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

New Model on Elliptic Curves for Public Key Cryptosystems

A Dissertation

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

By

Batool Hatem Akar Hussein

Supervised by

Asst. Prof.Dr. Ruma Kareem K. Ajeena

2023 A.D. **1445 A.H.**

Dedication

To my father and mother in the heavens...

They are the sun that never leaves my heart.

To my loving partner...My rock

My sisters and brothers ...

With love.

Batool H.

Acknowledgements

In the name of Allah, the most beneficent, the most merciful. I offer my thanks and appreciation the supervisor of this dissertation **Asst. Prof .Dr. Ruma Kareem K. Ajeena** who did not skimp on me with any support or assistance....

I would like to record my thanks to professors 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 supporting and encouragement during the period of this work, who whenever, I seek help from them, find them in front of me, they provide me with every strength Finally I would like to thank all of the other friends that I developed over the years. I am a lucky person to have the friendships that I have.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table 1: The Mathematical Symbols

Symbol	**Description**
F_p	Prime field
$GL_n(F)$	General Linear Group over Filed
kP	Scalar multiplication operation
m	plaintext
mod	Modulo
$mod\ p$	Arithmetic modulo p
O_E	Infinity point on elliptic curve
P	Prime number
$/$	Division

Table 2: The Abbreviations

Abbreviations	**Definitions**
COA	Cuckoo Optimization Algorithm
DLP	Discrete Logarithm Problem
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECDLP	Elliptic Curve Discrete Logarithm Problem
EEPKC	Elliptic Curve ElGamal Public Key Cryptosystem
EPKC	ElGamal Public Key Cryptosystem
ECC	Elliptic Curve Cryptography
HC	Huff curve
P	Prime number
PSO	Particle Swarm Optimization
ST	Spanning Tree

Abstract

This dissertation discusses the models on elliptic curve over a prime field especially Huff's curve and the proposed $BRH_{\alpha,\beta}$ (Batool-Ruma-Huff) and it applications in cryptographic schemes. The proposed $BRH_{\alpha,\beta}$ curve is proved as smooth curve with certain condition .An affine form of the $BRH_{\alpha,\beta}$ curve is derivated. The arithmetic on the $BRH_{\alpha,\beta}$ curve is done through computing the point doubling and addition that are defined which is an unified .The $BRH_{\alpha,\beta}$ curve under the addition operation $BRH_{\alpha,\beta}$ is proved as a group over aprime field F_p.

Some cryptographic schemes are proposed using $BRH_{\alpha,\beta}$ curve defined over F_p. One of these application is Diffie-Hellman key exchange scheme. Another one is the ElGamal public key cryptosystem. As well as, the $BRH_{\alpha,\beta}$ curve is applied to give new version of the digital signature scheme.

On the other hand, $BRH_{\alpha,\beta}$ curve is employed with graph theory to proposed and modified new versions of the cryptosystems that are named by the $BRH_{\alpha,\beta}$ ElGamal graphic cryptosystem, $BRH_{\alpha,\beta}$ matrix ElGamal graphic cryptosystem, and generated graph for text encryption scheme based on $BRH_{\alpha,\beta}$ curve in three cases.

This work also proposes two encryption methods through employing the optimization algorithms on the $BRH_{\alpha,\beta}$ curve, defined over F_p. These algorithms are called $BRH_{\alpha,\beta}$ ElGamal public cryptosystem based on the Particle Swarm Optimization algorithm (PSO), and $BRH_{\alpha,\beta}$ ElGamal public cryptosystem using the Cuckoo Optimization Algorithm (COA).

New experiment results of each the proposed encryption algorithms have been presented, some algorithms are implemented using the Matlab, other ones are implemented using Python.

The security considerations on all proposed encryption algorithms are determined based on the hard mathematical problems that are deponded on them.

CHAPTER 1

GENERAL INTRODUCTION

1.1. Introduction

Cryptography is the science that depends on hard mathematical problems to encrypt and decrypt data.Cryptography protects, stores, and transmits sensitive information across insecure networks so that only specific individuals can decode such information.Cryptography is utilized by spies and has applications in communications (e.g., phone, fax, and e-mail), bank transactions, bank security, passwords, and online credit card transactions. 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 [26]. 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 [49].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 [19].Elliptic curves have been widely studied as a subject of almost pure mathematical interest. The study of elliptic curves could be of various areas: Algebra, Algebraic Geometry, Number Theory, and Diophantine problems, etc. [40] mentions in his book that

" It is possible to write endlessly on elliptic curves. (This is not a treat.)"

