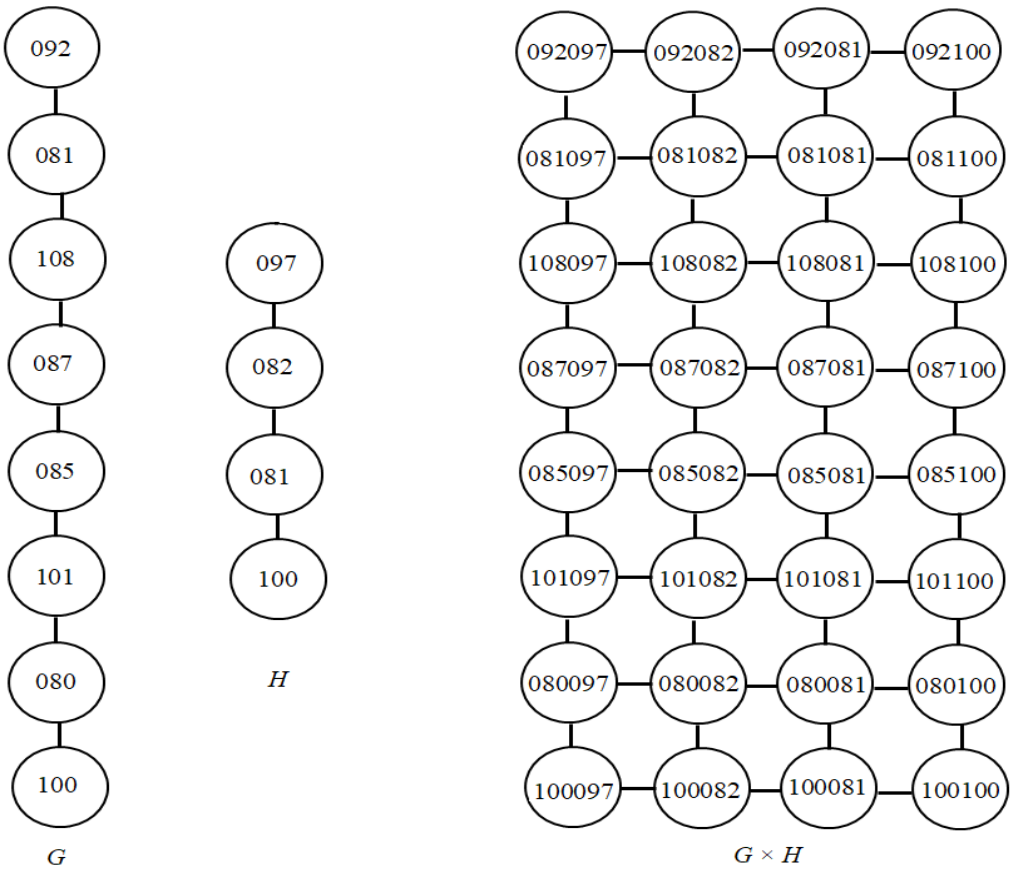

Figure 4.4. The CPG of path graphs *G* and *H* have 8 and 4 vertices respectively.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1:{092, 081, 108, 087, 085, 101, 080, 100} and List 2: {097, 082, 081, 100 }.

Based on a shared secret key the word **now** and the ASCII values, second user performs the following computations:

092-110=109→ m, 081-111= 097→a, 108-119=116→t,

087-110=104→ h, 085-111= 101→e, 101-119= 109→m,

080-110=097→ a, 100-111= 116→t, 097-119= 105→i,

082-110=099→ c, 081-111= 097→a, 100-119=108→l.

Thus, the plaintext *m* is a **mathematical**.

Chapter Five

More Examples

Example 5.1: Encryption Scheme Based on the CPG

Suppose m is the plaintext that is given by the word **Babylon**. Based on the English alphabet Table (3.1) of the letters, one can convert the letters in the plaintext m into numbers. So,

$$b \to 1, a \to 0, b \to 1, y \to 24, l \to 11, o \to 14, n \to 13.$$

The length K of m is equal to 7. Adding K to all of these numbers one by one gives us

$$b = 1 + 7 = 08, a = 0 + 7 = 07, b = 1 + 7 = 08, y = 24 + 7 = 05,$$

$$l = 11 + 7 = 18, o = 14 + 7 = 21, n = 13 + 7 = 20.$$

In other words, it is possible to write these numbers in two lists

List 1: {08, 07} and List 2: {08, 05, 18, 21,20}.

These lists can be represented by two graphs, say two path graphs as shown in Figure (5.1). The graphs G and H are used to form the CPG, $G \times H$, that is computed by

$$G \times H = \{(08,08), (08,05), (08,18), (08,21), (08,20),(07,08), (07,05), (07,18), (07,21), (07,20)\}.$$

The $G \times H$ is shown in Figure (5.1).

