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

New Design Cryptosystems via Graphs Theory

A Dissertation

Submitted to the Council of the College of Education for Pure Sciences in University of Babylon in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Education / Mathematics

By

Karrar Taher Radhi Ewad

Supervised by

Asst. Prof. Dr. Ruma Kareem K. Ajeena

2022 A.D. **1443 A.H.**

Contents

List of Tables

List of Figures

List of Algorithms

Table 1: The Mathematical Symbols.

Symbol	Description
p	Prime number.
F_p	Prime field.
g	Primitive root.
$\mid$	Division.
$\equiv$	Congruence.
$M_n(R)$	Matrix Ring.
$GL_n(F)$	General Linear Group over Filed.
O_E	Infinity point on elliptic curve.
$\triangle$	Discriminate of EC.
$EC(F_p)$	Elliptic curve group over F_p.
$\#EC(F_p)$	Order of elliptic curve group over F_p.
kP	Scalar multiplication operation.
m	plaintext.
mod	Modulo.
$mod\ p$	Arithmetic modulo p.
ϕ	Euler Phi function .
$gcd(a, b)$	Greatest common divisor of a and b.
■	End of proof.

Table 2: The abbreviations.

Abbreviations	Definitions
CESM	Complete Elliptic Scalar Multiplication
CSESM	Complete Symmetric Elliptic Scalar Multiplication
DHKE	Diffe-Helman key exchange
DLP	Discrete Logarithm Problem.
EC	Elliptic Curve.
ECC	Elliptic Curve Cryptosystem.
ECDLP	Elliptic Curve Discrete Logarithm Problem.
ECMF-DH	Elliptic Curve Matrix Function-Diffie – Hellman
EEPKC	Elliptic Curve ElGamal Public Key Cryptosystem.
EPKC	ElGamal Public Key Cryptosystem.
ESMMF	Elliptic Scalar Multiplication Matrix Function
GCD	The greatest common divisor.
ISD	Integer Sub-Decomposition.
L/R-ESMMF	Left-Right sides elliptic scalar multiplication matrix function
L-ESMMF	Left-side elliptic scalar multiplication matrix function
LMPF	A left-sided matrix power function
MP	Matrix Power.
MP-EPKC	Matrix Power ElGamal Public Key Cryptosystem.
MPF	Matrix Power Function.
MST	Minimum Spanning Tree.
RSA	Rivest–Shamir–Adleman.
SG-PKC	Soft Graph Public Key Cryptosystem.
UCG	Undirected Complete Graph.

Abstract

This work employed some graph theory concepts to design new cryptosystems. The proposed cryptosystems use the matrix power functions (MPF), which are considered more secure cryptosystems compared to other cryptosystems. Among these cryptosystems, the alternative version of the ElGamal public key cryptosystem (EPKC) that is being proposed depends on an extended DLP over matrices. Following that, a new public key cryptosystem based on an undirected complete graph (UCG) has been designed. Also, a new public key cryptosystem has been created to encrypt a weighted subgraph of an undirected graph in $n-$dimensional vector space V. The L/R MPF DLP has been proposed to display other cryptosystems revised versions of the original ones, and the L/R MPF DLP is employed with graph theory to draw other cryptosystem which is a hybrid G-L/R MPF-EPKC. As well, the soft graph is used with the MPF to design another cryptosystem that encrypts a subset of the prime field. The decryption process for the proposed cryptosystems has been proved theoretically.

On the other hand, new graphs have been defined as the main points to design new asymmetric cryptosystems. These graphs are formed based on the scalar multiplication operation on the elliptic and Edwards curves defined over a prime field. New public key cryptosystems are proposed on an elliptic and an Edwards curves graph. Several new experimental results of the proposed cryptosystems are introduced.

Publications

The publications of this work are:

1. Karrar Taher R. Aljamaly and Ruma Kareem K. Ajeena, The KR –Elliptic Curve Public Key Cryptosystem, at Ibn Al-Haitham International Conference for Pure and Applied Sciences (IHICPS). IOP Publishing, Journal of Physics, Conference Series. DOI:10.1088/1742-6596/1879/3/032046. (Indexed in Scopus).

2. Karrar Taher R. Aljamaly and Ruma Kareem K. Ajeena, Undirected Complete Graph to Design New Public Key Cryptosystem,IOP Publishing, Journal of Physics, Conference Series. DOI:10.1088/1742-6596/1897/1/012045 (Indexed in Scopus).

3. Karrar Taher R. Aljamaly and Ruma Kareem K. Ajeena, The Elliptic Scalar Multiplication Graph its Application in Elliptic Curve Cryptography, Taylor and Francis Publishing, Journal of Discrete Mathematical Sciences and Cryptography, DOI:10.1080/09720529.2021.1932896 (Indexed in Scopus).

4. Karrar Taher R. Aljamaly and Ruma Kareem K. Ajeena,The Elliptic Scalar Multiplication Digraph for Elliptic Curve Cryptographic Usages, The Scientific Journal of King Faisal University, DOI:10.37575/b/sci/0048. (submitted).

5. Karrar Taher R. Aljamaly and Ruma Kareem K. Ajeena, A Matrix Power Function to revise the Discrete Logarithm Encryption Schemes, Iraqi Journal of Science (submitted).

Dedication

To everyone who helped me achieve my dissertation;

to my professors;

to the soul that will stay with me .... my father;

to my mother and brothers;

to my wife and children;

to my supervisor, Dr. Ruma Kareem K. Ajeena;

to all my friends and relatives;

I dedicate this work.

Karrar Taher Radhi
2021

Acknowledgements

I would like to express my appreciation and great thanks to my supervisor, Asst. Prof. Dr. Ruma Kareem K. Ajeena, for her valuable instruction, patience, and support during the writing of this thesis. I would like to record my thanks to the Head, professors, and staff of the Mathematics Department at the Faculty of Education for Pure Sciences, University of Babylon, for supporting me throughout the progress of my studies. I wish to express my deepest thanks to my family for their support and encouragement during the period of this work. Finally, I would like to thank all those who have participated in one way or another in achieving this work.

Karrar Taher Radhi
2021

Chapter 1

General Introduction

1.1 Introduction

Cryptography is science to design and analysis the mathematical techniques that enable secure communications in the presence of malicious adversaries. It is a technique that enables secure communication in the face of attacks by hackers or other attackers. Diffie and Hellman, in 1976, introduced new directions in cryptography depended on the computations of the discrete logarithm problem (DLP) and exchanged the results of these computations between two entities [15]. The RSA cryptosystem was proposed by Rivest, R.L., et al, in 1978 by presenting a public key cryptosystem that depends on an integer factorization problem that factors a composite number into two large primes [35]. Elgamal in 1985 introduced another version of the cryptosystems which are public key cryptosystems and a signature scheme that is also based on the DLP [17].

Graph theory is widely used as a tool for encryption due to its various properties and its easy representation on computers as a matrix. It is considered as an essential tool in many cryptographic applications. Most of them focused on applying various concepts of graph theory to design the symmetric encryption algorithms [7, 42]. Some researchers proposed cryptographic algorithms using paths in any graph [39] and others proposed encryption algorithms using directed graphs [31].

The elliptic curve cryptography (ECC) is more interesting to many researchers, because it has been employed in different applications, such as mobile devices, wireless sensors, networks, image encryption, and others [1, 29, 41]. It has received more attention due to its smaller key size, which allows it to be much more efficient compared to other public key cryptosystems like RSA [35]. This makes it more attractive for applications in confined environments, as shorter key sizes translate into fewer power and storage requirements and shorter computing times. Miller in 1985, proposed the Diffie-Hellman key exchange protocol based on elliptic curves [30]. Koblitz in 1987 proposed the elliptic curve ElGamal public key cryptosystem

[27]. In 2000, Neal Koblitz introduced the state of ECC. He presented a survey about elliptic curve cryptosystems [28].

The matrix power function (MPF) is based on matrix powering by other matrices. This function is some generalization of a discrete exponent function by its expansion in matrix set. Sakalauskas and Luksys, in 2007, defined MPF as an action of two matrices powering some base matrices on the left and right, as shown in the mathematical background section in Equations (2.1) and (2.2) and introduced a matrix power S-box construction. They proposed a symmetric encryption based on the S-box construction MPF [37]. The same researchers in 2012 introduced the MPF and its application in the block cipher S-box construction. They used the system of multivariate quadratic polynomial equations [38].

1.2 Previous Studies

In 2009, Jao, et al., [22], introduced the expander graphs which depend on the generalized Riemann hypothesis (GRH) with its applications in ECC. They presented the construction of the expander graphs, their properties and the security of their proposition.

In 2012, Selvakumar and Gupta, [40], proposed their study using the fundamental circuits and cut-sets in cryptography. They presented an innovative algorithm for encryption and decryption using the connected graphs. The messages are represented by the connected graphs and encrypted by using a spanning tree of the graph. In the same year, Yamuna, et al., [45], introduced an encryption mechanism using Hamilton path properties. They encrypt the data twice, once using the Hamilton path and the second time using the complete graph to impose a more secure method.

In 2014, Al Etaiwi, [7], proposed a symmetric encryption algorithm to encrypt and decrypt the data using graph theory. He used graph theory properties such as complete graph and minimum spanning tree.

In 2015, Agarwal and Uniyal, [3], presented a definition of the prime weighted graph and proposed an encryption scheme based on the prime weighted graph with more secure communication.

In 2016, Amounas [8], introduced an innovative approach to enhance the security of Amazigh text using the elliptic curve cryptography based on graph theory.

In 2018, Amudha et al., [9], proposed an encryption technique that applied the graph theory in cryptography. Each character of the data based on this technique has been encrypted into an Euler Graph. Also, in 2018, Bhapka [11], introduced the applications of planar graph on the key in cryptography. He proposed a secret code for planar graph with respect to regions.

In 2020, Khaleel and Al-Shumam [25] introduced a study of graph theory applications in IT security. They used the Euler graph as a method that is employed in the remote method invocation (RMI) technique and compared this algorithm with the most popular algorithms, such as RSA and 3DES. Also in the same year, Akl, [5], proposed a study on how to encrypt graphs. The graph is considered as a message and its edge weights are the information.

In 2021, Perera and Wijesiri [33] introduced the encryption and decryption algorithms in the symmetric key cryptography using graph theory. They proposed a connection between the graph theory and symmetric cryptography to protect the information from unauthorized parties. Their proposed methodology used a matrix as the secret key, which added more security to the cryptosystem. It converted the plaintext into several graphs and represented these graphs in their matrix form. In the same year, Joseph and Bindhu [23], introduced a cryptographic method irrespective (CMI) of code order that used the graph theory. The proposed CMI of code order is used for encrypting and decrypting the data securely with the benefits of a defined graph and a matrix that used the binary strings of ASCII values of the given message. In 2021, Adnan, et al., [2], proposed securing text messages using the graph theory and steganography. This study aimed to secure text messages through two security principles: encryption and steganography. The system produced a novel method for encryption using the graph theory properties.

1.3 Objectives of the Dissertation

The aim of this dissertation is to propose new versions of cryptosystems that use graph theory concepts. One of them depended on an undirected complete graph and a minimum spanning tree, while another one used vector space associated with a graph. The soft graph was also applied in one of these cryptosystems. In addition, two curves, the elliptic curve and the Edwards curve defined over a prime field, are used to define new graphs. The graphs are applied to give a new cryptosystem. Also, design a new cryptosystem in ECC based on the factorization in number theory.

1.4 The Problem Statement

There are cryptosystems that can be broken by attackers due to the security of these systems. Graph theory can be used in different areas of cryptography because it increases the security of cryptosystems. In this work, it is proposed to give extended versions of the original cryptosystems like EPKC by using graph theory and matrices.

In addition, new ECC have been proposed, with higher security than previous systems. One of the themes is the KR-elliptic curve public key cryptosystem. The main idea behind this system is to design a cryptosystem that focuses on generating the elliptic curve group. Its order divides the Euler phi function that is computed securely based on two large primes. Faster and more secure computations to generate the keys, encryption and decryption processes, on the KR-ECPK algorithm are discussed. Compared with other algorithms like RSA and the ElGamal public key cryptosystem. Other versions of public key cryptography are proposed on an elliptic curve and an Edwards curve based on new graphs have been defined. These graphs are formed based on the scalar multiplication operation on the elliptic curve and Edwards curve defined over a prime field.

1.5 Dissertation Outline

The outline of this study is as follows: in addition to chapter one, it contains:

Chapter 2. This includes the basic facts of finite fields, general linear groups, and matrix power functions. Furthermore, this chapter presents some basic concepts of graph theory. In another part of this chapter, it includes an introduction to cryptography as well as some concepts and schemes in cryptography. Finally, it presents elliptic and Edward curves over the prime fields.

Chapter 3. It introduces some definitions about the discrete logarithm problem using matrices and then used it in cryptographic schemes such as Diffe-Hellman and Elgamal schemes after that used graph theory in these schemes.

Chapter 4. uses number theory, matrices, and graph theory, to propose a novel version of elliptic curve public key cryptosystem. Also proposed a new version of

Edward curve public key cryptosystem based on graph theory.

Chapter 5. includes some computational results on the proposed cryptosystem.

Chapter 6. draws the conclusions and future works.

Chapter 2

Mathematical Background

2.1 Introduction

This chapter first discusses some basic definitions, theorems, and examples of finite fields, the general linear group, and the matrix power function. Also, presents the important facts about the graph theory, it discusses some basic definitions and examples. In addition, the encryption schemes which depend on the DLP have been presented, one of them is the ElGamal public key cryptosystem (EPKC). Also, presents the important facts about elliptic and Edward curves defined over a prime field.

2.2 Basic Concepts

In this section, the mathematical concepts related to the fields, especially the finite fields, the general linear group, and matrix power function (MPF), are discussed as follows:

Definition 2.2.1 (Field)[21].

A field is an order triple $(F, +, \cdot)$, where F is a nonempty set, $+$ and $\cdot$ are two binary operations on F satisfying the following properties:

1. $(F, +)$ is an abelian group with (additive) identity denoted by 0.

2. $(F \setminus \{0\}, \cdot)$ is an abelian group with (multiplicative) identity denoted by 1.

3. The distributive law holds: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ for all $a, b, c \in F$.

Definition 2.2.2 (Finite Field or Galois field)[21].

The triple $(F_p, +, .)$, simply known as F_p or prime field, is a finite field. Finite fields are also called Galois fields.

Definition 2.2.3 (The Characteristic of a Field)[24], p, 358.

Let F be a field. The characteristic of F is the least positive integer p such that for every nonzero element α in F, we have $p\alpha = 0$. If no such p exists, we define the characteristic to be 0.

Theorem 2.2.1 (Primitive Root Theorem)[13].
Let p be a prime number. Then there exists an element $g \in F_p \setminus \{0\}$ whose powers give every element of $F_p \setminus \{0\}$, i.e.

$$F_p \setminus \{0\} = \left\{1, g, g^2, ..., g^{p-2}\right\}$$

Elements with this property are called generators of F_p.

Definition 2.2.4 (Relatively Prime) [20], p. 17.
Two integers a and b, not both of which are zero, are said to be relatively prime whenever $gcd(a, b) = 1$.

Definition 2.2.5 (The general linear group)[18].
Let F be a field. Then the general linear group $GL_n(F)$ is the group of invertible $n \times n$ matrices with entries in F under multiplication matrix.

Proposition 2.2.1 The number of elements in $GL_n(F_p)$ is $\prod_{k=0}^{n-1}(p^n - p^k)$ [43].

Example 2.2.1 The general linear group of 2×2 matrices over F_2 is

$$GL_2(F_2) = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \right\},$$

the number of element in $GL_2(F_2)$ is:

$$\#GL_2(F_2) = (2^2 - 1) \times (2^2 - 2) = 6.$$

Definition 2.2.6 (A left-sided matrix power function)[36].
A left-sided matrix power function (LMPF) over a prime field F_p is defined to correspond the matrix A powered by a matrix L on the left side with MPF value equal to the matrix $B = [b_{ij}]$. The LMPF is written by:

$$^{L}A = B, \quad \text{where} \;\; b_{ij} = \prod_{k=1}^{m} a_{jk}^{l_{ik}}, \tag{2.1}$$

such that A, B and $L \in GL_n(F_p)$.

Definition 2.2.7 (A right-sided matrix power function)[36].

The right-sided matrix power function (RMPF) over F_p defines by a matrix A powered by matrix R on the right side with MPF value equal to the matrix $E = [e_{ij}]$. The RMPF is given by

$$A^R = E, \quad \text{where } \; e_{ij} = \prod_{k=1}^{m} a_{ik}^{r_{kj}}, \tag{2.2}$$

such that A, E and $R \in GL_n(F_p)$.

The LMPF and RMPF have equality relation, namely

$$({}^L A)^R =^L (A^R). \tag{2.3}$$

Example 2.2.2 Let $A = \begin{pmatrix} 18 & 22 \\ 3 & 17 \end{pmatrix}$ and $L = \begin{pmatrix} 7 & 19 \\ 14 & 10 \end{pmatrix}$, then a matrix B over F_{23} is computed by

$$\begin{aligned} B =^L A &= {}^{\begin{pmatrix} 7 & 19 \\ 14 & 10 \end{pmatrix}} \begin{pmatrix} 18 & 22 \\ 3 & 17 \end{pmatrix} \\ &= \begin{pmatrix} 18^7.3^{19} & 22^7.17^{19} \\ 18^{14}.3^{10} & 22^{14}.17^{10} \end{pmatrix} \\ &\equiv \begin{pmatrix} 13 & 18 \\ 12 & 4 \end{pmatrix} (mod\ 23) \end{aligned}$$

Similarly, let $A = \begin{pmatrix} 18 & 22 \\ 3 & 17 \end{pmatrix}$ and $R = \begin{pmatrix} 15 & 11 \\ 21 & 3 \end{pmatrix}$, then a matrix E over F_{23} is computed by

$$\begin{aligned} E = A^R &= \begin{pmatrix} 18 & 22 \\ 3 & 17 \end{pmatrix}^{\begin{pmatrix} 15 & 11 \\ 21 & 3 \end{pmatrix}} \\ &= \begin{pmatrix} 18^{15}.22^{21} & 18^{11}.22^3 \\ 3^{15}.17^{21} & 3^{11}.17^3 \end{pmatrix} \\ &\equiv \begin{pmatrix} 19 & 22 \\ 21 & 14 \end{pmatrix} (mod\ 23). \end{aligned}$$

2.3 Graph Theory

In this section, some basic concepts of graph theory that have been used in this work are discussed as follows.

Definition 2.3.1 (Graph) [44], p, 9.

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

Example 2.3.1 Let $V = \{v_1, v_2, v_3, v_4\}$ be vertex set and the edge set is $E = \{e_1, e_2, e_3, e_4, e_5\}$, where $e_1 = v_1v_2$, $e_2 = v_2v_3$, $e_3 = v_3v_4$, $e_4 = v_1v_4$ and $e_5 = v_1v_3$ are formed the graph G. The graph $G(V, E)$ is shown in Figure 2.1.

Definition 2.3.2 (Subgraph) [34], p, 11.

Let H be a graph with vertex set $V(H)$ and edge set $E(H)$, and similarly let G be a graph with vertex set $V(G)$ and edge set $E(G)$. Then, we say that H is a subgraph of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

Definition 2.3.3 (Spanning subgraph) [34], p, 11.

A spanning subgraph of G is a subgraph H with $V(H) = V(G)$, that is H and G have exactly the same vertex set.

Definition 2.3.4 (Induced subgraph) [34] p, 13.

A subgraph $H \subseteq G$ is an induced subgraph, if $E_H = E_G \cap E_{V(H)}$. In this case, H is induced by its set V_H of vertices. In an induced subgraph $H \subseteq G$, the set E_H of edges consists of all $e \in E_G$, such that $e \in E(V_H)$.

Definition 2.3.5 (Order and Size of a Graph) [34] p, 1.

Let $G = (V, E)$ be a graph. The order of G is defined by $\mid V \mid= n$ and $\mid E \mid= m$ is defined to be the size of G.

In Figure 2.1, $\mid V \mid= 4$ and $\mid E \mid= 5$.

Definition 2.3.6 (Self-Loop and Parallel Edges) [34], p, 1.

The definition of a graph allows the possibility of the edge e having identical end vertices. Such an edge having the same vertex as both of its end vertices is called a self-loop (or simply a loop), see Figure 2.2. Also, note that the definition of graph allows that more than one edge is associated with a given pair of vertices, such edges are referred to as parallel edges as shown in Figure 2.2.

Definition 2.3.7 (Simple graph) [34] p, 1.

A graph, that has neither self-loops nor parallel edges, is called a simple graph. A simple graph is given in Figure 2.1.

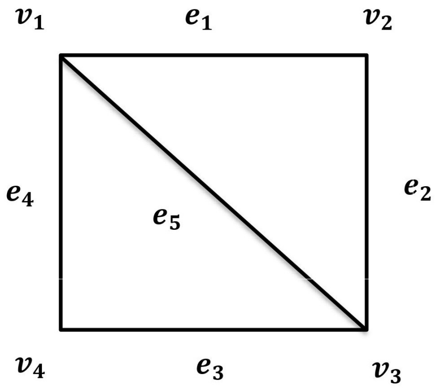

Figure 2.1: The simple graph.

Definition 2.3.8 (Multigraph) [34], p, 1.

A multigraph G is an ordered pair $G = (V, E)$ with V a set of vertices or nodes and E a multiset of unordered pairs of vertices which are called edges. A multigraph is shown in Figure 2.2.

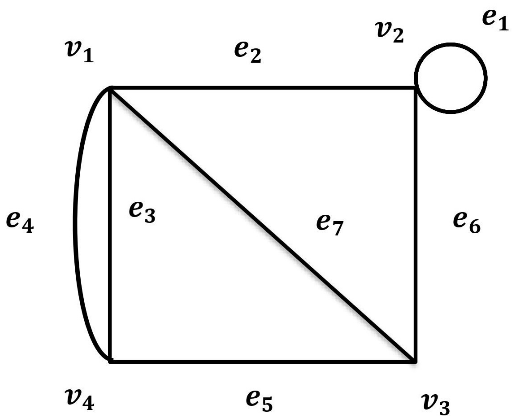

Figure 2.2: The multigraph.

Definition 2.3.9 (Adjacency Matrix representations) [34], p, 101.

Assume that G is a simple undirected graph of order n with vertex set $\{v_1, v_2, ..., v_n\}$. The adjacency matrix of G is the $n \times n$ matrix $A = [a_{ij}]$, whose entries a_{ij} are given by:

$$a_{ij} = \begin{cases} 0, & \text{if there is no edge between ith and jth vertices,} \\ 1, & \text{if there is an edge between them.} \end{cases}$$

Example 2.3.2 The symmetric adjacency matrix of graph that is given in Figure 2.1 is computed by

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Definition 2.3.10 (Complete Graph) [44], p, 17.

A simple graph in which there exists an edge between every pair of vertices is called a complete graph. The complete graph with n vertices can be denoted by K_n.

Example 2.3.3 The complete graph K_5 is given in Figure 2.3.

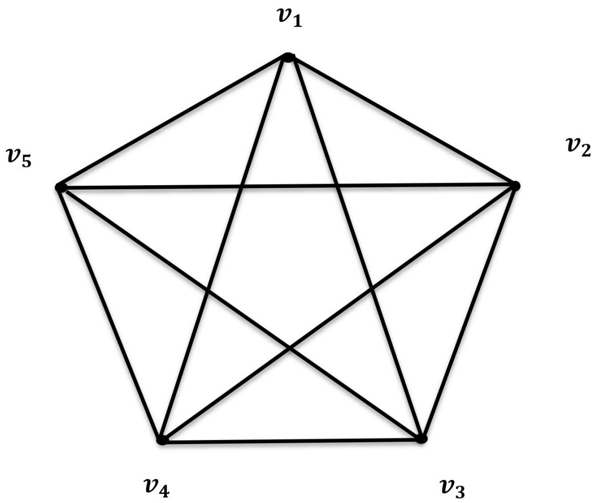

Figure 2.3: The complete graph K_5.

Definition 2.3.11 (Weighted graph) [44], p, 39.

A weighted graph is a graph in which each edge has a numerical weight. So, a weighted graph consider as a special type of a labeled graph in which the labels are numbers.

Example 2.3.4 The weighted graph is given in Figure 2.4.

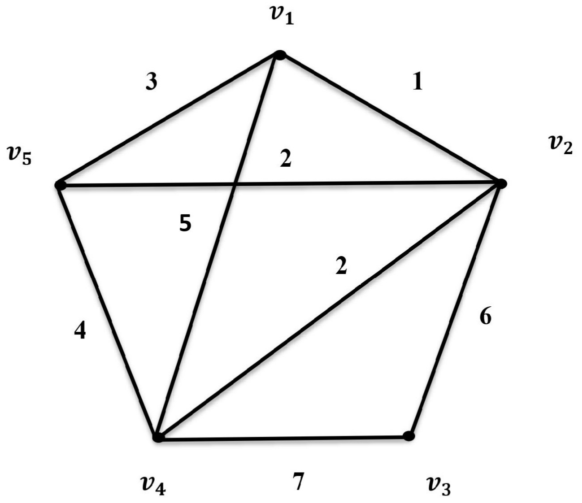

Figure 2.4: A weighted graph G.

Definition 2.3.12 (Tree) [34], p, 35.

A connected graph with no cycle is called a tree.

Definition 2.3.13 (Spanning Tree) [34], p, 40.

A tree T is called a spanning tree of a connected graph G if T is a subgraph of G and if T contains all the vertices of G. In other words, a spanning tree of a graph G is a spanning subgraph of G that is a tree.

Definition 2.3.14 (Minimum Spanning Tree)[34], p, 68.

Let G be a weighted graph in which each edge e has been assigned a real number $w(e)$, called the weight of the edge e. If H be a subgraph of a weighted graph, the weight $w(H)$ of H, is the sum of the weights $w(e_1) + w(e_2) + ... + w(e_k)$, where $e_1, e_2, ..., e_k$ is the set of edges of H.

A spanning tree T of a weighted graph G is called a minimal spanning tree if its weight is minimum. In other words, $w(T)$ is minimum, where $w(T) = w(e_1) + w(e_2) + ... + w(e_k)$ and $e_1, e_2, ..., e_k$ is the set of edges of T. Algorithm (1) is used for finding the minimal spanning tree.

Algorithm 1 Kruskal's Algorithm.

```
Let $G = (V, E)$ be a weighted connected graph.
Step-1: Select one edge $e_i$ of $G$ such that its weight $w(e_i)$ is minimum.
Step-2:
  1: If edges $e_1, e_2, ..., e_k$ have been chosen, then select an edge $e_{k+1}$ such that $e_{k+1} \neq e_i$ for $i = 1, 2, ..., k$.
  2: The edges $e_1, e_2, ..., e_k, e_{k+1}$ does not form a circuit.
  3: The weight of $w(e_{k+1})$ is as small as possible subject to the condition number 2 of step-2 above.
  4: Step-3: Stop, when all the vertices of $G$ are in $T$ which is the required spanning tree of $G$ with $n - 1$ edges.
```

Example 2.3.5 Figure 2.5 shows how determining the minimal spanning tree of the weighted graph in Figure 2.4 by using Kruskal's algorithm.

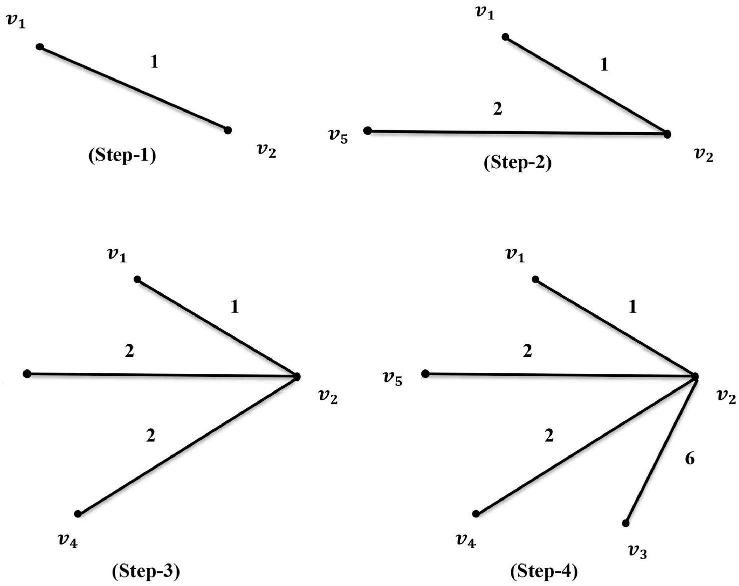

Figure 2.5: A step by step to find a minimal spanning tree.

Proposition 2.3.1 The number of spanning trees of K_n is n^{n-2} [44], p, 50.

Definition 2.3.15 (Directed graph) [44], p, 100.

A directed graph (digraph), $D = (V, Ar)$ consists of a nonempty finite set $V(D)$ of elements called vertices, and a finite family $Ar(D)$ of ordered pairs of elements of $V(D)$ called arcs. The digraph is given in Figure 2.6.

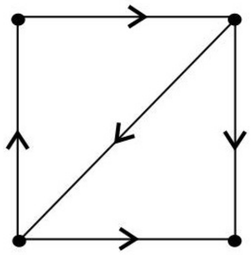

Figure 2.6: The digraph.

Definition 2.3.16 (The Adjacency Matrix of Digraph) [26].

The adjacency matrix, denoted by $A(D) = [a_{jk}]$, of the digraph $D = (V, Ar)$ is defined by:

$$a_{jk} = \begin{cases} 1, & \text{if } \; v_i v_k \in Ar \\ 0, & \text{otherwise.} \end{cases}$$

Example 2.3.6 The symmetric adjacency matrix of digraph that is given in Figure 2.6 is computed by:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Definition 2.3.17 (Complete symmetric digraph) [14].

A digraph $D = (V, Ar)$ is said to be complete symmetric if both uv and $vu \in Ar$, for all $u, v \in V$ and is denoted by K_n^*. The complete symmetric digraph K_4^* is given in Figure 2.7.

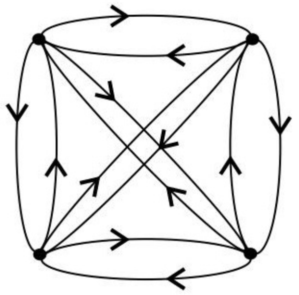

Figure 2.7: The complete symmetric digraph.

Definition 2.3.18 (Soft set) [32].

A pair (F, A) is called soft set over U, where $A \subseteq U$, F is a set-valued function $F : A \to P(U)$. In other words, a soft set over U is a parameterized family of subsets of U. For any $\epsilon \in A$, $F(\epsilon)$ may be considered as a set of $\epsilon - approximated$ elements of soft set (F, A).

Definition 2.3.19 (Soft graphs)[6].

A 4-tuple $G = (G^*, F, K, A)$ is called a soft graph if it satisfies the following conditions:

- $G^* = (V, E)$ is a simple graph,
- A is a nonempty set of parameters,
- (F, A) is a soft set over V,
- (K, A) is a soft set over E,
- $(F(a), K(a))$ is a subgraph of G^*, for all $a \in A$.

The subgraph $(F(a), K(a))$ is denoted by $H(a)$ for convenience. A soft graph can also be represented by:

$$G = <F, K, A> = \{H(x) : x \in A\}.$$

Example 2.3.7 Consider a graph G as shown in Figure 2.8.

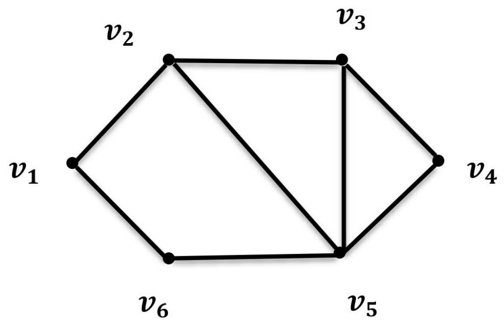

Figure 2.8: The simple graph G^*.

Let $A = \{v_4, v_5\} \subseteq V$ and (F, A) be a soft set over V with approximated function $F : A \to P(V)$ that is defined by:

$$F(x) = \{y \in V : xRy \leftrightarrow d(x, y) = 1\},$$

for all $x \in A$. That is, $F(v_4) = \{v_3, v_5\}$ and $F(v_5) = \{v_2, v_3, v_4, v_6\}$. Let (K, A) be a soft set over E with its approximate function $K : A \to P(E)$ defined by:

$$K(x) = \{uv \in E : \{u, v\} \subseteq F(x)\}$$

for all $x \in A$. In other words, we have $K(v_4) = \{v_3v_5\}$ and $K(v_5) = \{v_2v_3, v_3v_4\}$. Thus, the soft graph as shown in Figure 2.9.

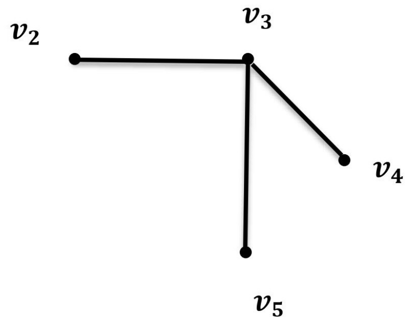

Figure 2.9: The soft graph.

Definition 2.3.20 (Vector space associated with a graph).

The vector space can be associated with a graph over finite field modulo 2 as presented in [34], p, 95.

The graph over finite field modulo p can be represented as well. Let $G(V, E)$ be a finite graph. Any subgraph H of G can be represented by a e-tuple. $X = (x_1, x_2, x_3, ..., x_e)$ such that

$$\begin{cases} x_i \in \{1, 2, ...p - 1\}, \text{if } e_i \text{ is in } H \\ x_i = 0, \text{if } e_i \text{ is not in } H \end{cases}$$

There is a vector space W_G associated with $G(V, E)$ and this vector space consists of

1. Finite field modulo p, F_p. In other words, the set $\{0, 1, 2, ..., p - 1\}$ with operation addition and multiplication modulo p.

2. There are p^e vectors (e-tuples), where e is the number of edges in G.

3. Let $X = (x_1, x_2, ..., x_e)$ and $Y = (y_1, y_2, ..., y_e) \in W_G$, then the vector sum is defined by:

$$X +_p Y = (x_1 + y_1, x_2 + y_2, ..., x_e + y_e)$$

where $+_p$ is the addition modulo p.

4. Suppose a scalar $c \in F_p$ in F_p and a vector $X = (x_1, x_2, ..., x_e) \in W_G$. A scalar multiplication is defined by

$$c \cdot_p X = (c \cdot x_1, c \cdot x_2, ..., c \cdot x_e),$$

where $\cdot_p$ is the multiplication modulo p.

Example 2.3.8 There are $7^3 = 343$ weighted sub graph of a graph G in Figure 2.1 over F_7 For instance, the subgraph H in Figure 2.10 of graph G will be represented by $(1, 0, 1, 0, 1), (1, 0, 1, 0, 2), ..., (6, 0, 6, 0, 6)$

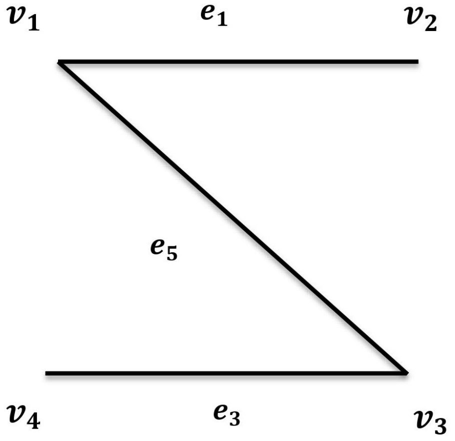

Figure 2.10: The subgraph H of graph G in Figure 2.1.

2.4 Cryptography

In this section, some basic concepts of cryptography and cryptosystems, are discussed.

2.4.1 Basic Concepts of Cryptography

Definition 2.4.1 (Symmetric Key Cryptosystem) [20].

In a symmetric key cryptosystem the sender and receiver of a ciphertext have a same key for both encryption and decryption process. This key is known as a secret key.

Definition 2.4.2 (Asymmetric Key Cryptosystem) [20].
In an asymmetric key cryptosystem, there are two keys used for the encryption and decryption of data. One of these keys is known to everybody. This key is called a public key. Where as, another key is kept secret which is called a private key.

2.4.2 The Cryptosystems based on the Discrete Logarithm Problem

The first discrete logarithm problem (DLP) was the key agreement protocol proposed by Diffie and Hellman in 1976 [15].

Definition 2.4.3 (The Discrete Logarithm Problem) [20], p, 62.
Let g be a generator element of F_p suppose h be a nonzero element in F_p. The DLP is the problem for finding an exponent x such that:

$$g^x \equiv h \ (mod \ p).$$

The number x is called the discrete logarithm of h to the base g and it is denoted by $DLP_g(h)$.

Example 2.4.1 The number $p = 56509$ is a prime, and one can check that $g = 2$ is a generator modulo p. The discrete logarithm x that gives a correct value $h = 38679$ is computed by:

$$2^2, 2^3, 2^4, ... \ (mod \ 56509)$$

until a power that equals to 38679. It would be difficult to do this by hand, but using a computer, the $DLP_g(h) = 11235$. This can be verified by calculating 2^{11235} $(mod \ 56509)$ and checking that which is equal to 38679.

The Diffie – Hellman Key Exchange [15].

The Diffie – Hellman key exchange (DHKE) has been proposed in 1976 by W. Diffie and M. Hellman. This kind of the key exchange depended on the DLP. So, it first requires to explain in this problem in the following definition.

Definition 2.4.4 Two entities, Alice and Bob are agreed to choose the public domain parameters of for achieving the DHKE. These parameters are: a large prime p and a generator element g over a prime field F_p. The entities choose the secret integers, a and b to compute the public keys A and B respectively.

$$A \equiv g^a (mod\ p) \text{ and } B \equiv g^b (mod\ p).$$

Alice sends A to Bob and Bob sends B to Alice. Then, Finally, Bob and Alice again use their secret integers to compute:

$$K \equiv B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b\ (mod\ p)$$

as a shared secret key.

Example 2.4.2 The DHKE can be explained as follows. Alice and Bob agree to use the prime $p = 9967$ and the generator $g = 7797\ (mod\ 9967)$. Alice chooses her secret key $a = 845$ and computes her public key $A = 3847 \equiv 7797^{845}\ (mod\ 9967)$. Similarly, Bob chooses his secret key $b = 5525$ and computes his public key $B = 8643 \equiv\ 7797^{5525}\ (mod\ 9967)$. Alice sends the number 3847 to Bob and he sends the number 8643 to Alice. Then, both of them able to compute the number $3692 \equiv 7797^{845 \cdot 5525} \equiv A^{5525} \equiv B^{845}\ (mod\ 9967)$ which is consider the encrypted key.

The RSA Public Key Cryptosystem [35].

The RSA cryptosystem proposed in 1978 by R. L. Rivest, A. Shamir, and L. Adleman. This cryptosestem depended on the integer factorization problem. A public key on the RSA cryptosystem is a pair of integers (n, e), where n is a product of two secret primes p and q and e is an integer satisfying $gcd(e, \phi(n)) = 1$, with $\phi(n) = (p-1)(q-1)$ is Euler phi function [13], which is defined by, for $n \geq 1$, where $\phi(n)$ denote the number of positive integers not exceeding n that are relatively prime to n. Whereas, a private key d is an integer satisfying $ed \equiv 1(mod\ \phi(n))$. The ciphertext c of a plaintext m is computed by $c \equiv m^e\ (mod\ n)$. The original plaintext can be recovered through the decryption process which is computed by $m \equiv c^d\ (mod\ n)$.

The ElGamal Public Key Cryptosystem [17].

Taher Elgamal in 1985, introduced another version of the cryptosystems which are public key cryptosystems and a signature scheme that is also based on the DLP. The ElGamal public key cryptosystem (EPKC) is closely related to the DHKE. The entities agree on the public domain of parameters which are: a large prime p and a generator g $(mod\ p)$. Alice chooses a secret key a and computes a public key by g^a $(mod\ p)$. Bob selects a plaintext m and a random ephemeral key k. He uses a public key A to compute:

$$C_1 \equiv g^k\ (mod\ p) \text{ and } C_2 \equiv mA^k\ (mod\ p),$$

and sends ciphertext (C_1, C_2) to Alice. The decryption process can be done by Alice. She uses her secret key to computes $m \equiv (C_1^a)^{-1} \times C_2\ (mod\ p)$.

Definition 2.4.5 (The DLP in Matrix Groups).
Let F_p be a prime field and $A, B \in GL_n(F_p)$. The DLP is the problem of finding an exponent integer $x \in [2, p-1]$ such that:

$$A^x \equiv B\ (mod\ p). \tag{2.4}$$

The number x is called the discrete logarithm of the matrix B to the base a matrix A and is denoted by $DLP_A(B)$.

2.5 Introduction to Elliptic Curves Cryptography

The ECC was discovered separately in 1985 by Victor Miller [30] and in 1987 by Neal Koblitz [27]. Since then, researchers and mathematical scientists have been interested in using it in cryptographic applications.

2.5.1 Basic Facts of the Elliptic Curve Over finite fields

In this section, some important facts related to elliptic curves over finite fields are discuss.

Definition 2.5.1 (The Elliptic Curves Over F_p) [20], p, 286.
Let F_p be a prime filed with characteristic not equal to 2 or 3, the equation of Weierstrass is simplified by

$$y^2 \equiv x^3 + ax + b\ (mod\ p) \tag{2.5}$$

where $a, b \in F_p$ and $\triangle \equiv 4a^3 + 27b^2 \ (mod\ p) \not\equiv 0$

The number of points on the EC is denoted by $\#EC(F_p)$ which is the number of the solutions $(x, y) \in F_p \times F_p$ plus the point O_E which is a point at infinity.