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 [1,41,54].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 [49].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 [43].Koblitz in 1987 proposed the elliptic curve ElGamal public key cryptosystem [38]. In 2000, Neal Koblitz introduced the state of ECC. He presented a survey about Elliptic Curve Cryptosystems [39].Elliptic Curve Cryptography (ECC) began to be employed for commercial applications.As a result, a significant amount of research has been devoted to analyzing the performance of various forms of Elliptic curves proposed in the mathematical literature, such as Weierstra cubics Hoffstein et al. [53],Jacobi intersections Billet and Joye [43], Hessian curves Bernstein et al. [10], or the more recent forms of elliptic curves due to Montgomery Montgomery [45], or Edwards Edwards [18]. In addition, a long-forgotten model of elliptic curves suggested by Huff's in 1948 was addressed in 2010 Joye et al. [33] . 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 [57]. Some researchers proposed cryptographic algorithms using paths in any graph [50] and others proposed encryption algorithms using directed graphs [44].Digital

Signature means to sign through some cryptographic operations, a computerized or digitized document, instead of handwritten signature or seal .The receiver of signature can verify the signature and that the document is not altered after being signed, thus ensuring the authenticity of information and the integrity of the document. Digital Signature [13] as a technique, signature receiver can verify whether the received signature indeed is signed by a legitimate signer, only signer can generate his own signature, signature file contains the information not to be used as the signatures of other documents and signer cannot deny his signature at any time. Digital Signature Algorithm (DSA) in general is composed of the signature algorithm and verification algorithm.The safety and security requirements of the signature will be higher and verification faster than signature, especially when dealing with online verification . It is possible to solve these hard optimization problems by inspirations from nature, since known that nature is a system of vast complexity and it always generates a near-optimum solution.The nature-inspired computational paradigms are nature-inspired met heuristics for search and optimization.The most prominent ones are particle swarm optimization, ant colony optimization, and immune algorithm.These include methods inspired by physical laws, chemical reaction, biological phenomena, social behaviors, and animal thinking. Met heuristics are a class of intelligent self-learning algorithms for finding near-optimum solutions to hard optimization problems.

1.2. Previous Studies

To study a Diophantine problem, a type of model elliptic curves was introduced by Huff [28]in 1948.In 2010, a development of Huff curve was proposed by Joye et al. [33] in a paper entitled "Huff's model for elliptic curves". In this paper, they presented fast explicit formulae for adding and doubling points on Huff curves.In 2011, Devigne and Joye [16]described the addition law for binary Huff curves. In the mean-time, Ciss and Sow [14] proposed a generalization of Huff curves and in the subsequent year, they presented Tate pairing computation on these generalized Huff curves. Gu et al. [24] also suggested efficient pairing computation on Huff curves in 2015.In 2017, Jafri and Islam [31]suggested an optimized architecture for unified binary Huff curves.In 2009, Jao, et al. [32], 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 [51], 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 2015, Agarwal and Uniyal [2], 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 [6], introduced an innovative approach

to enhance the security of Amazigh text using the elliptic curve cryptography based on graph theory.In 2015 Shubham Agarwal and Anand Singh Uniyal [2]came up with the idea of a prime weighted graph (PWG) as a way to make communication more secure (PWG).In 2021, Ruma Ajeena [3]used the connected sub-graphs of undirected simple graph to propose the soft graphic ISD (SG-ISD) method. In 2021, Karrar Aljamaly and Ruma Ajeena [5]presented a new public key cryptosystem based on undirected complete graph (UCG). And in [4], they defined a new graph based on the elliptic scalar multiplication over a prime field to design a new version of an asymmetric encryption scheme.In 2017 Waruhari, Philomena, and Lawrence Nderu [58] presented their study about designing the ECDSA and the possibility of implementation it in medical data encryption.In 2021, Balasubramanian Prabhu Kavin and Ganapathy Sannasi [35]suggested an improved EDSA to check the accuracy of the information saved in cloud databases.In2016,Chande, Manoj Kumar, and Cheng-Chi Lee [13] improve (ECDSA) scheme by using two random numbers for signature generation. This will reduce the probability of risk of exposure of secret key. Therefore, the improved scheme can enhance the security of the Junru's ECDSA.In2012,Yamuna M, Meenal Gogia, Ashish Sikka, Md. Jazib Hayat Khan [59]propose an encryption mechanism where nodes are organized in a Hamiltonian path. This method aims to use adjacency matrix as an additional parameter to encrypt and forward the data,use matrix properties for decryption. In 2016 Shankar, K., and P. Eswaran [52], presented method to the plaintext point is encrypted to produce an intermediate cipher. Then the

intermediate cipher is passed to the genetic functions crossover and mutation to produce the final cipher.In 2014,Devi, S. Pramela, and K. Sindhuja [15] propose a genetic algorithm based elliptic curve cryptography. Here the message (plaintext) is encoded as x-y point using elliptic curve. Then the key pair's private and public keys are calculated. Then using the above generated keys the plaintext point is encrypted to produce an intermediate cipher. Then the intermediate cipher is passed to the genetic functions crossover and mutation to produce the final cipher