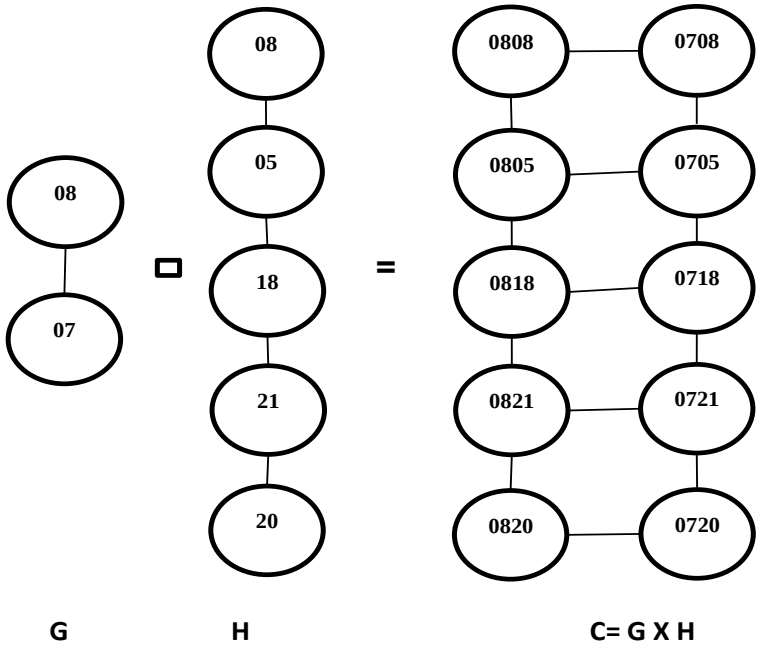

Figure 5.1. The ciphertext C as Cartesian product graph $G \times H$ of path graphs G and H with 2 and 5 vertices respectively.

The ciphertext C of a message m is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: {08, 07} and List 2: {08, 05, 18, 21,20}.

Since the length of m is K =7 , so he/ she the following computations: Based on the English alphabet Table (3.1), the previous numbers converted into

$08 - 7 = 01 \rightarrow b$, $07 - 7 = 00 \rightarrow a$, $08 - 7 = 01 \rightarrow b$, $05 - 7 = 24 \rightarrow y$,

$18 - 7 = 11 \rightarrow l$, $21 - 7 = 14 \rightarrow o$, $20 - 7 = 13 \rightarrow n$.

Thus, the plaintext m is Babylon.

Example 5.2: Encryption Scheme Based on the CPG

Suppose m is the plaintext that is given by the sentence m= help me. Based on the English alphabet Table (3.1) of the letters, one can convert the letters in the plaintext m into numbers. So,

$$h \rightarrow 07,\ e \rightarrow 04,\ l \rightarrow 11,\ p \rightarrow 15,\ m \rightarrow 12,\ e \rightarrow 04.$$

The length K of m is equal to 6. Adding K to all of these numbers one by one gives us

$$h = 07 + 6 = 13,\ e = 04 + 6 = 10,\ l = 11 + 6 = 17,$$

$$p = 15 + 6 = 21,\ m = 12 + 6 = 18,\ e = 04 + 6 = 10.$$

In other words, it is possible to write these numbers in two lists

List 1: {13, 10, 17, 21} and List 2: {18, 10}.

These lists can be represented by two graphs, say two path graphs as shown in Figure (5.2). The graph *G* and *H* are used to form the Cartesian product graph CPG, G × H, that is computed by

G × H={(13,18), (13,10), (10,18), (10,10), (17,18), (17,10), (21,18),(21,10)}.

The G × H shown Figure (5.2).

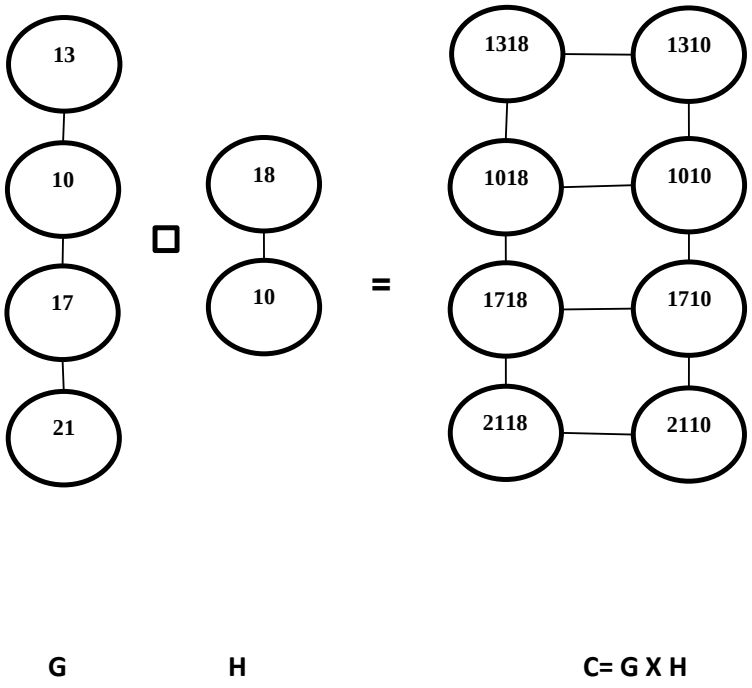

Figure 5.1. The CPG of path graphs *G* and *H* have 4 and 2 vertices respectively.