Example 2.5.1 Let F_{31} be a prime field. Suppose EC is an elliptic curve defined by

$$EC : y^2 \equiv x^3 + 4x + 3 \ (mod\ 31). \tag{2.6}$$

Note that $\triangle \equiv 4a^3 + 27b^2 \ (mod\ 31) \equiv 3 \ (mod\ 31) \not\equiv 0 \ (mod\ 31)$. The points in $EC(F_{31})$ are determined by:

$$\{(1,16),(1,-16),(2,9),(2,-9),(7,8),(7,-8),(8,19),(8,-19),(10,19),(10,-19),$$
$$(11,18),(11,-18),(13,19),(13,-19),(15,20),(15,-20),(16,3),(16,-3),(19,5)$$
$$,(19,-5),(24,2),(24,-2),(27,4),(27,-4),(29,7),(29,-7),O_E\}.$$

The order of $EC(F_{31})$ is equal to 27.

Definition 2.5.2 (Point Addition on EC $(mod\ p)$) [20], p, 285.

Suppose $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, where $P \neq \mp Q$, are two points lie on an elliptic curve EC defined over F_p. Adding the two points P and Q gives a third point $R = (x_3, y_3)$ which also lies on EC, by

$$\begin{cases} x_3 \equiv (\lambda^2 - x_1 - x_2) \ (mod\ p) \\ y_3 \equiv (\lambda(-x_1 - x_3) - y_1) \ (mod\ p) \end{cases} \tag{2.7}$$

$$\text{where} \quad \lambda \equiv [\frac{y_2 - y_1}{x_2 - x_1}] \ (mod\ p)$$

Definition 2.5.3 (Point Doubling on EC $(mod\ p)$) [20], p, 293.

Suppose $P = (x_1, y_1)$ is a point on an elliptic curve EC defined over F_p. The point $Q = 2P = (x_2, y_2)$ that results from doubling the point P is computed by defined over F_p.

$$\begin{cases} x_2 \equiv (\lambda^2 - 2x_1) \ (mod\ p) \\ y_2 \equiv (\lambda(x_1 - x_2) - y_1) \ (mod\ p) \end{cases} \tag{2.8}$$

$$\text{where} \quad \lambda \equiv [\frac{3x_1^2 + a}{2y_1}] \ (mod\ p)$$

Q should lies on the curve EC.

Example 2.5.2 (Points Addition and Doubling on $EC\ (mod\ p)$) .

With the same parameters in the Example (2.5.1), suppose the points $P = (10, 19)$ and $Q = (24, 29) \in EC(F_{31})$. Based on the relation in Equation (2.7), the computation of

$$P + Q = (7, 23),$$

which is in $EC(F_{31})$. Whereas, using the relation in Equation (2.8), the computation of $2P$ can be done by

$$2P = 2(10, 19) = (13, 19) \in EC(F_{31}).$$

Definition 2.5.4 (Elliptic Curve Scalar Multiplication) [20], p, 291.

Suppose P is a point on an elliptic curve EC defined over F_p which has a prime order n. Assume k is a positive integer, $k \in [1, n-1]$. The elliptic curve scalar multiplication operation is defined by

$$kP = \underbrace{P + P + ... + P}_{k \text{ times}} \tag{2.9}$$

Definition 2.5.5 (Order of Elliptic Point) [20].

The order of a point P on E defines by the smallest positive integer n such that $nP = O_E$, where O_E is a point at infinity.

Example 2.5.3 Based on the parameters of Example (2.5.1), the order of a point $P = (10, 19)$ is equal to 27, since $27P = O_E$.

Definition 2.5.6 (Elliptic Curve Discrete Logarithm Problem) [20], p, 295.

Let EC be an elliptic curve over the finite field F_p and let P and Q be points in $EC(F_p)$. The elliptic curve discrete logarithm problem (ECDLP) is the problem of finding an integer n such that $Q = nP$. We denote this integer n by

$$n = log_P(Q)$$

and we call n the elliptic discrete logarithm of Q with respect to P.

2.5.2 Elliptic Curve Cryptosystems

This section discusses the elliptic curve cryptosystem as follows:

Elliptic Diffie–Hellman Key Exchange (ECDH)[20], p, 296.

Alice and Bob agree to use a particular elliptic curve $EC(F_p)$ and a particular point $P \in EC(F_p)$. Alice chooses a secret integer n_A and Bob chooses a secret integer n_B. They compute the associated multiples, Alice computes this $Q_A = n_A P$ and Bob computes this $Q_B = n_B P$, and they exchange the values of Q_A and Q_B. Alice then uses her secret multiplier to compute $n_A Q_B$, and Bob similarly computes $n_B Q_A$. They now have the shared secret value $K = n_A Q_B = (n_A n_B)P = n_B Q_A$.

Example 2.5.4 Alice and Bob decide to use elliptic Diffie–Hellman with the following curve, and point: $EC(F_{1997}) = y^2 = x^3 + 5x + 2$, $P = (263, 249) \in EC(F_{1997})$. Alice and Bob choose respective secret values $n_A = 1192$ and $n_B = 1987$, and then Alice and Bob computes respective,

$$Q_A = 1192P = (1690, 304),$$

$$Q_B = 1987P = (1275, 1474),$$

Alice sends Q_A to Bob and Bob sends Q_B to Alice. Finally, Alice and Bob computes respective,

$$n_A Q_B = 1192(1275, 1474) = (1864, 1692),$$

$$n_B Q_A = 1987(1690, 304) = (1864, 1692).$$

Elliptic ElGamal Public Key Cryptosystem (EEPKC) [20], p, 299.

Alice and Bob agree to use a particular elliptic curve $EC(F_p)$ and a particular point $P \in EC(F_p)$. Alice chooses a secret multiplier n_A and publishes the point $Q_A = n_A P$ as her public key. Bob's plaintext is a point $M \in EC(F_p)$. He chooses an integer k to be his ephemeral key and computes

$$C_1 = kP$$

and

$$C_2 = M + Q_A.$$

He sends the two points (C_1, C_2) to Alice.

Finally, Alice computes

$$C_2 - n_A C_1 = M$$

to recover original plaintext.

2.6 Introduction to Edwards Curves

Several mathematicians have studied elliptic curves for over a hundred years. They met to solve a wide variety of mathematical problems. Edwards curves are a family of elliptic curves which are also used for cryptographic schemes. In 2007, Edwards proposed a new normal form for elliptic curves and gave an addition law of the points lying on it [16].

Definition 2.6.1 (Edwards Curves) [10], p, 21.
An Edwards curve over K, with $CharK \neq 2$ is a curve given by

$$x^2 + y^2 = 1 + dx^2y^2 \tag{2.10}$$

where $d \in K \backslash \{0, 1\}$

Example 2.6.1 Let E_7 be the Edwards curve $x^2 + y^2 = 1 + 7x^2y^2$ over F_{13}. The points on E_7 form a set

$$E_7(F_{13}) = \{(0,1),(0,12),(1,0),(12,0),(2,4),(11,4),(2,9),(11,9),(4,2),(9,2),(4,11),\\(9,11),(5,6),(8,6),(5,7),(8,7),(6,5),(7,5),(6,8),(7,8)\}$$

Definition 2.6.2 Point Addition [16].
The addition law for any two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ in $E_d(F_p)$ is defined by

$$P + Q = (\frac{x_1y_2 + x_2y_1}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2}) \tag{2.11}$$

Definition 2.6.3 Point Doubling [16].
Suppose $P = (x_1, y_1)$ is a point on an Edwards Curve $E_d(F_p)$, the point $Q = 2P = (x_2, y_2)$ that results from doubling the point P as:

$$\begin{cases} x_2 \equiv \dfrac{2x_1y_1}{x_1^2 + y_1^2} \ (mod\ p) \\ y_2 \equiv \dfrac{y_1^2 - x_1^2}{2 - (x_1^2 + y_1^2)} \ (mod\ p) \end{cases} \tag{2.12}$$

Example 2.6.2 With the same parameters in the Example (2.6.1), suppose the points $P = (9, 11)$ and $Q = (7, 5)$, based on the relation in Equation (2.11), the computation of $P + Q$ which is in $E_7(F_{13})$ such that

$$P + Q = (x_3, y_3),$$

where

$$\begin{aligned}
x_3 &\equiv \frac{x_1y_2 + x_2y_1}{1 + dx_1x_2y_1y_2}\ (mod\ 13) \\
&\equiv \frac{(9)(5) + (7)(11)}{1 + 7(9)(7)(11)(5)}(mod\ 13) \\
&\equiv 4\ (mod\ 13), \\
y_3 &\equiv \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2}\ (mod\ 13) \\
&\equiv \frac{(11)(5) - (9)(7)}{1 - 7(9)(7)(11)(5)}\ (mod\ 13) \\
&\equiv 11\ (mod\ 13), \\
P + Q &= (4, 11).
\end{aligned}$$

Whereas, using the relation in Equation (2.12), the computation of $2P$ which is in $E_7(F_{13})$ can be computed as follows.

$$2P = (x_2, y_2)$$

where

$$\begin{aligned}
x_2 &\equiv \frac{2x_1y_1}{x_1^2 + y_1^2}\ (mod\ 13) \\
&\equiv \frac{2(9)(11)}{(9)^2 + (11)^2}\ (mod\ 13) \\
&\equiv 6\ (mod\ 13),
\end{aligned}$$

$$\begin{aligned}
y_2 &\equiv \frac{y_1^2 - x_1^2}{2 - (x_1^2 + y_1^2)}\ (mod\ 13) \\
&\equiv \frac{(11)^2 - (9)^2}{2 - ((9)^2 + (11)^2)}\ (mod\ 13) \\
&\equiv 5\ (mod\ 13), \\
2P &= (6, 5).
\end{aligned}$$

Chapter 3

The Use of Graphs for Some Cryptosystems

3.1 Introduction

This chapter uses graph theory in cryptography after representing the graph as a matrix, because of the importance of using it in cryptography. One of the most famous uses of matrices in cryptography is the Hill Cipher Algorithms [19]. In Section (3.2), an alternative version of the ElGamal public key cryptosystem has been proposed by using the power of matrices in $GL_n(F_p)$. In Section (3.3), a new public key cryptosystem has been designed based on an undirected complete graph (UCG). The plaintext data is selected as a subset of the finite fields which corresponds to an undirected complete graph $G(V, E)$, where V and E are finite sets of the vertices and edges respectively. The plaintext is encrypted using the matrix power (MP) and the minimum spanning tree (MST) problem of UCG. In Section (3.4), encrypt a vector of vector space $(F_p)^n$ by using the matrices. In Section (3.5), a plaintext is a sub graph by representation of a graph in vector space. In Section (3.6), alternative versions of the DLP cryptosystem have been proposed. The first proposition used the application of the MPF concept to modify the Diffie-Hellman key exchange problem and the ElGamal cryptosystem. In the MPF Diffie-Hellman key exchange and MPF-ElGamal cryptosystem, the left L and right R matrices are chosen secretly. Another proposition is to employ graph theory with the MPF concept to draw a hybrid version of an asymmetric cryptosystem. The new alternative versions are considered more secure compared with the original ones. In Section (3.7), used the soft graph with the matrix power function to encrypt a subset of finite fields by representation of a weighted edge subgraph. In Section (3.8), discussed the security cases of the proposed schemes. In Section (3.9), discussed the summary.

3.2 The Matrix Power ElGamal Public Key Cryptosystem

In this section, the alternative version of the ElGamal public key cryptosystem (EPKC) has been proposed. The revised EPKC depended on an extended discrete logarithm problem over matrices, given in Definition (2.4.5). The proposed alternative version of EPKC based on matrix power is discussed.

The public parameters of the proposed version MP-EPKC, are a prime p and a public matrix $D \in GL_n(F_p)$. Two entities Alice and Bob want to communicate for exchanging the information. Alice chooses randomly a private key as a number a such that $a \in \{2, 3, ..., p-1\}$ and computes a public key A by:

$$A \equiv D^a \ (mod\ p) \tag{3.1}$$

Bob employs a public key A of the Alice to encrypt her plaintext which is considered as a matrix $M_{n\times m}$. He chooses an ephemeral key as a number b such that $b \in \{2, 3, ..., p-1\}$. The ciphertext C which is a pair (C_1, C_2) of two matrices are computed by:

$$C_1 \equiv D^b \ (mod\ p) \ \text{ and } \ C_2 \equiv A^b \times M \ (mod\ p) \tag{3.2}$$

The ciphertext $C = (C_1, C_2)$ has been sent to the Alice.

The decryption process is done by Alice to recover the original plaintext. She first compute $(C_1^a)^{-1}$ $(mod\ p)$ based on her secret key a. The multiplication matrix $(C_1^a)^{-1} \times C_2$ $(mod\, p)$ is computed to give the original plaintext M. Algorithms (2), (3) and (4) are used for obtaining the several numerical results of the revised MP-EPKC.

Algorithm 2 The MP-EPKC: Keys Generation Process.

Input: A prime p and a matrix $D \in GL_n(F_p)$.

Output: The public key A, where $A \in GL_n(F_p)$.

```
1: Alice chooses a number $a$ as her a private key, where $a \in \{2, 3, ..., p-1\}$.
2: She computes her public key $A \equiv D^a \ (mod\ p)$.
3: Alice keys are $(A, a)$.
```

Algorithm 3 The MP-EPKC: Encryption Process.

```
Input: A prime $p$ and a public key $A$.
Output: The ciphertext $(C_1, C_2)$, where $C_1$ and $C_2$ are two matrices.
1: Bob chooses an ephemeral secret key $b$, where $b \in \{2, 3, ..p - 1\}$.
2: He chooses his plaintext a matrix $M_{n\times m}$.
3: He computes the ciphertext through the computations of two matrices $C_1 \equiv D^b \ (mod\ p)$ and $C_2 \equiv A^b \times M_{n\times m} \ (mod\ p)$.
4: Bob sends the ciphertext pair $(C_1, C_2)$ to Alice.
```

Algorithm 4 The MP-EPKC: Decryption Process.

```
Input: A prime $p$ and a secret key $a$.
Output: The plaintext $M_{n\times m}$, where $M_{n\times m}$ is a matrix of size $n \times m$.
1: Alice first uses her secret key $a$ to compute $C_1^a \ (mod\ p)$.
2: She computes the matrix inverse $(C_1^a)^{-1} \ (mod\ p)$.
3: She computes the multiplication matrix $(C_1^a)^{-1} \times C_2 \ (mod\ p) \equiv M_{n\times m} \ (mod\ p)$.
```

Example 3.2.1 Alice and Bob agree to use the prime $p = 73$ and D a matrix of size 4×4 as a public parameters

$$D = \begin{pmatrix} 6 & 6 & 60 & 72 \\ 23 & 46 & 38 & 39 \\ 60 & 14 & 22 & 54 \\ 32 & 11 & 45 & 64 \end{pmatrix},$$

where $D \in GL_4(F_{73})$.

Keys generation process.

Alice performs the following steps:

- She chooses her secret key $a = 5$ and computes her public key

$$A = D^5 \ (mod\ 73) \equiv \begin{pmatrix} 44 & 1 & 49 & 44 \\ 17 & 27 & 50 & 65 \\ 48 & 65 & 44 & 11 \\ 71 & 33 & 53 & 18 \end{pmatrix}.$$

Encryption process.

Bob does the following steps:

- He chooses his private key $b = 7$ and computes

$$C_1 = D^7 \ (mod\ 73) \equiv \begin{pmatrix} 65 & 66 & 41 & 35 \\ 26 & 45 & 58 & 65 \\ 38 & 68 & 44 & 0 \\ 39 & 69 & 43 & 61 \end{pmatrix},$$

and

$$A^7 \ (mod\ 73) \equiv \begin{pmatrix} 64 & 53 & 38 & 34 \\ 34 & 48 & 49 & 21 \\ 32 & 1 & 72 & 13 \\ 19 & 4 & 53 & 62 \end{pmatrix}.$$

- He chooses his plaintext

$$M = \begin{pmatrix} 7 & 33 & 50 & 17 & 66 \\ 22 & 71 & 58 & 3 & 60 \\ 2 & 61 & 55 & 44 & 61 \\ 55 & 17 & 25 & 13 & 14 \end{pmatrix},$$

- He computes

$$C_2 = A^7 \times M \ (mod\ 73) \equiv \begin{pmatrix} 56 & 11 & 59 & 3 & 51 \\ 65 & 65 & 8 & 12 & 12 \\ 10 & 46 & 50 & 15 & 30 \\ 14 & 15 & 61 & 42 & 47 \end{pmatrix},$$

- He sends his ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She computes

$$(C_1^5)^{-1} \ (mod\ 73) \equiv \begin{pmatrix} 43 & 1 & 65 & 59 \\ 38 & 40 & 18 & 16 \\ 43 & 67 & 60 & 53 \\ 63 & 67 & 23 & 71 \end{pmatrix},$$

where $C_1^5 = (D^7)^5 = (D^5)^7 = A^7$,

- She computes

$$(C_1^5)^{-1} \times C_2 \; (mod\; 73) \equiv \begin{pmatrix} 7 & 33 & 50 & 17 & 66 \\ 22 & 71 & 58 & 3 & 60 \\ 2 & 61 & 55 & 44 & 61 \\ 55 & 17 & 25 & 13 & 14 \end{pmatrix} \equiv M.$$

More implemented results of MP-EPKC can be seen in Chapter (5).

3.3 The Undirected Complete Graph based Public Key Cryptosystem

In this section, a new public key cryptosystem has been designed based on the idea of MP-EPKC and employing the undirected complete graph. This cryptosystem is called the UCG public key cryptosystem. Two entities agreed to choose the public parameters. These parameters are a prime p and a square matrix $D \in GL_n(F_p)$. Alice generates the keys, public and private keys. She selects a number a in a secret way, where $a \in \{2, 3, ..., p-1\}$ which is a her private key. Depending on her private key a, she computes her public key $A \equiv D^a \; (mod\; p)$. So Alice's keys are given with a pair (A, a). For obtaining the several numerical results to generate the keys, one can use Algorithm (2).

Bob wants to communicate with Alice for sending the important information which is represented by a plaintext m that is a subset of a prime field F_p. Bob creates UCG $G(V, E)$ of a plaintext $m = \{m_1, m_2, ..., m_l\}$, where are vertices and edges sets of a graph G respectively, $V = \{v_1, v_2, ..., v_l\}$ and $E = \{e_1, e_2, ..., e_s\}$. Then the graph G is converted into a weighted graph through computing the distance between any two vertices. The MST graph G' of a graph G is determined. The MST graph G' is represented by a matrix M. Later on, a matrix M is converted into M' by adding the elements of m to the diagonal of a matrix M respectively. Bob chooses an ephemeral secret key b, where $b \in \{2, 3, ..., p-1\}$. The ciphertext C is computed through the computations of two square matrices $C_1 \equiv D^b \; (mod\; p)$ and $C_2 \equiv A^b \times M' \; (mod\; p)$. Bob sends the ciphertext pair $C = (C_1, C_2)$ to Alice. Several numerical results to compute the ciphertext are got using Algorithm (5).

Algorithm 5 The UCG Public Key Cryptosystem: Encryption Process.

```
Input: A prime $p$ and a public key $A$.
Output: The ciphertext $(C_1, C_2)$, where $C_1$ and $C_2$ are two matrices of size $n \times n$.
1: Bob selects his plaintext $m = \{m_1, m_2, ..., m_l\}$ which is choosed randomly as a subset of $F_p$.
2: He forms a complete graph $G(V, E)$ of a plaintext $m$, where $V = \{v_1, v_2, ..., v_l\}$ and $E = \{e_1, e_2, ..., e_s\}$. such that every vertices of $V$ has a code number of $m$ respectively.
3: He converts a complete graph $G$ into a weighted graph through computing the weights $w_i$, for $i = 1, 2, ...$, for all edges, which equal to the distance
       $w_i = |Code(v_i) - Code(v_i + 1)|, \; i = 1, 2, ..., n - 1.$
4: He determines MST $G'$ of a graph $G$.
5: He represents $G'$ by its corresponding matrix $M$.
6: He converts a matrix $M$ into $M'$ by adding the elements of $m$ to diagonal of a matrix $M$ respectively.
7: Bob chooses an ephemeral secret key $b$, where $b \in \{2, 3, ..., p - 1\}$.
8: He computes the ciphertext through the computations of two square matrices $C_1 \equiv D^b \ (mod \ p)$ and $C_2 \equiv A^b \times M' \ (mod \ p)$.
9: Bob sends the ciphertext pair $(C_1, C_2)$ to Alice.
```

Upon receiving Alice the ciphertext $C = (C_1, C_2)$, she uses her private key a to recover the plaintext. She first computes a matrix $C_1^a \ (mod \ p)$. Then, an inverse matrix $(C_1^a)^{-1} \ (mod \ p)$ is calculated. After that, the computation of the multiplication matrix $(C_1^a)^{-1} \times C_2 \equiv M' \ (mod \ p)$ is done. The diagonal elements, in M', are selected as a list which they form a set of the elements that gives the plaintext m. The decryption process of the proposed UCG-public key cryptosystem is proved as follows.

Proposition 3.3.1 The decryption process is computed by $(C_1^a)^{-1} \times C_2 \equiv M' \ (mod\ p)$.

Proof.

$$
\begin{aligned}
(C_1^a)^{-1} \times C_2 &\equiv ((D^b)^a)^{-1} \times C_2 \ (mod\ p), \ \text{[since } \ C_1 = D^b\text{]} \\
&\equiv (D^{ba})^{-1} \times (A^b \times M' \ (mod\ p)), \ \text{[since } \ C_2 = A^b \times M'\text{]} \\
&\equiv (D^{ba})^{-1} \times ((D^a)^b \times M' \ (mod\ p)), \ \text{[since } \ A = D^a\text{]} \\
&\equiv ((D^{ba})^{-1} \times (D^{ab}) \ (mod\ p)) \times M', \\
&\equiv I \times M' \ (mod\ p) \\
&\equiv M' \ (mod\ p) \qquad \blacksquare
\end{aligned}
$$

The implemented results for recovering the original plaintext can be got using Algorithm (6).

```
Algorithm 6 The UCG-Public Key Cryptosystem: Decryption Process
Input: A prime $p$, a secret key $a$ and a ciphertext $(C_1, C_2)$
Output: The plaintext $m$, where $m$ is a subset of $F_p$.
1: Alice first uses her secret key $a$ to compute $C_1^a \ (mod\ p)$.
2: She computes the multiplication matrix $(C_1^a)^{-1} \times C_2 \ (mod\ p) \equiv M' \ (mod\ p)$.
3: The diagonal elements of $M'$ are selected as a list which they form a set of the
   original plaintext elements $m$.
```

Example 3.3.1 Alice and Bob agree to use the a prime $p = 41$ and a matrix D of size 4×4

$$
D = \begin{pmatrix} 10 & 5 & 33 & 1 \\ 25 & 17 & 8 & 12 \\ 15 & 18 & 23 & 5 \\ 2 & 39 & 13 & 3 \end{pmatrix},
$$

where $D \in GL_4(F_{41})$, the procedure of the UCG-public key cryptosystem has been done as follows.

Keys generation process.

Alice performs the following:

- She chooses her secret key $a = 13$ and she computes her public key

$$A = D^{13}\ (mod\ 41) \equiv \begin{pmatrix} 28 & 9 & 2 & 31 \\ 3 & 17 & 34 & 25 \\ 15 & 18 & 30 & 17 \\ 12 & 31 & 19 & 22 \end{pmatrix},$$

  so, her keys are $a = 13$ and $A = D^{13}$.

Encryption process.

Bob does the following steps:

- He chooses his private key $b = 25$ and he computes

$$C_1 = D^{25}\ (mod\ 41) \equiv \begin{pmatrix} 35 & 5 & 11 & 20 \\ 30 & 14 & 14 & 22 \\ 16 & 40 & 13 & 30 \\ 21 & 3 & 33 & 17 \end{pmatrix}$$

  and

$$A^{25}\ (mod\ 41) \equiv \begin{pmatrix} 22 & 20 & 16 & 37 \\ 11 & 14 & 31 & 36 \\ 19 & 5 & 32 & 5 \\ 9 & 24 & 36 & 25 \end{pmatrix}.$$

- He chooses randomly his plaintext $m = \{39, 19, 23, 5\}$ as a subset of a prime field F_{41}.

- The plaintext m is represented by the UCG as shown in Figure 3.1.

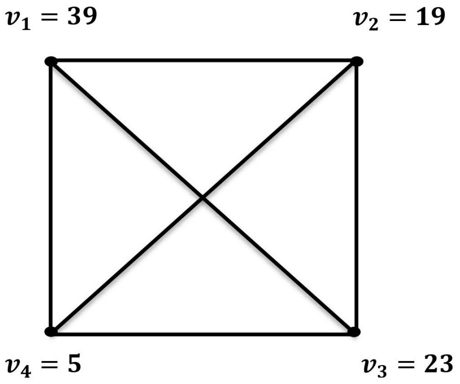

Figure 3.1: The UCG that corresponds to the plaintext.

- The weights of all edges on the UCG are computed as shown Figure 3.2.

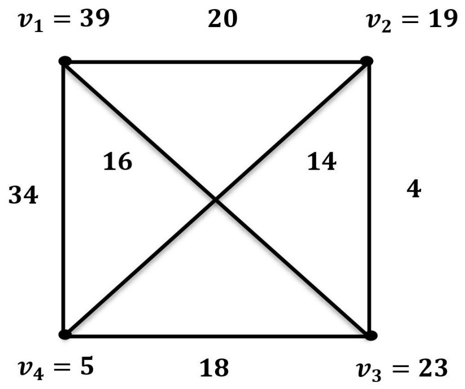

Figure 3.2: The weighted UCG.

- The MST sub-graph is computed as shown in Figure 3.3.

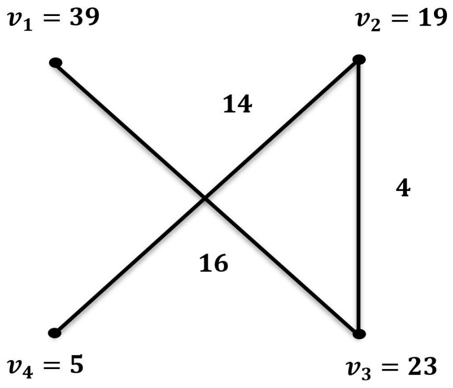

Figure 3.3: The MST sub-graph of the weighted UCG.

- The MST sub-graph is represented by the following matrix

$$M = \begin{pmatrix} 0 & 0 & 16 & 0 \\ 0 & 0 & 4 & 14 \\ 16 & 4 & 0 & 0 \\ 0 & 14 & 0 & 0 \end{pmatrix}.$$

- The modified matrix M' of M has been done by adding the elements of plaintext at the diagonal of it. So a matrix M' is given by

$$M' = \begin{pmatrix} 39 & 0 & 16 & 0 \\ 0 & 19 & 4 & 14 \\ 16 & 4 & 23 & 0 \\ 0 & 14 & 0 & 5 \end{pmatrix}.$$

- Bob computes

$$C_2 = A^{25} \times M' \ (mod\ 41) \equiv \begin{pmatrix} 7 & 19 & 21 & 14 \\ 23 & 33 & 2 & 7 \\ 23 & 6 & 35 & 13 \\ 25 & 7 & 2 & 10 \end{pmatrix}.$$

- He sends his ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She computes

$$(C_1^{13})^{-1} \ (mod\ 41) \equiv \begin{pmatrix} 29 & 13 & 28 & 0 \\ 27 & 8 & 28 & 20 \\ 21 & 22 & 29 & 20 \\ 40 & 15 & 0 & 16 \end{pmatrix},$$

  where $C_1^{13} = (D^{25})^{13} = (D^{13})^{25} = A^{25}$.

- She computes

$$(C_1^{13})^{-1} \times C_2 \ (mod\ 41) \equiv \begin{pmatrix} 39 & 0 & 16 & 0 \\ 0 & 19 & 4 & 14 \\ 16 & 4 & 23 & 0 \\ 0 & 14 & 0 & 5 \end{pmatrix} = M',$$

- The elements of diagonal in a matrix M' is the original plaintext $m = \{39, 19, 23, 5\}$.

More implemented results of the UCG public key cryptosystem can be seen in Chapter (5).

3.4 The n-Dimension Public Key Cryptosystem

A new public key cryptosystem has been designed in this section. The vector space V of n-dimension are employed for this design. Alice and Bob agreed to choose the public parameters. These parameters are a prime p, a square matrix $D \in GL_n(F_p)$ and a random vector v of

$$V = \{(x_1, x_2, ..., x_n) : x_i \in F_p, i = 1, ..., n\} \tag{3.3}$$

such that V is a $n-$dimensional vector space over the field F_p. Alice generates her keys, the public and private keys. She selects a number a in a secret way, where $a \in \{2, 3, ..., p-1\}$ which her a private key. Depending on her private key a, Alice computes her public key $A \equiv D^a$ $(mod\ p)$. So, Alice's keys are given with a pair (A, a). Algorithm (2) is used for obtaining the several numerical results to generate the Alice's keys.

Bob wants to communicate with Alice and send the plaintext m as a vector of a vector space $(F_p)^n$ defined over a prime field F_p. Bob chooses an ephemeral secret key b, where $b \in \{2, 3, ..., p-1\}$. He computes A^b and the transformation T of a matrix A^b. The vector v is used to compute a vector $v^\star$ through the transformation relation by

$$v^\star \equiv T(v)\ (mod\ p). \tag{3.4}$$

He chooses his plaintext m and converts it into $m^\star$ by

$$m^\star \equiv m +_p v^\star. \tag{3.5}$$

The ciphertext $C = (C_1, C_2)$ is computed by

$$C_1 \equiv D^b\ (mod\ p) \text{ and } C_2 \equiv T(m^\star)\ (mod\ p). \tag{3.6}$$

Bob sends the ciphertext pair $C = (C_1, C_2)$ to Alice. Several numerical results to compute the ciphertext are got using Algorithm (7).

Algorithm 7 The n-Dimension Public Key Cryptosystem: Encryption Process.

```
Input: A prime $p$ and a public key $A$.
Output: The ciphertext $(C_1, C_2)$, where $C_1$ is a square matrix and $C_2$ is a vector.
1: Bob chooses an ephemeral secret key $b$, where $b \in \{2, 3, ...p-1\}$.
2: He computes a matrix $C_1 \equiv D^b\ (mod\ p)$.
3: He chooses his plaintext $m$ as a vector belong to a vector space $V$.
4: He computes $A^b\ (mod\ p)$ .
5: He computes the transformation $T$ of a matrix $A^b$.
6: He computes $v^\star$ by $v^\star = T(v)$, where $v$ is a public vector.
7: He converts a plaintext $m$ into $m^\star$ computing $m^\star \equiv m +_p v^\star$.
8: He computes a vector $C_2 = T(m^\star)$.
9: He sends the ciphertext $(C_1, C_2)$ to Alice.
```

Upon receiving Alice the ciphertext $C = (C_1, C_2)$, she uses her private key a and the public vector v to recover the plaintext. She first computes a matrix $C_1^a\ (mod\ p)$. Then, an inverse matrix $(C_1^a)^{-1}\ (mod\ p)$ is calculated. She computes the transformation T of a matrix C_1^a and $T^\star$ of a matrix $(C_1^a)^{-1}$ respectively. After that, she computes $T(v)\ (mod\ p)$ and $T^\star(C_2)\ (mod\ p)$. Finally, the computation $T^\star(C_2) -_p T(v)$ is equal to m. The decryption process of the n-dimension public key cryptosystem is proved as follows.

Proposition 3.4.1 The decryption process of n-dimension public key cryptosystem is computed by

$$T^\star(C_2) -_p T(v) \equiv m. \tag{3.7}$$

Proof.

$$\begin{aligned}
T^\star(C_2) -_p T(v) &\equiv C_2 \times (C_1^a)^{-1} -_p v^\star, [\text{since } T^\star \text{ is transformation over } (C_1^a)^{-1} \text{ and } T(v) = v^\star] \\
&\equiv T(m^\star) \times ((D^b)^a)^{-1} -_p v^\star, [\text{since } C_2 = T(m^\star) \text{ and } C_1 = D^b] \\
&\equiv (m^\star \times A^b) \times (D^{ab})^{-1} -_p v^\star, [\text{since } T \text{ is transformation over } A^b] \\
&\equiv (m^\star \times (D^{ab})) \times (D^{ab})^{-1} -_p v^\star, [\text{since } A = D^a] \\
&\equiv m^\star \times ((D^{ab}) \times (D^{ab})^{-1}) -_p v^\star, \\
&\equiv m^\star \times I -_p v^\star \\
&\equiv m^\star -_p v^\star \\
&\equiv m\ (mod\ p). \qquad \blacksquare
\end{aligned}$$

More computational results can be got using Algorithm (8).

Algorithm 8 The n-Dimension Public Key Cryptosystem: Decryption Process.

```
Input: A prime $p$ , a secret key $a$, a ciphertext $(C_1, C_2)$ and a public vector $v$.
Output: The plaintext $m$.
1: Alice first uses her secret key $a$ to compute $C_1^a$ $(mod\ p)$.
2: She computes the matrix inverse $(C_1^a)^{-1}$ $(mod\ p)$.
3: She computes the transformations $T$ of a matrix $C_1^a$ and $T^\star$ of a matrix $(C_1^a)^{-1}$.
4: She computes $T(v)$ $(modp)$ and $T^\star(C_2)$ $(mod\ p)$.
5: She computes $T^\star(C_2) -_p T(v) \equiv m$.
```

Example 3.4.1 With a prime $p = 127$, a matrix D of size 3×3

$$D = \begin{pmatrix} 101 & 117 & 57 \\ 87 & 91 & 115 \\ 74 & 18 & 47 \end{pmatrix},$$

where $D \in GL_3(F_{127})$ and $v = (85, 97, 122)$, the procedure n-dimension public key cryptosystem is discussed as follows.

Keys generation process.

Alice performs the following:

- She chooses a secret key $a = 110$ and she computes her public key

$$A = D^{110}\ (mod\ 127) \equiv \begin{pmatrix} 115 & 83 & 69 \\ 83 & 9 & 72 \\ 86 & 2 & 125 \end{pmatrix},$$

Alice's keys are $a = 110$ and $A = D^{110}$.

Encryption process.

Bob does the following steps:

- He chooses his private key $b = 117$ and he computes

$$C_1 = D^{117}\ (mod\ 127) \equiv \begin{pmatrix} 120 & 39 & 117 \\ 97 & 27 & 88 \\ 19 & 94 & 81 \end{pmatrix},$$

and

$$A^{117}\ (mod\ 127) \equiv \begin{pmatrix} 26 & 95 & 119 \\ 110 & 40 & 55 \\ 18 & 10 & 21 \end{pmatrix}.$$

- He computes the transformation T on 3-dimensional over matrix A^{117}

$$T(a,b,c) = \begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} 26 & 95 & 119 \\ 110 & 40 & 55 \\ 18 & 10 & 21 \end{pmatrix}$$

$$= (26a + 110b + 18c, 95a + 40b + 10c, 119a + 55b + 21c),$$

- He computes $v^\star = T(v) = T(85, 97, 122)\ (mod\ 127) \equiv (90, 94, 105)$,

- He chooses his plaintext $m = (26, 87, 113)$ and converts m into $m^\star = m +_{127} v^\star \equiv (116, 54, 91)$.

- He computes $C_2 = T(m^\star) = T(116, 54, 91)\ (mod\ 127) \equiv (53, 120, 16)$,

- He sends his ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She computes

  $$(C_1^{110})^{-1}\ (mod\ 127) \equiv \begin{pmatrix} 9 & 21 & 21 \\ 51 & 109 & 121 \\ 95 & 45 & 124 \end{pmatrix},$$

  where $C_1^{110} = (D^{117})^{110} = (D^{110})^{117} = A^{117}$.

- Compute the transformation T on 3-dimensional over matrices C_1^{110} and $(C_1^{110})^{-1}$,

  $$T(a,b,c) = (26a + 110b + 18c, 95a + 40b + 10c, 119a + 55b + 21c)$$

  $$T^\star(a,b,c) = (9a + 51b + 95c, 21a + 109b + 45c, 21a + 121b + 124c)$$

- Using public key v to compute

  $$v^\star = T(v) = T(85, 97, 122)\ (mod\ 127) \equiv (90, 94, 105)$$

  and compute

  $$T^\star(C_2) = T^\star((53, 120, 16))\ (mod\ 127) \equiv (116, 54, 91) = m^\star$$

- Compute $m^\star -_{127} v^\star = (116, 54, 91) -_{127} (90, 94, 105) \equiv (24, 87, 113) = m$.

Another case study of the n-dimension public Key cryptosystem can be seen in Chapter (5).

3.5 The n-UG Public Key Cryptosystem

The sub graphs of undirected graphs are used to give the generalized case of the UG public key cryptosystem. On this case the vectors over a prime filed with n-dimension are represented as the sub graphs H of undirected graph G. So, the cryptosystem here is named by the n-UG public key cryptosystem. In this cryptosystem, two entities agreed to choose the public parameters. These parameters are a prime p, a square matrix $D \in GL_n(F_p)$, undirected graph $G(V, E)$ such that the size of this graph is n and a random vector v of $V = \{(x_1, x_2, ..., x_n) : x_i \in F_p, i = 1, ..., n\}$ such that V is a $n-$ dimensional vector space over a prime field F_p. Alice generates her public and private keys. She selects a number a in a secret way, where $a \in \{2, 3, ..., p-1\}$ which is her private key. Depending on her private key a, Alice computes her public key $A \equiv D^a$ $(mod\ p)$. So Alice's keys are given with a pair (a, A). Algorithm (2) is used for obtaining the several numerical results to generate those keys.

Bob wants to communicate with Alice for exchanging the information as (the plaintext) a weighted sub graph H of undirected graph $G(V, E)$. Bob chooses an ephemeral secret key b, where $b \in \{2, 3, ..., p-1\}$. He computes A^b and the transformation T of a matrix A^b. After that, A public vector v is used to compute a vector $v^\star$. Bob converts his plaintext H into vector m and converts m into $m^\star$ by computing $m^\star \equiv m +_p v^\star$. The ciphertext $C = (C_1, C_2)$ is computed through the computations of $C_1 \equiv D^b$ $(mod\ p)$ and $C_2 \equiv T(m^\star)$ $(mod\ p)$. Bob sends the ciphertext pair $C = (C_1, C_2)$ to Alice. Several numerical results to compute the ciphertext are got using Algorithm (9).

Algorithm 9 The n-UG Public Key Cryptosystem: Encryption Process.

```
Input: A prime $p$, a public key $A$, a public vector $v$ and an undirected graph $G(V, E)$.
Output: The ciphertext $(C_1, C_2)$, where $C_1$ is a square matrix and $C_2$ is a vector.
Bob chooses an ephemeral secret key $b$, where $b \in \{2, 3, ...p - 1\}$.
He represents an undirected graph $G(V, E)$ as the vectors in vector space $V$.
He computes a square matrix $C_1 \equiv D^b \ (mod\ p)$.
He chooses his plaintext as a weighted sub graph $H$.
He converts his plaintext $H$ into $m$ as a vector in $V$.
He computes $A^b \ (mod\ p)$.
He computes the transformation $T$ of a matrix $A^b$.
He uses a public vector $v$ to compute $v^\star$ by $v^\star \equiv T(v) \ (mod\ p)$.
He converts a plaintext $m$ into $m^\star \equiv m +_p v^\star$.
He computes $C_2 \equiv T(m^\star) \ (mod\ p)$.
Bob sends the ciphertext $(C_1, C_2)$ to Alice.
```

Upon receiving Alice the ciphertext $C = (C_1, C_2)$, she uses her private key a and a public vector v to recover the plaintext m. She first computes a matrix $C_1^a \ (mod\ p)$. Then, an inverse matrix $(C_1^a)^{-1} \ (mod\ p)$ is calculated. She computes the transformations T of a matrix C_1^a and $T^\star$ of a matrix $(C_1^a)^{-1}$. After that, she computes $T(v) \ (mod\ p)$ and $T^\star(C_2) \ (mod\ p)$. Then, computing $T^\star(C_2) -_p T(v)$ is equal to m. Finally, converting m into original plaintext gives a weighted sub graph H. For more numerical results, one can use Algorithm (10).

Algorithm 10 The n-UG Public Key Cryptosystem: Decryption Process.

```
Input: A prime $p$ , a secret key $a$, ciphertext $(C_1, C_2)$ and a public vector $v$.
Output: The plaintext as a weighted sub graph $H$.
Alice first uses her secret key $a$ to compute $C_1^a \ (mod\ p)$.
She computes an inverse matrix $(C_1^a)^{-1} \ (mod\ p)$.
She computes transformations $T$ of a matrix $C_1^a$ and $T^\star$ of a matrix $(C_1^a)^{-1}$.
She computes $T(v) \ (mod\ p)$ and $T^\star(C_2) \ (mod\ p)$.
She computes $T^\star(C_2) -_p T(v) \equiv m$.
She converts $m$ to original plaintext as a weighted sub graph $H$.
```

Example 3.5.1 Let $p = 97$ be a prime number, a matrix D of size 5×5

$$D = \begin{pmatrix} 10 & 67 & 17 & 55 & 93 \\ 3 & 44 & 54 & 61 & 18 \\ 91 & 12 & 92 & 37 & 22 \\ 87 & 61 & 36 & 77 & 33 \\ 42 & 14 & 29 & 27 & 13 \end{pmatrix},$$

where $D \in GL_5(F_{97})$, $V = (16, 28, 45, 83, 95)$ and a graph G as shown in Figure 3.4.

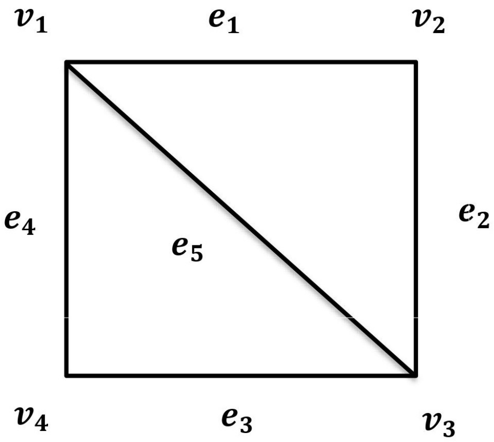

Figure 3.4: A graph is public key

Key Generation Process.