1.3. The Problem Statement

Elliptic curve cryptography is the most widely employed class of asymmetric cryptography algorithm. However, it is exposed to attacks .The unified crypto systems such as Binary Edward, Hessian and Huff curves provide resistance against attacks. Furthermore, Huff curves are more secure than Edward and Hessian curves but require more computational resources. In this work, it is proposed to give new versions of Huff curve with few computational resources. In addition, other versions of public key cryptography are proposed on propose curve based on new graphs have been defined. These graphs are formed based on the scalar multiplication operation on this curve over a prime field.To increase the security of encryption algorithms while ensuring the speed of calculations, optimization algorithms were used for the purpose of finding private keys.

1.4. Objectives of the Dissertation

The aim of this dissertation is to propose new models of Elliptic curve it was represent propose new version of Huff curve and it is applied with the Diffie- Hellman and ElGamal Algorithms as well as with the Digital Signature Algorithms.This curve used with new versions of cryptosystems that use graph theory concepts ,matrices and optimization. Used in graph an undirected complete graph and spanning tree. The graphs are applied to give a new cryptosystem and used the optimization algorithm to obtain an optimization of the cryptosystem algorithms applied to the proposed curve.

1.5. Dissertation Outline

This study's structure is as follows: In addition to the **first chapter**, it includes:

Chapter 2. This comprises the fundamentals of finite fields, linear groups in general. Moreover, this chapter presents some fundamental graph theory concepts. Another section of this chapter contains an introduction to the study of cryptography as well as some cryptographic concepts and schemes. It concludes with Huff curves and optimization algorithm.

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

schemes and generated graph for text encryption algorithm.

Chapter 4. It introduces some definitions about optimization algorithm and proposes algorithm connected between optimization algorithm and ElGamal schemes.

Chapter 5. Includes some computational results on the proposed cryptosystems.

Chapter 6. Draws the conclusions and future works.

CHAPTER 2

MATHEMATICAL BACKGROUND

2.1. Introduction

It is very necessary to start our study by including and clarifying some of the mathematical definitions and concepts that we need. Therefore, in this chapter, we will explain some concepts some basic definitions, theorems, and examples of finite fields, the general linear group. 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 Huff curve defined over a prime field and optimization algorithm.

2.2. Some Basic Definition

In this section, the mathematical concepts related to the fields, especially the finite fields, the general linear group, are discussed as follows:

Definition 2.2.1. (Field) [30]
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: $a \cdot (b + c) = (a \cdot b) + (a \cdot c) \quad \forall a, b, c \in F$.

Definition 2.2.2. (The Characteristic of a Field) [34]
Let F be a field. The characteristic of F is the least positive integer p such that for every nonzero element $\alpha \in F$, we have $p\ \alpha = 0$. If no such p exists, we define the characteristic to be 0.

Definition 2.2.3. (Relatively Prime) [27] :
Two integers a and b, not both of which are zero, are said to be relatively prime whenever: $g\ c\ d\ (a, b) = 1$.

Definition 2.2.4. (The general linear group) [23]:
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 $\prod_{k=0}^{n-1}(p^n - p^k)$ [48].

Example 2.2.1. The general linear group of 2×2 matrices over F_2 is $GL_2(F_2)$

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

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

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

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) [56] : 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

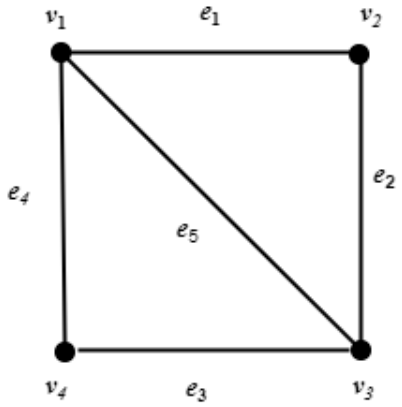


Figure 2.1: The simle graph

Definition 2.3.2. (Subgraph) [48]: 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) [48]: 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. (Order and Size of a Graph) [48]: Let $G = (V, E)$ be a graph. The order of G is defined by $|V| = n$ and $|E| = m$ is defined to be the size of G. In Figure 2.1 , $|V| = 4$ and $|E| = 5$.

Definition 2.3.5. (Simple graph) [48] A graph, that has neither self-loops nor parallel edges, is called a simple graph. A simple graph is given in Figure 2.1.