The ciphertext C of a message *m* is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

$$\text{List 1: } \{13, 10, 17, 21\} \quad \text{and} \quad \text{List 2: } \{18, 10\}.$$

computes Since the length of m is $K = 6$, so he/ she the following computations: Based on the English alphabet Table(3.1), the previous

and numbers converted into

$$13 - 6 = 07 \rightarrow h,\ 10 - 6 = 04 \rightarrow e,\ 17 - 6 = 11 \rightarrow l,$$

$$21 - 6 = 15 \rightarrow p,\ 18 - 6 = 12 \rightarrow m, 10 - 6 = 04 \rightarrow e.$$

Thus, the plaintext m help me.

Example 5.3: Encryption Scheme Based on the CPG

Suppose m is the plaintext that is given by the sentence **Museum**. Based on the English ASCII Table (3.2) of the letters, one can convert the letters in the plaintext m into numbers. So,

$$M \rightarrow 077,\ u \rightarrow 117,\ s \rightarrow 115,\ \text{e} \rightarrow 101,\ u \rightarrow 117,\ m \rightarrow 109.$$

The length K of m is equal to 6. Adding K to all of these numbers one by one gives us

$$077 + 6 = 083,\ 117 + 6 = 123,\ 115 + 6 = 121,$$

$$101 + 6 = 107,\ 117 + 6 = 123,\ 109 + 6 = 115.$$

In other words, it is possible to write these numbers in two lists

List 1: {083, 123, 121, 107} and List 2: {123, 115}.

These lists can be represented by two graphs, say two path graphs as shown in Figure (5.3). The graph *G* and *H* are used to form the Cartesian product graph (CPG), G × H, that is computed by

G × H={(083,123), (083,115), (123,123), (123,115), (121,123),(121,115),

(107,123), (107,115)}.

The G × H is shown Figure (5.3).

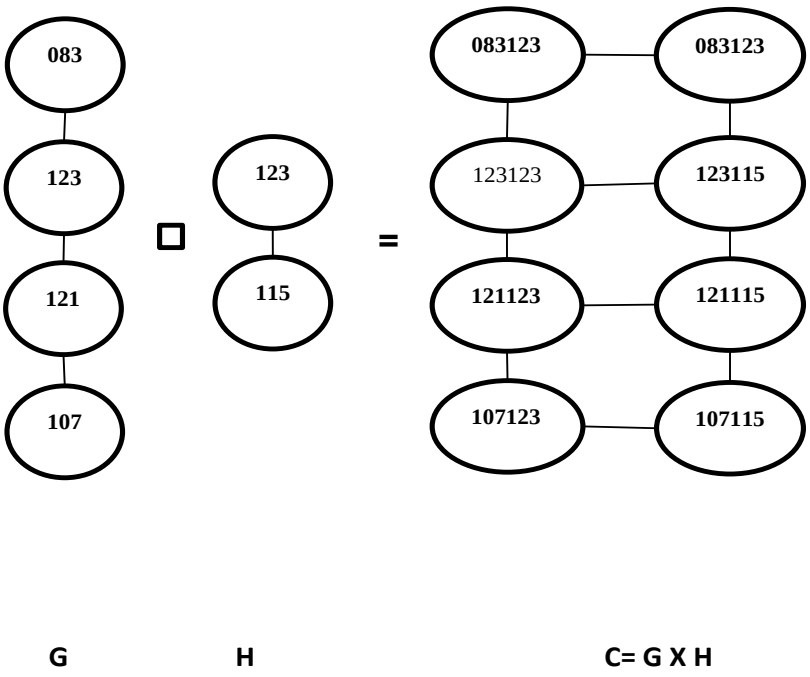

Figure 5.3. The CPG of path graphs *G* and *H* have 4and 2 vertices respectively.

The ciphertext C of a message *m* is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: {083, 123, 121, 107} and List 2: {123, 115}.

Since the length of m is $K = 6$, so he/ she computes the following computations:

$$083 - 6 = 077,\ 123 - 6 = 117,\ 121 - 6 = 115,$$

$$107 - 6 = 101,\ 123 - 6 = 117,\ 115 - 6 = 109.$$

Based on the English ASCII Table (3.2), the previous numbers converted into

$$077 \rightarrow M,\ 117 \rightarrow u \text{ and } 115 \rightarrow s,\ 101 \rightarrow e,\ 117 \rightarrow u,\ 109 \rightarrow m.$$

Thus, the plaintext m is **museum**.

Example 5.4: Encryption Scheme Based on the CPG

Suppose m is the plaintext that is given by the sentence m=**Central bank** . Based on the English ASCII Table (3.2), of the letters, one can convert the letters in the plaintext m into numbers. So,

$$C \rightarrow 067,\ e \rightarrow 101,\ n \rightarrow 110,\ t \rightarrow 116,\ r \rightarrow 114,\ a \rightarrow 097,$$

$$l \rightarrow 108,\ \ \rightarrow 032,\ b \rightarrow 098,\ a \rightarrow 097,\ n \rightarrow 110,\ k \rightarrow 107.$$

The length K of m is equal to 12. Adding K to all of these numbers one by one gives us

$C \rightarrow 067+12=079,\ e \rightarrow 101+12=113,\ n \rightarrow 110+12=122,\ t \rightarrow 116+12=001,$

r →114+12=126, *a*→097+12=109, *l*→ 108+12=120, →032+12=044,

b→ 098+12=110, *a*→ 097+12=109, *n*→110+12=122, *k*→107+12=119.