Alice performs the following:

- She chooses $a = 63$ as her a private key and computes her public key A by

$$A = D^{63}\ (mod\ 97) \equiv \begin{pmatrix} 56 & 71 & 23 & 60 & 39 \\ 63 & 79 & 27 & 56 & 25 \\ 47 & 91 & 2 & 5 & 46 \\ 31 & 11 & 86 & 51 & 56 \\ 45 & 48 & 74 & 0 & 82 \end{pmatrix},$$

So, Alice's private and public keys are $a = 63$ and $A = D^{63}$.

Encryption process.

Bob does the following steps:

- He chooses his an ephemeral secret key $b = 91$ and computes

$$C_1 = D^{91} \ (mod \ 97) \equiv \begin{pmatrix} 7 & 81 & 70 & 75 & 95 \\ 5 & 23 & 91 & 17 & 74 \\ 50 & 59 & 71 & 24 & 67 \\ 26 & 59 & 5 & 82 & 78 \\ 82 & 95 & 34 & 87 & 73 \end{pmatrix},$$

and

$$A^{91} \ (mod \ 97) \equiv \begin{pmatrix} 63 & 10 & 93 & 94 & 80 \\ 39 & 63 & 84 & 72 & 80 \\ 82 & 93 & 71 & 96 & 51 \\ 6 & 54 & 7 & 54 & 80 \\ 5 & 6 & 34 & 12 & 91 \end{pmatrix}.$$

- He computes the transformation T on 5-dimensional over matrix A^{91} through

$$T(a,b,c,d,e) = \begin{pmatrix} a & b & c & d & e \end{pmatrix} \begin{pmatrix} 63 & 10 & 93 & 94 & 80 \\ 39 & 63 & 84 & 72 & 80 \\ 82 & 93 & 71 & 96 & 51 \\ 6 & 54 & 7 & 54 & 80 \\ 5 & 6 & 34 & 12 & 91 \end{pmatrix}$$

$= (63a + 39b + 82c + 6d + 5e, 10a + 63b + 93c + 54d + 6e, 93a + 84b + 71c + 7d + 34e, 94a + 72b + 96c + 54d + 12e, 80a + 80b + 51c + 80d + 91e)$,

- He computes $v^\star = T(v) = T(16, 28, 45, 83, 95) \ (mod \ 97) \equiv (70, 6, 79, 76, 51)$,

- He chooses his plaintext as a sub graph in Figure 3.5 and represents it by a vector $m = (17, 0, 41, 0, 87)$.

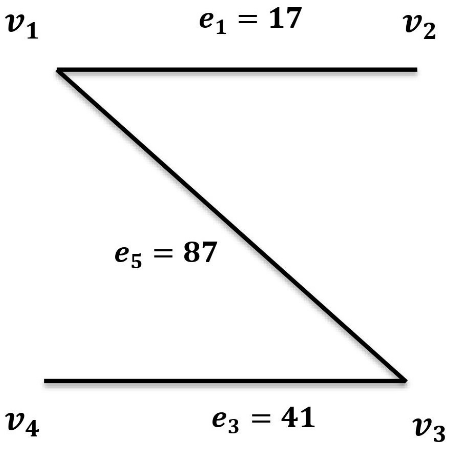

Figure 3.5: A plaintext m as a sub graph H of undirected graph G.

- He converts a vector m into $m^\star = m +_{97} v^\star \equiv (87, 6, 23, 76, 41)$.
- He computes $C_2 = T(m^\star) = T(87, 6, 23, 76, 41)\ (mod\ 97) \equiv (17, 74, 29, 88, 91)$.
- Bob sends his ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She computes

$$(C_1^{63})^{-1}\ (mod\ 97) \equiv \begin{pmatrix} 27 & 56 & 9 & 63 & 67 \\ 30 & 46 & 52 & 45 & 83 \\ 50 & 37 & 72 & 66 & 33 \\ 18 & 61 & 86 & 70 & 34 \\ 0 & 4 & 9 & 78 & 38 \end{pmatrix},$$

  where $C_1^{63} = (D^{91})^{63} = (D^{63})^{91} = A^{91}$.

- She computes the transformations T and $T^\star$ on 5-dimensional of the matrices C_1^{63} and $(C_1^{63})^{-1}$ respectively by:

  $T(a, b, c, d, e) = (63a + 39b + 82c + 6d + 5e, 10a + 63b + 93c + 54d + 6e, 93a + 84b + 71c + 7d + 34e, 94a + 72b + 96c + 54d + 12e, 80a + 80b + 51c + 80d + 91e)$,

  $T^\star(a, b, c, e, d) = (27a + 30b + 50c + 18d, 56a + 46b + 37c + 61d + 4e, 9a + 52b + 72c + 86d + 9e, 63a + 45b + 66c + 70d + 78e, 67a + 83b + 33c + 34d + 38e)$.

- Using public vector v to compute $v^\star = T(v) = T(16, 28, 45, 83, 95)\ (mod\ 97) \equiv (70, 6, 79, 76, 51)$ and computes $T^\star(C_2) = T^\star(17, 74, 29, 88, 91)\ (mod\ 97) \equiv (87, 6, 23, 76, 41) = m^\star$.

- She computes $m^\star -_{97} v^\star = (87, 6, 23, 76, 41) -_{97} (70, 6, 79, 76, 51) \equiv (17, 0, 41, 0, 87) = m$.
- She converts m into original plaintext as a subgraph as shown in Figure 3.5.

Another case study of the n-UG public key cryptosystem can be seen in Chapter (5).

3.6 The Matrix Power Function DLP Cryptosystems

In this section, the alternative versions of the cryptosystem, which are the MPF-Diffie-Hellman key exchange and the MPF-ElGamal public key cryptosystem, have been proposed. These cryptosystems are employed based on the proposed new extended definitions of the DLP. This extension is proposed based on the MPF that is applied to satisfy the DLP, as explained as follows.

Definition 3.6.1 Let A and B be two matrices in $GL_n(F_p)$. The left-side discrete logarithm problem (L-DLP) is the problem for finding a matrix L such that ${}^{L}A \equiv B\ (mod\ p)$. The matrix L is called the left-side discrete logarithm of B to the base matrix A.

Example 3.6.1 Suppose $p = 71$ is a prime number and $A = \begin{pmatrix} 50 & 43 \\ 14 & 61 \end{pmatrix}$ is a matrix in $GL_2(F_{71})$. How would we go about calculating the discrete logarithm of matrix

$$B =^L A \equiv \begin{pmatrix} 13 & 68 \\ 52 & 41 \end{pmatrix} mod\ (71)?$$

There is no method that is to compute $L = \begin{pmatrix} 17 & 55 \\ 47 & 61 \end{pmatrix}$.

Definition 3.6.2 Let A and B be two matrices in $GL_n(F_p)$. The right-side discrete logarithm problem (R-DLP) is the problem for finding a matrix R such that $A^R = B\ (mod\ p)$. The matrix R is called the right-side discrete logarithm of B to the base matrix A.

Example 3.6.2 Suppose $p = 71$ is a prime number and $A = \begin{pmatrix} 50 & 43 \\ 14 & 61 \end{pmatrix}$ is a matrix in $GL_2(F_{71})$. How would we go about calculating the discrete logarithm of matrix

$$B = A^R \equiv \begin{pmatrix} 48 & 12 \\ 4 & 27 \end{pmatrix} mod\ (71)?$$

There is no method that is to compute $R = \begin{pmatrix} 67 & 19 \\ 37 & 55 \end{pmatrix}$.

Now, the encryption schemes based on the MPF-DLP are discussed as follows.

3.6.1 The MPF-Diffie – Hellman Key Exchange

On a new version of Diffie-Hellman, exchange key which is named by the MPF-Diffie-Hellman key exchange (MPF-DHKE), the public parameters are a prime p and a public matrix $D \in GL_n(F_p)$. Alice and Bob, agree to choose the secret left L and right R matrices respectively. They compute $A \equiv^L D\ (mod\ p)$ and $B \equiv D^R\ (mod\ p)$. Alice sends A to Bob and Bob sends B to Alice. Both of them compute a shared key $K \equiv^L D^R$ through the following relation

$$K \equiv A^\star \equiv^L B \equiv^L (D^R) \equiv ({}^L D)^R \equiv^L D^R \equiv A^R \equiv B^\star. \tag{3.8}$$

Algorithm (11) is used for obtaining the several numerical results of the revised MPF-DHKE.

Algorithm 11 The MPF-Diffie – Hellman Key Exchange.

```
Input: A prime $p$ and a matrix $D \in GL_n(F_p)$.
Output:A shared key $K$, where $K$ is a matrix of size $n \times n$.
Alice chooses a matrix $L$ as her a private key, where $L \in GL_n(F_p)$.
She computes her public key $A \equiv^L D\ (mod\ p)$.
Alice keys are $(L, A)$.
Bob chooses a matrix $R$ his a private key, where $R \in GL_n(F_p)$.
He computes his public key $B \equiv D^R\ (mod\ p)$.
Bob keys are $(R, B)$.
Alice and Bob computes the share key $K$ as shown in Equation (3.8).
```

Example 3.6.3 Alice and Bob agree to use a prime $p = 97$. The matrix D of size 3×3 is given by

$$D = \begin{pmatrix} 15 & 77 & 90 \\ 56 & 80 & 19 \\ 12 & 14 & 16 \end{pmatrix},$$

where $D \in GL_3(F_{97})$.

Alice

- She chooses her secret key L by

$$L = \begin{pmatrix} 17 & 10 & 72 \\ 73 & 57 & 45 \\ 91 & 17 & 13 \end{pmatrix}$$

and computes her public key A as follows.

$$A =^L D \ (mod\ 97) \equiv \begin{pmatrix} 17 & 15 & 80 \\ 49 & 63 & 81 \\ 54 & 52 & 49 \end{pmatrix}.$$

Bob

- He chooses a secret key R by

$$R = \begin{pmatrix} 11 & 51 & 44 \\ 52 & 77 & 91 \\ 27 & 41 & 83 \end{pmatrix}$$

and compute his public key as follows.

$$B = D^R \ (mod\ 97) \equiv \begin{pmatrix} 3 & 74 & 75 \\ 94 & 29 & 95 \\ 65 & 69 & 60 \end{pmatrix}.$$

The sheared key K is computed by

$$K =^L B = A^R \ (mod\ 97) \equiv \begin{pmatrix} 27 & 15 & 54 \\ 31 & 14 & 67 \\ 32 & 59 & 67 \end{pmatrix}.$$

More implemented results of MPF-DHKE can be seen in Chapter (5).

3.6.2 The MPF-ElGamal Public Key Cryptosystem

The EPK cryptosystem can take alternative version based on the MPF-DLP. The public parameters, in the MPF-DLP, are a prime p and a public matrix $D \in GL_n(F_p)$. Alice chooses randomly a private key as a left matrix $L \in GL_n(F_p)$ and she computes a public key A by $A \equiv^L D \ (mod\ p)$.

Bob employs a public key A to encrypt a plaintext is a matrix $M_{n\times m}$. He chooses an ephemeral key as a right matrix $R \in GL_n(F_p)$. Using a public key A, the second entity computes a ciphertext $C = (C_1, C_2)$, where

$$C_1 \equiv D^R \ (mod\ p) \ \text{ and } \ C_2 \equiv A^R \times M \ (mod\ p). \tag{3.9}$$

The decryption to recover the original plaintext M is done using a secret key L through computing $({}^L C_1)^{-1}$ first. Then, computing the multiplication matrix $({}^L C_1)^{-1} \times C_2$ gives a plaintext M. Algorithm (12),(13) and (14) are used for obtaining the several numerical results of the revised MPF-EPKC.

Algorithm 12 The MPF-ElGamal Public Key Cryptosystem. Keys Generation Process:

```
Input: A prime $p$ and a matrix $D \in GL_n(F_p)$.
Output: The public key $A$, where $A \in GL_n(F_p)$.
1: Alice chooses a matrix $L$ as her a private key, where $L \in GL_n(F_p)$.
2: She computes her public key $A \equiv^L D \ (mod\ p)$.
3: Alice keys are $(L, A)$.
```

Algorithm 13 The MPF-ElGamal Public Key Cryptosystem. Encryption Process:

```
Input: A prime $p$, a public key $A$.
Output: The ciphertext $(C_1, C_2)$, where $C_1$ and $C_2$ are two matrices of size $n \times n$ and $n \times m$ respectively.
1: Bob chooses an ephemeral secret key $R$, where $R \in GL_n(F_p)$.
2: He chooses his message a matrix $M_{n\times m}$.
3: He computes the ciphertext through the computations of two matrices $C_1 \equiv D^R \ (mod\ p)$ and $C_2 \equiv A^R \times M_{n\times m} \ (mod\ p)$.
4: He sends the ciphertext pair $(C_1, C_2)$ to Alice.
```

The decryption process of the MPF-ElGamal public key cryptosystem is proved as follows.

Proposition 3.6.1 The decryption process to recover an original plaintext M is computed by

$$({}^{L}C_1)^{-1} \times C_2 \equiv M \ (mod\ p).$$

Proof.

$$\begin{aligned}
({}^{L}C_1)^{-1} \times C_2 &\equiv ({}^{L}(D^{R}))^{-1} \times C_2 \ (mod\ p), \ \text{[since } C_1 = D^{R}.\text{]} \\
&\equiv ({}^{L}D^{R})^{-1} \times ((A^{R} \times M) \ (mod\ p)) \ \text{[since } C_2 = A^{R} \times M.\text{]} \\
&\equiv ({}^{L}D^{R})^{-1} \times (({}^{L}D)^{R} \times M) \ (mod\ p)), \ \text{[since } A =^{L} D.\text{]} \\
&\equiv (({}^{L}D^{R})^{-1} \times ({}^{L}D^{R}) \ (mod\ p)) \times M, \\
&\equiv I \times M \ (mod\ p). \\
&\equiv M \ (mod\ p). \qquad \blacksquare
\end{aligned}$$

The decryption process can be implemented by Algorithm (14).

Algorithm 14 The MPF-ElGamal Public Key Cryptosystem. Decryption Process:

```
Input: A prime $p$ , a secret key $L$ and the ciphertext pair $(C_1, C_2)$.
Output: The plaintext $M_{n\times m}$, where $M_{n\times m}$ is a matrix.
1: Alice first uses her secret key $L$ to compute ${}^{L}C_1$ $(mod\ p)$,
2: She computes the matrix inverse $({}^{L}C_1)^{-1}$ $(mod\ p)$.
3: She computes the multiplication matrix $({}^{L}C_1)^{-1} \times C_2$ $(mod\ p) \equiv M_{n\times m}$ $(mod\ p)$.
```

Example 3.6.4 Alice and Bob agree to use the a prime $p = 773$ and a matrix D of size 4×4

$$D = \begin{pmatrix} 155 & 612 & 99 & 251 \\ 350 & 80 & 199 & 390 \\ 400 & 750 & 145 & 601 \\ 125 & 723 & 700 & 555 \end{pmatrix},$$

where $D \in GL_4(F_{773})$.

Alice performs the following:

- She chooses her secret key L by

$$L = \begin{pmatrix} 100 & 250 & 155 & 450 \\ 444 & 91 & 187 & 741 \\ 722 & 750 & 89 & 442 \\ 203 & 17 & 501 & 300 \end{pmatrix},$$

  and computes her public key A by

$$A =^L D \ (mod\ 773) \equiv \begin{pmatrix} 240 & 397 & 708 & 208 \\ 718 & 270 & 640 & 722 \\ 355 & 116 & 408 & 562 \\ 582 & 400 & 80 & 253 \end{pmatrix}.$$

Encryption process.

Bob does the following steps:

- He chooses an ephemeral key as a matrix

$$R = \begin{pmatrix} 200 & 711 & 666 & 556 \\ 350 & 655 & 250 & 185 \\ 522 & 177 & 360 & 233 \\ 700 & 450 & 120 & 400 \end{pmatrix}.$$

- He chooses his plaintext M by

$$M = \begin{pmatrix} 700 & 688 & 514 & 170 \\ 302 & 711 & 362 & 191 \\ 401 & 151 & 461 & 183 \\ 700 & 66 & 250 & 585 \end{pmatrix}.$$

- He computes

$$C_1 = D^R \ (mod\ 773) \equiv \begin{pmatrix} 551 & 711 & 384 & 35 \\ 287 & 359 & 68 & 518 \\ 667 & 444 & 640 & 531 \\ 646 & 590 & 449 & 327 \end{pmatrix}$$

  and

$$A^R \ (mod\ 773) \equiv \begin{pmatrix} 577 & 341 & 324 & 601 \\ 385 & 331 & 409 & 308 \\ 717 & 147 & 372 & 253 \\ 9 & 195 & 509 & 94 \end{pmatrix}.$$

- He computes

$$C_2 = A^R \times M \ (mod\ 773) \equiv \begin{pmatrix} 42 & 624 & 744 & 532 \\ 34 & 239 & 419 & 290 \\ 622 & 492 & 217 & 419 \\ 390 & 638 & 202 & 619 \end{pmatrix}.$$

- He sends his ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She used her private key L to compute

$$({}^L C_1)^{-1} \ (mod\ 773) \equiv \begin{pmatrix} 566 & 503 & 599 & 119 \\ 156 & 540 & 142 & 420 \\ 562 & 181 & 669 & 156 \\ 296 & 335 & 203 & 164 \end{pmatrix}$$

  such that ${}^L C_1 \ (mod\ 773) \equiv A^R$.

- Finally, she computes

$$({}^L C_1)^{-1} \times C_2 \ (mod\ 773) \equiv \begin{pmatrix} 700 & 688 & 514 & 170 \\ 302 & 711 & 362 & 191 \\ 401 & 151 & 461 & 183 \\ 700 & 66 & 250 & 585 \end{pmatrix} = M.$$

3.6.3 The Hybrid GMPF-ElGamal Public Key Cryptosystem

The graph theory can be employed with the matrix power function to draw another hybrid version of the ElGamal public key cryptosystem, which is called GMPF-ElGamal public key cryptosystem. Alice and Bob agree to use a prime p and a public matrix $D \in GL_n(F_p)$. Alice chooses randomly a private key left matrix $L \in GL_n(F_p)$ and computes her public key A by $A \equiv^L D \ (mod\ p)$ as computed in MPF-ElGamal public key cryptosystem.

Bob employs a public key A of the Alice to encrypt his plaintext m as an connected graph $G(V, E)$. He chooses an ephemeral key as a right matrix

$R \in GL_n(F_p)$ and represents his plaintext by adjacent matrix M. He computes a ciphertext $C = (C_1, C_2)$, where

$$C_1 \equiv D^R \ (mod\ p) \ \text{ and } \ C_2 \equiv A^R \times M \ (mod\ p). \tag{3.10}$$

The decryption to recover a plaintext is done using a secret key L to compute $({}^{L}C_1)^{-1}$ $(mod\ \ p)$. The computation of the multiplication matrix $({}^{L}C_1)^{-1} \times C_2$ $(mod\ p)$ gives a matrix M. she converted a matrix M into corresponding graph $G(V, E)$ which represents as an original a plaitext m. Algorithm (12),(15) and (16) is used for obtaining the several numerical results of the revised MPF-EPKC.

Algorithm 15 The Hybrid GMPF-ElGamal Public Key Cryptosystem: Encryption Process.

```
Input: A prime $p$ and a public key $A$ .
Output: The ciphertext $(C_1, C_2)$, where $C_1$ and $C_2$ are two matrices of size $n \times n$.
1: Bob chooses an ephemeral secret key $R$, where $R \in GL_n(F_p)$.
2: He chooses his plaintext as an connected graph $G(V, E)$.
3: He represents his plaintext by adjacent matrix $M$,
4: He computes the ciphertext $(C_1, C_2)$, where $C_1$ and $C_2$ are two matrices
   computed by $C_1 \equiv D^R \ (mod\ p)$ and $C_2 \equiv A^R \times M \ (mod\ p)$.
5: Bob sends the ciphertext pair $(C_1, C_2)$ to Alice.
```

Algorithm 16 The Hybrid GMPF-ElGamal Public Key Cryptosystem: Decryption Process.

```
Input: A prime $p$ and the ciphertext pair $(C_1, C_2)$.
Output: The plaitext as an connected graph $G(V, E)$.
1: Alice first uses her secret key $L$ to compute ${}^{L}C_1 \ (mod\ p)$.
2: She computes the matrix inverse $({}^{L}C_1)^{-1} \ (mod\ p)$.
3: She computes the multiplication matrix $({}^{L}C_1)^{-1} \times C_2 \ (mod\ p) \equiv M \ (mod\ p)$.
4: She converted a matrix $M$ into corresponding graph $G(V, E)$ which represents
   as an original plaintext $m$.
```

Example 3.6.5 Alice and Bob agree to use the prime $p = 613$ and a matrix D of size 4×4

$$D = \begin{pmatrix} 11 & 500 & 108 & 425 \\ 601 & 250 & 87 & 222 \\ 28 & 333 & 17 & 147 \\ 155 & 14 & 33 & 611 \end{pmatrix},$$

where $D \in GL_4(F_{613})$.

Alice performs the following:

- She chooses her secret key L by

$$L = \begin{pmatrix} 110 & 17 & 217 & 355 \\ 333 & 401 & 603 & 145 \\ 520 & 250 & 130 & 97 \\ 103 & 170 & 59 & 575 \end{pmatrix},$$

and computes her public key A by

$$A =^L D \ (mod\ 613) \equiv \begin{pmatrix} 137 & 378 & 457 & 47 \\ 437 & 196 & 20 & 612 \\ 409 & 404 & 426 & 156 \\ 530 & 49 & 114 & 483 \end{pmatrix}.$$

Encryption process.

Bob does the following steps:

- He chooses an ephemeral key as a matrix

$$R = \begin{pmatrix} 301 & 99 & 525 & 603 \\ 400 & 189 & 67 & 91 \\ 250 & 370 & 450 & 37 \\ 19 & 107 & 607 & 375 \end{pmatrix}.$$

- He chooses his plaintext as a connected graph G in Figure 3.6

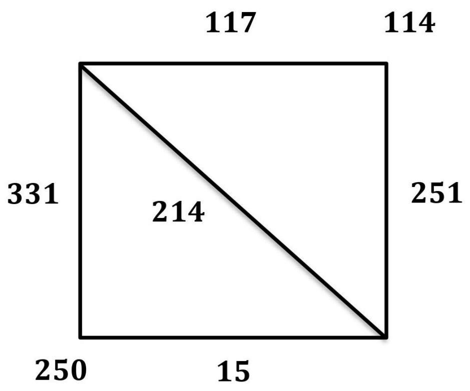

Figure 3.6: The weighted graph is plaintext

- He representation weighted graph by adjacent matrix

$$M = \begin{pmatrix} 0 & 117 & 214 & 331 \\ 117 & 114 & 251 & 0 \\ 214 & 251 & 0 & 15 \\ 331 & 0 & 15 & 250 \end{pmatrix}.$$

- He computes

$$C_1 = D^R \ (mod\ 613) \equiv \begin{pmatrix} 371 & 300 & 210 & 137 \\ 251 & 303 & 518 & 514 \\ 566 & 438 & 466 & 70 \\ 249 & 550 & 100 & 128 \end{pmatrix}$$

and

$$A^R \ (mod\ 613) \equiv \begin{pmatrix} 10 & 515 & 174 & 295 \\ 276 & 249 & 159 & 268 \\ 42 & 90 & 179 & 428 \\ 112 & 47 & 100 & 488 \end{pmatrix}.$$

- He computes

$$C_2 = A^R \times M \ (mod\ 613) \equiv \begin{pmatrix} 202 & 570 & 357 & 593 \\ 456 & 55 & 531 & 135 \\ 474 & 29 & 605 & 374 \\ 236 & 39 & 175 & 579 \end{pmatrix}.$$

- He sends his ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She used her private key L to compute

$$({}^{L}C_1)^{-1}\ (mod\ 613) \equiv \begin{pmatrix} 316 & 198 & 503 & 264 \\ 191 & 320 & 224 & 413 \\ 419 & 78 & 428 & 606 \\ 136 & 571 & 127 & 602 \end{pmatrix}$$

  such that ${}^{L}C_1\ (mod\ 613) \equiv A^R$.

- She computes

$$({}^{L}C_1)^{-1} \times C_2\ (mod\ 613) \equiv \begin{pmatrix} 0 & 117 & 214 & 331 \\ 117 & 114 & 251 & 0 \\ 214 & 251 & 0 & 15 \\ 331 & 0 & 15 & 250 \end{pmatrix} = M.$$

- Finally, converted a matrix M into corresponding graph $G(V, E)$ which represents as original plaintext graph in Figure 3.6.

3.7 The Soft Graph Public Key Cryptosystem

In this section, used the soft graph with the MPF to encrypt a subset of the finite field. Alice and Bob agree to use the prime p and a public matrix $D \in GL_n(F_p)$. Alice chooses randomly a private key left matrix $L \in GL_n(F_p)$ and computes her public key A by $A \equiv^{L} D\ (mod\ p)$. Bob chooses a connected graph $G(V, E)$ and computes a soft graph $G^*(F, K, N)$ of $G(V, E)$ by choosing N is a non empty set of V and defining approximate functions $F : N \to P(V)$ and $K : N \to P(E)$. He employs a public key A of Alice and a soft graph to encrypt his plaintext $m = \{m_1, m_2, ..., m_l\}$ as a subset of F_p such that l is equal to the number of edges in the soft graph. He converts the graph $G(V, E)$ to a weighted graph $G_w(V, E)$ after converting a soft graph G^* into a weighted soft graph, where the weight values of the edges are l elements of the plaintext $m = \{m_1, m_2, ..., m_l\}$ representing the other edges of G by the weight values that are chosen randomly from F_p. Then it represents the weighted graph $G_w(V, E)$ by the adjacent matrix M. After that, it chooses an ephemeral key as a right matrix $R \in GL_n(F_p)$ and computes a ciphertext $C = (C_1, C_2)$, where

$$C_1 \equiv D^R\ (mod\ p) \quad \text{and} \quad C_2 \equiv A^R \times M\ (mod\ p).$$

Finally, sends (C_1, C_2, N, F, K) and $deg(x)$ for all $x \in N$ to Alice.

The decryption to recover a plaintext is done using a secret key L to compute $({}^{L}C_1)^{-1}$ $(mod\ p)$. The computation of the multiplication matrix $({}^{L}C_1)^{-1} \times C_2$ $(mod\ p)$ gives a matrix M. She converted a matrix M into a corresponding weighted connected graph $G_w(V, E)$. After that, she uses public keys (N, F, K) to find a soft graph $G^*(F, K, N)$. Finally, the weights of the edges of a soft graph are representations of the plaintext m. The algorithms (12), (17) and (18) are used for obtaining the several numerical results of the revised SG-PKC.

Algorithm 17 The Soft Graph Public Key Cryptosystem. Encryption Process:

```
Input: A prime $p$ and a public key $A$.
Output: The ciphertext $(C_1, C_2)$, where $C_1$ and $C_2$ are two matrices.
1: Bob chooses an connected graph $G(V, E)$.
2: He chooses a nonempty sub set $N$ of $V$ such that $x \in N$ and $y \in V - N$,
   $deg(x) \neq deg(y), \forall x, y \in V$.
3: He defined approximate functions $F : N \to P(V)$ and $K : N \to P(E)$.
4: He selects his plaintext $m$ as a sub set of $F_p$, the length of $m$ is equal to $l$.
5: He forms a subgraph $G^*$ of $G$ as a soft graph $G^*(F, K, N)$ has $l$ edges.
6: He converted a soft graph $G^*$ into weighted soft graph, where the weights values
   of the edges are $l$ elements of the plaintext $m = \{m_1, m_2, ..., m_l\}$.
7: He represents other edges of $G$ by the weights values that are chosen randomly
   from $F_p$.
8: He represents a weighted connected graph $G_w(V, E)$ by adjacent matrix $M$.
9: He chooses an ephemeral secret key $R$, where $R \in GL_n(F_p)$.
10: He computes the ciphertext through the computations of two matrices $C_1 \equiv$
    $D^R$ $(mod\ p)$ and $C_2 \equiv A^R \times M$ $(mod\ p)$.
11: Bob sends $(C_1, C_2, N, F, K)$ and $deg(x)$ for all $x \in N$ to Alice.
```

Algorithm 18 The Soft Graph Public Key Cryptosystem. Decryption Process:

```
Input: A prime $p$, a secret key $L$ and $(C_1, C_2, N, F, K)$ .
Output: The plaintext $m = \{m_1, m_2, ..., m_l\}$.
1: Alice first uses her secret key $L$ to compute ${}^{L}C_1$ $(mod\ p)$.
2: She computes the matrix inverse $({}^{L}C_1)^{-1}$ $(mod\ p)$.
3: She computes the multiplication matrix $({}^{L}C_1)^{-1} \times C_2$ $(mod\ p) \equiv M$ $(mod\ p)$.
4: She converted a matrix $M$ into a corresponding weighted graph $G_w(V, E)$.
5: She uses public keys $(N, F, K)$ to find a soft graph $G^*(F, K, N)$.
6: The weighted edges of the soft graph represent the plaintext $m$.
```

Example 3.7.1 Alice and Bob agree to use the prime $p = 1987$ and a matrix D of size 6×6

$$D = \begin{pmatrix} 150 & 77 & 909 & 1010 & 17 & 555 \\ 56 & 808 & 1969 & 1972 & 22 & 1984 \\ 120 & 14 & 1600 & 105 & 9 & 1500 \\ 1900 & 1947 & 1 & 6 & 1983 & 701 \\ 1850 & 8 & 1977 & 3 & 300 & 607 \\ 3 & 22 & 17 & 18 & 21 & 1970 \end{pmatrix},$$

where $D \in GL_6(F_{1987})$.

Alice performs the following steps:

- She chooses her secret key L

$$L = \begin{pmatrix} 1700 & 109 & 1112 & 39 & 1500 & 1953 \\ 1973 & 777 & 450 & 22 & 993 & 250 \\ 902 & 17 & 1300 & 14 & 554 & 1450 \\ 405 & 239 & 123 & 1504 & 27 & 1015 \\ 222 & 1607 & 5 & 8 & 1 & 400 \\ 30 & 833 & 1980 & 1947 & 1972 & 13 \end{pmatrix},$$

  and computes her public key A by

$$A =^{L} D\ (mod\ 1987) \equiv \begin{pmatrix} 1385 & 253 & 989 & 710 & 143 & 1850 \\ 1973 & 707 & 1025 & 1973 & 711 & 1893 \\ 1805 & 138 & 179 & 91 & 777 & 1907 \\ 1810 & 995 & 502 & 1387 & 754 & 1971 \\ 1943 & 1665 & 1229 & 1832 & 1440 & 1785 \\ 1881 & 772 & 1942 & 1713 & 138 & 1781 \end{pmatrix}.$$

Encryption process.

Bob does the following steps:

- He chooses a connected graph in Figure 3.7.

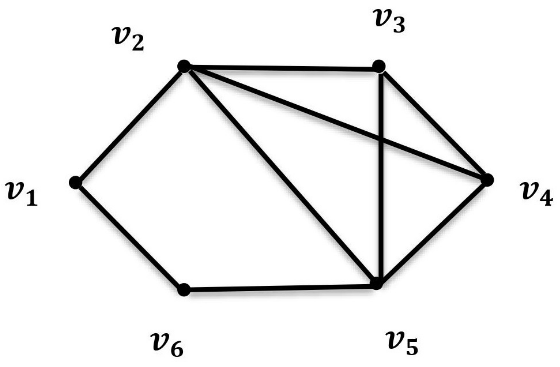

Figure 3.7: The connected graph.

- He chooses $N = \{v_3, v_4\} \subseteq V$ and (F, N) be a soft set over V with approximate function $F : N \to P(V)$ by:

$$F(x) = \{y \in V : xRy \leftrightarrow d(x, y) = 1\}$$

for all $x \in N$. That is, $F(v_3) = \{v_2, v_4, v_5\}$ and $F(v_4) = \{v_2, v_3, v_5\}$.

Let (K, N) be a soft set over E with approximate function $K : N \to P(E)$ by

$$K(x) = \{uv \in E : \{u, v\} \subseteq F(x)\}$$

for all $x \in A$. That is, $K(v_3) = \{v_2v_4, v_4v_5, v_5v_2\}$ and $K(v_4) = \{v_2v_3, v_3v_5, v_5v_2\}$.

Then the soft graph is shown in Figure 3.8.

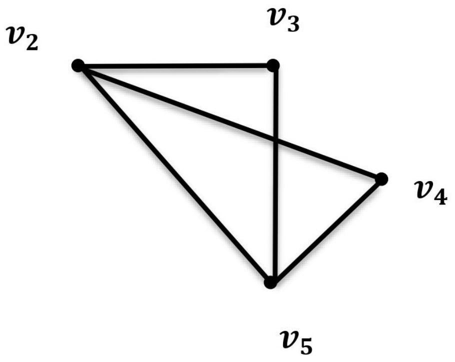

Figure 3.8: The soft graph.

- He chooses a plaintext $\{123, 361, 941, 1445, 1983\}$.

- He converts a connected graph in Figure 3.7 to weighted connected graph in Figure 3.9.

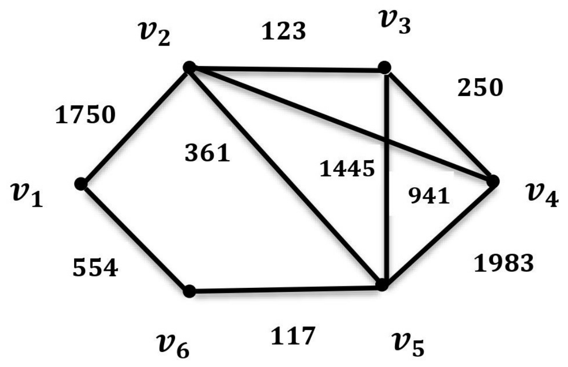

Figure 3.9: The weighted connected graph.

- He represents the weighted connected graph by the adjacent matrix M.

$$M = \begin{pmatrix} 0 & 1750 & 0 & 0 & 0 & 554 \\ 1750 & 0 & 123 & 1445 & 361 & 0 \\ 0 & 123 & 0 & 250 & 941 & 0 \\ 0 & 1445 & 250 & 0 & 1983 & 0 \\ 0 & 361 & 941 & 1983 & 0 & 117 \\ 554 & 0 & 0 & 0 & 117 & 0 \end{pmatrix}.$$

- He chooses an ephemeral key as a matrix

$$R = \begin{pmatrix} 17 & 516 & 1509 & 1955 & 317 & 121 \\ 1333 & 170 & 91 & 1499 & 87 & 1617 \\ 1919 & 415 & 1833 & 15 & 100 & 1250 \\ 1420 & 1718 & 213 & 122 & 1000 & 191 \\ 17 & 100 & 400 & 901 & 330 & 1400 \\ 13 & 812 & 1822 & 710 & 3 & 1887 \end{pmatrix}.$$

- He computes his cipher text

$$C_1 = D^R \ (mod\ 1987) \equiv \begin{pmatrix} 1769 & 1528 & 1969 & 638 & 46 & 822 \\ 1920 & 1649 & 1836 & 1588 & 1797 & 983 \\ 881 & 1103 & 247 & 1423 & 962 & 752 \\ 865 & 1343 & 748 & 1295 & 25 & 82 \\ 360 & 297 & 408 & 1936 & 1444 & 1373 \\ 1401 & 491 & 362 & 1351 & 652 & 124 \end{pmatrix}.$$

and

$$C_2 = A^R \times M \ (mod\ 1987) \equiv \begin{pmatrix} 875 & 1563 & 275 & 1560 & 637 & 1137 \\ 1223 & 116 & 430 & 734 & 98 & 974 \\ 1900 & 929 & 408 & 1165 & 1614 & 1401 \\ 588 & 1247 & 68 & 1697 & 232 & 789 \\ 483 & 1952 & 190 & 909 & 1050 & 1354 \\ 1546 & 259 & 1061 & 1053 & 480 & 1209 \end{pmatrix}$$

,such that A^R is the shear key

$$K =^L C_1 = A^R \ (mod\ 1987) \equiv \begin{pmatrix} 944 & 1301 & 1815 & 300 & 346 & 221 \\ 922 & 1539 & 307 & 731 & 228 & 926 \\ 1286 & 418 & 1119 & 311 & 1582 & 1538 \\ 1163 & 1096 & 1093 & 856 & 1344 & 1632 \\ 772 & 1648 & 937 & 1307 & 449 & 1104 \\ 488 & 1116 & 402 & 1099 & 315 & 430 \end{pmatrix}.$$

- He sends his ciphertext (C_1, C_2) to Alice

Decryption process.

Alice performs the following steps:

- She used her private key L to compute

$$({}^L C_1)^{-1} \ (mod\ 1987) \equiv \begin{pmatrix} 1284 & 838 & 1684 & 109 & 953 & 583 \\ 538 & 132 & 1828 & 1695 & 1870 & 1777 \\ 764 & 1150 & 1052 & 100 & 1008 & 1842 \\ 730 & 935 & 654 & 1121 & 945 & 846 \\ 505 & 545 & 1682 & 1144 & 1088 & 1695 \\ 384 & 332 & 409 & 428 & 1717 & 1772 \end{pmatrix}$$

such that ${}^L C_1 \ (mod\ 1987) \equiv A^R$.

- She computes

$$({}^{L}C_1)^{-1} \times C_2(mod\ 1987) \equiv \begin{pmatrix} 0 & 1750 & 0 & 0 & 0 & 554 \\ 1750 & 0 & 123 & 1445 & 361 & 0 \\ 0 & 123 & 0 & 250 & 941 & 0 \\ 0 & 1445 & 250 & 0 & 1983 & 0 \\ 0 & 361 & 941 & 1983 & 0 & 117 \\ 554 & 0 & 0 & 0 & 117 & 0 \end{pmatrix} = M.$$

- She converted a matrix M into a corresponding weighted connected graph in Figure 3.9.

- She uses public keys (N, F, K) to find a soft graph in Figure 3.10.

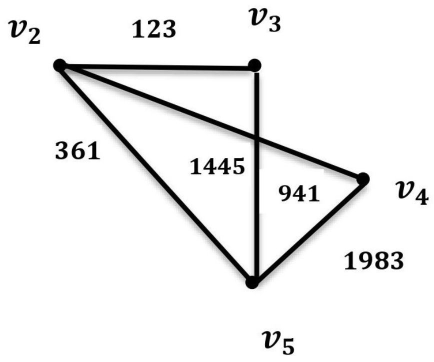

Figure 3.10: The weighted soft graph.

- Then the weights of edges of the soft graph are representation the original plaintext $\{123, 361, 941, 1445, 1983\}$.

Another case study of the soft graph public key cryptosystem can be seen in Chapter (5).

3.8 The Security Considerations on Proposed Cryptosystems

In this section, discussed the security cases of the proposed cryptosystems.

3.8.1 The Security Considerations on UCG Proposed Public Key Cryptosystems

More secure communications with the revised EPKC compared to the original one are obtained. The security of the revised EPKC is determined by the difficulty of finding the DLP that solves the matrices. Whereas, the security of the proposed UCG-based public key cryptosystem can be determined first by the hardness of solving the DLP that computing on the matrices and the difficulty of determining the correct MST graph among all other possible cases of the MST graphs that can be created from undirected complete subgraphs. Also, another point focuses on a good choice of the domain parameters of the revised EPKC and proposed UCG-based public key cryptosystem. Specifically, with a large prime p, it gives us the possibility to choose and generate the big size matrices over a set $GL_n(F_p)$ which helps us to implement the revised EPKC and the proposed UCG based public key cryptosystem more securely compared with other public key algorithms.

3.8.2 The Security on n-UG Public Key Cryptosystem

The security of the n-UGPKC is determined by the difficulty of finding the DLP that solves the matrices. Whereas, the security of the proposed n-UGPKC based public key cryptosystem can be determined first by the hardness of solving the DLP that computing on the matrices and the difficulty of determining the correct sub graph among all those that can be created from an undirected complete graph because its number is p^e.

3.8.3 The Security on MPF-ElGamal Public Key Cryptosystem

The security of the MPF-EPKC is determined by the difficulty of finding the left and right side DLP over matrices.

3.8.4 The Security on Soft Graph Public Key Cryptosystem

The security of the soft graph PKC is determined by the difficulty of solving the L-DLP and R-DLP over matrices. Eve if know the approximate functions that are defined over the undirected graph. She can not get the soft graph of the undirected graph because the graph is secret. Therefore, she does not get the plaintext.

3.9 Summary

In Chapter 3, new public key cryptosystems have been designed based on matrices and graph theory. Among these systems, an extended DLP over matrices is used in an alternative version of the ElGamal public key cryptosystem (EPKC) that is being presented. Following that, a new UCG-based public key cryptosystem was created. A new public key cryptosystem for encrypting a weighted subgraph of an undirected graph in n-dimensional vector space V has also been presented. The L/R MPF-Diffie Hellman and L/R MPF-ElGamal public key cryptosystems have been proposed to display other revised versions of the original ones, and the L/RMPDLP is employed with graph theory to draw other versions of the discrete logarithm encryption scheme which is a hybrid G-L/RMP-ElGamal public key cryptosystem. Another version of an asymmetric cryptosystem that encrypts a subset of the prime field is designed using the soft graph and the matrix power function.