Definition 2.3.6. (Adjacency Matrix representations) [48]: 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 $A = [aij]$, 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 2.3.7. (Complete Graph) [56]: 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 2.2

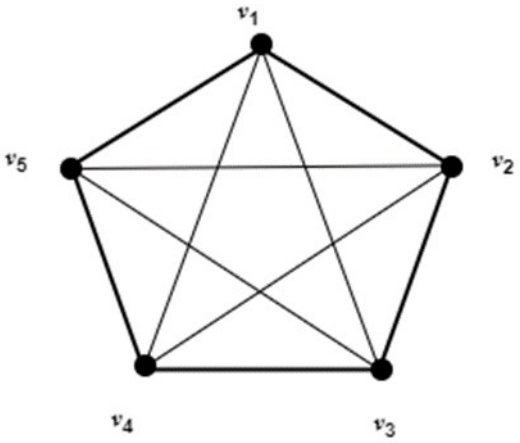


Figure 2.2: Complete Graph

Definition 2.3.8. (Weighted graph) [56]: 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 2.3.

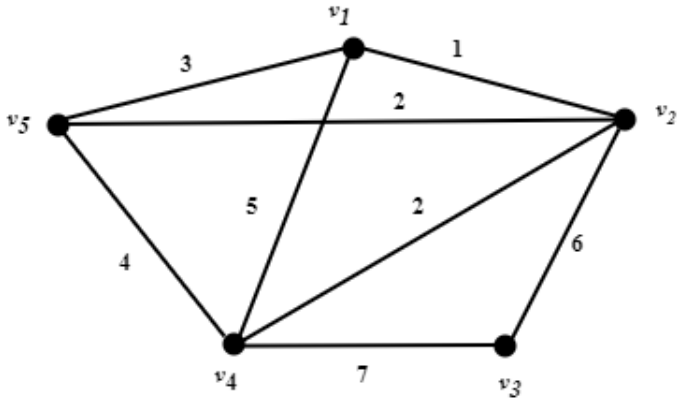


Figure 2.3: A weighted graph G.

Definition 2.3.9. (Tree) [48] A connected graph with no cycle is called a tree.

Definition 2.3.10. (Spanning Tree) [48]: 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.

2.4. Basic Concepts of Cryptography

Some basic concepts related to cryptography are discussed as follows.

Definition 2.4.1. (Cryptography) [25]:Is the design and analysis of mathematical techniques that enable secure communications in the presence of adversaries.

Definition 2.4.2. (Cryptosystem) [42]:A cryptographic system is specifically a set of methods (algorithms) for computing (implementing) the encryption and decryption

Definition 2.4.3. (Cryptanalysis) [55]: Is the study of analyzing cryptosystem in order to study the hidden aspects of the systems.

Definition 2.4.4. (Plaintext) [42]: The information which we want to protect from other people (attackers).

Definition 2.4.5. (Security) [42]:It means that the difficulty to know the information which transferred over the channel easily.

Definition 2.4.6. (Symmetric Key Cryptosystem) [27]: 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 2.4.7. (Asymmetric Key Cryptosystem) [27]: 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. Whereas, another key is kept secret which is called a private key.

2.5. Introduction to Elliptic Curves Cryptography (ECC)

In 1985 by Victor Miller [43] and in 1987 by Neal Koblitz [38] the ECC was discovered separately . 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

Some important definitions, theorems and examples related to the elliptic curves over finite fields are explained as follows.

Definition 2.5.1. (The Elliptic Curves Over Fp) [27]: Let F_p be a prime filed with characteristic $\neq$2,3, the equation of Weierstrass is simplified by:

$$y^2 \equiv x^3 + ax + b \ (mod\ p) \tag{2.1}$$

where a and b are coefficients in F_p and the discriminant of EC is computed $\triangle \equiv 4a^3 + 27b^2(modp) \neq 0$.

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_{41} be a prime field. Suppose EC is an elliptic curve defined by:

$$EC : y^2 \equiv x^3 + 7x + 5 \ (mod\ 41)$$

The discriminant of is :

$$\triangle \equiv (4(7)^3 + 27(5)^2) = 38 \neq 0 \ (mod\ 41)$$

The points in $EC(F_{41})$ are determined by:

$EC(F_{41}) = \{(0, 28), (0, 13), (5, 1), (5, 40), (8, 32), (8, 9), (9, 10), (9, 31),$
$(10, 38), (10, 3), (14, 10), (14, 31), (15, 0), (16, 20), (16, 21), (18, 10), (18, 31),$
$(23, 19), (23, 22), (24, 37), (24, 4), (25, 15), (25, 26), (26, 16), (26, 25), (27, 19),$

$(27, 22), (30, 14), (30, 27), (31, 1), (31, 40), (32, 19), (32, 22), (34, 33), (34, 8),$
$(36, 38), (36, 3), (37, 35), (37, 6), (38, 11), (38, 30), O_E\}$

Definition 2.5.2. (Point Addition on EC(mod p)) [27]: 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

$$\begin{cases} x_3 \equiv \left(\lambda^2 - x_1 - x_2\right) (\mathrm{mod} p) \\ y_3 \equiv \left(\lambda \left(-x_1 - x_3\right) - y_1\right) (\mathrm{mod} p) \end{cases} \tag{2.2}$$

where $\lambda \equiv \left[\dfrac{y_2 - y_1}{x_2 - x_1}\right] (\mathrm{mod} p)$.