In other words, it is possible to write these numbers in two lists
List 1: {079, 113, 122, 001, 126, 109} and List 2: {120, 044, 110, 109, 122, 119}.

These lists can be represented by two graphs, say two path graphs as shown in Figure (5.4). The graph *G* and *H* are used to form the Cartesian product graph (CPG), G × H, that is computed by

G × H = {(079,120), (079,044),(079,110), (079,109), (079,122), (079,119),…, (109,119)

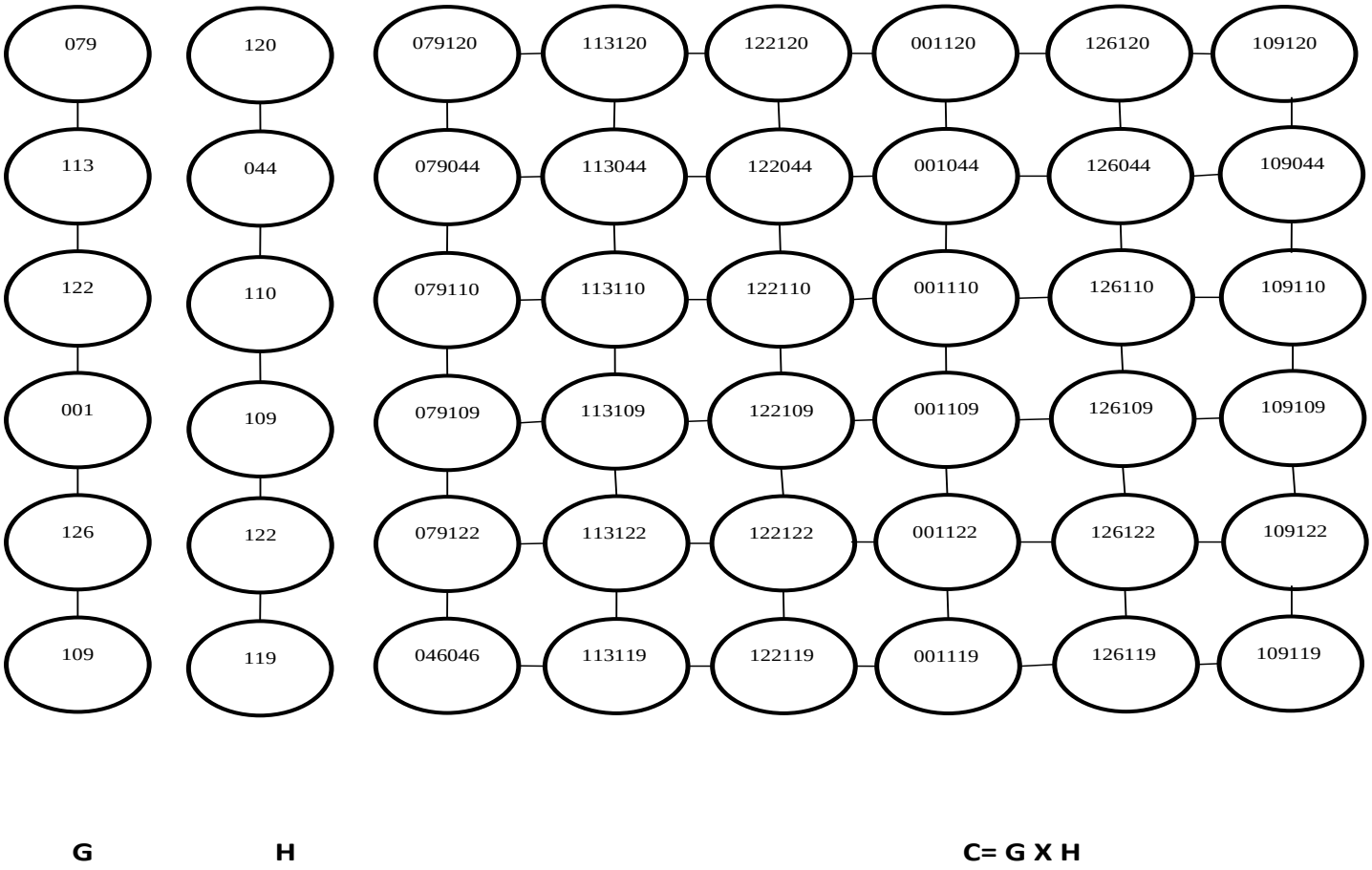

Figure 5.4. The CPG of path graphs *G* and *H* have 6 and 6 vertices respectively.

The ciphertext C of a message *m* is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the cipher text and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: {097, 113, 122, 001, 126, 109} and List 2: {120, 044, 110, 109, 122, 119}.