Chapter 4

Elliptic Curve Cryptography with Graph

4.1 Introduction

This chapter proposed new versions of elliptic curve public key cryptography by using number theory, matrices, and graph theory. Also, proposed a new version of Edward curve public key cryptography by using graph theory. In Section (4.2), a new version of the elliptic curve public key cryptosystem (ECPKC) has been proposed called KR-ECPKC. The main idea is to design this cryptosystem focused on generating the elliptic curve group. Its order divides the Euler phi function that is computed securely based on two large primes. In Section (4.3), proposed modifications to Diffie-Hellman and ElGamal schemes in elliptic curve cryptography are dependent on new definitions of left and right side elliptic curve DLP matrix functions. In Sections 4.4 and 4.5, new graphs based on an elliptic scalar multiplication operation are defined and designed for new versions of an asymmetric encryption scheme. These graphs are formed based on the scalar multiplication operation on an elliptic curve defined over a prime field. In Section (4.6), a new graph, which is named the Edwards curve graph. It has been defined as a bright point for designing a new version of an asymmetric encryption scheme on the Edwards curve. In Section (4.7), discussed the security cases of the proposed schemes. In Section (4.8), discussed the summary.

4.2 The KR –Elliptic Curve Public Key Cryptosystem

In this section, a new version of elliptic curve public key cryptosystem has been proposed. It called the KR-EC cryptosystem. The public parameters are a prime p and an elliptic curve EC defined over a prime field F_p. The KR-EC cryptosystem consists of three stages: key generation, encryption and decryption processes. These processes are discussed in the following subsections.

4.2.1 The KR-Elliptic Curve Cryptosystem: Keys Generation Process

Suppose EC is an elliptic curve over a prime field F_p. a set of the points lie on EC is computed. The order $\#EC(F_p)$ of a set $EC(F_p)$ should divide a positive integer n, where $n = (q-1)(t-1)$ with q and t are two large primes that are selected secretly by Alice to computes her private key n. The public key e is selected as an integer from the range $[2, n-1]$ such that $gcd(e, n) = 1$. So, the public and private keys are e and n respectively. Algorithm (19) is used to generate the keys of the KR-EC cryptosystem.

Algorithm 19 The KR-EC Cryptosystem: Keys Generation Process.

```
Input: The large primes $(q, t)$.
Output: The set of elliptic curve points and public key $e$.
Alice computes $n = (q-1)(t-1)$.
She selects $a, b \in [1, n-1]$ are coefficients of $EC$.
She select randomlys $e \in [2, n-1]$ , such that $gcd(e, n) = 1$.
She computes the elliptic curve set $EC(F_p)$.
She computes the order $\#EC(F_p) = h$ of $EC(F_p)$ under condition $h \mid n$.
Return $(e, n, EC(F_p))$.
```

4.2.2 The KR-Elliptic Curve Cryptosystem: Encryption Process

The encryption process is done as follows. A plaintext M is selected and represented as an elliptic point in $EC(F_p)$. Then a plaintext M is encrypted by computing $C = e'M$, where e' is computed by $e' \equiv e \pmod N$, with N is an order of M. The ciphertext C considers as a scalar multiplication which can be computed by the ISD method [4] or any other elliptic scalar multiplication algorithm. Algorithm (20) is employed to compute the ciphertext C.

Algorithm 20 The KR-EC Cryptosystem: Encryption Process.

```
Input: A set $EC(F_p)$ of elliptic points and a public key $e$.
Output: The ciphertext $C$.
1: Bob computes the order $N$ of a plaintext $M$.
2: He computes $e' \equiv e \ (mod\ N)$
3: He computes the ciphertext $C = e'M$.
4: Return $C$.
```

4.2.3 The KR-Elliptic Curve Cryptosystem: Decryption Process

Alice wants to decrypt the ciphertext and recover the original plaintext upon receiving the ciphertext. So, first she computes the key d secretly, where $d \in [2, n-1]$ and $ed \equiv 1 \ (mod\ n)$. Then d' is computed by $d' \equiv d \ (mod\ N)$ with N is an order of the C that is computed as an elliptic curve point. The original plaintext is recovered by $d'C = M$. The decryption process is proved mathematically in following proposition.

Proposition 4.2.1 The decryption process of KR-EC cryptosystem takes the expression $M = d'C$

Proof.

$d'C = d'(e'M)$, [since $C = e'M$]

$= deM$

$= (rn + 1)M$, [since $ed \equiv 1 \ (mod\ n)$], so based on that $rn + 1 = ed$, where $r \in Z^+$.

$= rnM + M$

$= r(sN)M + M$, [since $n = sN$], where N is order of a point M and every order of elliptic point divide n.

$= rs(NM) + M$

$= rsO_E + M$ [since $NM = O_E$], where N is order of a point M and O_E ia a point at infinity.

$= M$. ■

For the computational results to recover a message M, one can use Algorithm (21)

Algorithm 21 The KR-EC Cryptosystem: Decryption Process.

```
Input: A set $E(F_p)$ of elliptic points, public key $e$ and the ciphertext $C$.
Output: A plaintext $M$.
1: Alice chooses randomly $d \in [2, n-1]$, when $ed \equiv 1 \ (mod\ n)$.
2: She computes the order $N$ of an elliptic point $C$, where $C$ is a ciphertext.
3: She computes $d' \equiv d \ (mod\ N)$.
4: She computes $d'C = M$, where $M$ is an elliptic point.
5: Return $M$.
```

Example 4.2.1 (The KR-EC Cryptosystem).

With a prime $p = 8123$, an elliptic curve

$$EC : y^2 \equiv x^3 + 5x + 13 \ (mod\ 8123),$$

is defined. The set of $EC(F_{8123})$ of elliptic curve points is computed by:

$$EC(F_{8123}) = \{(1, 1548), (1, 6575), (3, 3067), (3, 5056), (4, 6052), (4, 2071), ..., \\ (8122, 5994), (8122, 2129), O_E.\}$$

The $\#EC(F_{8123}) = 8144$, where $8144 \mid n$. The parameter n is considered as a private key of the first user which is computed by $n = (q-1)(t-1) = 12123438 \times 31423268 = 3809574206353584$, with primes $q = 12123439$ and $t = 31423269$ that are selected secretly. The public key e is selected randomly from the range $[2, 3809574206353583]$ such that $gcd(12326106299, 3809574206353584) = 1$. So, the public and private keys are determined $e = 12326106299$ and $n = 3809574206353584$ respectively.

Now for encryption process, second user chooses a plaintext as point $M = (1000, 5149) \in EC(F_{8123})$. The order of a point M is 4072, namely $4072M = 4072(1000, 5149) = O_E$. He computes $e' = 3491$ through computing $e' \equiv e \ (mod\ 4072)$. The ciphertext C is computed as a scalar multiplication $C = e'M = 3491(1000, 5149) = (5375, 4448)$. The ciphertext $C = (5375, 4448) \in EC(F_{8123})$ will be sent to the first user.

After receiving the ciphertext C to the first user (Alice), she computes $d = 618131$ such that $ed \ (mod\ n) \equiv 12326106299 \times 618131 \ (mod\ 3809574206353584) \equiv 1$. The order of a ciphertext C is computed to be $N = 4072$, since $4072C = 4072(5375, 4448) = O_E$. The parameter $d' = 3259$ is computed, where $d' \equiv$

$d \ (mod\ 4072)$. Alice computes a scalar multiplication $d'C = 3259(5375, 4448) = (1000, 5149) = M$.

More implemented results of the KR-ECC can be seen in Chapter (5).

4.2.4 Comparison on KR-EC, RSA and EC-PK Cryptosystems

With the proposed KR-EC cryptosystem, there is no problem to choose the small primes q and t for computing $n = (q-1)(t-1)$, since n is a private key. In comparing with that, the RSA public key cryptosystem needs the choice of the large primes for increasing the security. The public key generation process in elliptic curve ElGamal public key cryptosystem (EC-EPKC) requires more computations to solve the DLP comparing with the new version KR-EC cryptosystem that is proposed to choose a public key as a number in the range $[2, n-1]$, with $gcd(e, n) = 1$. Also, computing a public key on the EC-EPKC requires finding the generator point on elliptic curve group in comparing to the KR-EC cryptosystem which is not required that.

As well as, the proposed version KR-EC cryptosystem is considered faster public key algorithm in compare with EC-EPKC, since on the KR-EC cryptosystem, the encryption and decryption processes need computing only the scalar multiplication operation while on the EC-EPKC requires for these processes computing the scalar multiplication and addition operations, so the last cryptosystem needs extra computations.

4.3 The Elliptic Scalar Multiplication Matrix Function for Elliptic Curve Cryptosystems

In this section, new concepts of left-side, right-side and two-side scalar multiplication matrix function are defined as main concepts for ECC. Through which we encrypt and decrypt the elliptic curve points in a matrix. Also proposed modification Diffie-Hellman and ElGamal schemes in ECC, thing this modification is more security from the original because the DLP is very hard to break. Used an elliptic curve group that order a prime number because all points generate it.

4.3.1 The Elliptic Scalar Multiplication Matrix Function (ESMMF)

New mathematical concepts depending on the points doubling and addition on elliptic curve group and matrices power functions are defined as the key points in this section. The essential concept is called the elliptic scalar multiplication matrix function (ESMMF). The ESMMF is a squre matrix, its elements are elliptic curve points. The discussion of these concepts is done as follows.

Definition 4.3.1 (Left-side elliptic scalar multiplication matrix function (L-ESMMF)).

Let $\star_L : M_n(F_{\#EC(F_p)}) \times EM_n(F_p) \to EM_n(F_p)$ be a map. Then, for all $L \in M_n(F_{\#EC(F_p)})$ and $A \in EM_n(F_p)$, there exists $B \in EM_n(F_p)$ such that

$$\star_L(A) = L \star A = B.$$

The elements that form a L-ESMMF are computed by

$$B_{ij} = \sum_{k=1}^{n} l_{ik} A_{kj} \tag{4.1}$$

Definition 4.3.2 (Right-side elliptic scalar multiplication matrix function (R-ESMMF)).

Let $\star_R : EM_n(F_p) \times M_n(F_{\#EC(F_p)}) \to EM_n(F_p)$ be a map. Then, for all $R \in M_n(F_{\#EC(F_p)})$ and $A \in EM_n(F_p)$, there exists $C \in EM_n(F_p)$ such that

$$\star_R(A) = A \star R = C.$$

The elements that form a R-ESMMF are computed by

$$C_{ij} = \sum_{t=1}^{n} r_{ij} A_{it} \tag{4.2}$$

Definition 4.3.3 (Left-Right sides elliptic scalar multiplication matrix function (L/R-ESMMF)). Let $\star_{LR} : M_n(F_{\#EC(F_p)} \times EM_n(F_p) \times M_n(F_{\#EC(F_p)}) \to EM_n(F_p)$ be a map. Then, for all $A \in EM_n(F_p)$ and $L, R \in M_n(F_{\#EC(F_p)+1})$, there exists $D \in EM_n(F_p)$ such that $\star_{LR}(A) = L \star A \star R = D$, where L, A, R and D are square matrices with the same sizes. The elements that form a L/R-ESMMF are computed by

$$D_{ij} = \sum_{t=1}^{n} \sum_{k=1}^{n} l_{it} r_{kj} A_{tk} \tag{4.3}$$

Theorem 4.3.1 Properties on the L/R-ESMMF

(1) The L/R-ESMMF is an associative. This means that $(L\star A)\star R = L\star(A\star R)$

(2) $L_1 \star (L_2 \star A) = (L_1.L_2) \star A$ and $(A \star R_1) \star R_2 = A \star (R_1.R_2)$

(3) $L^{-1} \star (L \star A) = A$ and $(A \star R) \star R^{-1} = A$

(4) $L_2 \star (L_1 \star A \star R_1) \star R_2 = (L_2.L_1) \star A \star (R_1.R_2)$

Proof:

(1)

$$\begin{aligned}(L \star A) \star R &= (\sum_{k=1}^{n} l_{ik}A_{kj}) \star R \\ &= \sum_{t=1}^{n} r_{tj}(\sum_{k=1}^{n} l_{ik}A_{kt}) \\ &= \sum_{t=1}^{n}\sum_{k=1}^{n} r_{tj}l_{ik}A_{kt} \\ &= \sum_{k=1}^{n}\sum_{t=1}^{n} l_{ik}r_{tj}A_{kt} \\ &= \sum_{k=1}^{n} l_{ik}(\sum_{t=1}^{n} r_{tj}A_{kt}) \\ &= L \star (\sum_{t=1}^{n} r_{tj}A_{kt}) \\ &= L \star (A \star R).\end{aligned}$$

(2)

$$\begin{aligned}L_1 \star (L_2 \star A) &= L_1 \star (\sum_{k=1}^{n} l^2_{ik}A_{kj}) \\ &= \sum_{t=1}^{n} l^1_{ij}(\sum_{k=1}^{n} l^2_{ik}A_{kj}) \\ &= \sum_{k=1}^{n}(\sum_{t=1}^{n} l^1_{ij}l^2_{ik})A_{kj} \\ &= (L_1.L_2) \star A.\end{aligned}$$

Similarly, it can be shown that $(A \star R) \star R^{-1} = A$.

(3)

$$\begin{aligned} L^{-1} \star (L \star A) &= (L^{-1}.L) \star A \text{ by } (2) \\ &= I \star A \\ &= A. \end{aligned}$$

Similarly, it can be shown that $(A \star R) \star R^{-1} = A$.

(4)

$$\begin{aligned} L_2 \star (L_1 \star A \star R_1) \star R_2 &= (L_2.L_1) \star (A \star R_1) \star R_2 \text{ by } (2) \\ &= (L_2.L_1) \star A \star (R_1.R_2) \text{ by } (2). \end{aligned}$$ ■

4.3.2 The Elliptic Scalar Multiplication Matrix Function Cryptosystems

In this section, the alternative versions of the elliptic curve DLP (MF-ECDLP) has been defined. This extension is proposed based on the elliptic scalar multiplication matrix function (ESMMF) that is applied to satisfy the DLP as explained in the following definitions.

Definition 4.3.4 Let EC be an elliptic curve over a prime field F_p and Let A and B be matrices in $EM_n(F_p)$. The left-side matrix function elliptic curve discrete logarithm problem (L- MF-ECDLP) is to find a matrix $L \in M_n(F_{\#EC(F_p)})$ such that $L \star A = B$. The left matrix L is called the L- MF-ECDLP of B to the matrix A.

Definition 4.3.5 Let EC be an elliptic curve over a prime field F_p and Let A and B be matrices in $EM_n(F_p)$. The right-side matrix function elliptic curve discrete logarithm problem (R- MF-ECDLP) is to find a matrix $R \in M_n(F_{\#EC(F_p)})$ such that $A \star R = B$. The right matrix R is called the R- MF-ECDLP of B to the matrix A.

Now, the alternative version of the cryptosystems based on the new proposed problems which are defined in Definitions (4.3.4) and (4.3.5) are discussed as follows.

The Elliptic Curve Matrix Function-Diffie – Hellman Key Exchange

On an alternative version of the ECDH key exchange which is named ECMF-DH key exchange, the public parameters are a prime number p, elliptic curve EC over F_p

and a matrix $D \in EM_n(F_p)$. Alice and Bob, agree to choose the secret left and right matrices $L \in M_n(F_{\#EC(F_p)})$ and $R \in M_n(F_{\#EC(F_p)})$ respectively. They compute $A = L \star D$ and $B = D \star R$. Alice sends A to Bob and Bob sends B to Alice. Both of them compute a shared secret key $K = L \star D * R$ by:

$$K = A^* = L \star B = L \star (D \star R) = (L \star D) \star R = A \star R = B^*$$

Algorithm (22) is used for obtaining the several numerical results of the revised ECMF-DH key exchange.

Algorithm 22 The ECMF-DH Key Exchange.

```
Input: A set $EC(F_p)$ of elliptic points and a matrix $D \in EM_n(F_p)$.
Output: A shared key $K$, where $K \in EM_n(F_p)$.
Alice:
  1: She chooses a matrix $L$ as her a private key, where $L \in M_n(F_{\#EC(F_p)})$.
  2: She computes her public key $A = L \star D$.
  3: Alice keys are $(L, A)$.
Bob:
  1: He chooses a matrix $R$ his a private key, where $R \in M_n(F_{\#EC(F_p)})$.
  2: He computes his public key $B = D \star R$.
  3: Bob keys are $(R, B)$.
Alice and Bob
  1: Computes the shared secret key $K = A^* = L \star B = L \star (D \star R) = (L \star D) \star R =$
     $A \star R = B^*$.
```

Example 4.3.1 Alice and Bob agree to to use EC is an elliptic curve defined by

$$EC : y^2 = x^3 + 13x + 2$$

over a prime field F_{131}, such that the set of all points satisfy the elliptic curve equation is given by

$$EC(F_{131}) = \{(1,4), (1,127), (2,6), (2,125), (8,15), (8,116), ..., O_E\}$$

and a public matrix

$$D = \begin{pmatrix} (2,6) & (66,126) \\ (129,19) & (68,130) \end{pmatrix} \in EM_2(F_{131}).$$

The order $\#EC(F_{131})$ of $EC(F_{131})$ is equal to 151.

- Alice chooses her secret key $L = \begin{pmatrix} 149 & 98 \\ 101 & 125 \end{pmatrix}$ and computes

$$A = L \star D = \begin{pmatrix} (108, 80) & (105, 64) \\ (51, 73) & (25, 25) \end{pmatrix},$$

as her public key.

- Bob chooses his secret key $R = \begin{pmatrix} 122 & 150 \\ 145 & 88 \end{pmatrix}$, and computes

$$B = D \star R = \begin{pmatrix} (64, 42) & (114, 34) \\ (129, 19) & (17, 114) \end{pmatrix},$$

as his public key.

Both of them compute the shared secret key

$$K = L \star B = A \star R = \begin{pmatrix} (105, 64) & (73, 51) \\ (96, 65) & (90, 90) \end{pmatrix}.$$

More implemented results of the ECMF-DH key exchange. can be seen in Chapter (5).

The Elliptic Curve Function Matrix-ElGamal Public Key Cryptosystem

The EEPKC can take alternative version based on the MF-ECDLP as explained in the following. Alice and Bob agreed to use an elliptic curve EC and a matrix $D \in EM_n(F_p)$. Alice chooses randomly her private key as a left matrix $L \in M_n(F_{\#EC(F_p)})$ and she computes her public key A by

$$A \equiv L \star D \ (mod\ p).$$

Bob employs a public key A to encrypt his plaintext, which is a matrix $M \in EM_n(F_p)$. He chooses an ephemeral key as a right matrix $R \in M_n(F_{\#EC(F_p)})$. The ciphertext C which is a pair (C_1, C_2) of two matrices are computed by

$$C_1 \equiv D \star R \ (mod\ p) \ \text{ and } \ C_2 \equiv M + A \star R \ (mod\ p).$$

The decryption process is done by Alice to recover the original plaintext. She first computes $-L \star C_1$ based on her secret key L. The matrix $C_2 - L \star C_1$ is computed to give the original plaintext M. Algorithms (23), (24) and (25) are used for obtaining the several numerical results of the revised ECMF-EPKC.

Algorithm 23 The ECMF-EPKC: Keys Generation Process.

```
Input: A set $EC(F_p)$ of elliptic points and a matrix $D \in EM_n(F_p)$.
Output: The public key $A$, where $A \in EM_n(F_p)$.
1: Alice chooses randomly her private key as a left matrix $L$, where $L \in M_n(F_{\#EC(F_p)})$.
2: She computes her public key $A \equiv L \star D \ (mod\ p)$.
3: Alice keys are $(L, A)$.
```

Algorithm 24 The ECMF-EPKC: Encryption Process.

```
Input: A set $EC(F_p)$ of elliptic points , a matrix $D \in EM_n(F_p)$ and a public key $A$.
Output: The ciphertext $(C_1, C_2)$, where $C_1$ and $C_2$ are two matrices in $EM_n(F_p)$.
1: Bob chooses an ephemeral key as a right matrix $R$ , where $R \in M_n(F_{\#EC(F_p)})$.
2: He chooses a plaintext $M \in EM_n(F_p)$.
3: He computes the ciphertext through the computations of two matrices $C_1 \equiv D \star R \ (mod\ p)$ and $C_2 = M + A \star R \ (mod\ p)$.
4: Bob sends the ciphertext pair $(C_1, C_2)$ to Alice.
```

The decryption process of the ECMF-EPKC is proved as follows.

Proposition 4.3.1 The decryption process is computed by $C_2 - L \star C_1 = M$.

Proof.

$$\begin{aligned} C_2 - L * C_1 &= M + A \star R - L \star C_1, \text{ since } C_2 = M + A \star R \\ &= M + A \star R - L * D \star R, \text{ since } C_1 = D \star -R \\ &= M + L \star D * R - L \star D * R, \text{ since } A = L \star D \\ &= M. \end{aligned}$$ ■

```
Algorithm 25 The ECMF-EPKC: Decryption Process.
Input: A set $EC(F_p)$ of elliptic points and the ciphertext pair $(C_1, C_2)$.
Output: The plaintext $M$.
1: Alice first uses her secret key $L$ to compute $-L \star C_1$.
2: She computes the matrix $C_2 - L \star C_1 = M$.
```

Example 4.3.2 Alice and Bob agreed to to use an elliptic curve EC which is defined by $EC : y^2 = x^3 + 2x + 7$. over a prime field F_{941}. The set of all points satisfy the elliptic curve equation is given by:

$$EC(F_{941}) = \{(4,199),(4,742),(6,519),(6,422),(8,522),(8,419),(9,80),(9,861),...,O_E\}$$

and a public matrix

$$D = \begin{pmatrix} (315,925) & (467,613) & (743,84) \\ (691,850) & (47,357) & (6,519) \\ (4,199) & (400,292) & (9,80) \end{pmatrix},$$

where $D \in EM_3(F_{941})$. The order $\#EC(F_{941})$ of $EC(F_{941})$ is equal to 887.

Keys generation process.

Alice performs the following:

- She chooses her secret key

$$L = \begin{pmatrix} 700 & 250 & 550 \\ 520 & 13 & 121 \\ 144 & 750 & 800 \end{pmatrix} \in M_3(F_{887})$$

and computes her public key

$$A = L \star D \ (mod\ 941) \equiv \begin{pmatrix} (116,937) & (30,890) & (90,525) \\ (456,785) & (624,534) & (692,381) \\ (703,147) & (501,887) & (323,122) \end{pmatrix}.$$

Encryption process.

Bob does the following steps:

- He choose his plaintext M as matrix to send Alice the plaintext

$$M = \begin{pmatrix} (894,58) & (570,585) & (550,755) \\ (856,491) & (636,417) & (86,339) \\ (455,680) & (455,680) & (521,568) \end{pmatrix} \in EM_3(F_{941}).$$

- He chooses his ephemeral key as a matrix

$$R = \begin{pmatrix} 450 & 150 & 639 \\ 333 & 275 & 750 \\ 720 & 885 & 115 \end{pmatrix} \in M_3(F_{887}).$$

- He computes

$$A \star R \ (mod\ 941) \equiv \begin{pmatrix} (529,514) & (189,75) & (464,462) \\ (407,82) & (738,300) & (277,609) \\ (433,729) & (35,685) & (777,45) \end{pmatrix}.$$

- He computes his ciphertext

$$C_1 = D \star R \ (mod\ 941) \equiv \begin{pmatrix} (659,254) & (445,871) & (90,416) \\ (141,493) & (894,883) & (600,224) \\ (196,102) & (873,105) & (240,550) \end{pmatrix}$$

and

$$C_2 = M + A \star R \ (mod\ 941) \equiv \begin{pmatrix} (177,392) & (515,726) & (681,57) \\ (75,430) & (642,863) & (140,772) \\ (196,839) & (407,82) & (568,730) \end{pmatrix}.$$

- He sends his ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She uses her private key L, first computes

$$-L \star C_1 \ (mod\ 941) \equiv \begin{pmatrix} (529,427) & (189,866) & (464,479) \\ (407,859) & (738,641) & (277,332) \\ (433,212) & (35,256) & (777,896) \end{pmatrix}.$$

- Finally, she computes $C_2 - L \star C_1$ which is equal to original plaintext

$$M = \begin{pmatrix} (894,58) & (570,585) & (550,755) \\ (856,491) & (636,417) & (86,339) \\ (455,680) & (455,680) & (521,568) \end{pmatrix}.$$

4.4 The Elliptic Scalar Multiplication Graph and its Applications in Elliptic Curve Cryptosystems

In this section, a new graph based on an elliptic scalar multiplication operation is defined which is called elliptic scalar multiplication (ESM) graph. A new contribution of an asymmetric encryption algorithm has been proposed. This algorithm is built using the elliptic curve defined over a prime field and undirected complete graphs.

4.4.1 The Scalar Multiplication Graph of Elliptic Curve Group

A scalar multiplication kR on an elliptic curve EC defined over a prime field is employed to form a new graph, its vertices are the elliptic curve points and its edges are the elliptic scalar multiplication kR. This graph is defined as follows.

Definition 4.4.1 An elliptic scalar multiplication graph $G_{ESM}(EC(F_p), kR)$ of the elliptic curve points is a graph whose vertex set is $EC(F_p)$ and a scalar multiplication operation set as an edge set of two distinct vertices P and Q which are adjacent if and only if there exist positive integer such that either $nP = Q$ or $nQ = P$, $n \in [2, \#EC(F_p)]$. In other word, the graph $G_{ESM}(EC(F_p), kR)$ is given by:

$$G_{ESM}(EC(F_p), kR) = \{V = \{\forall\ P \in EC(F_p)\}, E = \{e_i = (P, Q) = kR \text{ such that } nP = Q \text{ or } nQ = P,\ \forall\ P, Q \in EC(F_p) \text{ and } n \in [2, \#E(F_p)]\}\}$$

Example 4.4.1 If EC is an elliptic curve defined by $y^2 = x^3 + x + 1$ over F_3. All elliptic points which lie on EC form a set

$$EC(F_3) = \{(0,1), (0,2), (1,0), O_E\}$$

Since $2(0,1) = (1,0)$, $3(0,1) = (0,2)$, $4(0,1) = O_E$, $2(0,2) = (1,0)$, $3(0,2) = (0,1)$, $4(0,2) = O_E$ and $2(1,0) = O_E$. Then the elliptic scalar multiplication graph $G_{ESM}(EC(F_3), kR)$ of the elliptic points $EC(F_3)$ has four vertices and six edges as shown in Figure 4.1.

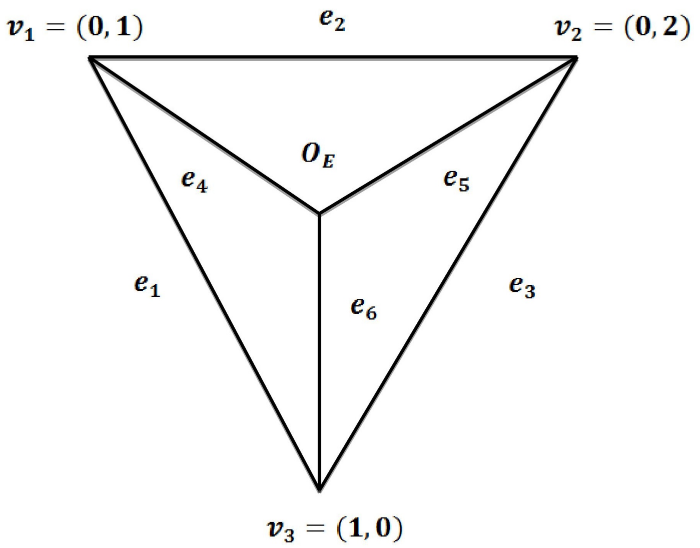

Figure 4.1: The elliptic scalar multiplication graph $G_{ESM}(EC(F_p), kR)$ of $EC(F_3)$.

Proposition 4.4.1 Let $EC(F_p)$ be an elliptic curve group. Then, a scalar multiplication graph $G_{ESM}(EC(F_p), kR)$ is a complete if and only if for any two subgroup $\langle P\rangle$, $\langle Q\rangle$ of $EC(F_p)$, then either $\langle P\rangle \subseteq \langle Q\rangle$ or $\langle Q\rangle \subseteq \langle P\rangle$, with $P \neq \mp Q$.

Proof: Suppose $G_{ESM}(EC(F_p), kR)$ is a complete graph and let $\langle P\rangle$ and $\langle Q\rangle$ be two subgroups of $EC(F_p)$ such that $P \neq \mp Q$, by assumption. Then the elliptic points P and Q are adjacent. So, either $nP = Q$ or $nQ = P$ is satisfied, where $n \in [2, \#EC(F_p)]$. This means that, either $Q \in \langle P\rangle$ or $P \in \langle Q\rangle$. Hence, either $\langle Q\rangle \subseteq \langle P\rangle$ or $\langle P\rangle \subseteq \langle Q\rangle$.

Conversely, suppose for any two subgroups $\langle P\rangle, \langle Q\rangle$ from $EC(F_p)$, then either $\langle P\rangle \subseteq \langle Q\rangle$ or $\langle Q\rangle \subseteq \langle P\rangle$, $\forall P, Q \in EC(F_p)$, with $P \neq \mp Q$. To prove $G_{ESM}(EC(F_p), kR)$ is a complete graph, let P' and Q' are two distinct points in $EC(F_p)$, then by assumption either $\langle P'\rangle \subseteq \langle Q'\rangle$ or $\langle Q'\rangle \subseteq \langle P'\rangle$. Thus, either $nP' = Q'$ or $nQ' = P'$. Hence, P' adjacent to Q' in $G_{ESM}(EC(F_p), kR)$. Therefore, the graph $G_{ESM}(EC(F_p), kR)$ is a complete. ■

Theorem 4.4.1 Let $EC(F_p)$ be an elliptic curve group. Then, a scalar multiplication graph $G_{ESM}(EC(F_p), kR)$ is a complete if and only if a group $EC(F_p)$ of order q^t, where q is a prime number and t is any positive integer.

Proof: Let $G_{ESM}(EC(F_p), kR)$ be a complete graph. The proof of this theorem will be done by contradiction. Suppose α and β are two distinct primes such that $\alpha\beta = \#EC(F_p)$. Then by [12], p 131, $EC(F_p)$ must have two distinct subgroups

$\langle P\rangle$ and $\langle Q\rangle$ with order α and β respectively. In this case $\langle P\rangle \nsubseteq \langle Q\rangle$ and $\langle Q\rangle \nsubseteq \langle P\rangle$, that is, P and Q are not adjacent in $G_{ESM}(EC(F_p), kR)$. Thus, this leads to a contradiction. Therefore $\#EC(F_p) = q^t$.

Conversely, if $\#EC(F_p) = q^t$, Then by Lagrange Theorem any P in $EC(F_p)$ is order q^k for some $k \leq t$. Hence if P and Q are two points with order q^j, q^k respectively, then if $j < k$ implies $\langle P\rangle \subseteq \langle Q\rangle$ and $k < j$ implies $\langle Q\rangle \subseteq \langle P\rangle$. Thus, by Proposition (4.4.1), the graph $G_{ESM}(EC(F_p), kR)$ is a complete. ■

Corollary 4.4.1 Let $EC(F_p)$ be an elliptic curve group. Its order $\#EC(F_p) = q^t$, where q is a prime number and t is a positive integer. Then every subset of $EC(F_p)$ corresponds to the complete induced subgraph.

Proof: Let S be a subset of $EC(F_p)$, then by Theorem (4.4.1), $EC(F_p)$ corresponds to the complete graph $G_{ESM}(EC(F_p), kR)$. Thus, every two elements P', Q' in S are adjacent in the induced subgraph $H_{ESM}(EC(F_p), kR)$. Thus, S corresponds to a complete induced subgraph. ■

4.4.2 The Complete Elliptic Scalar Multiplication Graph Public Key Cryptosystem

In this section, an asymmetric cryptosystem has been proposed based on the complete elliptic scalar multiplication (CESM) graph, which is formed depending on the elliptic points and the complete graph.

The public parameters of the proposed cryptosystem are an elliptic curve EC over a prime field F_p with it's a prime order l, the matrices $D \in GL_n(F_p)$, and $K \in M_n(EC(F_p))$, where $EC(F_p)$ is a set of all points lie of an elliptic curve EC. Alice and Bob want to communicate for exchanging the information. Alice chooses randomly a private key as a number a such that $a \in \{2, 3, ..., p-1\}$ and computes a public key $A \equiv D^a \ (mod\ p)$. So Alice's keys are given with a pair (a, A). For obtaining the several numerical results to generate the keys, one can use Algorithm (2).

Bob wants to communicate with Alice for sending the important information which is represented by plaintext $M(F_p)$ that is considered as a subset of $EC(F_p) \backslash \{O_E\}$ such that $P, -P \in M(F_p)$. Bob converts his plaintext $M(F_p)$ into $M^{\bullet}(F_p)$ such that for all $P \in M(F_p)$ if $P \in M^{\bullet}(F_p)$ then $-P \notin M^{\bullet}(F_p)$. Then, the set $M^{\bullet}(F_p)$ converted into a complete graph by using Theorem (4.4.1) and Corollary

(4.4.1). He selects a code number of each elliptic point in $M^{\bullet}(F_p)$ based on $x-$ coordinates of these points. Then, the complete graph converted into a weighted complete graph through computing the weights w_i, for $i = 1, 2, 3, ...$ for all edges, which equal to the distance $\mid cod(v_j) - cod(v_{j+1}) \mid$, $j = 1, 2, ..., n-1$. MST of a weighted complete graph is determined and represented it by an adjacent matrix B_1. Later on, a matrix B_1 is converted into B_2 by adding a code numbers to diagonal B_1. Bob chooses an ephemeral key as secret key b such that $b \in \{2, 3, ..., p-1\}$. He uses Alice's public key A and a matrix B_2 to compute $B_3 = A^b \times B_2$. The ciphertext C which is a pair (C_1, C_2) of two matrices are computed by $C_1 \equiv D^b \ (mod\ p)$ and $C_2 = B_3 \times K = \{b_{11}(x_{11}, y_{11}), b_{12}(x_{12}, y_{12}), ..., b_{nn}(x_{nn}, y_{nn})\}$. Bob sends the ciphertext pair $C = (C_1, C_2)$ to Alice. Several numerical results to compute the ciphertext are got using Algorithm (26).

Algorithm 26 The CESM Graph Public Key Cryptosystem: Encryption Process.

```
Input: The matrices $D \in GL_n(F_p)$, $K \in M_n(EC(F_p))$ and a public key $A$.
Output: The ciphertext $(C_1, C_2)$, where $C_1$ and $C_2$ are two matrices .
Bob selects his plaintext $M(F_p)$ which is chosen randomly as a subset of $EC(F_p)\setminus\{O_E\}$ such that if $P \in M(F_p)$ then $-P \in M(F_p)$.
He converts a plaintext $M(F_p)$ into $M^{\bullet}(F_p)$ such that for all $P \in M(F_p)$ if $P \in M^{\bullet}(F_p)$ then $-P \notin M^{\bullet}(F_p)$.
He converts $M^{\bullet}(F_p)$ to complete graph by using Theorem (4.4.1) and Corollary (4.4.1).
He selects a code number of each elliptic point in $M^{\bullet}(F_p)$ based on $x-$coordinates of these points.
He converts a complete graph into a weighted complete graph through computing the weights $w_i$, with $i = 1, 2, 3, ...$, for all edges, which equal to the distance
    $$w_i =\mid cod(v_j) - cod(v_{j+1}) \mid, \; j = 1, 2, ..., n-1.$$
He determines a MST of a weighted complete graph and represents it by an adjacent matrix $B_1$.
He converts an adjacent matrix $B_1$ into $B_2$ through adding a code numbers to diagonal $B_1$.
He chooses an ephemeral key as secret key $b$ such that $b \in \{2, 3, ..., p-1\}$.
He uses Alice's public key $A$ and a matrix $B_2$ to compute $B_3 \equiv A^b \times B_2 \; (mod \; p)$.
He computes $C_1 \equiv D^b \; (mod \; p)$ and $C_2$ through multiplying the elements $b_{ij}$ in a matrix $B_3$ by the points $(x_{nn}, y_{nn})$ in a matrix $K$ one by one. Bob can apply any elliptic scalar multiplication algorithm such the ISD algorithm [4]. In other words, $C_2 = B_3 \times K = \{b_{11}(x_{11}, y_{11}), b_{12}(x_{12}, y_{12}), ..., b_{nn}(x_{nn}, y_{nn})\}$.
Bob sends the ciphertext pair $(C_1, C_2)$ to Alice.
```

Upon receiving Alice the ciphertext $C = (C_1, C_2)$, she uses her private key a to recover the plaintext. She first computes a matrix $C_1^a \; (mod \; p)$. Then, an inverse matrix $(C_1^a)^{-1} \; (mod \; p)$ is calculated. After that, a matrix B_3 computed based on C_2 through solving the ECDLP using the Baby step and Giant step algorithm [20]. Then the computation of the multiplication matrix $(C_1^a)^{-1} \times B_3 \; (mod \; p)$ which is equal to B_2. The diagonal elements, in B_2, are selected as a list that forms a set of the code numbers of elliptic points. Finally, she converts the code numbers into the plaintext $M(F_p)$. The implemented results for recovering the original plaintext can be got using Algorithm (27).

Algorithm 27 The CESM Graph Public Key Cryptosystem: Decryption Process

```
Input: A prime $p$ and a ciphertext $(C_1, C_2)$
Output: The plaintext $M(F_p)$, where $M(F_p)$ is a subset of $EC(F_p)\backslash \{O_E\}$.
1: Alice first uses her secret key $a$ to compute $C_1^a$ $(mod\ p)$.
2: She computes the elements of a matrix $B_3$ based on $C_2$ through solving the
   ECDLP using the Baby step and Giant step algorithm [20].
3: She computes the multiplication matrix $(C_1^a)^{-1} \times B_3$ $(mod\ p) \equiv B_2$.
4: The diagonal elements of $B_2$ are selected as a list that forms a set of the code
   numbers of elliptic points.
5: She converts the code numbers into original plaintext $M(F_p)$.
```

Example 4.4.2 (The CESM graphic public key cryptosystem)

Suppose $EC : y^2 = x^3 + 3x + 2$ over a prime field F_{151}. The set of all points satisfy the elliptic curve equation is given by:

$$EC(F_{151}) = \{(2,4),(2,147),(3,76),(3,75),(4,94),(4,57),...,O_E\}.$$

The $\#EC(F_{151}) = 163$ is a prime. A public matrix

$$D = \begin{pmatrix} 155 & 3 & 61 & 7 & 120 \\ 16 & 9 & 160 & 70 & 125 \\ 13 & 4 & 80 & 43 & 21 \\ 55 & 135 & 140 & 161 & 29 \\ 77 & 99 & 14 & 5 & 101 \end{pmatrix} \in GL_5(F_{163})$$

and

$$K = \begin{pmatrix} (88,107) & (6,38) & (15,10) & (35,131) & (4,57) \\ (86,105) & (53,1) & (27,102) & (16,89) & (45,80) \\ (40,20) & (86,46) & (41,48) & (71,135) & (61,45) \\ (31,21) & (2,4) & (88,44) & (83,79) & (33,10) \\ (7,8) & (3,76) & (12,16) & (19,24) & (68,16) \end{pmatrix},$$

where K is a square matrix, its elements are elliptic curve points that are chosen randomly from $EC(F_{151})$ as a public parameter.

Keys generation process.

Alice performs the following:

- She chooses her secret key $a = 110$ and she computes her public key

$$A = D^{110}\ (mod\ 163) \equiv \begin{pmatrix} 123 & 101 & 8 & 118 & 32 \\ 12 & 6 & 39 & 148 & 112 \\ 69 & 77 & 50 & 9 & 72 \\ 149 & 107 & 59 & 33 & 90 \\ 15 & 153 & 9 & 51 & 32 \end{pmatrix},$$

  so, her keys are $a = 110$ and $A = D^{110}$.

Encryption process.

Bob does the following steps:

- He chooses his private key $b = 151$ and computes

$$C_1 = D^{151}\ (mod\ 163) \equiv \begin{pmatrix} 3 & 137 & 111 & 109 & 149 \\ 50 & 37 & 61 & 44 & 91 \\ 38 & 28 & 105 & 64 & 63 \\ 66 & 63 & 3 & 99 & 20 \\ 55 & 38 & 116 & 148 & 55 \end{pmatrix}$$

  and

$$A^{151}\ (mod\ 163) \equiv \begin{pmatrix} 119 & 120 & 74 & 132 & 18 \\ 36 & 121 & 30 & 91 & 102 \\ 73 & 25 & 13 & 72 & 108 \\ 26 & 20 & 28 & 83 & 144 \\ 9 & 48 & 85 & 127 & 3 \end{pmatrix}.$$

- He chooses randomly his plaintext

$$M(F_{151}) = \{(80,72),(80,79),(29,4),(29,147),(47,57),(47,94),(56,18),$$
$$(56,133),(140,65),(140,86)\},$$

  as a subset of elliptic curve group $EC(F_{151})$.

- He converts a plaintext $M(F_{151})$ into

$$M^{\bullet}(F_{151}) = \{(80,72),(29,4),(47,57),(56,18),(140,65)\}$$

by using Theorem (4.4.1), the set $EC(F_{151})$) corresponds to a complete elliptic curve graph.

- According to Corollary (4.4.1), $M^{\bullet}(F_{151})$ corresponds to a CESM subgraph as seen in Figure 4.2.

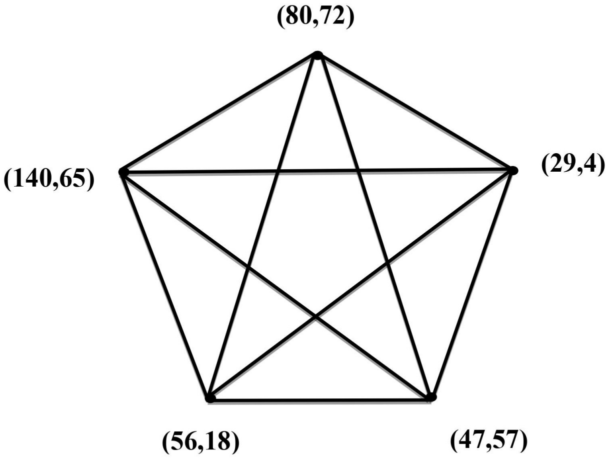

Figure 4.2: The CESM subgraph of $M^{\bullet}(F_{151})$

- Each elliptic point in a set $M^{\bullet}(F_{151})$ has a code numbers given in Table (4.1).