Definition 2.5.3. (Point Doubling on EC(mod p)) [27]: 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.

$$\begin{cases} x_2 \equiv \left(\lambda^2 - 2x_1\right) (\mathrm{mod} p) \\ y_2 \equiv \left(\lambda \left(-x_1 - x_2\right) - y_1\right) (\mathrm{mod} p) \end{cases} \tag{2.3}$$

where $\lambda \equiv \left[\dfrac{3x_1^2 + a}{2y_1}\right] (\mathrm{mod} p)$, 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 $P_1 = (14, 10)$ and $P_2 = (24, 4) \in E(F_{41})$. Using the relation in Equation2.2, the

computation of

$$P_1 + P_2 = (14, 10) + (24, 4) = (5, 1)$$

Which is in $E(F_{41})$. Whereas, based on the relation in Equation2.3, the computation of $2P$ can be done by:

$$2P = 2(14, 10) = (14, 31) \in E(F_{41})$$

Definition 2.5.4. (Elliptic Curve Scalar Multiplication) [27]: 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 define:

$$\underbrace{KP = P + P + \ldots + P}_{\text{ktimes}} \tag{2.4}$$

Definition 2.5.5. (Elliptic Curve Discrete Logarithm Problem) [27]: 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:

2.5.2.1 Elliptic Diffie–Hellman Key Exchange(ECDH)

[27]:

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 $K = n_A Q_B = (n_A n_B)P = n_B Q_A$.

2.5.2.2 Elliptic ElGamal Public Key Cryptosystem (EEPKC)

[27]

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

$$C_2 = M + KQ_A$$

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 Huff Curve

In this section, we will give the details of the models of Huff curves, especially their group structure and related formulae with associated computational costs. In [28] ,Huff investigated the Huff elliptic curves over rational fields Q in 1948. Joye et al. [33] [46] improved these curves to the finite field F with $char(F) \neq 2$ that are given by:

Definition 2.6.1. (Huff Curve) [46]: Let F be a field with a characteristic different than 2.Huff curve over F is defined as follows:

$$HC : ax(y^2 - 1) = by(x^2 - 1) \tag{2.5}$$

Where $a, b \in F$ and $a^2 \neq b^2$ The number of points on the HC is denoted by $\#HC(F)$ which is the number of the solutions $(x, y) \in F \times F$ plus the point O_H which is a point $(0, 0)$.

Example 2.6.1. Let F_{31} be a prime field. Suppose HC is an elliptic curve defined by:

$$HC : 2x(y^2 - 1) = 3y(x^2 - 1)(mod\ 31)$$

The discriminant of is $\triangle = (22 - 32) = 26(mod 31) \neq 0$ The points in $HC(F_{31})$ are determined by:

$HC(F_{31}) = \{(1,1),(1,30),(3,8),(3,27),(4,11),(4,14),(5,13),(5,19),$
$(6,13),(6,19),(8,17),(8,20),(10,8),(10,27),(12,16),(12,29),(13,2),(13,15),$
$(18,16),(18,29),(19,2),(19,15),(21,4),(21,23),(23,11),(23,14),(25,12),$
$(25,18),(26,12),(26,18),(27,17),(27,20),(28,4),(28,23),(30,1),(30,30),O_H\}$

Definition 2.6.2. (Point Addition on HC (F)) [33]: Suppose $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are two points lie on an Huff curve HC defined over F. Adding the two points P and Q gives a third point $R = (x_3, y_3)$ which also lies on HC, by:

1. ***Dedicated addition:*** Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points on HC and $P \neq Q$. The addition $P + Q$ is computed by:

$$P + Q \equiv \left(\frac{(x_1 - x_2)(y_1 + y_2)}{(1 - x_1 x_2)(y_1 - y_2)}, \frac{(y_1 - y_2)(x_1 + x_2)}{(x_1 - x_2)(1 - y_1 y_2)} \right) \tag{2.6}$$

2. ***Unified addition:*** The addition of the points P and Q can be computed by alternative formula that is given by

$$P + Q \equiv \left(\frac{(x_1 + x_2)(1 + y_1 y_2)}{(1 + x_1 x_2)(1 - y_1 y_2)}, \frac{(y_2 + y_1)(1 + x_1 x_2)}{(1 - x_1 x_2)(1 + y_1 y_2)} \right) \tag{2.7}$$