Computes Since the length of m is $K = 12$, so he/ she the following computations:
Based on the English ASCII Table(3.2), the previous and numbers converted into

$$079 - 12 = 067 \rightarrow C, 113 - 12 = 101 \rightarrow e, 122 - 12 = 110 \rightarrow n,$$

$$001 - 12 = 116 \rightarrow t, 126 - 12 = 114 \rightarrow r, 109 - 12 = 097 \rightarrow a,$$

$$120 - 12 = 108 \rightarrow l, \ 044 - 12 = 032 \rightarrow , 110 - 12 = 098 \rightarrow b,$$

$$109 - 12 = 097 \rightarrow a, 122 - 12 = 110 \rightarrow n, 119 - 12 = 107 \rightarrow k .$$

Thus, the plaintext m is Central bank .

Example 5.5: Encryption Scheme Based on the CPG

Suppose m is the plaintext that is given by the sentence m= diploma. Based on the English alphabet Table (3.1) of the letters, one can convert the letters in the plaintext m into numbers. So,

$$d \rightarrow 3, i \rightarrow 4, p \rightarrow 15, l \rightarrow 11, o \rightarrow 14, m \rightarrow 12, a \rightarrow 0.$$

some rules on the key K are determined *by*

1- Sift first letters three positions to its right.

2- Sift the second letters four positions to its left.

In more details, the letter *d* moves three position into up to become *g*, *i* letter moves four positions into dawn to become *e*, the letter p moves three position into up to become *s*. *l* letter moves four positions into dawn to become *h*.

$$di \rightarrow g\ e,\ p\ l \rightarrow s\ h,\ o\ m \rightarrow r\ i,\ a \rightarrow d\ .$$

In other words, it is possible to write these numbers in two lists

List 1: $\{g, e, s\}$ and List 2: $\{h, r, i, d\}$.

These lists can be represented by two graphs, say two path graphs as shown in Figure (5.5). The graph *G* and *H* are used to form the Cartesian product graph (CPG), G × H, that is computed by

C = G × H ={(g, h),(g, r),(g, i),(g, d),…,(s, d)}.

The G × H shown Figure (5.5).

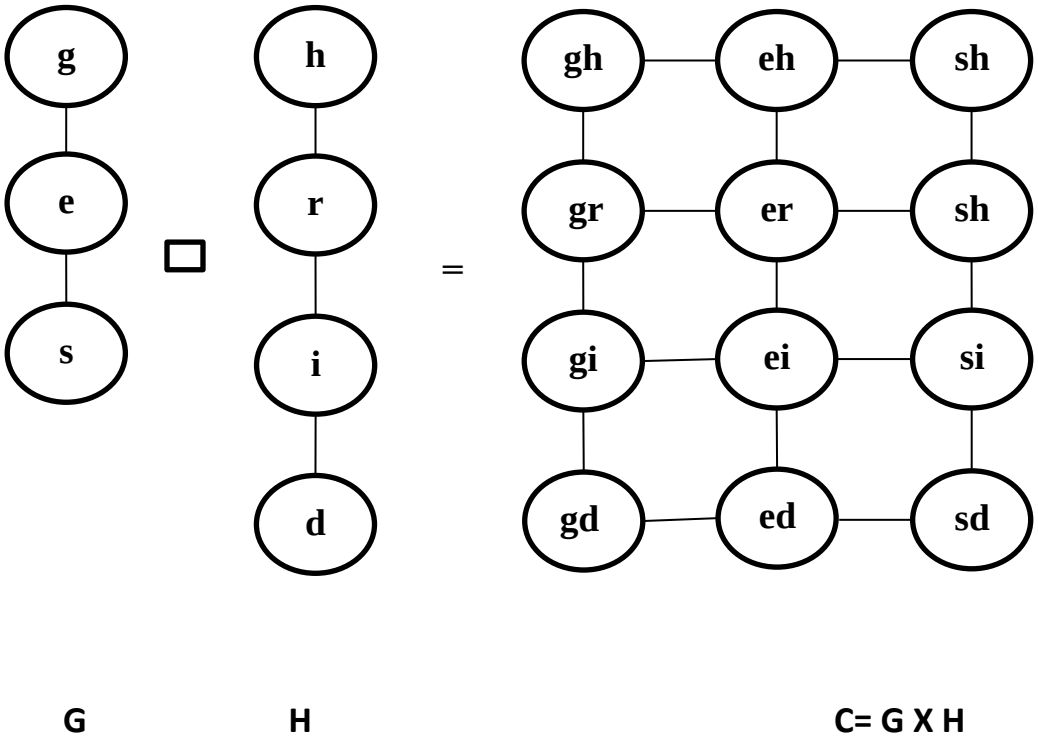

Figure 5.5. The CPG of path graphs *G* and *H* have 3 and 4 vertices respectively.

The ciphertext C of a message *m* is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: $\{g, e, s\}$ and List 2: $\{h, r, i, d\}$.

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

1- Sift first letters three positions to its left.

2- Sift the second letters four positions to its right.