Table 4.1: Code numbers of elliptic curve points in a set $M^{\bullet}(F_{151})$.

Points in $M^{\bullet}(F_{151})$	Code number
$(80, 72)$	80
$(29, 4)$	29
$(47, 9)$	47
$(56, 18)$	56
$(80, 72)$	80

- Each edge of CESM subgraph has a weight which is computed as a distance between two connected vertices dependent on the coding number table. For instance, the weight w_{12} is computed by

$$w_{12} =\mid cod(v_1) - cod(v_2) \mid=\mid 80 - 29 \mid= 51.$$

The weights computations of all edges are given in Table (4.2).

Table 4.2: The weights of all edges of a CESM subgraph G.

The weight	Weight value
w_{12}	51
w_{13}	33
w_{14}	24
w_{15}	60
w_{23}	18
w_{24}	27
w_{25}	111
w_{34}	9
w_{35}	93
w_{45}	86

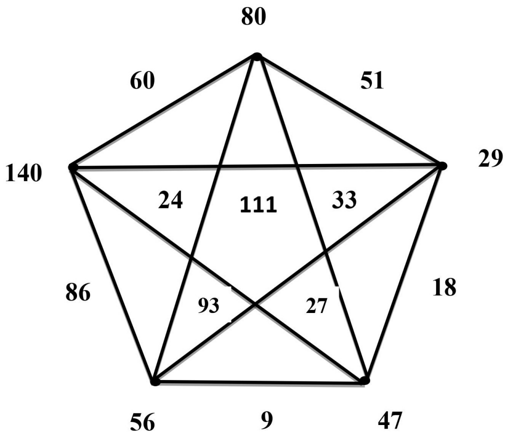

Figure 4.3: The weighted CESM subgraph of $M^{\bullet}(F_{151})$

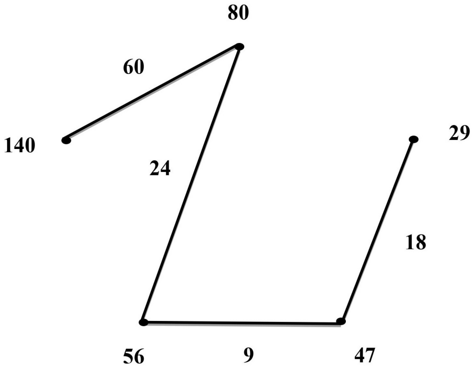

Figure 4.4: The MST graph of the weighted CESM subgraph.

The adjacent matrix of MST graph is

$$B_1 = \begin{pmatrix} 0 & 0 & 0 & 24 & 60 \\ 0 & 0 & 18 & 0 & 0 \\ 0 & 18 & 0 & 9 & 0 \\ 24 & 0 & 9 & 0 & 0 \\ 60 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

- The modified matrix B_2 of the matrix B_1 is done through adding the code numbers of elliptic points in a set $M^{\bullet}(F_{151})$ at the diagonal. So

$$B_2 = \begin{pmatrix} 80 & 0 & 0 & 24 & 60 \\ 0 & 29 & 18 & 0 & 0 \\ 0 & 18 & 47 & 9 & 0 \\ 24 & 0 & 9 & 56 & 0 \\ 60 & 0 & 0 & 0 & 140 \end{pmatrix}.$$

- Multiplying the matrices A^{151} and B_2 to find a matrix B_3 as follows.

$$B_3 = A^{151} \times B_2 \ (mod\ 163) \equiv \begin{pmatrix} 76 & 85 & 143 & 156 & 43 \\ 100 & 137 & 6 & 36 & 140 \\ 30 & 144 & 79 & 33 & 103 \\ 161 & 106 & 141 & 145 & 41 \\ 36 & 151 & 134 & 106 & 145 \end{pmatrix}.$$

- He computes

$$C_2 = B_3 \times K = \begin{pmatrix} (102,52) & (140,65) & (140,86) & (144,98) & (68,135) \\ (128,65) & (106,61) & (12,16) & (29,147) & (47,57) \\ (118,31) & (63,22) & (141,123) & (97,110) & (2,147) \\ (105,15) & (56,133) & (47,57) & (16,89) & (136,31) \\ (6,113) & (88,44) & (2,147) & (114,124) & (7,143) \end{pmatrix}.$$

- He sends the ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She computes $(C_1^{110})^{-1}$,

$$((C_1)^{110})^{-1}\ (mod\ 163) \equiv \begin{pmatrix} 61 & 77 & 40 & 35 & 90 \\ 132 & 63 & 145 & 0 & 159 \\ 92 & 53 & 64 & 17 & 68 \\ 132 & 37 & 94 & 152 & 147 \\ 51 & 94 & 26 & 42 & 12 \end{pmatrix},$$

  where

$$(C_1)^{110} = A^{151}\ (mod\ 163) \equiv \begin{pmatrix} 119 & 120 & 74 & 132 & 18 \\ 36 & 121 & 30 & 91 & 102 \\ 73 & 25 & 13 & 72 & 108 \\ 26 & 20 & 28 & 83 & 144 \\ 9 & 48 & 85 & 127 & 3 \end{pmatrix},$$

- She computes a matrix B_3 from C_2 which its elements represent the ECDLPs, using the Baby step and Giant step algorithm [20].

- She computes a matrix

$$(C_1^3)^{-1} \times B_3\ (mod\ 163) \equiv \begin{pmatrix} 80 & 0 & 0 & 24 & 60 \\ 0 & 29 & 18 & 0 & 0 \\ 0 & 18 & 47 & 9 & 0 \\ 24 & 0 & 9 & 56 & 0 \\ 60 & 0 & 0 & 0 & 140 \end{pmatrix},$$

which is equal to a matrix B_2 The elements diagonal of the matrix B_2 represent the x-coordinates of the elliptic points. For instance, the first element 80 in a matrix B_2 gives two possibilities of elliptic points, $(80, 72)$, $(80, 79)$. The computations for other elements can be seen in Table (4.3).

Table 4.3: The elliptic curve points that correspond to the diagonal elements of the matrix B_2.

Diagonal elements of the matrix B_2	Elliptic points
29	$(29, 4), (29, 147)$
47	$(47, 57), (47, 94)$
56	$(56, 18), (56, 133)$
140	$(140, 65), (140, 86)$

Then the original plaintext

$$M(F_{151}) = \{(80, 72), (80, 79), (29, 4), (29, 147), (47, 57), (47, 94), (56, 18), (56, 133), (140, 65), (140, 86)\}$$

4.5 The Elliptic Scalar Multiplication Digraph for Elliptic Curve Cryptographic Usages

In this section, a new digraph has been defined to design an alternative asymmetric cryptosystem. This digraph, which is called an elliptic scalar multiplication (ESM) digraph, is created based on an elliptic scalar multiplication operation defined over a prime field. The ESM digraph is proved theoretically as a complete symmetric (CS) digraph. An asymmetric cryptosystem has been implemented on the complete symmetric elliptic scalar multiplication (CSESM) digraph. The adjacent matrix representation and the MST of the CSESM digraph are used to compute the ciphertext. Fast computations are obtained on the proposed cryptosystem. More secure communications are determined with the CSESM di-graphic cryptosystem compared, to the previous asymmetric cryptosystems.

4.5.1 The Scalar Multiplication Digraph of Elliptic Curve Group

In this section, the scalar multiplication kR on the elliptic curve E defined over a prime field is employed to form a new digraph, its vertices are the elliptic curve points and its arcs are the elliptic scalar multiplication kR. This digraph is defined as follows.

Definition 4.5.1 An ESM digraph $DG_{ESM}(EC(F_P), kR)$ of the elliptic curve points is a graph whose vertex set is $EC(F_p)\backslash\{O_E\}$ and a scalar multiplication operation set as an arc set of two distinct vertices P and Q. There exist an arc from P to Q if there exist a positive integer such that $nP = Q$ with $n \in [2, \#EC(F_p)]$.

Example 4.5.1 if E is an elliptic curve defined by $y^2 = x^3 + x + 1$ over F_3. All elliptic points which lie on E form a set

$$EC(F_3) = \{(0,1), (0,2), (1,0), O_E\}$$

Since $2(0,1) = (1,0)$, $3(0,1) = (0,2)$, $4(0,1) = O_E$, $2(0,2) = (1,0)$, $3(0,2) = (0,1)$, $4(0,2) = O_E$ and $2(1,0) = O_E$. Then the ESM digraph $DG_{ESM}(EC(F_3), kR)$ of the elliptic points set $EC(F_3)$ has three vertices $v_1 = (0,1), v_2 = (0,2)$ and $v_3 = (1,0)$ and four arcs e_1, e_2, e_3 and e_4 as shown in Figure 4.5.

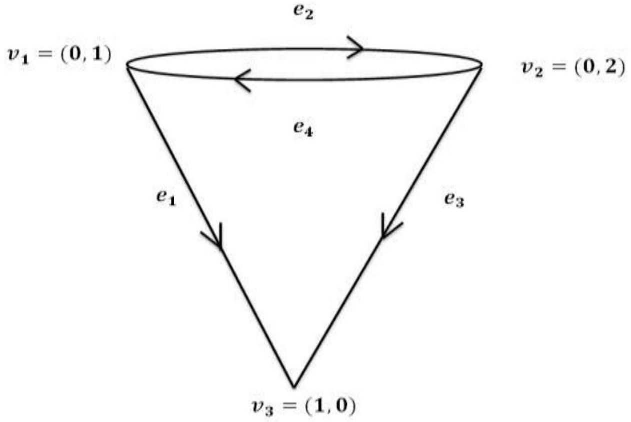

Figure 4.5: The ESM digraph of $EC(F_3)$.

Theorem 4.5.1 Let $EC(F_p)$ be an elliptic curve group. Then an ESM digraph $DG_{ESM}(EC(F_P), kR)$ is a CS digraph if and only if an order of a group $EC(F_p)$ is a prime.

Proof. Let $DG_{ESM}(EC(F_P), kR)$ be a CS digraph. The proof of this theorem will be done by contradiction. Suppose α and β are two distinct primes such that $\alpha\beta = \#EC(F_p)$. Then by [12], p 131, $EC(F_p)$ must have two distinct subgroups $\langle P\rangle$ and $\langle Q\rangle$ with order α and β respectively. In this case $\langle P\rangle \nsubseteq \langle Q\rangle$ and $\langle Q\rangle \nsubseteq \langle P\rangle$. This means that, there are no positive integers n_1 and n_2 such that $P = n_1 Q$ and $Q = n_2 P$. So, there is no an arc between P and Q in $DG_{ESM}(EC(F_P), kR)$. This contradicts our hypothesis. Therefore, an order of a group $EC(F_p)$ is a prime. Conversely, let $EC(F_p)$ be an elliptic curve group of a prime order ℓ. Since an order of $EC(F_p)$ is a prime, then $\langle P\rangle = EC(F_p)$, for all point P in $EC(F_p)$. It follows that there exist two positive integers n_1 and n_2 such that $P = n_1 Q$ and $Q = n_2 P$, for all $P, Q \in EC(F_p)$. Hence, the digraph $DG_{ESM}(EC(F_P), kR)$ is a CS digraph. ■

Corollary 4.5.1 Let $EC(F_p)$ be an elliptic curve group. Its prime order $\#EC(F_p) = \ell$. Then, for each subset of $EC(F_p)$ corresponds to a CS induced sub-digraph.

Proof. Let S be a subset of $EC(F_p)$, then by Theorem (4.5.1) $EC(F_p)$ corresponds to the cs digraph $DG_{ESM}(EC(F_P), kR)$. Thus, every two points $P, Q \in EC(F_p)$ are adjacent in the this graph. This means that, there are arcs from P into Q and from Q into P. So, every two elements P', Q' in S are adjacent in an induced sub-digraph $H_{ESM}(EC(F_P), kR)$. Thus, there exist arcs from P' to Q' and also from Q' to P'. Hence, S corresponds to a CS sub-digraph. ■

4.5.2 The Complete Symmetric Elliptic Scalar Multiplication Digraph Cryptosystem

An asymmetric cryptosystem has been proposed based on the CSESM digraph. The proposed cryptosystem depended on the elliptic points and the CS digraph. The public parameters are an elliptic curve $EC(F_p)$ with a prime order ℓ, the matrices $D \in GL_n(F_p)$, and $K \in M_n(EC(F_p))$. Alice and Bob want to communicate for exchanging the information. Alice chooses randomly a private key as a number a such that $a \in \{2, 3, ..., p-1\}$ and computes a public key $A \equiv D^a \ (mod\ p)$. So Alice's keys are given with a pair (a, A). For obtaining the several numerical results to generate the keys, one can use Algorithm (2).

Bob wants communicate with Alice for sending the important information which is represented by plaintext $M(F_p)$ that is considered as a subset of $EC(F_p)\setminus\{O_E\}$ such that $P, -P \in M(F_p)$. Bob converts his plaintext $M(F_p)$ into $M^{\bullet}(F_p)$ such that for all $P \in M(F_p)$ if $P \in M^{\bullet}(F_p)$ then $-P \notin M^{\bullet}(F_p)$. Then the set $M^{\bullet}(F_p)$ converted into a CS digraph by using Theorem (4.5.1) and Corollary (4.5.1). He converted a CS digraph into a weighted complete symmetric (WCS) digraph. This conversion is done by computing the weights w_i, with $i = 1, 2, 3, ...$ for all arcs, using the scalar multiplication between any two vertices. MST of a WCS digraph is determined and represented by an adjacent matrix B_1. Later on, a matrix B_1 is converted into B_2 through adding the x -coordinates of each elliptic point in $M^{\bullet}(F_p)$ into the diagonal of a matrix B_1 respectively. Bob chooses an ephemeral key as secret key b such that $b \in \{2, 3, ..., p-1\}$. He uses Alice's public key A and a matrix B_2 to compute $B_3 = A^b \times B_2$. The ciphertext C which is a pair (C_1, C_2) of two matrices are computed by $C_1 \equiv D^b \ (mod\ p)$ and $C_2 = B_3 \times K = \{b_{11}(x_{11}, y_{11}), b_{12}(x_{12}, y_{12}), ..., b_{nn}(x_{nn}, y_{nn})\}$. Bob sends the ciphertext pair $C = (C_1, C_2)$ to Alice. Several numerical results to compute the ciphertext are got using Algorithm (28).

Algorithm 28 The CS-ESMD Public Key Cryptosystem: Encryption Process.

```
Input: The matrices $D \in GL_n(F_p)$ and $K \in M_n(EC(F_p))$ and a public key $A$.
Output: The ciphertext $(C_1, C_2)$, where $C_1$ and $C_2$ are two matrices .
1: Bob selects his plaintext $M(F_p)$ which is chosen randomly as a subset of $EC(F_p)\backslash \{O_E\}$ such that if $P \in M(F_p)$ then $-P \in M(F_p)$.
2: He converts a plaintext $M(F_p)$ into $M^{\bullet}(F_p)$. In other words, if $P \in M^{\bullet}(F_p)$ then $-P \notin M^{\bullet}(F_p)$, for all $P \in M(F_p)$.
3: Using Theorem (4.5.1) and Corollary (4.5.1), the $M^{\bullet}(F_p)$ is converted into a CS digraph.
4: A CS digraph is converted into a WCS digraph. This conversion is done by computing the weights $w_i$, with $i = 1, 2, 3, ...$ for all arcs, using the scalar multiplication between any two vertices.
5: He determines a MST of a WCS digraph and represents it by an adjacent matrix $B_1$.
6: He converts an adjacent matrix $B_1$ into a matrix $B_2$ by adding the x-coordinates of each elliptic point in $M^{\bullet}(F_p)$ into the diagonal of a matrix $B_1$.
7: He chooses an ephemeral key as secret key $b$ such that $b \in \{2, 3, ..., p-1\}$.
8: He uses Alice's public key $A$ and a matrix $B_2$ to compute $B_3 \equiv A^b \times B_2 \ (mod\ p)$.
9: He computes $C_1 \equiv D^b \ (mod\ p)$ and $C_2$ through multiplying the elements $b_{ij}$ in a matrix $B_3$ by the points $(x_{nn}, y_{nn})$ in a matrix $K$ one by one. Bob can apply any elliptic scalar multiplication algorithm such the ISD algorithm [4]. In other words, $C_2 = B_3 \times K = \{b_{11}(x_{11}, y_{11}), b_{12}(x_{12}, y_{12}), ..., b_{nn}(x_{nn}, y_{nn})\}$
10: Bob sends the ciphertext pair $(C_1, C_2)$ to Alice.
```

Upon receiving Alice the ciphertext $C = (C_1, C_2)$, she uses her private key a to recover the plaintext. She first computes a matrix $C_1^a \ (mod\ p)$. Then, an inverse matrix $(C_1^a)^{-1} \ (mod\ p)$ is calculated. After that, a matrix B_3 computed based on C_2 through solving the ECDLP using the Baby step and Giant step algorithm [20]. Then the computation of the multiplication matrix $(C_1^a)^{-1} \times B_3 \ (mod\ p)$ which is equal to B_2. The diagonal elements, in B_2, are selected as a list which they form a set of the code numbers of elliptic points. Finally, Converts the code numbers into the plaintext $M(F_p)$. The implemented results for recovering the original plaintext can be got using Algorithm (27).

Example 4.5.2 Suppose EC is an elliptic curve defined by $EC : y^2 = x^3 + 5x + 5$ over a prime field F_{941}. All points lying on EC form a set $EC(F_{941})$ which is computed by:

$$EC(F_{941}) = \{(0, 486), (0, 455), (3, 762), (3, 179), ..., O_E\}.$$

The order of a set $EC(F_{941})$ is a prime number that is equal to 997. The public matrices

$$D = \begin{pmatrix} 552 & 123 & 336 & 911 \\ 471 & 962 & 14 & 654 \\ 12 & 852 & 136 & 7 \\ 125 & 177 & 2 & 25 \end{pmatrix} \in GL_4(F_{997})$$

and

$$K = \begin{pmatrix} (154, 807) & (884, 269) & (19, 833) & (849, 486) \\ (338, 298) & (416, 211) & (148, 333) & (321, 241) \\ (911, 929) & (775, 253) & (903, 544) & (163, 12) \\ (252, 229) & (522, 121) & (510, 247) & (60, 481) \end{pmatrix},$$

where K is a square matrix. The, elements are elliptic curve points chosen randomly from $EC(F_{941)}$.

Now, Alice performs the following steps:

- She chooses a secret key $a = 5$.
- She computes her public key A by

$$A = D^5 \ (mod\ 997) \equiv \begin{pmatrix} 417 & 913 & 734 & 20 \\ 403 & 559 & 846 & 651 \\ 633 & 286 & 200 & 993 \\ 816 & 506 & 77 & 469 \end{pmatrix},$$

  so, her keys are $a = 5$ and $A = D^5$.

The encryption process.

Bob does the following steps:

- He chooses a secret key $b = 3$.

- He computes

$$C_1 = D^3 \ (mod\ 997) \equiv \begin{pmatrix} 639 & 551 & 127 & 340 \\ 842 & 652 & 208 & 907 \\ 639 & 994 & 541 & 231 \\ 206 & 836 & 640 & 258 \end{pmatrix},$$

$$A^3 \ (mod\ 997) \equiv \begin{pmatrix} 912 & 569 & 266 & 15 \\ 494 & 519 & 350 & 223 \\ 884 & 890 & 845 & 529 \\ 846 & 74 & 992 & 293 \end{pmatrix}.$$

- He chooses his plaintext

$$M(F_{941}) = \{(3,179),(3,762),(165,205),(165,736),(775,688),(775,253),$$
$$(900,175),(900,766)\},$$

  which is a subset of $EC(F_{941})$.

- He converts a plaintext $M(F_{941})$ into

$$M^{\bullet}(F_{941}) = \{(3,179),(165,205),(775,688),(900,175)\}$$

- Based on Theorem (4.5.1), The set $EC(F_{941})$ corresponds to a CS digraph and according to Corollary (4.5.1), $M^{\bullet}(F_{941})$ corresponds to a CSESM sub-digraph as seen in Figure 4.6.

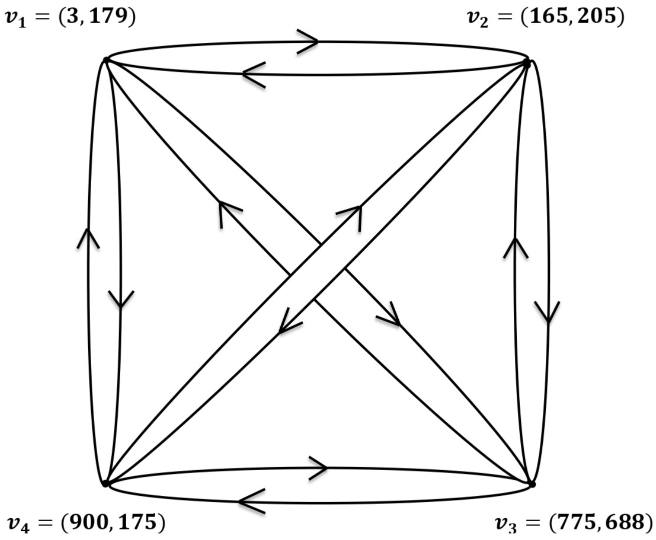

Figure 4.6: The CSESM sub-digraph.

- Each edge of the CSESM sub-digraph has a weight which is computed as a scalar multiplication between two connected vertices that is depended on the elliptic curve discrete logarithm problems (ECDLPs), using the Baby step and Giant step algorithm [20]. For instance, the weight w_{12} is equal to 128, because $(3, 179) = 128(165, 205)$. The weights computations of all arcs are given in Table 4.4.

Table 4.4: The weights of all arcs of CSESM sub-digraph.

The weight	Weight value
w_{12}	128
w_{13}	240
w_{14}	642
w_{21}	148
w_{23}	625
w_{24}	301
w_{31}	943
w_{32}	67
w_{34}	227
w_{41}	865
w_{42}	53
w_{43}	224

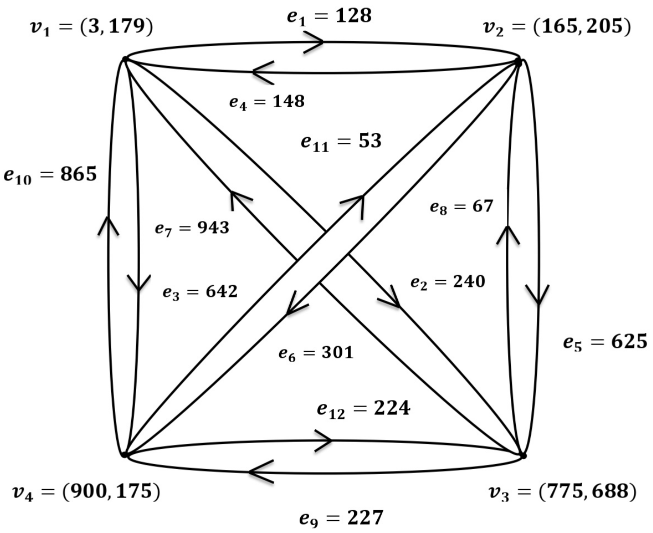

Figure 4.7: The weighted CSESM sub-digraph.

The adjacent matrix of MST of sub-digraph is

$$B_1 = \begin{pmatrix} 0 & 128 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 67 & 0 & 0 \\ 0 & 53 & 0 & 0 \end{pmatrix}.$$

The modified matrix B_2 of the matrix B_1 is done through the x-coordinates of each elliptic point in $M^{\bullet}(F_{941})$ at the diagonal. So

$$B_2 = \begin{pmatrix} 3 & 128 & 0 & 0 \\ 0 & 165 & 0 & 0 \\ 0 & 67 & 775 & 0 \\ 0 & 53 & 0 & 900 \end{pmatrix}.$$

Multiplying the matrices A^3 and B_2 to find a matrix B_3 as follows.

$$B_3 = A^3 \times B_2 \ (mod\ 997) \equiv \begin{pmatrix} 742 & 925 & 768 & 539 \\ 485 & 688 & 66 & 303 \\ 658 & 49 & 843 & 531 \\ 544 & 100 & 113 & 492 \end{pmatrix}.$$

$$C_2 = B_3 \times K = \begin{pmatrix} (548,170) & (263,357) & (101,566) & (488,250) \\ (347,309) & (162,582) & (692,550) & (561,789) \\ (13,746) & (361,161) & (561,152) & (633,821) \\ (777,841) & (40,790) & (794,207) & (20,442) \end{pmatrix}.$$

- He sends the ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She computes $(C_1^5)^{-1}$, where

$$C_1^5 = A^3 \ (mod\ 997) \equiv \begin{pmatrix} 912 & 569 & 266 & 15 \\ 494 & 519 & 350 & 223 \\ 884 & 890 & 845 & 529 \\ 846 & 74 & 992 & 293 \end{pmatrix},$$

and

$$(C_1^5)^{-1}\ (mod\ 997) \equiv \begin{pmatrix} 320 & 136 & 712 & 313 \\ 319 & 975 & 618 & 831 \\ 865 & 943 & 953 & 505 \\ 960 & 170 & 581 & 644 \end{pmatrix}.$$

- She computes a matrix B_3 from C_2 using the Baby step and Giant step algorithm [20]. Its elements are the ECDLPs.
- She computes a matrix

$$(C_1^5)^{-1} \times B_3\ (mod\ 997) \equiv \begin{pmatrix} 3 & 128 & 0 & 0 \\ 0 & 165 & 0 & 0 \\ 0 & 67 & 775 & 0 \\ 0 & 53 & 0 & 900 \end{pmatrix},$$

which is equal to a matrix B_2. The diagonal elements of the matrix B_2 are the x-coordinates of the elliptic points. For instance, the first element 3 in a matrix B_2 gives two possibilities of elliptic points, $(3, 179)$ and $(3, 762)$. Other possibilities of elliptic points are given in Table 4.5.

Table 4.5: The elliptic curve points that correspond to the diagonal elements of the matrix B_2.

Diagonal elements of the matrix B_2	Elliptic points
165	$(165, 205), (165, 736)$
775	$(775, 688), (775, 253)$
900	$(900, 175), (900, 766)$

Then, the original message is

$$M(F_{941}) = \{(3, 179), (3, 762), (165, 205), (165, 736), (775, 688), (775, 253), (900, 175), (900, 766)\},$$

4.5.3 Comparison of CSESM Cryptosystem with Other Cryptosystems

This work proposes an asymmetric cryptosystem, which used graph theory concepts, and compared it to previous works that used symmetric cryptosystems

([7],[8]). The proposed CSESM digraph cryptosystem increased security, as will be discussed in Section (4.7.2). Its security depends on the difficulty of computing the ECDLPs and the DLP over matrices while the security of other methods [27] depended on the ECDLPs only. So, the proposed method is more secure. Also, more than one point of the elliptic curve is encrypted simultaneously using the proposed algorithm, compared with other algorithms [27] that encrypt only one point. So, the CSESM digraph encryption algorithm saves time by encrypting many elliptic curve points compared to other algorithms.

4.6 The Graphic Representation of Edward Curve Cryptosystem

This section introduces a new graph, which is named the Edwards curve (EdC) graph. It has been defined as a bright point for designing a new version of an asymmetric cryptosystem. The vertices of the EdC graph are the Edwards curve points that are formed an Edwards curve group which has even order over a prime field. With a cyclic Edward curve group of even order n, then the EdC graph is formed with $(n-2)/2$ triangles and a line. This fact is considered as a new theoretical framework which is proved mathematically. On the EdC graph, the EdC subgraph is also defined and it is used to represent a message that is an Edwards curve subgroup. Using the EdC subgraph, fast computations on the proposed algorithm are obtained through the matrices representation. The EdC graphic cryptosystem is a more secure computation compared with the previous asymmetric cryptosystem. So, the EdC graphic asymmetric cryptosystem is considered a new sight for Edwards curve cryptographic usage.

4.6.1 The Graphic Representation of Edwards Curve over Prime Field

The Edward points over a prime field have been used to define a new graph. This graph is called an Edwards curve (EdC) graph which is defined as follows.

Definition 4.6.1 Let $E_d(F_p)$ be an Edward curve group. If $P +_d Q = (0, 1)$, where $(0, 1)$ is the identity element and $Q = -P$ such that P and Q are two different points in $E_d(F_p)$, then P and Q are adjacent or can be joined by an edge in the graph.

In addition, every Edwards point is adjoined with the identity element. Then it is possible to form the EdC graph $E_{dg}(F_p)$ that is corresponding to $E_d(F_p)$.

Example 4.6.1 Let E_3C be the Edward curve $x^2 + y^2 = 1 + 3x^2y^2$ over F_7. The points on E_3 form a set $E_3(F_7) = \{(0,1),(0,6),(1,0),(6,0)\}$. The Edwards curve graph $E_{3g}(F_7)$ of $E_3(F_7)$ is shown in Figure 4.8.

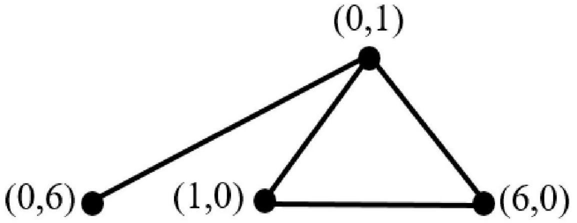

Figure 4.8: The EdC graph $E_{3g}(F_7)$ of $E_3(F_7)$.

Example 4.6.2 Let E_7C be the Edward curve $x^2 + y^2 = 1 + 7x^2y^2$ over F_{13}. The points on E_7C form a set

$$E_3(F_{13}) = \{(0,1),(0,12),(1,0),(12,0),(11,4),(2,9),(4,2),(9,2),(4,11),(9,11),(5,6),(8,6),(5,7),(8,7),(6,5),(7,5),(6,8),(7,8)\}.$$

The Edwards curve graph $E_{7g}(F_{13})$ of $E_7(F_{13})$ is shown in Figure 4.9.

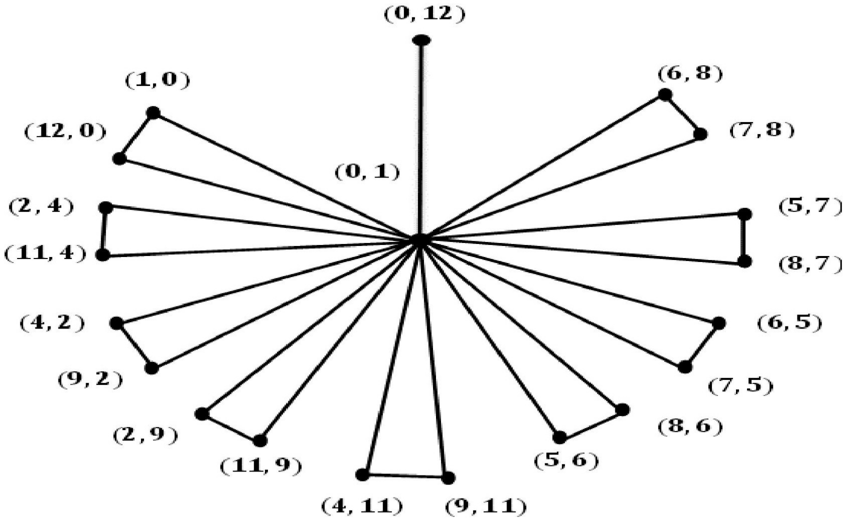

Figure 4.9: The EdC graph $E_{7g}(F_{13})$ of $E_7(F_{13})$.

Definition 4.6.2 Let $E_d(F_p)$ be an Edward curve group. The set $\langle P \rangle$ is a subgroup of $E_d(F_p)$ then the EdC graph for the subgroup $\langle P \rangle$ is known as the EdC subgraph of $E_{dg}(F_p)$.

Theorem 4.6.1 Let $E_d(F_p)$ be an Edward curve group.

1) If $P = (0, 1)$ then $\langle P \rangle = \{(0, 1)\}$.

2) If $P = (0, -1)$ then $\langle P \rangle = \{(0, 1), (0, -1)\}$.

3) If $P = (\pm 1, 0)$ then $\langle P \rangle = \{(0, 1), (0, -1), (1, 0), (-1, 0)\}$.

Proof.

1) Since $2P = (0, 1)$, by doubling law on the Edward curve points, so $(0, 1) \in \langle P \rangle$.

2) From computing $2P = (0, \frac{(-1)^2}{2 - (-1)^2}) \equiv (0, 1)\ (mod\ p)$.

3) Since $2P = (0, -1), 3P = P + 2P = (\pm 1, 0) + (0, -1) = (\mp 1, 0)$ and $4P = P + 3P = (\pm 1, 0) + (\mp 1, 0) = (0, 1)$.

Theorem 4.6.2 The order of an Edward curve group over a prime field F_p is even.
Proof. If $x = 0$ then $(0, 1)$ and $(0, -1)$ belong in $E_d(F_p)$. If Edward curve is a square $y^2 = a^2$ with $x \neq 0$, then (x, a) and $(x, -a)$ belong to $E_d(F_p)$, thus $\#E_d(F_p)$ is even. ■

Theorem 4.6.3 If $E_d(F_p) = \{P : nP = (0, 1)\}$ is a cyclic Edward curve group of order n, then the EdC graph $E_{dg}(F_p)$ of $E_d(F_p)$ is formed with $(n - 2)/2$ triangles and a line.
Proof. Let $E_d(F_p)$ be a cyclic Edward curve group of order n, then by Theorem (4.6.2), n is even, so there exist point $Q = (0, p-1)$ in $E_d(F_p)$ and $Q = 2P = (0, 1)$. Thus, Q joint in one edge only in EdC graph $E_{dg}(F_p)$. Hence, the $E_{dg}(F_p)$ has a line. If R is different point of Q in $E_d(F_p)$ then R has inverse point. Hence the $E_{dg}(F_p)$ has a triangle. The EdC graph $E_{dg}(F_p)$ associated with $E_d(F_p)$ that is given in Figure 4.10. So, it is easily to see that the EdC graph $E_{dg}(F_p)$ contains exactly $\frac{n-2}{2}$ triangles and one line. ■

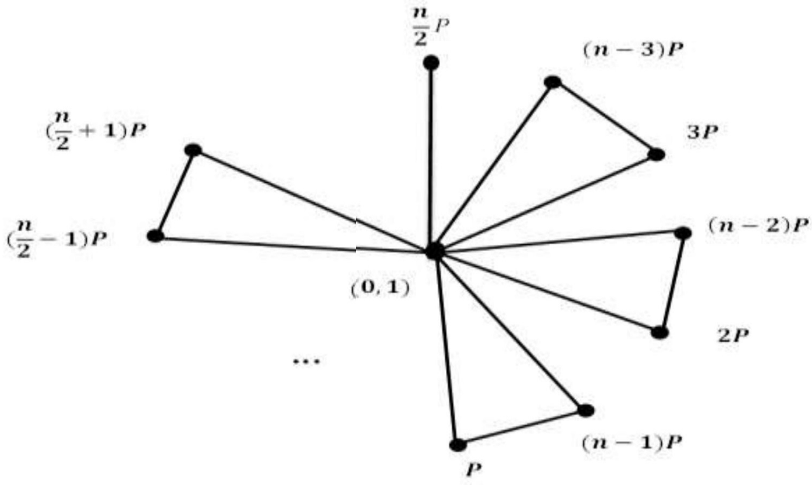

Figure 4.10: The EdC graph $E_{dg}(F_p)$ of $E_d(F_p)$.

4.6.2 The Edwards Curve Graphic Cryptosystem

An asymmetric cryptosystem has been proposed based on the Edward points and the EdC graph $E_{dg}(F_p)$. The public parameters are an Edward curve group $E_d(F_p)$ and the matrix $D \in GL_n(F_p)$. For obtaining the several numerical results to generate the keys, one can use Algorithm (2), to compute the ciphertext using Algorithm (29) and to recover the original plaintext can be done using Algorithm (30).

Algorithm 29 The EdC Graph Public Key Cryptosystem: Encryption Process

```
Input: The matrices $D \in GL_n(F_p)$ and a public key $A$.
Output: The ciphertext $(C_1, C_2)$, where $C_1$ and $C_2$ are two matrices.
1: Bob selects his plaintext $M = \langle P \rangle$ which is the Edwards curve subgroup of $E_d(F_p)$ such that if $(x, y) \in M$ then $(x, -y) \in M$.
2: He representation his plaintext by EdC subgraph $E_{dsg}(F_p)$.
3: He converted the $E_{dsg}(F_p)$ into a weighted graph. This converting is done through computing the weights for all edges, if $(x, y)$ and $(-x, y)$ two points between any edge then the weight of this edges is $w = min\{x, -x\}$, if $(x, y)$ and $(0, 1)$ two points between any edge then the weight of this edges is $w = 1$.
4: He converts a weighted graph into another graph after deleting one of the triangles that have the same weights and represents it by an adjacent matrix $B$.
5: He chooses his secret key $b \in \{2, 3, ..., p - 1\}$.
6: He computes the ciphertext through the computations of two square matrices $C_1 = D^b \ (mod\ p)$ and $C_2 = A^b \times B$.
7: Bob sends the ciphertext pair $(C_1, C_2)$ to Alice.
```

Algorithm 30 The EdC Graph Public Key Cryptosystem: Decryption process

```
Input: A prime $p$ and a ciphertext $(C_1, C_2)$
Output: The plaintext $M = \langle P \rangle$ which is the Edwards curve subgroup of $E_d(F_p)$.
1: Alice computes $(C_1^a)^{-1} \ (mod\ p)$.
2: She computes the multiplication matrix $(C_1^a)^{-1} \times C_2 \equiv B \ (mod\ p)$.
3: She represents matrix $B$ by weighted graph.
4: She gets the Edward's points by using the weights of all edges that give the original plaintext.
```

Example 4.6.3 Suppose E_d is Edward curve as defined in Example (4.6.2). The public matrix

$$D = \begin{pmatrix} 4 & 7 & 12 & 3 & 5 & 12 \\ 0 & 6 & 8 & 10 & 6 & 11 \\ 7 & 3 & 11 & 4 & 7 & 4 \\ 5 & 5 & 10 & 1 & 3 & 9 \\ 2 & 11 & 7 & 12 & 11 & 4 \\ 3 & 8 & 9 & 0 & 6 & 2 \end{pmatrix} \in GL(6, F_{13}).$$

Now, Alice performs the following steps:

1) She chooses her privet key $a = 8$.

2) She computes her public key A by

$$A = D^8 \ (mod\ 13) \equiv \begin{pmatrix} 11 & 5 & 5 & 6 & 3 & 9 \\ 1 & 11 & 9 & 7 & 5 & 2 \\ 0 & 7 & 1 & 3 & 7 & 4 \\ 11 & 7 & 5 & 11 & 2 & 2 \\ 1 & 4 & 9 & 6 & 6 & 12 \\ 2 & 5 & 0 & 11 & 12 & 1 \end{pmatrix}.$$

Alice's keys are $a = 8$ and a matrix A.

Now, Bob does the following steps:

1) He chooses his plaintext

$$M = \{(0, 1), (0, 12), (2, 4), (2, 9), (6, 8), (6, 5), (7, 5), (7, 8), (11, 9), (11, 4)\}.$$

2) He representation his plaintext by Edwards curve weighted subgraph $E_{7sg}(F_{13})$ as seen in Figure 4.11.

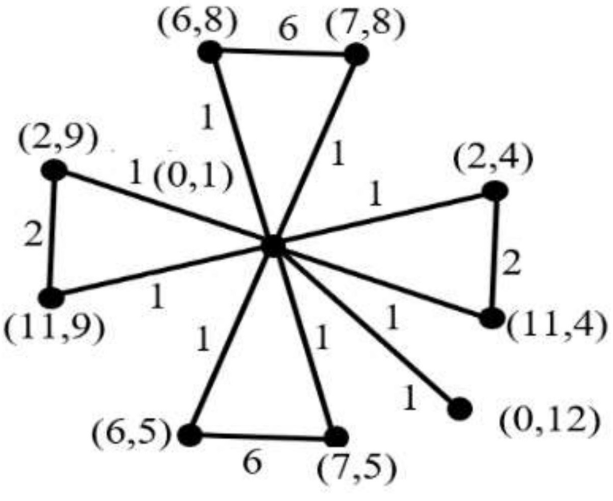

Figure 4.11: The Edwards curve weighted subgraph $E_{7sg}(F_{13})$.

3) He converts Edwards curve weighted subgraph $E_{7sg}(F_{13})$ into another Edwards curve graph $E^{\bullet}_{7sg}(F_{13})$ as seen in Figure 4.12.

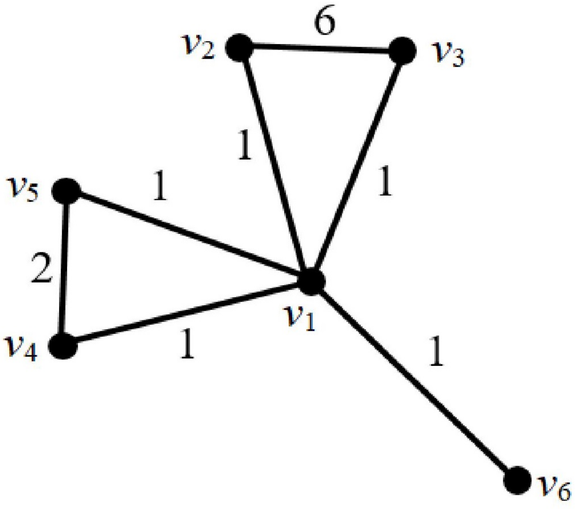

Figure 4.12: The Edwards curve weighted subgraph $E^{\bullet}_{7sg}(F_{13})$.