Definition 2.6.3. (Point Doubling on HC (F)) [33] The doubling point $2P$ of the point P is computed by:

$$2P \equiv \left(\frac{2x_1 \left(1 + y_1^2\right)}{\left(1 - y_1^2\right)\left(1 + x_1^2\right)}, \frac{2y_1^2 \left(1 + x_1^2\right)}{\left(1 - x_1^2\right)\left(1 + y_1^2\right)} \right) \tag{2.8}$$

Example 2.6.2. **(Points Addition and Doubling on $HC(F_p)$)** With the same parameters in the example (2.6.1), suppose the points $P = (4, 14)$

and $Q = (10, 8) \in HC(F_{31})$. Using the relation in equation (2.6), the computation of

$$P + Q = (4, 14) + (10, 8) = (26, 18) \in HC(F_{31})$$

Which is in $HC(F_{31})$. Whereas, based on the relation in equation (2.8), the computation of $2P$ can be done by:

$$2P = 2(4, 14) = (18, 29) \in HC(F_{31})$$

Theorem 2.6.1. *If (x, y) is a rational point on 2.5 and $x \neq 0, y \neq 0$, then $(-x, -y), (\pm x, \mp\frac{1}{y}), (\pm\frac{1}{x}, \mp y)$, and $(\pm\frac{1}{x}, \pm\frac{1}{y})$ all are rational points on [46]*

2.7. Optimization

Artificial Intelligence (AI) includes all types of intelligence provided by machines. The main topics of AI is the study of unconventional optimization techniques. It is considered Computational Intelligence (CI) is a branch of AI and is the basic principle of all optimization algorithms known as metheuristic algorithms, they are a trial-and-error method of production that provides an acceptable solution to a complex problem in a reasonable practical time [9].Optimization is the process of maximizing or minimizing a desired objective function while satisfying the prevailing constraints. The companies were based on the concept of improvement to strive for excellence. Solutions to their problems have mostly been based

on judgment and experience. However, competition requires that the solutions be perfect and not merely feasible. Often, improving the design process saves money for producing companies [12, 60]. Optimization problems have three basic elements. The first is the goal, may be a single scalar quantity, or an objective function, that must be minimized or maximized, increased or decreased. The second is a set of variables whose values affect the value of the goal, and can be manipulated to improve the goal. The third is a set of constraints, which are constraints on the values that variables can take. [8, 11, 60].

CHAPTER 3

A NEW MODELS OF HUFF CURVE FORM ELLIPTIC CURVE FOR ENCRYPTION SCHEMES

3.1. Introduction

The Diophantine problem has been studied in 1948. Huff considered the distance of subsets of a set S of the plane R^2 such that for all $s_1, s_2 \in S$, the distance between s_1 and s_2 is a rational number [46]. If $a, b \in Q$, then$S$ contains four points $(0, \pm a)$ and $(0, \pm b)$ on the y-axis, and (x, 0) on the x-axis for some $x \in Q$. The point $(x, 0)$ must satisfy the Equation $x^2+b^2 = v^2$ with $u, v \in Q$. The homogeneous Equation is of the form $x^2+a^2z^2 = u^2$ and $x^2 + b^2z^2 = v^2$. Later Huff and his student [46] provided the examples of curves that had positive rank over Q. The curve is equivalent the Equation

$$ax(y^2 - 1) = by(x^2 - 1) \tag{3.1}$$

where $a, b \in Q$. It is clearly seen that Equation (3.1) over any finite field K of odd characteristic defines an elliptic curve if $a^2 \neq b^2$ and $a, b \neq 0$. The Huff curve Equation which is defined as a similar with the other curve is presented but it differs in the speed of calculating the addition and multiplication. In this chapter,especilly in section (3.2) a study about a new curve of the Huff curve is proposed which is called Batool, Ruma and Huff (BRH) curve. The addition for two points as well as a doubling ponit Formulas are presented.The BRH curve is proved as a smooth curve.The identity and inverse points are determined.The BRH curve is used for cryptographic application such as $DHKE$ and ElGamal public key cryptosystem.The BRH curve is also used with in digital signature scheme and used BRH curve with graph theory cryptosystems.An alternative version was provided text encryption algorithm based on BRH

curve by using number theory, matrices, and graph theory.

3.2. The Proposed *BRH* Curve

In this section,a new version of Huff's curve is presented which is called BRH curve. An affine form of the BRH curve is proved mathematically as a smooth curve. The addition points on the BRH curve are proved as well as the doubling point on the BRH curve. The group law on the BRH curve is discussed. The starting will be with the definition of BRH.