Based on the key and ASCII value in table (3.1), the second user performs the following computation:

$$g\ e \rightarrow \text{d i},\ s\ h \rightarrow \text{p l},\ \text{r i} \rightarrow \text{o m},\ \text{d} \rightarrow \text{a}.$$

Thus, the plaintext *m* is diploma.

Example 5.6: Encryption Scheme Based on the CPG

Suppose *m* is the plaintext that is given by the sentence **books**. Based on the English alphabet Table (3.1) of the letters, one can convert the letters in the plaintext *m* into numbers. So,

$$b \rightarrow 1,\ o \rightarrow 14,\ o \rightarrow 14,\ k \rightarrow 10,\ s \rightarrow 18.$$

The length *K* of *m* is equal to 5. Adding *K* to all of these numbers one by one gives us

$$1 + 5 = 06,\ 14 + 5 = 19,\ \ 14 + 5 = 19,\ 10 + 5 = 15,\ 18 + 5 = 23.$$

In other words, it is possible to write these numbers in two lists

$$\text{List 1: } \{06, 19\} \quad \text{and} \quad \text{List 2: } \{19, 15, 23\}.$$

These lists can be represented by two graphs, say two path graphs as shown in Figure (5.6). The graph G and H are used to form the CPG, G×H, is computed by

$$G \times H = \{(06,19), (06,15), (06,23), (19,19), (19,15), (19,23)\}.$$

The $G \times H$ shown in Figure (5.6).

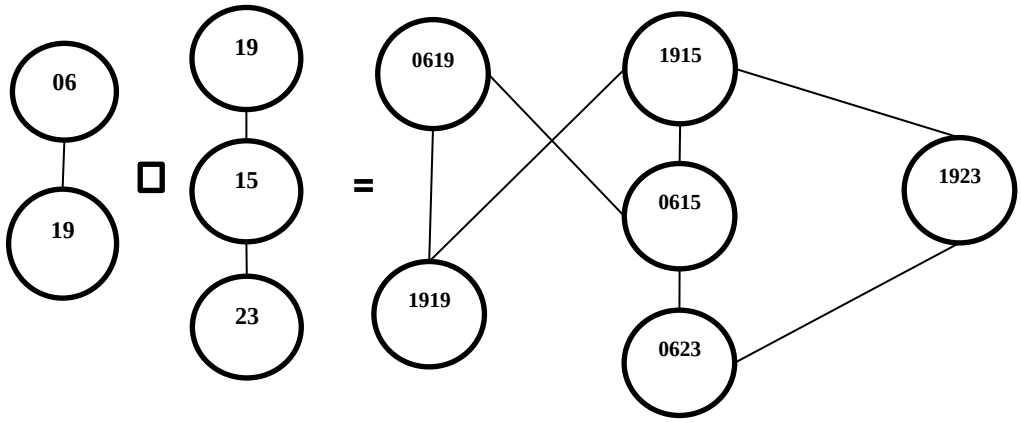

Figure 5.6. The ciphertext C as Cartesian product graph $G \times H$ of path graphs G and H with 2 and 3 vertices respectively.

The ciphertext C of a message m is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1: {06, 19} and List 2: {19, 15, 23}.

Since the length of m is $K = 5$, so he/ she computes the following computations:

$$06 - 5 = 01,\ 19 - 5 = 14,\ 19 - 5 = 14,\ 15 - 5 = 10,\ 23 - 5 = 18.$$

Based on the English alphabet Table (3.1), the previous numbers converted into

$$01 \rightarrow b,\ 14 \rightarrow o \text{ and } 14 \rightarrow o,\ 10 \rightarrow k,\ 18 \rightarrow s.$$

Thus, the plaintext m is **books**.

Example 5.7. The CPG for Polyalphabetic Encryption Scheme Based on ASCII Values

Suppose m is the plaintext that is given by the word **Baghdad city**. Based on the ASCII Table (3.2) of the letters, one can convert the letters in the plaintext m into numbers. So,

$B \rightarrow 066$, $a \rightarrow 097$, g $\rightarrow$ 103, $h \rightarrow 104$, $d \rightarrow 100$,a $\rightarrow$ 097,

$$d \rightarrow 100,\ c \rightarrow 099,\ i \rightarrow 105,\ t \rightarrow 116,\ \text{y} \rightarrow 121.$$

The shared secret key K is determined as the word **sky**. Adding K to all of these numbers one by one gives us

Bag hda dci ty

B+ s $\rightarrow$066+115=054, a+ k $\rightarrow$097+107=077, g+ y $\rightarrow$103+121=097,

h+ s $\rightarrow$104+115=092, d+ k $\rightarrow$100 +107=080, a+ y $\rightarrow$097+121=091,

d+ s $\rightarrow$100+115=088, c+ k $\rightarrow$099+107=079, i+ y $\rightarrow$105+121=099,

t+ s $\rightarrow$116+115=104, y+ k $\rightarrow$121+107=101.

In other words, it is possible to write these numbers in two lists

List 1:{054, 077, 097, 092, 080, 091} and List 2: {088, 079, 099, 104, 101}.

These lists can be represented by two graphs, say two path graphs as shown in Figure (5.7). The graph *G* and *H* are used to form the Cartesian product is computed by

C = G×H= {(054, 097), …, (091, 101)}.