4) He represents $E^{\bullet}_{7sg}(F_{13})$ by adjacent matrix

$$B = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 6 & 0 & 0 & 0 \\ 1 & 6 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

5) He chooses a secret key $b = 12$.

6) He computes the ciphertext

$$C_1 = D^{12} \equiv \ (mod\ 13) \begin{pmatrix} 4 & 5 & 8 & 8 & 11 & 8 \\ 2 & 10 & 11 & 9 & 3 & 9 \\ 6 & 8 & 0 & 1 & 12 & 0 \\ 6 & 5 & 3 & 12 & 0 & 3 \\ 9 & 2 & 5 & 6 & 7 & 11 \\ 12 & 10 & 4 & 5 & 6 & 7 \end{pmatrix}.$$

and

$$C_2 = A^{12} \times B \ (mod\ 13) \equiv \begin{pmatrix} 12 & 11 & 8 & 0 & 10 & 4 \\ 3 & 12 & 5 & 3 & 0 & 11 \\ 2 & 9 & 0 & 9 & 5 & 10 \\ 8 & 9 & 3 & 7 & 4 & 0 \\ 11 & 4 & 7 & 5 & 5 & 3 \\ 7 & 10 & 3 & 7 & 8 & 3 \end{pmatrix}.$$

7) He sends the ciphertext pair (C_1, C_2) to Alice.

To decrypt and recover the original plaintext, Alice performs the following steps:

1) She computes

$$(C_1^8)^{-1} \ (mod\ 13) \equiv \begin{pmatrix} 11 & 8 & 10 & 5 & 6 & 8 \\ 3 & 3 & 12 & 11 & 2 & 8 \\ 12 & 6 & 10 & 1 & 5 & 6 \\ 8 & 5 & 1 & 10 & 1 & 10 \\ 11 & 10 & 8 & 0 & 11 & 2 \\ 8 & 6 & 2 & 0 & 6 & 11 \end{pmatrix},$$

where $C_1^8 = A^{12}$.

2) She computes a matrix B by

$$(C_1^a)^{-1} \times C_2 \ (mod\ 13) \equiv \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 6 & 0 & 0 & 0 \\ 1 & 6 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = B.$$

3) She represents matrix B by weighted graph in Figure 4.12.

4) She gets the original plaintext

$$M = \{(0,1), (0,12), (2,4), (2,9), (6,8), (6,5), (7,5), (7,8), (11,9), (11,4)\}$$

by using the weighted of edges graph.

4.7 The Security Considerations

In this section, discussed the security cases of the proposed cryptosystems over elliptic and Edwards curves.

4.7.1 The Security Considerations of Proposed KR-EC Public Key Cryptosystem

The security considerations of the proposed cryptosystem are determined through some main points. The first one is about the difficulty for solving ECDLP. The second one is the hardness to find the private key d such that $ed \equiv 1 \ (mod \ n)$, where e is a public key and n is a secretly. The element d here represents the inverse element of e modulo n, and since n is computed secretly, so it is more difficult to compute the value of d modulo n.

4.7.2 The Security Considerations of CESM graphic Cryptosystem

The security of the proposed CESM graphic cryptosystem has been determined by some points. First point is the difficulty to determine the correct MST graph among all other possible cases of the MST graphs that can be created from CESM subgraph. The hardness also to compute the elements of a matrix B_3 from C_2 which represented as ECDLPs secondly. As well as, a third point focuses on a good choice of the domain parameters of the proposed CESM graphic cryptosystem. Specifically, with a large prime p that creates an elliptic curve group with large prime order which gives the possibility to choose and generate the big size matrices that help to implement the CESM graphic asymmetric cryptosystem with more secure in compare with other asymmetric cryptosystem.

4.7.3 The Security Consideration of Edwards Curve Graph Cryptosystem

The graph theory concepts here considered as an essential tool to increase security. The security of the proposed EdC sub-graphic cryptosystem depends on the random choice of the EdC sub-graph that corresponds to the EdC subgroup of an EdC group

$E_d(F_p)$. With a large prime number p, the $E_d(F_p)$ has a large even order n. So, the EdC graph has a large number of the vertices. Thus, it is possible to form a large number of the EdC subgraphs. Choosing one of these subgraph randomly to represent a plaintext gives more secure to communicate using the proposed EdC sub-graphic cryptosystem. Eve needs to compute many cases to reach the correct choice of the EdC sub-graph. In other words, Eve should compute roughly the number $2^n + 2^e - 1$ at least, where n is a number of vertices and e is the number of edges. Thus, the EdC sub-graphic cryptosystem is more secure and suitable for Edwards cryptographic communication schemes.

4.8 Summary

In Chapter 4, new public key cryptosystems have been designed based on graph theory and elliptic and Edwards curves. The main points for designing new versions of asymmetric encryption schemes have been defined as new graphs. The scalar multiplication operation on the elliptic and the Edwards curves defined over a prime field are used to construct these graphs.

Chapter 5

More Implemented Results about Suggested Cryptosystems

In this chapter, some computations of the suggested cryptosystems have been done. The programming languages MATLAB and Python were used to calculate the results.

5.1 The Results of Matrix Power ElGamal Public Key Cryptosystem

Some simple computations of the MP-ElGamal public key cryptosystem have been done. The experimental samples with different values of a prime p are chosen. The computational results to generate the keys, encryption and decryption processes are shown by Tables (5.1), (5.2) and (5.3) respectively.

Table 5.1: The experimental results of MP-ElGamal public key cryptosystem: key generation process.

p	Shear key D	A private key a	A public key $A \equiv D^a\ (mod\ p)$
89	$\begin{pmatrix} 81 & 77 & 19 & 25 \\ 14 & 6 & 87 & 11 \\ 3 & 61 & 18 & 8 \\ 22 & 9 & 15 & 37 \end{pmatrix}$	61	$\begin{pmatrix} 44 & 0 & 87 & 6 \\ 70 & 86 & 13 & 87 \\ 1 & 54 & 36 & 57 \\ 45 & 12 & 53 & 17 \end{pmatrix}$
127	$\begin{pmatrix} 113 & 13 & 17 & 55 \\ 3 & 75 & 100 & 60 \\ 55 & 85 & 32 & 19 \\ 10 & 90 & 29 & 120 \end{pmatrix}$	47	$\begin{pmatrix} 72 & 43 & 41 & 72 \\ 91 & 30 & 13 & 123 \\ 10 & 19 & 95 & 80 \\ 48 & 47 & 101 & 3 \end{pmatrix}$
151	$\begin{pmatrix} 120 & 99 & 104 & 25 \\ 17 & 87 & 77 & 61 \\ 133 & 14 & 6 & 141 \\ 12 & 37 & 51 & 3 \end{pmatrix}$	87	$\begin{pmatrix} 126 & 144 & 81 & 12 \\ 42 & 122 & 49 & 50 \\ 41 & 86 & 59 & 118 \\ 89 & 117 & 135 & 14 \end{pmatrix}$
167	$\begin{pmatrix} 3 & 77 & 10 & 14 & 50 \\ 17 & 150 & 33 & 1 & 19 \\ 10 & 15 & 9 & 125 & 40 \\ 100 & 7 & 60 & 4 & 23 \\ 17 & 5 & 85 & 15 & 71 \end{pmatrix}$	125	$\begin{pmatrix} 74 & 80 & 28 & 75 & 22 \\ 15 & 58 & 88 & 109 & 56 \\ 77 & 97 & 43 & 32 & 155 \\ 146 & 140 & 108 & 49 & 163 \\ 161 & 52 & 24 & 26 & 25 \end{pmatrix}$
179	$\begin{pmatrix} 175 & 10 & 2 & 15 & 75 \\ 1 & 145 & 31 & 99 & 17 \\ 120 & 13 & 100 & 77 & 151 \\ 14 & 55 & 13 & 167 & 17 \\ 103 & 170 & 37 & 105 & 23 \end{pmatrix}$	103	$\begin{pmatrix} 103 & 34 & 73 & 15 & 51 \\ 27 & 119 & 0 & 150 & 130 \\ 64 & 35 & 137 & 24 & 92 \\ 104 & 82 & 58 & 25 & 8 \\ 90 & 68 & 47 & 39 & 17 \end{pmatrix}$

Table 5.2: The computational results of MP-ElGamal public key cryptosystem: encryption process.

b	$C_1 \equiv D^b\ (mod\ p)$	A plaintext M	$C_2 \equiv A^b \times M\ (mod\ p)$
75	$\begin{pmatrix} 17 & 13 & 3 & 68 \\ 70 & 62 & 38 & 81 \\ 55 & 37 & 62 & 44 \\ 13 & 29 & 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 85 & 11 & 64 & 72 \\ 10 & 37 & 45 & 19 \\ 25 & 40 & 54 & 59 \\ 68 & 12 & 2 & 29 \end{pmatrix}$	$\begin{pmatrix} 68 & 36 & 81 & 33 \\ 58 & 75 & 62 & 13 \\ 72 & 87 & 18 & 39 \\ 26 & 69 & 36 & 22 \end{pmatrix}$
31	$\begin{pmatrix} 57 & 112 & 76 & 95 \\ 1 & 70 & 48 & 43 \\ 6 & 117 & 68 & 51 \\ 125 & 57 & 37 & 90 \end{pmatrix}$	$\begin{pmatrix} 120 & 97 & 119 & 12 \\ 13 & 77 & 3 & 116 \\ 87 & 123 & 17 & 5 \\ 56 & 45 & 61 & 103 \end{pmatrix}$	$\begin{pmatrix} 74 & 29 & 11 & 91 \\ 47 & 88 & 91 & 122 \\ 105 & 10 & 125 & 97 \\ 82 & 76 & 106 & 67 \end{pmatrix}$
141	$\begin{pmatrix} 57 & 139 & 48 & 74 \\ 133 & 80 & 115 & 129 \\ 114 & 144 & 13 & 55 \\ 67 & 48 & 112 & 46 \end{pmatrix}$	$\begin{pmatrix} 149 & 13 & 17 & 113 & 87 \\ 91 & 18 & 132 & 22 & 75 \\ 15 & 125 & 102 & 3 & 53 \\ 83 & 72 & 149 & 69 & 140 \end{pmatrix}$	$\begin{pmatrix} 51 & 104 & 46 & 78 & 71 \\ 79 & 27 & 148 & 81 & 2 \\ 122 & 137 & 6 & 130 & 34 \\ 42 & 79 & 100 & 55 & 27 \end{pmatrix}$
64	$\begin{pmatrix} 83 & 3 & 98 & 6 & 139 \\ 116 & 110 & 72 & 70 & 117 \\ 23 & 33 & 147 & 86 & 140 \\ 8 & 99 & 117 & 34 & 160 \\ 28 & 144 & 34 & 42 & 88 \end{pmatrix}$	$\begin{pmatrix} 100 & 120 & 56 & 33 & 10 \\ 161 & 114 & 77 & 51 & 63 \\ 12 & 107 & 3 & 11 & 17 \\ 55 & 87 & 27 & 136 & 86 \\ 17 & 93 & 44 & 165 & 2 \end{pmatrix}$	$\begin{pmatrix} 51 & 96 & 56 & 2 & 121 \\ 114 & 152 & 125 & 116 & 37 \\ 87 & 61 & 20 & 87 & 29 \\ 42 & 15 & 157 & 67 & 100 \\ 107 & 28 & 166 & 63 & 74 \end{pmatrix}$
127	$\begin{pmatrix} 74 & 73 & 100 & 177 & 153 \\ 115 & 126 & 52 & 170 & 152 \\ 142 & 126 & 68 & 17 & 47 \\ 0 & 168 & 4 & 135 & 177 \\ 101 & 70 & 129 & 149 & 144 \end{pmatrix}$	$\begin{pmatrix} 17 & 63 & 151 & 122 & 8 \\ 160 & 0 & 67 & 53 & 81 \\ 145 & 17 & 23 & 93 & 1 \\ 13 & 25 & 5 & 32 & 12 \\ 55 & 67 & 43 & 104 & 131 \end{pmatrix}$	$\begin{pmatrix} 20 & 126 & 132 & 70 & 84 \\ 112 & 145 & 117 & 54 & 56 \\ 59 & 112 & 141 & 27 & 90 \\ 147 & 20 & 165 & 117 & 37 \\ 34 & 28 & 30 & 30 & 66 \end{pmatrix}$

Table 5.3: The computational results of MP-ElGamal public key cryptosystem: decryption process..

$C_1^a\ (mod\ p)$	$(C_1^a)^{-1}\ (mod\ p)$	$(C_1^a)^{-1} \times C_2\ (mod\ p) \equiv M$
$\begin{pmatrix} 76 & 45 & 60 & 30 \\ 31 & 62 & 11 & 17 \\ 30 & 13 & 81 & 85 \\ 58 & 59 & 32 & 20 \end{pmatrix}$	$\begin{pmatrix} 57 & 20 & 47 & 76 \\ 32 & 53 & 26 & 59 \\ 20 & 51 & 78 & 9 \\ 2 & 20 & 36 & 54 \end{pmatrix}$	$\begin{pmatrix} 85 & 11 & 64 & 72 \\ 10 & 37 & 45 & 19 \\ 25 & 40 & 54 & 59 \\ 68 & 12 & 2 & 29 \end{pmatrix}$
$\begin{pmatrix} 85 & 65 & 109 & 111 \\ 18 & 79 & 47 & 41 \\ 8 & 30 & 63 & 84 \\ 126 & 11 & 95 & 112 \end{pmatrix}$	$\begin{pmatrix} 34 & 108 & 45 & 113 \\ 30 & 25 & 81 & 92 \\ 119 & 65 & 11 & 70 \\ 113 & 118 & 16 & 46 \end{pmatrix}$	$\begin{pmatrix} 120 & 97 & 119 & 12 \\ 13 & 77 & 3 & 116 \\ 87 & 123 & 17 & 5 \\ 56 & 45 & 61 & 103 \end{pmatrix}$
$\begin{pmatrix} 12 & 88 & 42 & 26 \\ 74 & 97 & 0 & 11 \\ 72 & 132 & 134 & 149 \\ 126 & 99 & 72 & 53 \end{pmatrix}$	$\begin{pmatrix} 120 & 104 & 139 & 53 \\ 54 & 80 & 75 & 68 \\ 116 & 90 & 41 & 114 \\ 103 & 105 & 92 & 142 \end{pmatrix}$	$\begin{pmatrix} 149 & 13 & 17 & 113 & 87 \\ 91 & 18 & 132 & 22 & 75 \\ 15 & 125 & 102 & 3 & 53 \\ 83 & 72 & 149 & 69 & 140 \end{pmatrix}$
$\begin{pmatrix} 23 & 154 & 155 & 75 & 71 \\ 71 & 94 & 16 & 54 & 6 \\ 63 & 56 & 126 & 137 & 75 \\ 20 & 30 & 114 & 19 & 58 \\ 150 & 94 & 30 & 0 & 79 \end{pmatrix}$	$\begin{pmatrix} 19 & 138 & 108 & 134 & 40 \\ 92 & 128 & 46 & 84 & 113 \\ 125 & 91 & 14 & 149 & 128 \\ 58 & 123 & 137 & 7 & 57 \\ 67 & 50 & 71 & 145 & 75 \end{pmatrix}$	$\begin{pmatrix} 100 & 120 & 56 & 33 & 10 \\ 161 & 114 & 77 & 51 & 63 \\ 12 & 107 & 3 & 11 & 17 \\ 55 & 87 & 27 & 136 & 86 \\ 17 & 93 & 44 & 165 & 2 \end{pmatrix}$
$\begin{pmatrix} 126 & 22 & 48 & 101 & 105 \\ 103 & 76 & 53 & 24 & 170 \\ 97 & 125 & 145 & 99 & 74 \\ 138 & 59 & 5 & 85 & 35 \\ 22 & 74 & 134 & 57 & 92 \end{pmatrix}$	$\begin{pmatrix} 75 & 37 & 72 & 72 & 109 \\ 16 & 117 & 18 & 63 & 17 \\ 55 & 141 & 25 & 97 & 145 \\ 146 & 92 & 56 & 68 & 110 \\ 36 & 160 & 119 & 60 & 72 \end{pmatrix}$	$\begin{pmatrix} 17 & 63 & 151 & 122 & 8 \\ 160 & 0 & 67 & 53 & 81 \\ 145 & 17 & 23 & 93 & 1 \\ 13 & 25 & 5 & 32 & 12 \\ 55 & 67 & 43 & 104 & 131 \end{pmatrix}$

5.2 The Results of UCG based Public Key Cryptosystem

The computation on the UCG public key cryptosystem have been done with several numerical results. Some experimental samples with different values of a prime p are chosen. The computational results to generate the keys, encryption and decryption processes are given by Tables (5.4),(5.5) and (5.6) respectively.

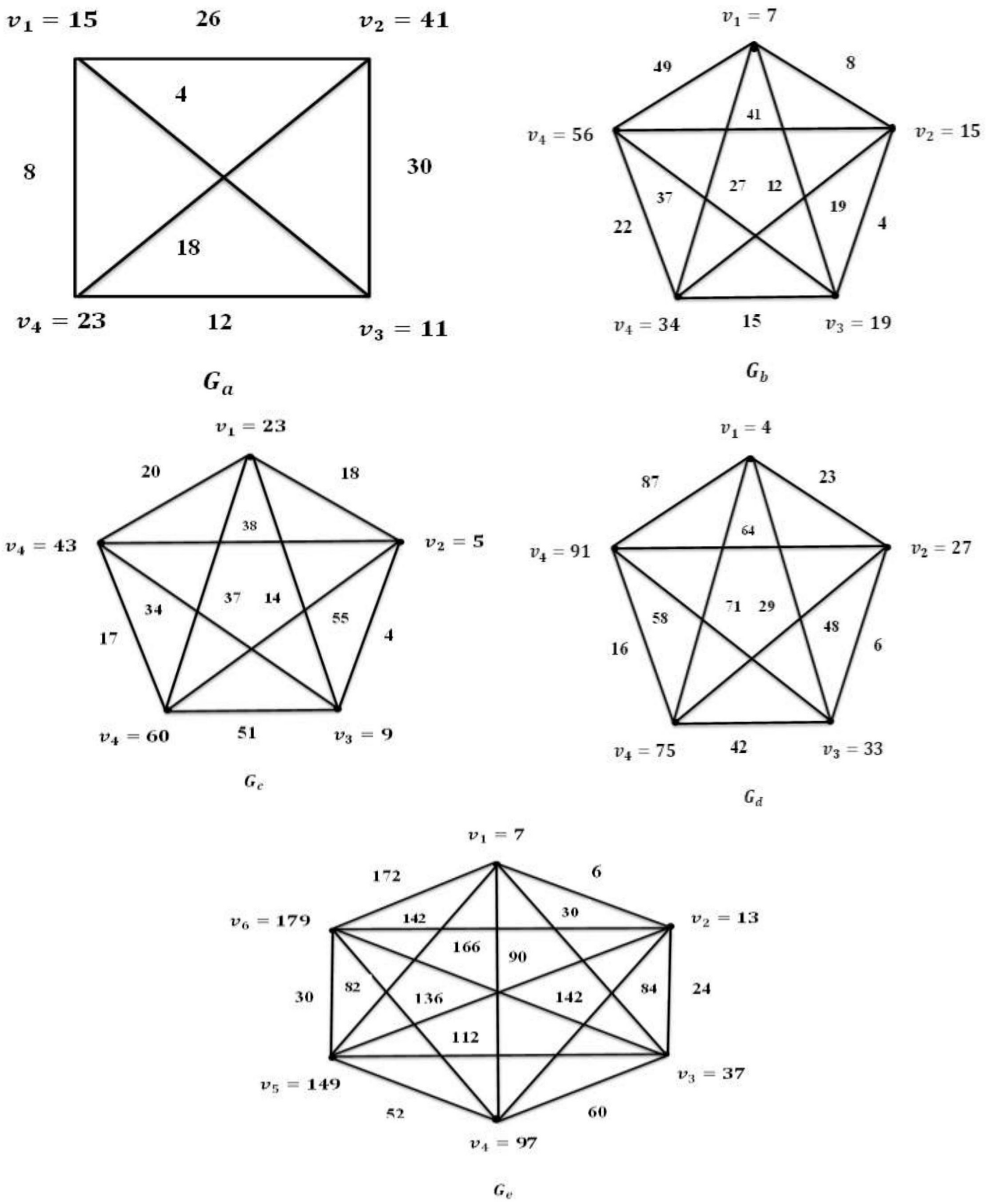

Figure 5.1: The UCGs G_a, G_b, G_c, G_d, G_e that correspond to Table (5.5).

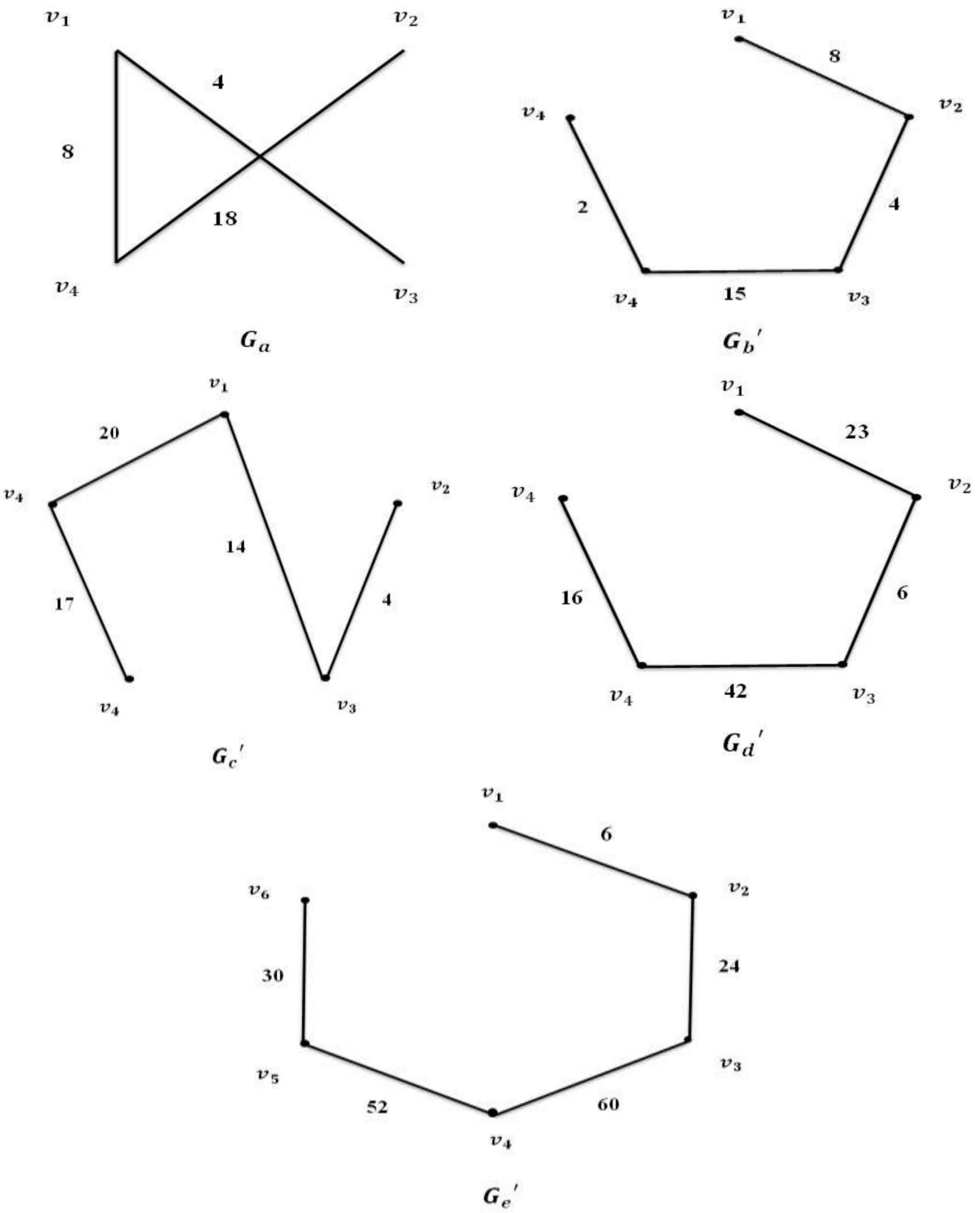

Figure 5.2: The MSTs $G'_a, G'_b, G'_c, G'_d, G'_e$ that correspond to Table (5.5).

Table 5.4: The experimental results of UCG public key cryptosystem: key generation process.

p	Shear key D	A private key a	A public key $A \equiv D^a \ (mod\ p)$
47	$\begin{pmatrix} 12 & 15 & 21 & 7 \\ 2 & 45 & 3 & 40 \\ 14 & 25 & 18 & 22 \\ 33 & 7 & 46 & 31 \end{pmatrix}$	13	$\begin{pmatrix} 45 & 3 & 0 & 46 \\ 28 & 1 & 34 & 8 \\ 19 & 35 & 24 & 40 \\ 0 & 1 & 28 & 0 \end{pmatrix}$
59	$\begin{pmatrix} 1 & 0 & 42 & 12 & 5 \\ 3 & 10 & 1 & 26 & 9 \\ 41 & 0 & 33 & 1 & 58 \\ 3 & 7 & 5 & 21 & 1 \\ 1 & 4 & 0 & 1 & 30 \end{pmatrix}$	7	$\begin{pmatrix} 5 & 27 & 0 & 16 & 47 \\ 57 & 6 & 46 & 40 & 19 \\ 29 & 49 & 25 & 6 & 35 \\ 56 & 57 & 26 & 58 & 55 \\ 23 & 44 & 27 & 48 & 45 \end{pmatrix}$
61	$\begin{pmatrix} 1 & 0 & 55 & 12 & 5 \\ 3 & 10 & 1 & 26 & 9 \\ 60 & 0 & 1 & 1 & 59 \\ 33 & 7 & 5 & 21 & 1 \\ 1 & 4 & 0 & 1 & 30 \end{pmatrix}$	4	$\begin{pmatrix} 55 & 53 & 47 & 17 & 36 \\ 49 & 35 & 6 & 28 & 48 \\ 9 & 20 & 18 & 23 & 39 \\ 5 & 9 & 31 & 22 & 38 \\ 43 & 4 & 43 & 40 & 31 \end{pmatrix}$
97	$\begin{pmatrix} 88 & 91 & 0 & 1 & 2 \\ 1 & 0 & 1 & 77 & 9 \\ 60 & 2 & 1 & 93 & 3 \\ 12 & 23 & 5 & 21 & 1 \\ 1 & 5 & 3 & 55 & 8 \end{pmatrix}$	6	$\begin{pmatrix} 54 & 29 & 41 & 0 & 77 \\ 35 & 42 & 45 & 18 & 78 \\ 70 & 33 & 81 & 38 & 87 \\ 4 & 8 & 50 & 55 & 29 \\ 28 & 33 & 57 & 53 & 53 \end{pmatrix}$
191	$\begin{pmatrix} 185 & 123 & 2 & 99 & 14 & 3 \\ 77 & 15 & 1 & 190 & 5 & 45 \\ 3 & 18 & 13 & 85 & 91 & 10 \\ 10 & 6 & 7 & 14 & 11 & 2 \\ 87 & 25 & 12 & 85 & 152 & 52 \\ 45 & 32 & 9 & 135 & 23 & 13 \end{pmatrix}$	5	$\begin{pmatrix} 163 & 147 & 172 & 75 & 68 & 66 \\ 67 & 31 & 181 & 148 & 56 & 123 \\ 76 & 57 & 42 & 168 & 184 & 97 \\ 73 & 114 & 68 & 67 & 55 & 133 \\ 164 & 58 & 35 & 164 & 165 & 8 \\ 39 & 48 & 185 & 77 & 170 & 20 \end{pmatrix}$

Table 5.5: The experimental results of UCG public key cryptosystem : encryption process.

A plaintext m	UCG	MST	M	M'	b	$C_1 \equiv D^b \ (mod\ p)$	$C_2 \equiv A^b \times M' \ (mod\ p)$
$\{15, 41, 11, 23\}$	G_a	G'_a	$\begin{pmatrix} 0 & 0 & 4 & 8 \\ 0 & 0 & 18 & 0 \\ 4 & 18 & 0 & 0 \\ 8 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 15 & 0 & 4 & 8 \\ 0 & 41 & 18 & 0 \\ 4 & 18 & 11 & 0 \\ 8 & 0 & 0 & 23 \end{pmatrix}$	11	$\begin{pmatrix} 27 & 19 & 25 & 26 \\ 20 & 23 & 23 & 6 \\ 44 & 35 & 33 & 1 \\ 40 & 17 & 41 & 16 \end{pmatrix}$	$\begin{pmatrix} 40 & 36 & 4 & 12 \\ 11 & 40 & 11 & 30 \\ 25 & 22 & 35 & 44 \\ 11 & 16 & 31 & 20 \end{pmatrix}$
$\{7, 15, 19, 34, 56\}$	G_b	G'_b	$\begin{pmatrix} 0 & 8 & 0 & 0 & 0 \\ 8 & 0 & 4 & 0 & 0 \\ 0 & 4 & 0 & 15 & 0 \\ 0 & 0 & 15 & 0 & 22 \\ 0 & 0 & 0 & 22 & 0 \end{pmatrix}$	$\begin{pmatrix} 7 & 8 & 0 & 0 & 0 \\ 8 & 15 & 4 & 0 & 0 \\ 0 & 4 & 19 & 15 & 0 \\ 0 & 0 & 15 & 34 & 22 \\ 0 & 0 & 0 & 22 & 56 \end{pmatrix}$	4	$\begin{pmatrix} 54 & 22 & 32 & 26 & 16 \\ 1 & 33 & 18 & 0 & 48 \\ 39 & 18 & 22 & 39 & 45 \\ 47 & 2 & 43 & 47 & 15 \\ 31 & 39 & 35 & 47 & 19 \end{pmatrix}$	$\begin{pmatrix} 56 & 40 & 24 & 57 & 33 \\ 8 & 33 & 10 & 47 & 50 \\ 23 & 47 & 11 & 50 & 49 \\ 5 & 3 & 40 & 50 & 40 \\ 25 & 34 & 44 & 24 & 25 \end{pmatrix}$
$\{23, 5, 9, 60, 43\}$	G_c	G'_c	$\begin{pmatrix} 0 & 0 & 14 & 0 & 20 \\ 0 & 0 & 4 & 0 & 0 \\ 14 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 17 \\ 20 & 0 & 0 & 17 & 0 \end{pmatrix}$	$\begin{pmatrix} 23 & 0 & 14 & 0 & 20 \\ 0 & 5 & 4 & 0 & 0 \\ 14 & 4 & 9 & 0 & 0 \\ 0 & 0 & 0 & 60 & 17 \\ 20 & 0 & 0 & 17 & 43 \end{pmatrix}$	7	$\begin{pmatrix} 31 & 17 & 31 & 9 & 10 \\ 30 & 59 & 42 & 4 & 27 \\ 29 & 9 & 2 & 13 & 18 \\ 14 & 18 & 54 & 25 & 9 \\ 30 & 31 & 31 & 4 & 48 \end{pmatrix}$	$\begin{pmatrix} 19 & 51 & 1 & 34 & 37 \\ 44 & 25 & 9 & 1 & 10 \\ 22 & 3 & 13 & 22 & 53 \\ 29 & 9 & 17 & 10 & 16 \\ 42 & 57 & 16 & 53 & 12 \end{pmatrix}$
$\{4, 27, 33, 75, 91\}$	G_d	G'_d	$\begin{pmatrix} 0 & 23 & 0 & 0 & 0 \\ 23 & 0 & 6 & 0 & 0 \\ 0 & 6 & 0 & 42 & 0 \\ 0 & 0 & 42 & 0 & 16 \\ 0 & 0 & 0 & 16 & 0 \end{pmatrix}$	$\begin{pmatrix} 4 & 23 & 0 & 0 & 0 \\ 23 & 27 & 6 & 0 & 0 \\ 0 & 6 & 33 & 42 & 0 \\ 0 & 0 & 42 & 75 & 16 \\ 0 & 0 & 0 & 16 & 91 \end{pmatrix}$	5	$\begin{pmatrix} 51 & 68 & 85 & 35 & 66 \\ 58 & 10 & 4 & 60 & 72 \\ 76 & 92 & 53 & 40 & 9 \\ 55 & 92 & 83 & 59 & 54 \\ 15 & 62 & 34 & 17 & 88 \end{pmatrix}$	$\begin{pmatrix} 62 & 76 & 39 & 91 & 84 \\ 38 & 31 & 43 & 73 & 24 \\ 27 & 63 & 11 & 42 & 78 \\ 86 & 77 & 4 & 26 & 73 \\ 55 & 59 & 7 & 88 & 38 \end{pmatrix}$
$\{7, 13, 37, 97, 149, 179\}$	G_e	G'_e	$\begin{pmatrix} 0 & 6 & 0 & 0 & 0 & 0 \\ 6 & 0 & 24 & 0 & 0 & 0 \\ 0 & 24 & 0 & 60 & 0 & 0 \\ 0 & 0 & 60 & 0 & 52 & 0 \\ 0 & 0 & 0 & 52 & 0 & 30 \\ 0 & 0 & 0 & 0 & 30 & 0 \end{pmatrix}$	$\begin{pmatrix} 7 & 6 & 0 & 0 & 0 & 0 \\ 6 & 13 & 24 & 0 & 0 & 0 \\ 0 & 24 & 37 & 60 & 0 & 0 \\ 0 & 0 & 60 & 97 & 52 & 0 \\ 0 & 0 & 0 & 52 & 149 & 30 \\ 0 & 0 & 0 & 0 & 30 & 179 \end{pmatrix}$	3	$\begin{pmatrix} 109 & 54 & 32 & 134 & 22 & 185 \\ 165 & 7 & 57 & 106 & 143 & 83 \\ 101 & 15 & 22 & 156 & 78 & 70 \\ 112 & 23 & 181 & 150 & 72 & 18 \\ 0 & 11 & 42 & 182 & 110 & 32 \\ 161 & 117 & 50 & 67 & 99 & 3 \end{pmatrix}$	$\begin{pmatrix} 26 & 108 & 115 & 45 & 132 & 30 \\ 159 & 132 & 49 & 125 & 36 & 129 \\ 142 & 69 & 48 & 94 & 182 & 55 \\ 125 & 3 & 176 & 13 & 162 & 81 \\ 120 & 109 & 160 & 53 & 113 & 126 \\ 8 & 17 & 96 & 123 & 134 & 168 \end{pmatrix}$

Table 5.6: The experimental results of UCG public key cryptosystem : decryption process.

$(C_1^a)^{-1}$	$(C_1^a)^{-1} \times C_2\ (mod\ p) \equiv M'$	A plaintext m
$\begin{pmatrix} 27 & 5 & 14 & 4 \\ 25 & 14 & 16 & 7 \\ 20 & 11 & 33 & 28 \\ 45 & 16 & 20 & 4 \end{pmatrix}$	$\begin{pmatrix} 15 & 0 & 4 & 8 \\ 0 & 41 & 18 & 0 \\ 4 & 18 & 11 & 0 \\ 8 & 0 & 0 & 23 \end{pmatrix}$	$\{15, 41, 11, 23\}$
$\begin{pmatrix} 3 & 9 & 41 & 33 & 3 \\ 51 & 10 & 5 & 26 & 43 \\ 57 & 4 & 36 & 43 & 37 \\ 53 & 23 & 44 & 19 & 9 \\ 51 & 35 & 56 & 24 & 33 \end{pmatrix}$	$\begin{pmatrix} 7 & 8 & 0 & 0 & 0 \\ 8 & 15 & 4 & 0 & 0 \\ 0 & 4 & 19 & 15 & 0 \\ 0 & 0 & 15 & 34 & 22 \\ 0 & 0 & 0 & 22 & 56 \end{pmatrix}$	$\{7, 15, 19, 34, 56\}$
$\begin{pmatrix} 0 & 14 & 54 & 7 & 37 \\ 9 & 25 & 51 & 49 & 37 \\ 4 & 44 & 52 & 15 & 47 \\ 34 & 46 & 59 & 43 & 8 \\ 11 & 7 & 16 & 46 & 25 \end{pmatrix}$	$\begin{pmatrix} 23 & 0 & 14 & 0 & 20 \\ 0 & 5 & 4 & 0 & 0 \\ 14 & 4 & 9 & 0 & 0 \\ 0 & 0 & 0 & 60 & 17 \\ 20 & 0 & 0 & 17 & 43 \end{pmatrix}$	$\{23, 5, 9, 60, 43\}$
$\begin{pmatrix} 43 & 62 & 81 & 52 & 54 \\ 59 & 21 & 12 & 67 & 68 \\ 15 & 10 & 31 & 45 & 7 \\ 62 & 16 & 50 & 57 & 47 \\ 36 & 49 & 7628 & 42 & \end{pmatrix}$	$\begin{pmatrix} 4 & 23 & 0 & 0 & 0 \\ 23 & 27 & 6 & 0 & 0 \\ 0 & 6 & 33 & 42 & 0 \\ 0 & 0 & 42 & 75 & 16 \\ 0 & 0 & 0 & 16 & 91 \end{pmatrix}$	$\{4, 27, 33, 75, 91\}$
$\begin{pmatrix} 189 & 119 & 76 & 11 & 60 & 68 \\ 167 & 123 & 155 & 15 & 124 & 142 \\ 166 & 92 & 143 & 49 & 38 & 85 \\ 101 & 5 & 118 & 124 & 6 & 106 \\ 110 & 35 & 67 & 57 & 44 & 27 \\ 89 & 14 & 137 & 29 & 3 & 60 \end{pmatrix}$	$\begin{pmatrix} 7 & 6 & 0 & 0 & 0 & 0 \\ 6 & 13 & 24 & 0 & 0 & 0 \\ 0 & 24 & 37 & 60 & 0 & 0 \\ 0 & 0 & 60 & 97 & 52 & 0 \\ 0 & 0 & 0 & 52 & 149 & 30 \\ 0 & 0 & 0 & 0 & 30 & 179 \end{pmatrix}$	$\{7, 13, 37, 97, 149, 179\}$

5.3 Case Study of n-Dimension Public Key Cryptosystem

With a prime $p = 173$, a matrix D of size 4×4

$$D = \begin{pmatrix} 150 & 43 & 89 & 91 \\ 91 & 13 & 100 & 170 \\ 17 & 107 & 75 & 165 \\ 9 & 22 & 143 & 18 \end{pmatrix},$$

where $D \in GL_4(F_{173})$ and $v = (77, 115, 145, 163)$, the procedure n-dimension public key cryptosystem is discussed as follows.

Keys generation process.

Alice performs the following:

- She chooses a secret key $a = 131$ and she computes her public key

$$A = D^{131}\ (mod\ 173) \equiv \begin{pmatrix} 11 & 165 & 154 & 98 \\ 11 & 155 & 132 & 143 \\ 33 & 5 & 16 & 35 \\ 12 & 7 & 86 & 153 \end{pmatrix},$$

Alice's keys are $a = 131$ and $A = D^{131}$.

Encryption process.

Bob does the following steps:

- He chooses his private key $b = 167$ and he computes

$$C_1 = D^{167}\ (mod\ 173) \equiv \begin{pmatrix} 101 & 7 & 97 & 167 \\ 139 & 137 & 97 & 43 \\ 42 & 36 & 132 & 60 \\ 144 & 143 & 76 & 117 \end{pmatrix},$$

and

$$A^{167}\ (mod\ 173) \equiv \begin{pmatrix} 74 & 51 & 126 & 53 \\ 87 & 95 & 80 & 159 \\ 89 & 86 & 71 & 89 \\ 79 & 70 & 147 & 18 \end{pmatrix}.$$

- He computes the transformation T on 4-dimensional over matrix A^{167}

$$T(a, b, c, d) = (74a + 87b + 89c + 79d, 51a + 95b + 86c + 70d,$$
$$126a + 80b + 70c + 147d, 53a + 159b + 89c + 18d),$$

- He computes $v^\star = T(v) \equiv (138, 153, 47, 145)\ (mod\ 173)$,

- He chooses his plaintext $m = (6, 87, 125, 149)$ and converts m into $m^\star = m +_{173} v^\star \equiv (144, 67, 172, 121)$.

- He computes $C_2 = T(m^\star) \equiv (5, 122, 46, 133)\ (mod\ 173)$,

- He sends his ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She computes

$$(C_1^{167})^{-1}\ (mod\ 173) \equiv \begin{pmatrix} 10 & 32 & 113 & 23 \\ 66 & 104 & 92 & 66 \\ 97 & 140 & 101 & 112 \\ 51 & 61 & 13 & 131 \end{pmatrix},$$

  where $C_1^{167} = A^{131}$.

- Compute the transformation T on 4-dimensional over matrices C_1^{167} and $(C_1^{167})^{-1}$,

$$\begin{aligned} T(a, b, c, d) = (&74a + 87b + 89c + 79d, 51a + 95b + 86c + 70d, \\ &126a + 80b + 70c + 147d, 53a + 159b + 89c + 18d), \end{aligned}$$

$$\begin{aligned} T^\star(a, b, c, d) = (&10a + 66b + 97c + 51d, 32a + 104b + 140c + 61d, \\ &113a + 92b + 101c + 13d, 23a + 66b + 112c + 131d), \end{aligned}$$

- Using public key v to compute $v^\star = T(v) \equiv (138, 153, 47, 145)\ (mod\ 173)$ and compute $T^\star(C_2) \equiv m^\star\ (mod\ 173)$.

- Compute $m^\star -_{173} v^\star \equiv (24, 87, 113) = m$.

5.4 Case Study of n-UG Public Key Cryptosystem

Let $p = 181$ be a prime number, a matrix D of size 5×5

$$D = \begin{pmatrix} 79 & 15 & 103 & 120 & 14 \\ 171 & 160 & 17 & 55 & 81 \\ 91 & 13 & 14 & 6 & 18 \\ 130 & 22 & 67 & 145 & 107 \\ 150 & 18 & 23 & 39 & 70 \end{pmatrix},$$

where $D \in GL_5(F_{181})$, $V = (27, 45, 91, 143, 175)$ and a graph G as shown in Figure 5.3.