Definition 3.2.1. Let F be a field with a characteristic different than 2 and $\alpha, \beta \in F$.The $BRH_{\alpha,\beta}$ curve over F is defined by

$$BRH_{\alpha,\beta} : x(\alpha y^2 - 1) = \beta y(x^2 - 1). \tag{3.2}$$

$BRH_{\alpha,\beta}$ is a non-singular if and only if the condition $\beta\alpha(\beta^2 - \alpha) \neq 0$.

Shown in a Figure (3.1) and (3.2)

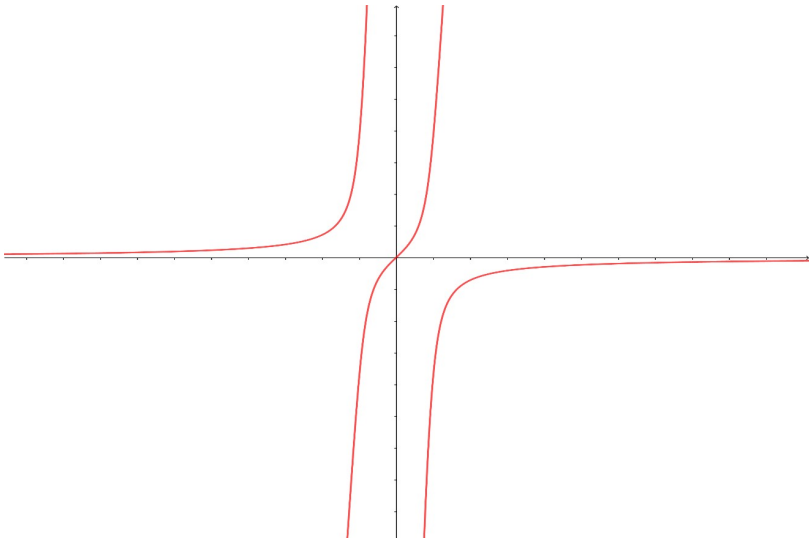


Figure 3.1: $BRH_{\alpha,\beta} : x(\alpha y^2 - 1) = \beta y(x^2 - 1)$.

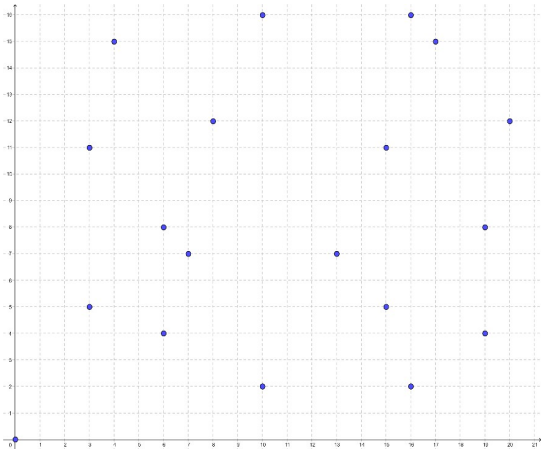


Figure 3.2: $HBR_{5,7} : x(5y^2 - 1) = 7y(x^2 - 1)(mod\, 23)$.

Proposition 3.2.1. Let F be a field with char $(F) \neq 2$ and $\alpha, \beta \in F$. The affine form of $BRH_{\alpha,\beta}$ is defined as given in Equation (3.2) with $\beta\alpha(\beta^2 - \alpha) \neq 0$ is smooth.

Proof. Let $P = (x, y)$ be a point lies on $BRH_{\alpha,\beta}$ curve .This point verifies $BRH_{\alpha,\beta}(x, y) = 0$.

So,

$$\frac{dBRH_{\alpha,\beta}(x,y)}{dx}0$$

and

$$\frac{dBRH_{\alpha,\beta}(x,y)}{dy} = 0$$

Where $BRH_{\alpha,\beta}$ carve is defined in Equation (3.2) then

1. $\alpha y^2 - 1 - 2\beta yx = 0$
2. $2\alpha xy - \beta x^2 + \beta = 0$

If $x = 0$, so $y = 0$ and a point $(0, 0)$ is not a solution of the previous condition. Multiplying (2) by y and using Equation (3.2) yields that

$x(\alpha y^2 + 1) = 0$ thus $y^2 = -1 \setminus \alpha$ and by (1),then $\beta xy + 1 = 0$. Similarly, $(\alpha \setminus \beta)xy + 1 = 0$. Combining these last two Equations given $\alpha = \beta^2$ which contradicts to the hypothesis. □

3.2.1 Affine Form of The $BRH_{\alpha,\beta}$ Curve

Let $P = (x_1, y_1)$, $Q = (x_2, y_2)$ and $R = (x_3, y_3)$ are a points on $BRH_{\alpha,\beta}$. The point R is the sum point of P and Q. The slope of λ is computed by $\lambda = (y_2 - y_1) \setminus (x_2 - x_1)$. Thus, the secant going the line Equation y can be written by $y = \lambda x + \mu$, where $\mu = y_1 - \lambda x_1$. Substituting in Equation (3.2) gives

$$\alpha x(\lambda x + \mu)^2 - x - \beta x^2(\lambda x + \mu) - \beta(\lambda x + \mu) = 0 \tag{3.3}$$

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

$A = \mu(2\alpha\lambda - \beta)$and $B = \lambda(\alpha\lambda - \beta)$.