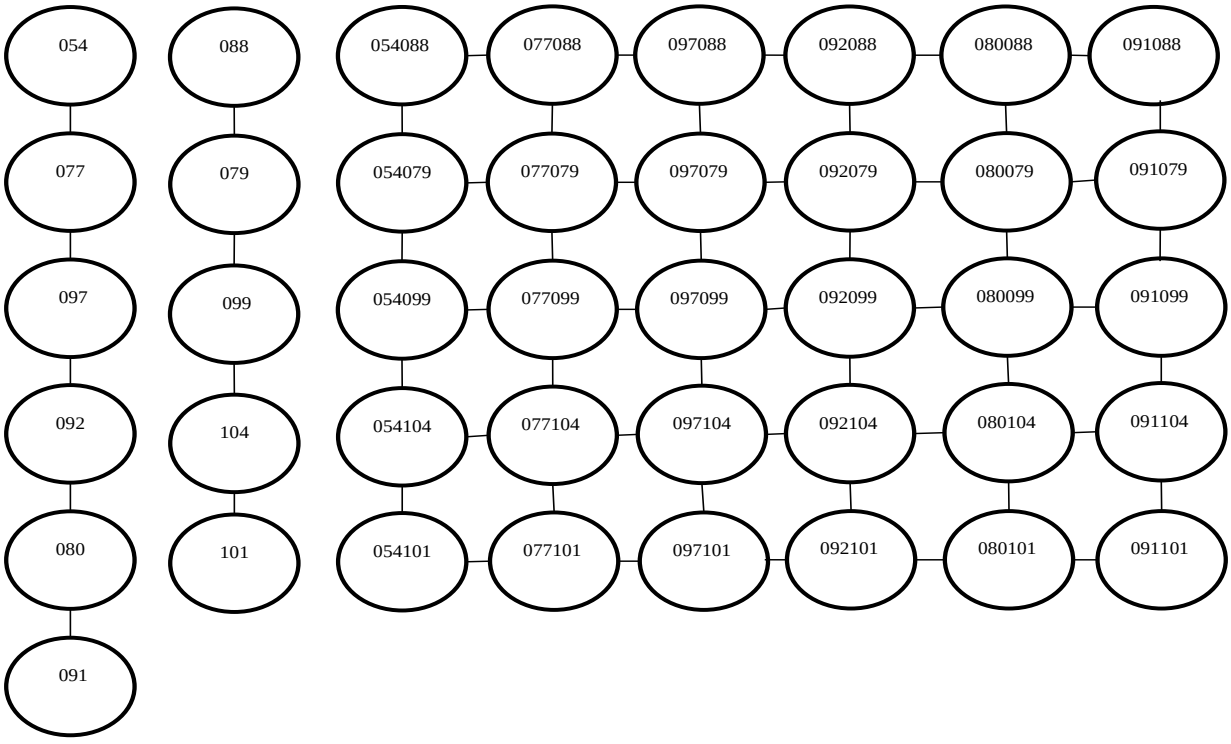

Figure 5.7. The CPG of path graphs *G* and *H* have 6 and 5 vertices respectively.

The ciphertext C of a message *m* is considered as the CPG which is sent to receiver by sender.

The second user (receiver) receives the CPG. He/ She wants to decrypt the ciphertext and recover the original plaintext. He/ She first writes down the labeled vertices into two lists.

List 1:{054, 077, 097, 092, 080, 091} and List 2: {088, 079, 099, 104, 101}.

Based on a shared secret key the word **sky** and the ASCII values, second user performs the following computations:

$$054\text{-}115=066\rightarrow B,\ 077\text{-}107=097\rightarrow a,\ 097\text{-}121=103\rightarrow g,$$

$$092\text{-}115=104\rightarrow h,\ 080\text{-}107=100\rightarrow d,\ 091\text{-}121=097\rightarrow a,$$

$$088\text{-}115=100\rightarrow d,\ 079\text{-}107=099\rightarrow c,\ 099\text{-}121=105\rightarrow i,$$

$$104\text{-}115=116\rightarrow t,\ 101\text{-}107=121\rightarrow y.$$

Thus, the plaintext *m* is a **Baghdad city** .

Chapter Six

Conclusions and Future Works

6.1 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 synthetic encryption schemes. This application used the Cartesian product graph (CPG) to design these versions with more secure level to create the cipher text of the original message. These versions are CPG encryption schemes based on English alphabet values and CPG encryption schemes based on ASCII values on other hand these graphs are applied to modify the polyalphabetic substitution cipher.

In the proposed method, while constructing the graph, the selection of number of vertices and the assignment of edges, so it is hardly possible to predict for an unauthorized person that how many vertices should be chosen and which edge will come between which pair of vertices. In addition, it is very difficult to find the sequence of edges while decryption. Therefore, the proposed algorithm is very useful for secure communication and has enormous potential to grow in future.

6.2 Future Works

It is possible to apply the same idea of the proposed encryption schemes with other kinds of symmetric and a symmetric encryption schemes and also it can use other types of graph.