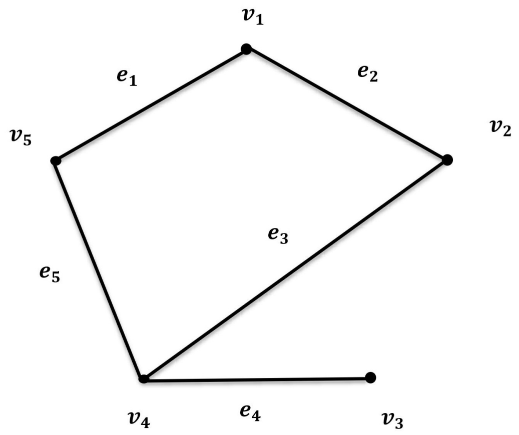

Figure 5.3: A graph is public key

Key Generation Process.

Alice performs the following:

- She chooses $a = 89$ as her a private key and computes her public key A by

$$A = D^{89}\ (mod\ 181) \equiv \begin{pmatrix} 69 & 152 & 159 & 8 & 103 \\ 179 & 171 & 119 & 132 & 77 \\ 110 & 59 & 169 & 70 & 65 \\ 63 & 174 & 157 & 115 & 55 \\ 148 & 94 & 147 & 159 & 51 \end{pmatrix},$$

So, Alice's private and public keys are $a = 89$ and $A = D^{89}$.

Encryption process.

Bob does the following steps:

- He chooses his an ephemeral secret key $b = 121$ and computes

$$C_1 = D^{121}\ (mod\ 181) \equiv \begin{pmatrix} 159 & 107 & 78 & 65 & 0 \\ 142 & 113 & 72 & 145 & 6 \\ 174 & 94 & 165 & 120 & 55 \\ 152 & 69 & 40 & 3 & 121 \\ 109 & 90 & 106 & 180 & 95 \end{pmatrix}$$

and

$$A^{121}\ (mod\ 181) \equiv \begin{pmatrix} 75 & 37 & 44 & 97 & 140 \\ 125 & 40 & 26 & 144 & 126 \\ 163 & 38 & 36 & 124 & 132 \\ 50 & 143 & 92 & 83 & 35 \\ 97 & 128 & 70 & 122 & 108 \end{pmatrix}.$$

- He computes the transformation T on 5-dimensional over matrix A^{121} through

$$\begin{aligned} T(a,b,c,d) = (&75a + 125b + 1632c + 50d + 97e, 37a + 40b + 38c + 143d + 128e, \\ &44a + 26b + 36c + 92d + 70e, 97a + 144b + 124c + 83d + 122e, \\ &140a + 126b + 132c + 35d + 108e), \end{aligned}$$

- He computes $v^{\star} = T(v) \equiv (91, 55, 89, 26, 117)\ (mod\ 181)$,

- He chooses his plaintext as a sub graph in Figure 5.4 and represents it by a vector $m = (114, 0, 45, 84, 156)$.

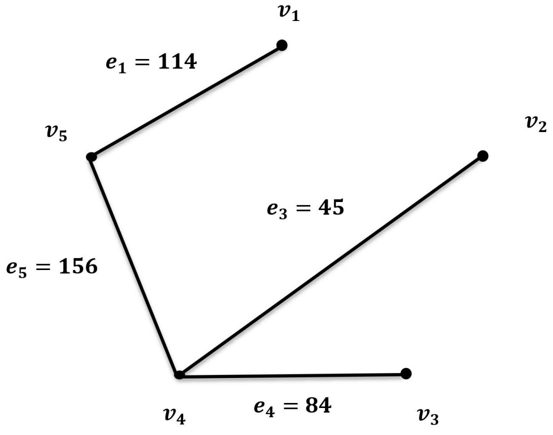

Figure 5.4: A plaintext m as a sub graph H of undirected graph G.

- He converts a vector m into $m^\star = m +_{181} v^\star \equiv (24, 55, 134, 110, 92)$.

- He computes $C_2 = T(m^\star) =\equiv (53, 92, 159, 158, 134)$ $(mod\ 181)$.

- Bob sends his ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She computes

$$(C_1^{89})^{-1}\ (mod\ 181) \equiv \begin{pmatrix} 38 & 55 & 30 & 41 & 26 \\ 17 & 153 & 5 & 75 & 2 \\ 135 & 170 & 137 & 101 & 122 \\ 33 & 139 & 21 & 165 & 93 \\ 12 & 154 & 106 & 147 & 3 \end{pmatrix},$$

  where $C_1^{89} = A^{121}$.

- She computes the transformations T and $T^\star$ on 5-dimensional of the matrices C_1^{89} and $(C_1^{89})^{-1}$ respectively by

  $T(a, b, c, d, e) = (75a + 125b + 1632c + 50d + 97e, 37a + 40b + 38c + 143d + 128e, 44a + 26b + 36c + 92d + 70e, 97a + 144b + 124c + 83d + 122e, 140a + 126b + 132c + 35d + 108e)$,

  $T^\star(a, b, c, e, d) = (38a + 17b + 135c + 33d + 12e, 55a + 153b + 170c + 139d + 154e, 30a + 5b + 137c + 21d + 106e, 41a + 75b + 101c + 165d + 147e, 26a + 2b + 122c + 93d + 3e)$.

- Using public vector v to compute $v^\star = T(v) \equiv (91, 55, 89, 26, 117)$ $(mod\ 181)$ and computes $T^\star(C_2) \equiv m^\star$.

- She computes $m^\star -_{181} v^\star \equiv m$.

- She converts m into original plaintext as a subgraph as shown in Figure 5.4.

5.5 The Results of MPF-Diffie – Hellman Key Exchange

With different values of a small (or a large) prime p, the computations of the MPF- Diffie-Hellman encryption scheme have been done as shown in Tables (5.7) and (5.8). respectively.

Table 5.7: The experimental results of MPF-Diffie – Hellman key exchange (part 1).

p	Shear key D	Alice's secret key L	$A \equiv^{L} D\ (mod\ p)$
41	$\begin{pmatrix} 21 & 8 & 1 \\ 3 & 35 & 12 \\ 40 & 9 & 22 \end{pmatrix}$	$\begin{pmatrix} 1 & 32 & 5 \\ 13 & 4 & 21 \\ 7 & 15 & 2 \end{pmatrix}$	$\begin{pmatrix} 20 & 40 & 19 \\ 5 & 10 & 15 \\ 11 & 6 & 30 \end{pmatrix}$
59	$\begin{pmatrix} 55 & 38 & 25 \\ 13 & 44 & 51 \\ 14 & 57 & 29 \end{pmatrix}$	$\begin{pmatrix} 49 & 37 & 25 \\ 50 & 33 & 32 \\ 17 & 12 & 8 \end{pmatrix}$	$\begin{pmatrix} 32 & 5 & 49 \\ 52 & 55 & 1 \\ 2 & 40 & 1 \end{pmatrix}$
127	$\begin{pmatrix} 112 & 14 & 66 \\ 85 & 126 & 51 \\ 26 & 3 & 120 \end{pmatrix}$	$\begin{pmatrix} 110 & 40 & 17 \\ 87 & 91 & 11 \\ 45 & 123 & 18 \end{pmatrix}$	$\begin{pmatrix} 87 & 75 & 42 \\ 81 & 45 & 84 \\ 19 & 64 & 32 \end{pmatrix}$
181	$\begin{pmatrix} 10 & 99 & 65 \\ 173 & 20 & 180 \\ 165 & 13 & 59 \end{pmatrix}$	$\begin{pmatrix} 175 & 7 & 80 \\ 54 & 152 & 179 \\ 85 & 15 & 32 \end{pmatrix}$	$\begin{pmatrix} 64 & 59 & 138 \\ 43 & 126 & 135 \\ 125 & 45 & 4 \end{pmatrix}$
211	$\begin{pmatrix} 199 & 33 & 87 \\ 200 & 14 & 153 \\ 16 & 205 & 123 \end{pmatrix}$	$\begin{pmatrix} 165 & 89 & 210 \\ 199 & 203 & 175 \\ 166 & 145 & 89 \end{pmatrix}$	$\begin{pmatrix} 8 & 110 & 12 \\ 85 & 161 & 42 \\ 108 & 116 & 104 \end{pmatrix}$

Table 5.8: The experimental results of MPF-Diffie – Hellman key exchange (part 2).

Bob's secret key R	$B \equiv D^{R}\ (mod\ p)$	${}^{L}B \equiv A^{R}\ (mod\ p)$
$\begin{pmatrix} 9 & 25 & 13 \\ 3 & 15 & 7 \\ 19 & 6 & 33 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 10 \\ 13 & 4 & 17 \\ 35 & 25 & 12 \end{pmatrix}$	$\begin{pmatrix} 26 & 25 & 7 \\ 27 & 4 & 11 \\ 12 & 31 & 27 \end{pmatrix}$
$\begin{pmatrix} 18 & 32 & 3 \\ 58 & 43 & 41 \\ 29 & 16 & 47 \end{pmatrix}$	$\begin{pmatrix} 49 & 14 & 12 \\ 15 & 38 & 28 \\ 12 & 22 & 6 \end{pmatrix}$	$\begin{pmatrix} 51 & 20 & 38 \\ 27 & 24 & 28 \\ 7 & 50 & 35 \end{pmatrix}$
$\begin{pmatrix} 109 & 32 & 77 \\ 44 & 99 & 14 \\ 124 & 9 & 55 \end{pmatrix}$	$\begin{pmatrix} 5 & 74 & 35 \\ 118 & 81 & 120 \\ 71 & 91 & 88 \end{pmatrix}$	$\begin{pmatrix} 13 & 102 & 70 \\ 120 & 118 & 49 \\ 76 & 47 & 47 \end{pmatrix}$
$\begin{pmatrix} 100 & 60 & 17 \\ 169 & 46 & 98 \\ 133 & 117 & 125 \end{pmatrix}$	$\begin{pmatrix} 64 & 135 & 130 \\ 177 & 20 & 92 \\ 135 & 144 & 49 \end{pmatrix}$	$\begin{pmatrix} 168 & 122 & 62 \\ 121 & 34 & 87 \\ 56 & 79 & 180 \end{pmatrix}$
$\begin{pmatrix} 207 & 81 & 62 \\ 185 & 109 & 175 \\ 145 & 113 & 111 \end{pmatrix}$	$\begin{pmatrix} 38 & 187 & 165 \\ 69 & 4 & 34 \\ 10 & 197 & 157 \end{pmatrix}$	$\begin{pmatrix} 202 & 158 & 31 \\ 170 & 185 & 203 \\ 41 & 85 & 41 \end{pmatrix}$

5.6 The Computational Results on MPF-ElGamal public key cryptosystem

Some simple computations of MPF- ElGamal public key cryptosystem have been done. Some experimental samples with different values of a prime p are chosen. The computational results to generate the keys, encryption and decryption processes are shown in Tables (5.9), (5.10) and (5.11).

Table 5.9: The experimental results of MPF-ElGamal public key cryptosystem: key generation process.

p	Shear key D	The private key L	A public key $A \equiv^L D\ (mod\ p)$
53	$\begin{pmatrix} 51 & 15 & 19 & 25 \\ 12 & 35 & 45 & 1 \\ 13 & 34 & 17 & 32 \\ 52 & 7 & 18 & 35 \end{pmatrix}$	$\begin{pmatrix} 12 & 8 & 22 & 38 \\ 25 & 35 & 50 & 40 \\ 16 & 44 & 45 & 32 \\ 32 & 40 & 51 & 10 \end{pmatrix}$	$\begin{pmatrix} 28 & 4 & 25 & 24 \\ 47 & 35 & 47 & 10 \\ 44 & 33 & 11 & 39 \\ 46 & 39 & 42 & 26 \end{pmatrix}$
61	$\begin{pmatrix} 2 & 31 & 19 & 5 \\ 12 & 3 & 25 & 13 \\ 1 & 59 & 7 & 32 \\ 35 & 7 & 6 & 23 \end{pmatrix}$	$\begin{pmatrix} 25 & 14 & 9 & 21 \\ 2 & 3 & 60 & 13 \\ 18 & 55 & 7 & 32 \\ 15 & 1 & 51 & 17 \end{pmatrix}$	$\begin{pmatrix} 12 & 17 & 41 & 9 \\ 7 & 51 & 29 & 32 \\ 60 & 23 & 29 & 11 \\ 60 & 32 & 56 & 45 \end{pmatrix}$
113	$\begin{pmatrix} 112 & 31 & 19 & 69 \\ 110 & 73 & 100 & 70 \\ 14 & 44 & 7 & 72 \\ 10 & 87 & 69 & 73 \end{pmatrix}$	$\begin{pmatrix} 35 & 87 & 80 & 84 \\ 102 & 90 & 79 & 73 \\ 107 & 88 & 83 & 13 \\ 53 & 92 & 77 & 110 \end{pmatrix}$	$\begin{pmatrix} 78 & 40 & 5 & 86 \\ 89 & 82 & 30 & 19 \\ 54 & 72 & 33 & 86 \\ 69 & 14 & 33 & 25 \end{pmatrix}$
139	$\begin{pmatrix} 100 & 125 & 109 & 65 \\ 110 & 137 & 10 & 73 \\ 123 & 100 & 17 & 82 \\ 20 & 101 & 16 & 14 \end{pmatrix}$	$\begin{pmatrix} 100 & 112 & 8 & 17 \\ 136 & 91 & 85 & 93 \\ 107 & 11 & 15 & 63 \\ 106 & 3 & 86 & 55 \end{pmatrix}$	$\begin{pmatrix} 30 & 70 & 117 & 21 \\ 41 & 111 & 67 & 43 \\ 4 & 53 & 119 & 102 \\ 39 & 53 & 104 & 55 \end{pmatrix}$
173	$\begin{pmatrix} 169 & 111 & 171 & 65 \\ 155 & 131 & 64 & 73 \\ 14 & 87 & 86 & 15 \\ 125 & 14 & 96 & 25 \end{pmatrix}$	$\begin{pmatrix} 154 & 112 & 10 & 88 \\ 136 & 172 & 105 & 93 \\ 123 & 102 & 1 & 144 \\ 99 & 11 & 13 & 55 \end{pmatrix}$	$\begin{pmatrix} 83 & 47 & 23 & 88 \\ 99 & 153 & 65 & 142 \\ 159 & 100 & 159 & 65 \\ 144 & 104 & 23 & 155 \end{pmatrix}$

Table 5.10: The experimental results of MPF-ElGamal public key cryptosystem: encryption process.

The ephemeral key R	$C_1 \equiv D^R \ (mod\ p)$	A plaintext M	$C_2 \equiv A^R \times M \ (mod\ p)$
$\begin{pmatrix} 20 & 50 & 7 & 2 \\ 40 & 10 & 14 & 44 \\ 15 & 13 & 29 & 21 \\ 25 & 26 & 51 & 11 \end{pmatrix}$	$\begin{pmatrix} 26 & 51 & 36 & 2 \\ 21 & 30 & 44 & 8 \\ 27 & 17 & 18 & 3 \\ 17 & 20 & 42 & 29 \end{pmatrix}$	$\begin{pmatrix} 2 & 5 & 11 & 17 \\ 19 & 23 & 3 & 31 \\ 41 & 17 & 47 & 29 \\ 17 & 37 & 43 & 19 \end{pmatrix}$	$\begin{pmatrix} 11 & 31 & 11 & 17 \\ 41 & 0 & 31 & 4 \\ 3 & 34 & 15 & 21 \\ 49 & 13 & 15 & 47 \end{pmatrix}$
$\begin{pmatrix} 1 & 17 & 33 & 60 \\ 14 & 25 & 12 & 37 \\ 1 & 15 & 13 & 40 \\ 21 & 17 & 12 & 3 \end{pmatrix}$	$\begin{pmatrix} 21 & 58 & 10 & 7 \\ 22 & 3 & 12 & 4 \\ 34 & 29 & 7 & 4 \\ 26 & 19 & 39 & 25 \end{pmatrix}$	$\begin{pmatrix} 0 & 12 & 55 & 60 \\ 47 & 53 & 29 & 32 \\ 53 & 37 & 39 & 25 \\ 21 & 17 & 19 & 1 \end{pmatrix}$	$\begin{pmatrix} 8 & 10 & 8 & 22 \\ 15 & 27 & 26 & 37 \\ 57 & 26 & 49 & 52 \\ 3 & 16 & 18 & 12 \end{pmatrix}$
$\begin{pmatrix} 10 & 111 & 45 & 9 \\ 110 & 98 & 13 & 91 \\ 75 & 6 & 97 & 87 \\ 80 & 12 & 88 & 65 \end{pmatrix}$	$\begin{pmatrix} 55 & 51 & 17 & 55 \\ 22 & 89 & 100 & 54 \\ 32 & 7 & 13 & 112 \\ 1 & 58 & 75 & 82 \end{pmatrix}$	$\begin{pmatrix} 83 & 5 & 41 & 7 \\ 53 & 67 & 101 & 23 \\ 61 & 93 & 17 & 92 \\ 67 & 3 & 93 & 29 \end{pmatrix}$	$\begin{pmatrix} 107 & 7 & 8 & 101 \\ 110 & 95 & 13 & 50 \\ 11 & 14 & 101 & 48 \\ 14 & 46 & 74 & 37 \end{pmatrix}$
$\begin{pmatrix} 55 & 122 & 41 & 75 \\ 35 & 102 & 129 & 112 \\ 130 & 87 & 99 & 100 \\ 80 & 76 & 70 & 89 \end{pmatrix}$	$\begin{pmatrix} 24 & 48 & 105 & 38 \\ 102 & 84 & 129 & 64 \\ 59 & 17 & 86 & 47 \\ 10 & 41 & 53 & 60 \end{pmatrix}$	$\begin{pmatrix} 99 & 43 & 41 & 17 \\ 13 & 133 & 122 & 23 \\ 93 & 93 & 19 & 29 \\ 97 & 83 & 92 & 29 \end{pmatrix}$	$\begin{pmatrix} 37 & 45 & 74 & 46 \\ 66 & 38 & 77 & 112 \\ 101 & 25 & 43 & 137 \\ 28 & 8 & 45 & 137 \end{pmatrix}$
$\begin{pmatrix} 75 & 125 & 149 & 75 \\ 170 & 169 & 129 & 112 \\ 145 & 158 & 171 & 18 \\ 100 & 45 & 166 & 81 \end{pmatrix}$	$\begin{pmatrix} 104 & 140 & 51 & 11 \\ 127 & 52 & 117 & 94 \\ 53 & 11 & 150 & 96 \\ 5 & 3 & 99 & 75 \end{pmatrix}$	$\begin{pmatrix} 169 & 93 & 45 & 7 \\ 125 & 97 & 101 & 5 \\ 123 & 101 & 19 & 29 \\ 33 & 107 & 7 & 2 \end{pmatrix}$	$\begin{pmatrix} 47 & 12 & 2 & 138 \\ 18 & 106 & 170 & 43 \\ 138 & 39 & 77 & 46 \\ 103 & 69 & 154 & 23 \end{pmatrix}$

Table 5.11: The experimental results of MPF-ElGamal public key cryptosystem: decryption process.

${}^{L}C_1 \ (mod\ p)$	$({}^{L}C_1)^{-1} \ (mod\ p)$	$({}^{L}C_1)^{-1} \times C_2 \equiv M \ (mod\ p)$
$\begin{pmatrix} 43 & 7 & 38 & 52 \\ 42 & 4 & 40 & 15 \\ 34 & 11 & 50 & 41 \\ 35 & 47 & 39 & 33 \end{pmatrix}$	$\begin{pmatrix} 9 & 20 & 51 & 29 \\ 26 & 24 & 30 & 4 \\ 16 & 44 & 24 & 39 \\ 10 & 50 & 29 & 1 \end{pmatrix}$	$\begin{pmatrix} 2 & 5 & 11 & 17 \\ 19 & 23 & 3 & 31 \\ 41 & 17 & 47 & 29 \\ 17 & 37 & 43 & 19 \end{pmatrix}$
$\begin{pmatrix} 9 & 50 & 41 & 29 \\ 59 & 12 & 12 & 22 \\ 49 & 11 & 30 & 36 \\ 19 & 18 & 45 & 38 \end{pmatrix}$	$\begin{pmatrix} 19 & 13 & 0 & 47 \\ 58 & 21 & 25 & 13 \\ 44 & 1 & 22 & 6 \\ 57 & 53 & 52 & 5 \end{pmatrix}$	$\begin{pmatrix} 0 & 12 & 55 & 60 \\ 47 & 53 & 29 & 32 \\ 53 & 37 & 39 & 25 \\ 21 & 17 & 19 & 1 \end{pmatrix}$
$\begin{pmatrix} 101 & 92 & 6 & 4 \\ 52 & 33 & 37 & 95 \\ 27 & 93 & 15 & 92 \\ 42 & 97 & 92 & 12 \end{pmatrix}$	$\begin{pmatrix} 21 & 103 & 36 & 41 \\ 27 & 111 & 101 & 79 \\ 92 & 79 & 17 & 41 \\ 67 & 67 & 93 & 109 \end{pmatrix}$	$\begin{pmatrix} 83 & 5 & 41 & 7 \\ 53 & 67 & 101 & 23 \\ 61 & 93 & 17 & 92 \\ 67 & 3 & 93 & 29 \end{pmatrix}$
$\begin{pmatrix} 114 & 41 & 56 & 14 \\ 110 & 99 & 109 & 73 \\ 70 & 75 & 57 & 130 \\ 136 & 8 & 14 & 82 \end{pmatrix}$	$\begin{pmatrix} 79 & 50 & 64 & 27 \\ 35 & 58 & 124 & 95 \\ 119 & 69 & 18 & 105 \\ 69 & 120 & 116 & 45 \end{pmatrix}$	$\begin{pmatrix} 99 & 43 & 41 & 17 \\ 13 & 133 & 122 & 23 \\ 93 & 93 & 19 & 29 \\ 97 & 83 & 92 & 29 \end{pmatrix}$
$\begin{pmatrix} 84 & 126 & 106 & 172 \\ 126 & 24 & 134 & 86 \\ 23 & 69 & 21 & 71 \\ 51 & 142 & 123 & 30 \end{pmatrix}$	$\begin{pmatrix} 160 & 115 & 90 & 97 \\ 111 & 30 & 43 & 12 \\ 21 & 128 & 126 & 91 \\ 68 & 20 & 38 & 103 \end{pmatrix}$	$\begin{pmatrix} 169 & 93 & 45 & 7 \\ 125 & 97 & 101 & 5 \\ 123 & 101 & 19 & 29 \\ 33 & 107 & 7 & 2 \end{pmatrix}$

5.7 The Computational Results on Hybrid GMP-ElGamal public key cryptosystem

Some simple computations of hybrid GMP-ElGamal public key cryptosystem have been done. Five experimental samples with different values of a prime pare chosen. The computational results to generate the keys, encryption and decryption processes are shown in Tables (5.12), (5.13) and (5.14) respectively. Figure 5.5 shows the original plaintext that are chosen and represented as the graphs G_a, G_b, G_c, G_d and G_e to encrypt in Table (5.13) and to recover in Table (5.14).

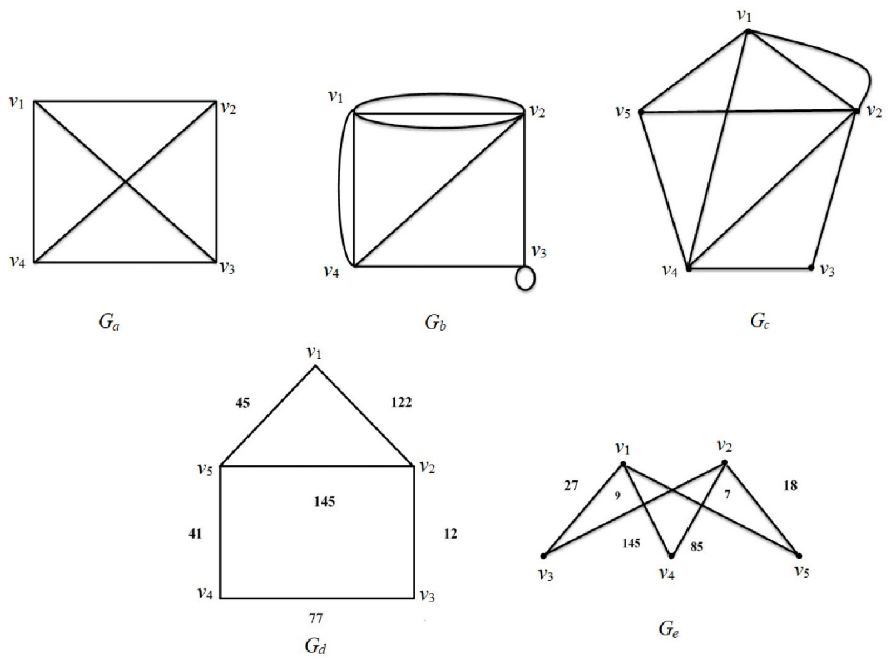

Figure 5.5: The original plaintext as the graphs that correspond to Tables (5.13), (5.14).

Table 5.12: The experimental results of hybrid GMPF-ElGamal public key cryptosystem: key generation process.

p	Shear key D	Alice's secret key L	$A \equiv^L D\ (mod\ p)$
61	$\begin{pmatrix} 2 & 31 & 19 & 5 \\ 12 & 3 & 25 & 13 \\ 1 & 59 & 7 & 32 \\ 35 & 7 & 6 & 23 \end{pmatrix}$	$\begin{pmatrix} 25 & 14 & 9 & 21 \\ 2 & 3 & 60 & 13 \\ 18 & 55 & 7 & 32 \\ 15 & 1 & 51 & 17 \end{pmatrix}$	$\begin{pmatrix} 12 & 17 & 41 & 9 \\ 7 & 51 & 29 & 32 \\ 60 & 23 & 29 & 11 \\ 60 & 32 & 56 & 45 \end{pmatrix}$
97	$\begin{pmatrix} 12 & 31 & 92 & 25 \\ 96 & 16 & 77 & 13 \\ 55 & 1 & 65 & 4 \\ 90 & 87 & 93 & 20 \end{pmatrix}$	$\begin{pmatrix} 96 & 33 & 9 & 21 \\ 34 & 44 & 70 & 74 \\ 18 & 95 & 88 & 15 \\ 17 & 8 & 10 & 17 \end{pmatrix}$	$\begin{pmatrix} 85 & 55 & 28 & 8 \\ 93 & 44 & 2 & 85 \\ 45 & 83 & 17 & 94 \\ 58 & 83 & 14 & 15 \end{pmatrix}$
137	$\begin{pmatrix} 121 & 6 & 8 & 16 & 111 \\ 13 & 101 & 53 & 101 & 32 \\ 45 & 56 & 9 & 3 & 66 \\ 2 & 99 & 17 & 1 & 12 \\ 135 & 1 & 23 & 87 & 45 \end{pmatrix}$	$\begin{pmatrix} 25 & 106 & 47 & 16 & 111 \\ 130 & 75 & 12 & 14 & 6 \\ 23 & 60 & 15 & 91 & 11 \\ 4 & 90 & 122 & 81 & 3 \\ 99 & 15 & 18 & 22 & 4 \end{pmatrix}$	$\begin{pmatrix} 86 & 90 & 89 & 70 & 13 \\ 51 & 78 & 110 & 9 & 63 \\ 86 & 71 & 10 & 27 & 68 \\ 99 & 121 & 67 & 101 & 15 \\ 71 & 110 & 96 & 123 & 111 \end{pmatrix}$
149	$\begin{pmatrix} 10 & 6 & 8 & 10 & 53 \\ 13 & 142 & 53 & 13 & 141 \\ 10 & 146 & 9 & 3 & 66 \\ 2 & 89 & 17 & 2 & 12 \\ 135 & 90 & 14 & 87 & 41 \end{pmatrix}$	$\begin{pmatrix} 125 & 16 & 25 & 16 & 110 \\ 130 & 55 & 12 & 15 & 60 \\ 25 & 60 & 15 & 92 & 10 \\ 144 & 90 & 44 & 88 & 133 \\ 91 & 142 & 132 & 91 & 4 \end{pmatrix}$	$\begin{pmatrix} 85 & 137 & 84 & 130 & 62 \\ 145 & 134 & 28 & 1 & 64 \\ 103 & 38 & 18 & 46 & 91 \\ 139 & 91 & 83 & 97 & 70 \\ 85 & 92 & 50 & 129 & 93 \end{pmatrix}$
151	$\begin{pmatrix} 100 & 6 & 59 & 10 & 100 \\ 149 & 14 & 50 & 13 & 55 \\ 130 & 146 & 19 & 130 & 66 \\ 149 & 89 & 17 & 1 & 65 \\ 135 & 14 & 59 & 135 & 150 \end{pmatrix}$	$\begin{pmatrix} 25 & 33 & 125 & 43 & 22 \\ 149 & 55 & 1 & 15 & 60 \\ 70 & 60 & 15 & 92 & 70 \\ 144 & 33 & 32 & 88 & 122 \\ 99 & 150 & 132 & 93 & 14 \end{pmatrix}$	$\begin{pmatrix} 75 & 45 & 99 & 121 & 36 \\ 93 & 74 & 1 & 69 & 148 \\ 23 & 6 & 47 & 87 & 114 \\ 141 & 14 & 37 & 133 & 136 \\ 115 & 128 & 29 & 25 & 92 \end{pmatrix}$

Table 5.13: The experimental results of hybrid GMPF-ElGamal public key cryptosystem: encryption process.

A plaintext is a graph	The ephemeral key R	adjacent matrix M	$C_1 \equiv D^R\ (mod\ p)$	$C_2 \equiv A^R \times M$
G_a	$\begin{pmatrix} 1 & 17 & 33 & 60 \\ 14 & 25 & 12 & 37 \\ 1 & 15 & 13 & 40 \\ 21 & 17 & 12 & 3 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 21 & 58 & 10 & 7 \\ 22 & 3 & 12 & 4 \\ 34 & 29 & 7 & 4 \\ 26 & 19 & 39 & 25 \end{pmatrix}$	$\begin{pmatrix} 59 & 18 & 27 & 39 \\ 46 & 32 & 32 & 22 \\ 16 & 54 & 35 & 29 \\ 40 & 41 & 14 & 21 \end{pmatrix}$
G_b	$\begin{pmatrix} 61 & 95 & 14 & 11 \\ 81 & 25 & 70 & 3 \\ 44 & 95 & 33 & 91 \\ 34 & 15 & 6 & 29 \end{pmatrix}$	$\begin{pmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 61 & 78 & 68 & 74 \\ 3 & 44 & 37 & 93 \\ 34 & 60 & 47 & 46 \\ 62 & 10 & 94 & 60 \end{pmatrix}$	$\begin{pmatrix} 45 & 18 & 58 & 48 \\ 82 & 12 & 91 & 28 \\ 59 & 1 & 24 & 12 \\ 41 & 44 & 46 & 29 \end{pmatrix}$
G_c	$\begin{pmatrix} 44 & 45 & 16 & 70 & 2 \\ 112 & 16 & 5 & 9 & 125 \\ 130 & 121 & 7 & 27 & 32 \\ 3 & 33 & 17 & 132 & 15 \\ 71 & 25 & 96 & 117 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 52 & 102 & 102 & 60 & 69 \\ 128 & 64 & 31 & 117 & 77 \\ 135 & 114 & 116 & 13 & 105 \\ 90 & 45 & 128 & 53 & 94 \\ 82 & 136 & 84 & 136 & 97 \end{pmatrix}$	$\begin{pmatrix} 63 & 75 & 23 & 70 & 30 \\ 128 & 68 & 43 & 130 & 47 \\ 34 & 132 & 77 & 44 & 135 \\ 80 & 135 & 83 & 39 & 34 \\ 7 & 82 & 16 & 2 & 45 \end{pmatrix}$
G_d	$\begin{pmatrix} 120 & 50 & 8 & 77 & 123 \\ 111 & 16 & 133 & 141 & 125 \\ 12 & 142 & 18 & 27 & 1 \\ 145 & 55 & 47 & 41 & 15 \\ 11 & 25 & 17 & 117 & 55 \end{pmatrix}$	$\begin{pmatrix} 0 & 122 & 0 & 0 & 45 \\ 122 & 0 & 12 & 0 & 145 \\ 0 & 12 & 0 & 77 & 0 \\ 0 & 0 & 77 & 0 & 41 \\ 45 & 145 & 0 & 41 & 0 \end{pmatrix}$	$\begin{pmatrix} 3 & 98 & 117 & 66 & 94 \\ 73 & 69 & 17 & 72 & 3 \\ 97 & 102 & 108 & 6 & 110 \\ 10 & 76 & 48 & 103 & 81 \\ 23 & 133 & 15 & 21 & 79 \end{pmatrix}$	$\begin{pmatrix} 112 & 133 & 16 & 81 & 59 \\ 96 & 20 & 101 & 146 & 2 \\ 91 & 132 & 141 & 135 & 42 \\ 106 & 125 & 147 & 77 & 25 \\ 102 & 19 & 52 & 29 & 138 \end{pmatrix}$
G_e	$\begin{pmatrix} 150 & 25 & 15 & 15 & 23 \\ 36 & 101 & 37 & 141 & 11 \\ 149 & 142 & 18 & 25 & 12 \\ 3 & 55 & 47 & 41 & 15 \\ 150 & 125 & 17 & 35 & 55 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 27 & 9 & 85 \\ 0 & 0 & 145 & 7 & 18 \\ 27 & 145 & 0 & 0 & 0 \\ 9 & 7 & 77 & 0 & 0 \\ 85 & 18 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 20 & 82 & 82 & 65 & 109 \\ 26 & 147 & 118 & 108 & 147 \\ 26 & 10 & 106 & 59 & 125 \\ 85 & 107 & 55 & 130 & 70 \\ 73 & 72 & 93 & 39 & 123 \end{pmatrix}$	$\begin{pmatrix} 146 & 22 & 69 & 140 & 76 \\ 25 & 114 & 138 & 111 & 95 \\ 61 & 143 & 36 & 28 & 13 \\ 93 & 3 & 142 & 49 & 86 \\ 111 & 109 & 69 & 31 & 58 \end{pmatrix}$

Table 5.14: The experimental results of hybrid GMPF-ElGamal public key cryptosystem: decryption process.

${}^{L}C_1\ (mod\ p)$	$({}^{L}C_1)^{-1}\ (mod\ p)$	$({}^{L}C_1)^{-1} \times C_2 \equiv M\ (mod\ p)$	A plaintext is a graph
$\begin{pmatrix} 9 & 50 & 41 & 29 \\ 59 & 12 & 12 & 22 \\ 49 & 11 & 30 & 36 \\ 19 & 18 & 45 & 38 \end{pmatrix}$	$\begin{pmatrix} 19 & 13 & 0 & 47 \\ 58 & 21 & 25 & 13 \\ 44 & 1 & 22 & 6 \\ 57 & 53 & 52 & 5 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$	G_a
$\begin{pmatrix} 30 & 33 & 52 & 70 \\ 53 & 44 & 72 & 72 \\ 72 & 45 & 17 & 59 \\ 48 & 2 & 28 & 66 \end{pmatrix}$	$\begin{pmatrix} 22 & 34 & 73 & 11 \\ 31 & 14 & 5 & 87 \\ 2 & 66 & 20 & 5 \\ 41 & 85 & 0 & 24 \end{pmatrix}$	$\begin{pmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix}$	G_b
$\begin{pmatrix} 7 & 81 & 23 & 79 & 96 \\ 4 & 123 & 41 & 57 & 99 \\ 58 & 92 & 29 & 122 & 2 \\ 88 & 106 & 91 & 114 & 28 \\ 29 & 51 & 119 & 102 & 77 \end{pmatrix}$	$\begin{pmatrix} 68 & 71 & 78 & 121 & 27 \\ 102 & 122 & 5 & 75 & 80 \\ 59 & 38 & 6 & 112 & 68 \\ 33 & 58 & 96 & 107 & 124 \\ 126 & 22 & 128 & 97 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$	G_c
$\begin{pmatrix} 133 & 135 & 81 & 14 & 57 \\ 74 & 148 & 71 & 8 & 18 \\ 33 & 126 & 22 & 137 & 71 \\ 137 & 69 & 139 & 115 & 57 \\ 36 & 98 & 69 & 57 & 71 \end{pmatrix}$	$\begin{pmatrix} 109 & 56 & 102 & 83 & 120 \\ 80 & 34 & 10 & 143 & 29 \\ 74 & 115 & 131 & 60 & 131 \\ 122 & 41 & 122 & 128 & 116 \\ 38 & 78 & 45 & 143 & 132 \end{pmatrix}$	$\begin{pmatrix} 0 & 122 & 0 & 0 & 45 \\ 122 & 0 & 12 & 0 & 145 \\ 0 & 12 & 0 & 77 & 0 \\ 0 & 0 & 77 & 0 & 41 \\ 45 & 145 & 0 & 41 & 0 \end{pmatrix}$	G_d
$\begin{pmatrix} 39 & 13 & 148 & 142 & 96 \\ 44 & 24 & 17 & 141 & 83 \\ 149 & 136 & 142 & 69 & 62 \\ 14 & 140 & 124 & 129 & 92 \\ 68 & 68 & 96 & 88 & 29 \end{pmatrix}$	$\begin{pmatrix} 137 & 138 & 17 & 143 & 83 \\ 119 & 3 & 1 & 133 & 69 \\ 15 & 101 & 138 & 7 & 73 \\ 103 & 102 & 103 & 102 & 99 \\ 6 & 145 & 84 & 124 & 44 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 27 & 9 & 85 \\ 0 & 0 & 145 & 7 & 18 \\ 27 & 145 & 0 & 0 & 0 \\ 9 & 7 & 0 & 0 & 0 \\ 85 & 18 & 0 & 0 & 0 \end{pmatrix}$	G_e

5.8 Case Study of Soft Graph Public Key Cryptosystem

Alice and Bob agree to use the a prime $p = 2011$ and a matrix D of size 6×6

$$D = \begin{pmatrix} 20 & 400 & 150 & 1500 & 13 & 1700 \\ 2000 & 15 & 170 & 103 & 1033 & 750 \\ 1200 & 87 & 2003 & 121 & 100 & 625 \\ 33 & 900 & 1500 & 333 & 707 & 3 \\ 1750 & 405 & 201 & 91 & 1800 & 18 \\ 1800 & 170 & 110 & 404 & 520 & 2005 \end{pmatrix},$$

where $D \in GL_6(F_{2011})$.

Keys generation process.

Alice performs the following:

- She chooses her secret key

$$L = \begin{pmatrix} 17 & 150 & 23 & 1032 & 250 & 1300 \\ 107 & 200 & 450 & 313 & 1600 & 207 \\ 100 & 1801 & 550 & 2001 & 250 & 1700 \\ 555 & 666 & 89 & 87 & 304 & 444 \\ 750 & 2007 & 1250 & 1000 & 201 & 41 \\ 33 & 15 & 901 & 19 & 1300 & 99 \end{pmatrix},$$

  and computes her public key A by

$$A =^{L} D\ (mod\ 2011) \equiv \begin{pmatrix} 1627 & 1005 & 1676 & 401 & 1222 & 1479 \\ 17 & 336 & 1267 & 369 & 1311 & 1456 \\ 117 & 221 & 1527 & 44 & 781 & 1252 \\ 91 & 1729 & 1457 & 407 & 1665 & 1092 \\ 570 & 301 & 1693 & 1726 & 1898 & 1936 \\ 1811 & 837 & 414 & 751 & 459 & 1892 \end{pmatrix}.$$

Encryption process.

Bob does the following steps:

- He chooses a connected graph in Figure 5.6.

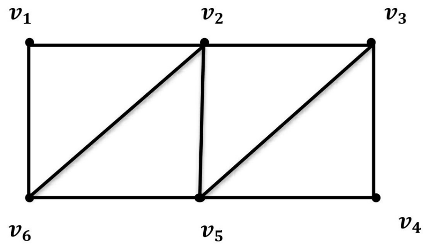

Figure 5.6: The connected graph.

- He chooses $N = \{v_1, v_4\} \subseteq V$ and (F, N) be a soft set over V with approximate function $F : N \to P(V)$ by:

$$F(x) = \{y \in V : xRy \Leftrightarrow d(x, y) \leq 1\}$$

  for all $x \in N$. That is, $F(v_1) = \{v_1, v_2, v_6\}$ and $F(v_4) = \{v_3, v_4, v_5\}$. Let (K, N) be a soft set over E with approximate function $K : N \to P(E)$ by:

$$K(x) = \{uv \in E : \{u, v\} \subseteq F(x)\}$$

for all $x \in A$. That is, $K(v_1) = \{v_1v_2, v_1v_6, v_2v_6\}$ and $K(v_4) = \{v_3v_4, v_3v_5, v_4v_5\}$. Then the soft graph is shown in Figure 5.7.

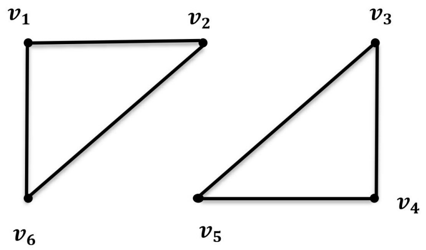

Figure 5.7: The soft graph.

- He chooses a plaintext $\{355, 675, 991, 1773, 1991, 2001\}$.
- He converts a connected graph in Figure 5.6 to weighted connected graph in Figure 5.8.

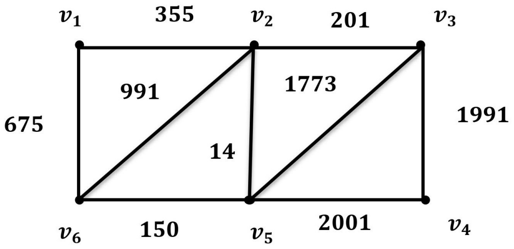

Figure 5.8: The weighted connected graph.

- He represents the weighted connected graph by the adjacent matrix M.

$$M = \begin{pmatrix} 0 & 355 & 0 & 0 & 0 & 675 \\ 355 & 0 & 201 & 0 & 14 & 991 \\ 0 & 201 & 0 & 1991 & 1773 & 0 \\ 0 & 0 & 1991 & 0 & 2001 & 0 \\ 0 & 14 & 1773 & 2001 & 0 & 150 \\ 675 & 991 & 0 & 0 & 150 & 0 \end{pmatrix}.$$

- He chooses an ephemeral key as a matrix

$$R = \begin{pmatrix} 85 & 22 & 330 & 1400 & 2000 & 650 \\ 333 & 15 & 881 & 130 & 700 & 850 \\ 14 & 103 & 132 & 1240 & 87 & 75 \\ 65 & 1987 & 670 & 277 & 1900 & 900 \\ 1011 & 170 & 1100 & 1301 & 1953 & 57 \\ 1920 & 41 & 37 & 200 & 1600 & 5 \end{pmatrix}.$$

- He computes his cipher text

$$C_1 = D^R \ (mod\ 2011) \equiv \begin{pmatrix} 1097 & 607 & 988 & 1496 & 407 & 46 \\ 97 & 1235 & 842 & 2008 & 1633 & 501 \\ 1027 & 620 & 468 & 761 & 1055 & 690 \\ 907 & 422 & 174 & 1082 & 1660 & 1283 \\ 25 & 708 & 1200 & 211 & 1856 & 1537 \\ 121 & 1124 & 171 & 1644 & 1260 & 1613 \end{pmatrix}.$$

and

$$C_2 = A^R \times M \ (mod\ 2011) \equiv \begin{pmatrix} 488 & 115 & 355 & 580 & 228 & 306 \\ 1930 & 96 & 1214 & 1647 & 1989 & 212 \\ 215 & 285 & 330 & 1466 & 1392 & 394 \\ 769 & 734 & 834 & 220 & 597 & 848 \\ 55 & 1126 & 1455 & 1983 & 499 & 43 \\ 1064 & 1006 & 144 & 145 & 211 & 1691 \end{pmatrix}$$

,such that A^R is the shear key

$$K =^L C_1 = A^R \ (mod\ 2011) \equiv \begin{pmatrix} 804 & 496 & 535 & 794 & 883 & 2010 \\ 719 & 1139 & 692 & 1702 & 1870 & 1534 \\ 267 & 536 & 641 & 1778 & 1789 & 210 \\ 1920 & 60 & 536 & 1715 & 917 & 628 \\ 394 & 52 & 597 & 890 & 1222 & 509 \\ 171 & 230 & 468 & 41 & 55 & 256 \end{pmatrix}.$$

- He sends his ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She used her private key L to compute

$$({}^{L}C_1)^{-1}\ (mod\ 2011) \equiv \begin{pmatrix} 417 & 900 & 1034 & 1848 & 1654 & 856 \\ 1475 & 1826 & 1551 & 383 & 909 & 1667 \\ 767 & 27 & 1811 & 902 & 1826 & 627 \\ 448 & 1637 & 1959 & 698 & 1445 & 1053 \\ 1566 & 1354 & 1314 & 1918 & 684 & 225 \\ 113 & 463 & 317 & 35 & 230 & 1796 \end{pmatrix}$$

such that ${}^{L}C_1 \equiv A^R\ (mod\ 2011)$.

- She computes

$$({}^{L}C_1)^{-1} \times C_2\ (mod\ 2011) \equiv \begin{pmatrix} 0 & 355 & 0 & 0 & 0 & 675 \\ 355 & 0 & 201 & 0 & 14 & 991 \\ 0 & 201 & 0 & 1991 & 1773 & 0 \\ 0 & 0 & 1991 & 0 & 2001 & 0 \\ 0 & 14 & 1773 & 2001 & 0 & 150 \\ 675 & 991 & 0 & 0 & 150 & 0 \end{pmatrix} \equiv M.$$

- She converted a matrix M into a corresponding weighted connected graph in Figure 5.8.

- She uses public keys (N, F, K) to find a soft graph in Figure 5.9.

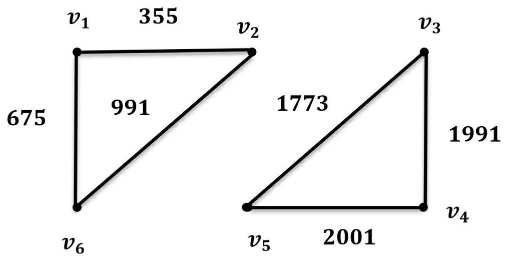

Figure 5.9: The weighted soft graph.

- Then the weights of edges of the soft graph are representation the original plaintext $\{355, 675, 991, 1773, 1991, 2001\}$.

5.9 The Results of KR –Elliptic Curve Public Key Cryptosystem

Some simple computational results about KR-EC cryptosystem have been done. The experimental samples with different values of a primes q and t are chosen The computational results to generate the keys, encryption and decryption processes are shown in Tables (5.15), (5.16) and (5.17) respectively.

Table 5.15: The experimental results of KR-EC cryptosystem: key generation process.

q	t	$n = (q-1)(t-1)$	e	F_p	$E(a,b)$	$\#E(F_p)$
733	853	623664	313399	F_{563}	$(6,10)$	568
773	941	725680	113093	F_{197}	$(1,22)$	193
911	997	906360	9673	F_{1163}	$(2,23)$	1162
1193	1321	1573440	90239	F_{1657}	$(20,14)$	1639
3253	7253	23583504	11039087	F_{2179}	$(25,2)$	2168
71527	91513	6545487312	10767257	F_{2099}	$(46,5)$	2096
112327	322537	36229178736	584663	F_{3089}	$(17,3)$	3088
1312153	312161	409601368320	167161	F_{1283}	$(9,4)$	1280
1211141	1121333	1358090038480	10672613269	F_{1657}	$(4,16)$	1640
1232453	1392221	1715844323440	868781935919	F_{1847}	$(37,20)$	1844

Table 5.16: The experimental results of KR-EC cryptosystem: encryption process.

$M \in E(F_p)$	N is the order of M	$e^{\backprime} \equiv e\ (mod\ p)$	$C = e^{\backprime}M$
$(395,229)$	284	147	$(131,403)$
$(151,67)$	193	188	$(41,172)$
$(999,123)$	1162	377	$(200,260)$
$(650,313)$	1639	94	$(1146,1645)$
$(2021,1131)$	271	173	$(1787,2080)$
$(527,600)$	1048	105	$(1761,402)$
$(993,111)$	386	259	$(2920,1652)$
$(1176,533)$	640	121	$(820,356)$
$(1611,101)$	1640	29	$(358,704)$
$(1715,1227)$	922	887	$(1595,55)$

Table 5.17: The experimental results of KR-EC cryptosystem: decryption process.

d	N is the order of C	$d^{\backprime} \equiv d \ (mod \ p)$	$M = d^{\backprime}C$
199	284	199	$(395, 229)$
77	193	77	$(151, 67)$
937	1162	937	$(999, 123)$
959	1639	959	$(650, 313)$
47	271	47	$(2021, 1131)$
45593	1048	529	$(1761, 402)$
743591	386	155	$(527, 600)$
4900681	640	201	$(1176, 533)$
509	1640	509	$(1611, 101)$
79	922	79	$(1715, 1227)$

5.10 The Results of ECMF-Diffie-Hellman Key Exchange

With different values of a small prime p, simple computations of the ECMF-DH Key Exchange have been done as shown in Tables (5.18) and (5.19).

Table 5.18: The experimental results of ECMF-DH key exchange (part 1).

p	$E(a, b)$	Shear key $D \in EM_nE(F_p)$	Alice's secret key L	A public key $A = L \star D$
61	$(4, 1)$	$\begin{pmatrix} (0,1) & (6,27) \\ (8,22) & (60,22) \end{pmatrix}$	$\begin{pmatrix} 3 & 32 \\ 6 & 16 \end{pmatrix}$	$\begin{pmatrix} (42,9) & (41,20) \\ (15,52) & (39,60) \end{pmatrix}$
67	$(2, 14)$	$\begin{pmatrix} (1,33) & (61,56) \\ (20,58) & (59,25) \end{pmatrix}$	$\begin{pmatrix} 80 & 60 \\ 65 & 77 \end{pmatrix}$	$\begin{pmatrix} (21,65) & (2,48) \\ (1,34) & (29,41) \end{pmatrix}$
79	$(2, 6)$	$\begin{pmatrix} (77,28) & (62,73) \\ (20,15) & (42,47) \end{pmatrix}$	$\begin{pmatrix} 75 & 78 \\ 66 & 45 \end{pmatrix}$	$\begin{pmatrix} (5,46) & (17,56) \\ (62,73) & (41,11) \end{pmatrix}$
97	$(3, 2)$	$\begin{pmatrix} (96,80) & (10,16) \\ (78,78) & (14,13) \end{pmatrix}$	$\begin{pmatrix} 102 & 86 \\ 99 & 65 \end{pmatrix}$	$\begin{pmatrix} (47,46) & (54,87) \\ (9,51) & (60,39) \end{pmatrix}$
131	$(13, 12)$	$\begin{pmatrix} (2,6) & (66,126) \\ (129,19) & (68,130) \end{pmatrix}$	$\begin{pmatrix} 149 & 98 \\ 101 & 125 \end{pmatrix}$	$\begin{pmatrix} (108,80) & (105,64) \\ (51,73) & (25,25) \end{pmatrix}$

Table 5.19: The experimental results of ECMF-DH key exchange (part 2).

Bob's secret key R	A public key $B = D \star R$	Exchanged key $K = L \star B = A \star R$
$\begin{pmatrix} 6 & 12 \\ 2 & 16 \end{pmatrix}$	$\begin{pmatrix} (22,1) & (38,53) \\ (48,58) & (38,8) \end{pmatrix}$	$\begin{pmatrix} (56,51) & (24,47) \\ (15,52) & (29,17) \end{pmatrix}$
$\begin{pmatrix} 28 & 82 \\ 75 & 56 \end{pmatrix}$	$\begin{pmatrix} (39,7) & (39,60) \\ (19,55) & (3,39) \end{pmatrix}$	$\begin{pmatrix} (59,25) & (20,58) \\ (38,25) & (7,6) \end{pmatrix}$
$\begin{pmatrix} 56 & 77 \\ 72 & 44 \end{pmatrix}$	$\begin{pmatrix} (2,52) & (77,51) \\ (28,17) & (2,27) \end{pmatrix}$	$\begin{pmatrix} (58,5) & (1,3) \\ (31,32) & (27,25) \end{pmatrix}$
$\begin{pmatrix} 80 & 101 \\ 57 & 69 \end{pmatrix}$	$\begin{pmatrix} (37,61) & (74,45) \\ (58,67) & (26,14) \end{pmatrix}$	$\begin{pmatrix} (20,60) & (23,4) \\ (58,88) & (9,46) \end{pmatrix}$
$\begin{pmatrix} 122 & 150 \\ 145 & 88 \end{pmatrix}$	$\begin{pmatrix} (64,42) & (114,34) \\ (129,19) & (17,114) \end{pmatrix}$	$\begin{pmatrix} (105,64) & (73,51) \\ (96,65) & (90,90) \end{pmatrix}$

5.11 The Results of ECMF-ElGamal Public Key Cryptosystem

Simple computations of ECMF-ElGamal public key cryptosystem have been obtained. Different experimental samples with various values of a prime p are chosen. The computational results to generate the keys, encryption and decryption processes are shown in Tables (5.20), (5.21) and (5.22) respectively.

Table 5.20: The experimental results of ECMF-ElGamal public key cryptosystem: key generation process.

p	$E(a,b)$	Shear key $D \in EM_nE(F_p)$	Alice's secret key L	A public key $A = L \star D$
139	$(5,10)$	$\begin{pmatrix} (133,96) & (6,16) & (8,80) \\ (14,34) & (100,108) & (40,100) \\ (11,80) & (122,4) & (17,2) \end{pmatrix}$	$\begin{pmatrix} 145 & 85 & 99 \\ 125 & 130 & 121 \\ 140 & 148 & 100 \end{pmatrix}$	$\begin{pmatrix} (92,87) & (67,112) & (62,92) \\ (61,46) & (66,84) & (92,87) \\ (68,124) & (67,112) & (62,47) \end{pmatrix}$
149	$(10,2)$	$\begin{pmatrix} (2,46) & (17,67) & (81,5) \\ (96,41) & (63,43) & (29,29) \\ (100,13) & (5,37) & (30,66) \end{pmatrix}$	$\begin{pmatrix} 120 & 13 & 4 \\ 99 & 130 & 101 \\ 136 & 5 & 17 \end{pmatrix}$	$\begin{pmatrix} (44,10) & (2,46) & (6,24) \\ (30,66) & (142,6) & (11,20) \\ (15,139) & (9,134) & (100,136) \end{pmatrix}$
151	$(3,2)$	$\begin{pmatrix} (0,105) & (13,107) & (7,8)) \\ (124,87) & (137,38) & (144,98) \\ (149,21) & (50,107) & (100,57) \end{pmatrix}$	$\begin{pmatrix} 100 & 130 & 97 \\ 88 & 150 & 135 \\ 146 & 87 & 145 \end{pmatrix}$	$\begin{pmatrix} (99,140) & (88,107) & (29,147) \\ (53,1) & (35,20) & (142,1) \\ (80,79) & (86,46) & (140,65) \end{pmatrix}$
157	$(2,8)$	$\begin{pmatrix} (1,47) & (100,78) & (154,17)) \\ (155,101) & (52,26) & (7,88) \\ (122,148) & (68,133) & (10,20) \end{pmatrix}$	$\begin{pmatrix} 156 & 140 & 88 \\ 126 & 99 & 56 \\ 113 & 87 & 149 \end{pmatrix}$	$\begin{pmatrix} (66,144) & (47,73) & (67,90) \\ (47,73) & (103,149) & (124,127) \\ (33,134) & (101,77) & (65,135) \end{pmatrix}$
941	$(2,7)$	$\begin{pmatrix} (315,925) & (467,613) & (743,84)) \\ (691,850) & (47,357) & (6,519) \\ (4,199) & (400,292) & (9,80) \end{pmatrix}$	$\begin{pmatrix} 700 & 250 & 550 \\ 520 & 13 & 121 \\ 144 & 750 & 800 \end{pmatrix}$	$\begin{pmatrix} (116,937) & (30,890) & (90,525) \\ (456,785) & (624,534) & (692,381) \\ (703,147) & (501,887) & (323,122) \end{pmatrix}$

Table 5.21: The experimental results of ECMF-ElGamal public key cryptosystem: encryption process.

The ephemeral key R	$C_1 = D \star R$	$A \star R$	The plaintext M	$C_2 = M + A \star R$
$\begin{pmatrix} 111 & 144 & 65 \\ 79 & 147 & 115 \\ 125 & 135 & 148 \end{pmatrix}$	$\begin{pmatrix} (70,13) & (127,86) & (8,80) \\ (9,28) & (81,43) & (40,100) \\ (31,40) & O_E & (124,70) \end{pmatrix}$	$\begin{pmatrix} (16,4) & (58,123) & (133,96) \\ (124,69) & (94,20) & (29,91) \\ (8,80) & (3,57) & (2,81) \end{pmatrix}$	$\begin{pmatrix} (56,99) & (91,129) & (11,80) \\ (77,70) & (40,100) & (100,108) \\ (2,81) & (92,52) & (107,134) \end{pmatrix}$	$\begin{pmatrix} (14,105) & (67,27) & (132,132) \\ (114,55) & (3,82) & (123,2) \\ (102,121) & (101,100) & (97,63) \end{pmatrix}$
$\begin{pmatrix} 111 & 56 & 87 \\ 9 & 15 & 18 \\ 126 & 131 & 45 \end{pmatrix}$	$\begin{pmatrix} (134,84) & (81,5) & (17,67) \\ (63,43) & (126,11) & (96,108) \\ (107,128) & (84,101) & (63,106) \end{pmatrix}$	$\begin{pmatrix} (98,14) & (98,14) & (77,89) \\ (141,30) & (85,90) & (26,16) \\ (73,73) & (98,14) & (147,102) \end{pmatrix}$	$\begin{pmatrix} (9,10) & (35,100) & (75,19) \\ (143,60) & (49,122) & (30,66) \\ (15,139) & (5,112) & (147,102) \end{pmatrix}$	$\begin{pmatrix} (73,73) & (17,67) & (30,66) \\ (90,139) & (128,31) & (63,43) \\ (77,60) & (77,60) & (28,35) \end{pmatrix}$
$\begin{pmatrix} 142 & 121 & 91 \\ 56 & 150 & 47 \\ 87 & 133 & 122 \end{pmatrix}$	$\begin{pmatrix} (122,130) & (29,147) & (58,14) \\ (136,131) & (103,10) & (124,64) \\ (65,105) & (135,20) & (129,3) \end{pmatrix}$	$\begin{pmatrix} (65,105) & (7,8) & (35,131) \\ (128,86) & (61,106) & (6,38) \\ (133,55) & (41,48) & (124,64) \end{pmatrix}$	$\begin{pmatrix} (19,127) & (6,113) & (89,70) \\ (50,107) & (105,136) & (144,98) \\ (134,25) & (118,120) & (56,133) \end{pmatrix}$	$\begin{pmatrix} (7,143) & (140,65) & (13,107) \\ (110,95) & (2,147) & (50,107) \\ (29,147) & (31,130) & (110,95) \end{pmatrix}$
$\begin{pmatrix} 133 & 130 & 71 \\ 155 & 97 & 145 \\ 125 & 54 & 105 \end{pmatrix}$	$\begin{pmatrix} (54,100) & (102,147) & (81,3) \\ (137,140) & (12,70) & (131,146) \\ (5,65) & (11,133) & (68,133) \end{pmatrix}$	$\begin{pmatrix} (34,75) & (154,140) & (94,67) \\ (122,9) & (14,116) & (152,117) \\ (153,90) & (100,78) & (154,17) \end{pmatrix}$	$\begin{pmatrix} (68,133) & (27,80) & (133,100) \\ (89,142) & (52,26) & (79,142) \\ (119,139) & (83,139) & (75,86) \end{pmatrix}$	$\begin{pmatrix} (103,149) & (137,140) & (84,26) \\ (14,41) & (94,90) & (31,119) \\ (7,69) & (1,47) & (132,87) \end{pmatrix}$
$\begin{pmatrix} 450 & 150 & 639 \\ 333 & 275 & 750 \\ 720 & 885 & 115 \end{pmatrix}$	$\begin{pmatrix} (659,254) & (445,871) & (90,416) \\ (141,493) & (894,883) & (600,224) \\ (196,102) & (873,105) & (240,550) \end{pmatrix}$	$\begin{pmatrix} (529,514) & (189,75) & (464,462) \\ (407,82) & (738,300) & (277,609) \\ (433,729) & (35,685) & (777,45) \end{pmatrix}$	$\begin{pmatrix} (894,58) & (570,585) & (550,755) \\ (856,491) & (636,417) & (86,339) \\ (455,680) & (455,680) & (521,568) \end{pmatrix}$	$\begin{pmatrix} (177,392) & (515,726) & (681,57) \\ (75,430) & (642,863) & (140,772) \\ (196,839) & (407,82) & (568,730) \end{pmatrix}$

Table 5.22: The experimental results of ECMF-ElGamal public key cryptosystem: decryption process.

$-L \star C_1$	$C_2 = C_2 - L \star C_1 = M$
$\begin{pmatrix} (16,135) & (58,16) & (133,43) \\ (124,70) & (94,119) & (29,48) \\ (8,59) & (3,82) & (2,58) \end{pmatrix}$	$\begin{pmatrix} (56,99) & (91,129) & (11,80) \\ (77,70) & (40,100) & (100,108) \\ (2,81) & (92,52) & (107,134) \end{pmatrix}$
$\begin{pmatrix} (98,135) & (98,135) & (77,60) \\ (141,119) & (85,59) & (26,133) \\ (73,76) & (98,135) & (147,47) \end{pmatrix}$	$\begin{pmatrix} (9,10) & (35,100) & (75,19) \\ (143,60) & (49,122) & (30,66) \\ (15,139) & (5,112) & (147,102) \end{pmatrix}$
$\begin{pmatrix} (65,46) & (7,143) & (35,20) \\ (128,65) & (61,45) & (6,113) \\ (133,96) & (41,103) & (124,87) \end{pmatrix}$	$\begin{pmatrix} (19,127) & (6,113) & (89,70) \\ (50,107) & (105,136) & (144,98) \\ (134,25) & (118,120) & (56,133) \end{pmatrix}$
$\begin{pmatrix} (34,82) & (154,17) & (94,90) \\ (122,148) & (14,41) & (152,40) \\ (153,67) & (100,79) & (154,140) \end{pmatrix}$	$\begin{pmatrix} (68,133) & (27,80) & (133,100) \\ (89,142) & (52,26) & (79,142) \\ (119,139) & (83,139) & (75,86) \end{pmatrix}$
$\begin{pmatrix} (529,427) & (189,866) & (464,479) \\ (407,859) & (738,641) & (277,332) \\ (433,212) & (35,256) & (777,896) \end{pmatrix}$	$\begin{pmatrix} (894,58) & (570,585) & (550,755) \\ (856,491) & (636,417) & (86,339) \\ (455,680) & (455,680) & (521,568) \end{pmatrix}$

5.12 Case Study of CESM graphic public key cryptosystem

Suppose EC is an elliptic curve defined by $EC : y^2 = x^3 + 3x + 2$ over a prime field F_{2017}. The set of all points satisfy the elliptic curve equation is given by:

$$EC(F_{2017}) = \{(0,986),(0,1031),(1,246),(1,1771),(2,4),(2,2013),...,O_E\}.$$

The $\#EC(F_{2017}) = 2027$ is a prime. A public matrix

$$D = \begin{pmatrix} 1000 & 170 & 550 & 413 & 17 & 1550 \\ 2000 & 2019 & 1100 & 1920 & 777 & 100 \\ 350 & 19 & 615 & 1300 & 217 & 220 \\ 100 & 160 & 1554 & 71 & 2011 & 1250 \\ 18 & 700 & 887 & 1700 & 87 & 130 \\ 1999 & 117 & 1450 & 91 & 555 & 3 \end{pmatrix} \in GL_6(F_{2027})$$

and

$$K = \begin{pmatrix} (5,1411) & (68,1849) & (2,4) & (11,1329) & (106,332) & (12,269) \\ (5,1411) & (965,934) & (108,641) & (77,715) & (45,1244) & (534,1400) \\ (1076,791) & (16,654) & (50,1166) & (20,142) & (212,230) & (909,846) \\ (1910,1417) & (320,1602) & (688,63) & (300,329) & (690,216) & (400,599) \\ (127,945) & (91,1760) & (100,1608) & (341,331) & (30,1162) & (24,1908) \\ (400,1418) & (555,1935) & (922,706) & (212,1787) & (28,681) & (18,1442) \end{pmatrix},$$

where K is a square matrix, its elements are elliptic curve points that are chosen randomly from $EC(F_{2017})$ as a public parameter.

Keys generation process.

Alice performs the following:

- She chooses her secret key $a = 533$ and she computes her public key

$$A = D^{533}\ (mod\ 2027) \equiv \begin{pmatrix} 1761 & 1995 & 1317 & 1040 & 386 & 1686 \\ 314 & 535 & 1150 & 430 & 89 & 1644 \\ 181 & 1803 & 134 & 1918 & 607 & 1326 \\ 1408 & 1768 & 183 & 1964 & 152 & 1258 \\ 1572 & 181 & 232 & 1342 & 1499 & 324 \\ 1943 & 56 & 652 & 1738 & 160 & 1504 \end{pmatrix},$$

so, her keys are $a = 533$ and $A = D^{533}$.

Encryption process.

Bob does the following steps:

- He chooses his private key $b = 771$ and computes

$$C_1 = D^{771}\ (mod\ 2027) \equiv \begin{pmatrix} 1450 & 1525 & 1224 & 492 & 1209 & 663 \\ 750 & 1391 & 1102 & 993 & 1317 & 1614 \\ 87 & 1301 & 1157 & 1826 & 1529 & 181 \\ 897 & 1016 & 1697 & 1806 & 476 & 842 \\ 1974 & 1208 & 1681 & 1757 & 962 & 350 \\ 38 & 70 & 1862 & 1344 & 1804 & 1038 \end{pmatrix}$$

and

$$A^{771}\ (mod\ 2027) \equiv \begin{pmatrix} 1215 & 246 & 152 & 1788 & 1708 & 167 \\ 1712 & 1516 & 622 & 666 & 429 & 866 \\ 1624 & 608 & 1850 & 1115 & 558 & 1915 \\ 1814 & 384 & 1655 & 1705 & 1122 & 1741 \\ 108 & 1854 & 922 & 740 & 903 & 934 \\ 1571 & 548 & 1707 & 657 & 650 & 187 \end{pmatrix}.$$

- He chooses randomly his plaintext

  $M(F_{2017}) = \{(60, 821), (60, 1196), (75, 1414), (75, 603), (350, 1689), (350, 328),$
  $(411, 1298), (411, 719), (625, 1155), (625, 862), (1500, 534), (1500, 1483)\},$

  as a subset of elliptic curve group $EC(F_{2017})$.

- He converts a plaintext $M(F_{2017})$ into

  $M^{\bullet}(F_{2017}) = \{(60, 821), (75, 1414), (350, 1689), (411, 1298), (625, 1155), (1500, 534)\}$

  by Theorem (4.4.1). The set $EC(F_{2017})$) corresponds to a complete elliptic curve graph.

- According to Corollary (4.4.1), $M^{\bullet}(F_{2017})$ corresponds to a CESM subgraph as seen in Figure 5.12.

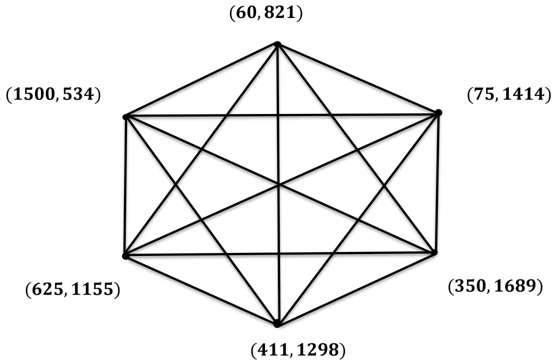

Figure 5.10: The CESM subgraph of $M^{\bullet}(F_{2017})$.

- Each elliptic point in a set $M^{\bullet}(F_{2017})$ has a code numbers given in Table (5.23).

Table 5.23: Code numbers of elliptic curve points in a set $M^{\bullet}(F_{2017})$.

Points in $M^{\bullet}(F_{2017})$	Code number
$(60, 821)$	60
$(75, 1414)$	75
$(350, 1689)$	350
$(411, 1298)$	411
$(625, 1155)$	625
$(1500, 534)$	1500

- Each edge of CESM subgraph has a weight which is computed as a distance between two connected vertices dependent on the coding number table. For instance, the weight w_{12} is computed by

$$w_{12} = \mid cod(v_1) - cod(v_2) \mid = \mid 60 - 75 \mid = 15.$$

The weights computations of all edges are given in Figure 5.12.

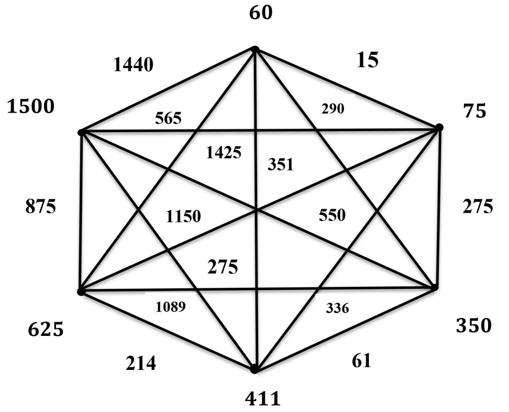

Figure 5.11: The weighted CESM subgraph of $M^{\bullet}(F_{2017})$.

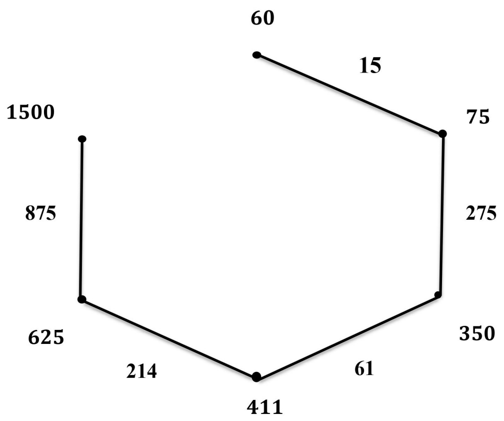

Figure 5.12: The MST graph of the weighted CESM subgraph.

The adjacent matrix of MST graph is

$$B_1 = \begin{pmatrix} 0 & 15 & 0 & 0 & 0 & 0 \\ 15 & 0 & 275 & 0 & 0 & 0 \\ 0 & 275 & 0 & 61 & 0 & 0 \\ 0 & 0 & 61 & 0 & 214 & 0 \\ 0 & 0 & 0 & 214 & 0 & 875 \\ 0 & 0 & 0 & 0 & 875 & 0 \end{pmatrix}.$$

- The modified matrix B_2 of the matrix B_1 is done through adding the code numbers of elliptic points in a set $M^{\bullet}(F_{2017})$ at the diagonal. So

$$B_2 = \begin{pmatrix} 60 & 15 & 0 & 0 & 0 & 0 \\ 15 & 75 & 275 & 0 & 0 & 0 \\ 0 & 275 & 350 & 61 & 0 & 0 \\ 0 & 0 & 61 & 411 & 214 & 0 \\ 0 & 0 & 0 & 214 & 625 & 875 \\ 0 & 0 & 0 & 0 & 875 & 1500 \end{pmatrix}.$$

- Multiplying the matrices A^{771} and B_2 to find a matrix B_3 as follows.

$$B_3 = A^{771} \times B_2 \ (mod\ 2027) \equiv \begin{pmatrix} 1591 & 1449 & 867 & 883 & 1008 & 1780 \\ 1813 & 299 & 235 & 101 & 847 & 73 \\ 1156 & 1015 & 970 & 1347 & 853 & 2011 \\ 1088 & 331 & 352 & 1967 & 1016 & 1406 \\ 1858 & 982 & 2026 & 253 & 1492 & 1965 \\ 1130 & 989 & 1751 & 423 & 1023 & 1964 \end{pmatrix}.$$

- He computes

$$C_2 = \begin{pmatrix} (1982,160) & (2004,675) & (783,165) & (1896,1569) & (1358,740) & (62,81) \\ (1557,131) & (1714,1877) & (1240,494) & (1829,349) & (1865,585) & (48,1883) \\ (1441,589) & (1324,674) & (1505,168) & (1677,1160) & (237,1873) & (1150,496) \\ (1108,1848) & (1874,4) & (1481,428) & (1022,1257) & (1857,2010) & (1133,532) \\ (58,1700) & (391,614) & (100,409) & (119,633) & (585,1559) & (973,48) \\ (990,1152) & (1872,1240) & (172,1290) & (450,629) & (1222,187) & (1322,576) \end{pmatrix}.$$

- He sends the ciphertext (C_1, C_2) to Alice.

Decryption process.

Alice performs the following steps:

- She computes $(C_1^{533})^{-1}$,

$$(C_1^{533})^{-1}\ (mod\ 2027) \equiv \begin{pmatrix} 1439 & 1243 & 969 & 148 & 1070 & 312 \\ 853 & 1794 & 715 & 1893 & 878 & 361 \\ 283 & 1727 & 1723 & 1921 & 1322 & 1517 \\ 307 & 1249 & 1730 & 1331 & 1704 & 773 \\ 1402 & 1924 & 1901 & 489 & 407 & 1279 \\ 1536 & 70 & 1218 & 1437 & 1856 & 1576 \end{pmatrix},$$

  where

$$(C_1)^{533} \equiv A^{771}\ (mod\ 2027).$$

- She computes a matrix B_3 from C_2 which its elements represent the ECDLPs, using the Baby step and Giant step algorithm [20].

- She computes a matrix

$$(C_1^{553})^{-1} \times B_3\ (mod\ 2027) \equiv \begin{pmatrix} 60 & 15 & 0 & 0 & 0 & 0 \\ 15 & 75 & 275 & 0 & 0 & 0 \\ 0 & 275 & 350 & 61 & 0 & 0 \\ 0 & 0 & 61 & 411 & 214 & 0 \\ 0 & 0 & 0 & 214 & 625 & 875 \\ 0 & 0 & 0 & 0 & 875 & 1500 \end{pmatrix},$$

  which is equal to a matrix B_2 The elements diagonal of the matrix B_2 represent the x-coordinates of the elliptic points. For instance, the first element 60 in a matrix B_2 gives two possibilities of elliptic points, $(60, 821)$, $(60, 1196)$. The computations for other elements can be seen in Table (5.24).

Table 5.24: The elliptic curve points that correspond to the diagonal elements of the matrix B_2.

Diagonal elements of the matrix B_2	Elliptic points
75	$(1414, 4), (75, 603)$
350	$(350, 1689), (350, 328)$
411	$(411, 1298), (411, 719)$
625	$(625, 1155), (625, 862)$
1500	$(1500, 534), (1500, 1483)$

Then the original plaintext

$$\begin{aligned} M(F_{2017}) = \{&(60, 821), (60, 1196), (75, 1414), (75, 603), (350, 1689), (350, 328), \\ &(411, 1298), (411, 719), (625, 1155), (625, 862), (1500, 534), (1500, 1483)\} \end{aligned}$$

Chapter 6

Conclusion and Future works

The aim of this work is to study the graph in cryptosystems. This study includes introducing new cryptosystems such as the matrix power ElGamal PKC, the undirected complete graph based PKC, the n-Dimension PKC, the n-UG PKC, the MPF-Diffie–Hellman key exchange, the MPF-ElGamal PKC, the hybrid GMPF-ElGamal PKC, and the soft graph PKC. These cryptosystems are more secure than the original cryptosystems because are dependent on new definitions of DLP that have high efficiency. Some results have also been obtained for the development of encryption for elliptic curves by representing the elliptic curve in a graph based on scalar multiplication. This representation is considered as an approach to encryption of more than one point at the same time and highly secure because of the used matrix power with the operations of the elliptic curve. These cryptosystems are the complete elliptic scalar multiplication graph PKC and the complete symmetric elliptic scalar multiplication digraph cryptosystem. Also, the study has defined a new graph of Edward's curve dependent on Edward's points, a new form of Edwards curve public key cryptography. It is called the Edwards curve graphic cryptosystem.

The mathematical complexity for some suggested cryptosystems with some related works is.

1. Although [7] employs fewer mathematical operations than the UCG public key cryptosystem, the UCG is superior because it is more secure because it is dependent on the power of the matrix, Eve cannot arrive at the inverse of the matrix (she cannot arrive at plaintext) in UCG PKC, whereas Eve can arrive at plaintext in [7] because the matrix K is public key.

2. The KR-EC cryptosystem is considered a faster public key algorithm in comparison with the EC-EPKC [30], since the encryption and decryption processes need only compute the scalar multiplication operation, while the EC-EPKC requires these processes to compute the scalar multiplication and

addition operations, so the EC-EPKC needs extra computations.

3. The CESM graph and CSESM digraph public key cryptosystem are considered faster than other algorithms like the EC-EPKC [30], because more than one point of the elliptic curve is encrypted at the same time, compared with other algorithms [30] that are used to encrypt only one point. So, these cryptosystems save time by encrypting many elliptic curve points compared with other algorithms.

For future work, researchers can apply these results to image encryption and character encryption. Some results can also be modified over the extension of Galois fields and can be applied to other curves. So, other types of graphs can be used in the suggested cryptosystems. They can also apply graph theory to digital signatures.

References

[1] A. A. Abd El-Latif and X. Niu. A hybrid chaotic system and cyclic elliptic curve for image encryption. *AEU-International Journal of Electronics and Communications*, 67(2):136–143, 2013.

[2] S. A. Abdul-Ghani, R. D. Abdul-Wahhab, and E. W. Abood. Securing text messages using graph theory and steganography. *Baghdad Science Journal*, 19(1):0189–0189, 2021.

[3] S. Agarwal and A. S. Uniyal. Prime weighted graph in cryptographic system for secure communication. *International Journal of Pure and Applied Mathematics*, 105(3):325–338, 2015.

[4] R. K. K. Ajeena. *Integer sub-decomposition (ISD) method for elliptic curve scalar multiplication*. PhD thesis, Universiti Sains Malaysia, 2015.

[5] S. G. Akl. How to encrypt a graph. *International Journal of Parallel, Emergent and Distributed Systems*, 35(6):668–681, 2020.

[6] M. Akram and S. Nawaz. Certain types of soft graphs. *Scientific Bulletin-University Politehnica of Bucharest, Series A*, 78:67–82, 2016.

[7] W. M. Al Etaiwi. Encryption algorithm using graph theory. *Journal of Scientific Research and Reports*, pages 2519–2527, 2014.

[8] F. Amounas. An innovative approach for enhancing the security of amazigh text using graph theory based ecc. 2016.

[9] P. Amudha, A. C. Sagayaraj, and A. S. Sheela. An application of graph theory in cryptography. *International Journal of Pure and Applied Mathematics*, 119(13):375–383, 2018.

[10] D. J. Bernstein and T. Lange. Inverted edwards coordinates. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pages 20–27. Springer, 2007.

[11] H. Bhapkar. Applications of planar graph to key cryptography. *International Journal of Pure and Applied Mathematics*, 120(8):89–97, 2018.

[12] D. M. Burton. *Introduction to modern abstract algebra.* Addison-wesley publishing company, 1967.

[13] D. M. Burton. *Elementary number theory.* Tata McGraw-Hill Education, 2006.

[14] G. Chartrand, L. Lesniak, and P. Zhang. *Graphs and digraphs*, volume 22. Chapman and Hall London, 1996.

[15] W. Diffie and M. Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[16] H. Edwards. A normal form for elliptic curves. *Bulletin of the American mathematical society*, 44(3):393–422, 2007.

[17] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[18] J. A. Green. The characters of the finite general linear groups. *Transactions of the American Mathematical Society*, 80(2):402–447, 1955.

[19] L. S. Hill. Cryptography in an algebraic alphabet. *The American Mathematical Monthly*, 36(6):306–312, 1929.

[20] J. Hoffstein, J. Pipher, J. H. Silverman, and J. H. Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[21] N. Jacobson. *Lectures in Abstract Algebra: III. Theory of Fields and Galois Theory*, volume 32. Springer Science & Business Media, 2012.

[22] D. Jao, S. D. Miller, and R. Venkatesan. Expander graphs based on grh with an application to elliptic curve cryptography. *Journal of Number Theory*, 129(6):1491–1504, 2009.

[23] B. Joseph and B. K. Thomas. A new cryptographic method irrespective of code order using graph theory. *Malaya Journal of Matematik (MJM)*, 9(1):470–473, 2021.

[24] T. W. Judson. *Abstract Algebra: Theory and Applications.* Number 00A05 JUD. 2013.

[25] T. A. Khaleel and A. A. Al-Shumam. A study of graph theory applications in it security. *Iraqi Journal of Science*, pages 2705–2714, 2020.

[26] M. Khan, R. Farooq, and J. Rada. Complex adjacency matrix and energy of digraphs. *Linear and Multilinear Algebra*, 65(11):2170–2186, 2017.

[27] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[28] N. Koblitz, A. Menezes, and S. Vanstone. The state of elliptic curve cryptography. *Designs, codes and cryptography*, 19(2):173–193, 2000.

[29] A. Liu and P. Ning. Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 245–256. IEEE, 2008.

[30] V. S. Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[31] L. Mittenthal. Sequencings and directed graphs with applications to cryptography. In *Sequences, Subsequences, and Consequences*, pages 70–81. Springer, 2007.

[32] D. Molodtsov. Soft set theory—first results. *Computers & Mathematics with Applications*, 37(4-5):19–31, 1999.

[33] P. Perera and G. Wijesiri. Encryption and decryption algorithms in symmetric key cryptography using graph theory. *Psychology and Education Journal*, 58(1):3420–3427, 2021.

[34] S. S. Ray. *Graph theory with algorithms and its applications: in applied science and technology.* Springer Science & Business Media, 2012.

[35] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[36] E. Sakalauskas. Enhanced matrix power function for cryptographic primitive construction. *Symmetry*, 10(2):43, 2018.

[37] E. Sakalauskas and K. Luksys. Matrix power s-box construction. *IACR Cryptol. ePrint Arch.*, 2007:214, 2007.

[38] E. Sakalauskas and K. Luksys. Matrix power function and its application to block cipher s-box construction. *Int. J. Inn. Comp., Inf. Contr*, 8(4):2655–2664, 2012.

[39] G. Samid. Denial cryptography based on graph theory, Nov. 23 2004. US Patent 6,823,068.

[40] R. Selvakumar and N. Gupta. Fundamental circuits and cut-sets used in cryptography. *Journal of Discrete Mathematical Sciences and Cryptography*, 15(4-5):287–301, 2012.

[41] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab. Nanoecc: Testing the limits of elliptic curve cryptography in sensor networks. In *European conference on Wireless Sensor Networks*, pages 305–320. Springer, 2008.

[42] V. Ustimenko. Cryptim: Graphs as tools for symmetric encryption. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pages 278–286. Springer, 2001.

[43] A. Weir. Sylow p-subgroups of the general linear group over finite fields of characteristic p. *Proceedings of the American Mathematical Society*, 6(3):454–464, 1955.

[44] R. J. Wilson. *Introduction to graph theory.* Pearson Education India, 1979.

[45] M. Yamuna, M. Gogia, A. Sikka, and M. J. H. Khan. Encryption using graph theory and linear algebra. *International Journal of Computer Application*, 5(2):102–107, 2012.