References

1. BONDY, John Adrian, et al. Graph theory with applications. London: Macmillan, 1976.
2. RIVEST, Ronald L. Cryptography. In: Algorithms and complexity. Elsevier, 1990. p. 717-755.
3. WARNER, Seth. Modern Algebra, chapter 1. 1990.
4. BELLARE, Mihir; ROGAWAY, Phillip. Introduction to modern cryptography. Ucsd Cse, 2005, 207: 207.
5. SFYRAKIS, Ioannis; GROSS, Thomas. GSL: A Cryptographic Library for the strong RSA Graph Signature Scheme. arXiv preprint arXiv:2005.12447, 2020.
6. COMTET, Louis. Advanced Combinatorics: The art of finite and infinite expansions. Springer Science & Business Media, 2012.
7. SUEN, Stephen; TARR, Jennifer. An improved inequality related to Vizing's conjecture. the electronic journal of combinatorics, 2012, P8-P8.
8. PEACOCK, Thea, et al. Verifiable voting systems. In: Computer and information security handbook. Morgan Kaufmann, 2013. p. e293-e315.
9. LESTRINGANT, Pierre; GUIHÉRY, Frédéric; FOUQUE, Pierre-Alain. Automated identification of cryptographic primitives in binary code with data flow graph isomorphism. In: Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security. 2015. p. 203-214.
10. AGARWAL, Shubham; UNIYAL, Anand Singh. Prime weighted graph in cryptographic system for secure communication.

International Journal of Pure and Applied Mathematics, 2015, 105.3: 325-338.

11. MENEZES, Alfred J.; VAN OORSCHOT, Paul C.; VANSTONE, Scott A. Handbook of applied cryptography. CRC press, 2018.
12. AMUDHA, P.; SAGAYARAJ, AC Charles; SHEELA, AC Shantha. An application of graph theory in cryptography. International Journal of Pure and Applied Mathematics, 2018, 119.13: 375-383.
13. RAY, Santanu Saha. Graph theory with algorithms and its applications: in applied science and technology. Springer Science & Business Media, 2012.
14. JOY PERSIAL, G.; PRABHU, M.; SHANMUGALAKSHMI, R. Side channel attack-survey. Int J Adva Sci Res Rev, 2011, 1.4: 54-57.
15. KINARIWALA, Bharat; DOBRY, Tep. Programming in C1. 1993.
16. PRONGJIT, Sirinya; SODSIRI, Wijarn. Applications of δ-fine tagged partitions in real analysis. Far East J. Math. Sci.(FJMS), 2014, 91.1: 97-109.
17. WEST, Douglas Brent, et al. Introduction to graph theory. Upper Saddle River: Prentice hall, 2001.
18. DEO, Narsingh. Graph theory with applications to engineering and computer science. Courier Dover Publications, 2017.
19. PERERA, P. A. S. D.; WIJESIRI, G. S. Encryption and Decryption Algorithms in Symmetric Key Cryptography Using Graph Theory. Psychology and Education Journal, 2021, 58.1: 3420-3427.
20. TOEMEH, Ragheb; ARUMUGAM, Subbanagounder. Applying Genetic Algorithms for Searching Key-Space of Polyalphabetic Substitution Ciphers. International Arab Journal of Information Technology (IAJIT), 2008, 5.1.

21. NASUTION, Surya Darma, et al. Data Security Using Vigenere Cipher and Goldbach Codes Algorithm. Int. J. Eng. Res. Technol, 2017, 6.1: 360-363.
22. SUN, Yuefang; LI, Xueliang; LI, Hengzhe. The generalized 3-connectivity of Cartesian product graphs. Discrete Mathematics & Theoretical Computer Science, 2012, 14.

الملخص

هذا العمل يقترح إصدارات جديدة من أنظمة التشفير المتماثل بناءا على ناتج الرسم البياني لحاصل الضرب الديكارتي .وهذه الإصدارات مخططات لتشفير الرسومات الناتجة من حاصل الضرب الديكارتي يعتمد على قيم الأحرف الأبجدية الإنكليزية,ونظام التشفير موضوع على أساس قيم موضوعه بجدول معين وكذلك نظام التشفير متعدد الأبجدية.

يتم اختيار الرسالة والتي هي عبارة عن كلمه أو جمله من اللغة الإنكليزية تعتبر الشفرات للرسالة الأصلية بمثابة ناتج حاصل الضرب الديكارتي والتي يتم إرسالها إلى المستلم عن طريق المرسل .

تمت مناقشه العديد من النتائج التجريبية لأنظمة التشفير المقترحة.

وكذلك تم تحديد الاعتبارات الأمنية لأنظمة التشفير المقترح.

جمهوريه العراق

وزاره التعليم العالي والبحث العلمي

جامعه بابل كليه- التربية للعلوم الصرفة

نظام تشفير امن باستخدام نظرية البيان

بحث مقدم إلى
قسم الرياضيات - كليه التربية للعلوم الصرفة - جامعه بابل

وهو جزء من متطلبات نيل شهادة الدبلوم العالي تربية / الرياضيات

من قبل
مروه علي حسين علي

بأشراف

ا.د. احمد عبد علي عمران