Suppose

$$x_1 + x_2 + x_3 = -\frac{A}{B}.$$

Namely

$$x_1 + x_2 + x_3 = -\frac{\mu(2\alpha\lambda - \beta)}{\lambda(\alpha\lambda - \beta}.$$

Hence

$$x_1 + x_2 + x_3 = -\frac{(y_1x_2 - y_2x_1)(2\alpha(y_2 - y_1) - \beta(x_2 - x_1))}{(y_2 - y_1)(\alpha(y_2 - y_1) - \beta(x_2 - x_1))}$$

Noting that

$$\begin{aligned}
&(\alpha(y_2-y_1)-\beta(x_2-x_1))\,y_1y_2(x_2+x_1)\\
&=(x_2-\beta y_2)\,y_1-(x_1-\beta y_1)\,y_2+\alpha y_1y_2(y_2x_1-x_2y_1)\\
&=(x_2y_1-x_1y_2)+\alpha y_1y_2(y_2x_1-y_1x_2)\\
&=(x_1y_2-x_2y_1)(\alpha y_1y_2-1)\\
&=\left(\alpha(y_2x_1+x_1y_2-x_2y_1-x_1y_1)-\beta x_2^2+\beta x_1^2\right)y_1y_2
\end{aligned}$$

This lands to

$$\begin{aligned}
-x_3&=x_1+x_2-\frac{\alpha(x_1+x_2)\,y_1y_2}{\alpha y_1y_2-1}+\frac{(\alpha(y_2-y_1)-\beta(x_2-x_1))(x_2+x_1)\,y_1y_2}{(y_1-y_2)(\alpha y_1y_2-1)}\\
&=\frac{x_1y_1-x_2y_2}{y_1-y_2}-\frac{\alpha(x_1+x_2)\,y_1y_2}{\alpha y_1y_2-1}.
\end{aligned}$$

From

$$\begin{aligned}
&(y_1-y_2)(\alpha x_1x_2(y_1+y_2)+\beta(x_1+x_2))\\
&=\left(\alpha x_1y_1^2+\beta y_1\right)x_2-\left(\alpha x_2y_2^2-\beta y_2\right)x_1+\beta y_1x_1-\beta x_2y_2\\
&=\left(\beta y_1x_1^2+x_1\right)x_2-\left(\beta y_2x_2^2+x_2\right)x_1-\beta(x_1y_1-x_2y_2)\\
&=\beta x_1x_2(x_1y_1-x_2y_2)+\beta(x_1y_1-x_2y_2)\\
&=\beta(x_1y_1-x_2y_2)(x_1x_2+1).
\end{aligned}$$

This gives

$$\frac{x_1y_1-x_2y_2}{y_1-y_2}=\frac{\alpha x_1x_2(y_1+y_2)+\beta(x_1+x_2)}{\beta(x_1x_2+1)}.$$

Then

$$
\begin{aligned}
-x_3 &= \frac{\alpha x_1 x_2 (y_1 + y_2) + \beta (x_1 + x_2)}{\beta (x_1 x_2 + 1)} - \frac{\alpha (x_2 + x_1) y_1 y_2}{\alpha y_1 y_2 - 1} \\
&= \frac{(\alpha x_1 x_2 (y_1 + y_2) - (x_1 + x_2)) (\alpha y_1 y_2 - 1) - \alpha (x_2 + x_1) y_1 y_2 \beta (x_1 x_2 + 1)}{\beta (x_1 x_2 + 1) (\alpha y_1 y_2 - 1)} \\
x_3 &= -\frac{(x_1 + x_2) (\alpha y_1 y_2 + 1)}{(x_1 x_2 + 1) (1 - \alpha y_1 y_2)}.
\end{aligned}
$$

In similar way

$$
y_3 = -\frac{(y_1 + y_2) (x_1 x_2 + 1)}{(1 - x_1 x_2) (\alpha y_1 y_2 + 1)}
$$

So that

$$
R = (-x_3, -y_3) = \left(\frac{(x_1 + x_2) (\alpha y_1 y_2 + 1)}{(x_1 x_2 + 1) (1 - \alpha y_1 y_2)}, \frac{(y_1 + y_2) (x_1 x_2 + 1)}{(1 - x_1 x_2) (\alpha y_1 y_2 + 1)} \right)
$$

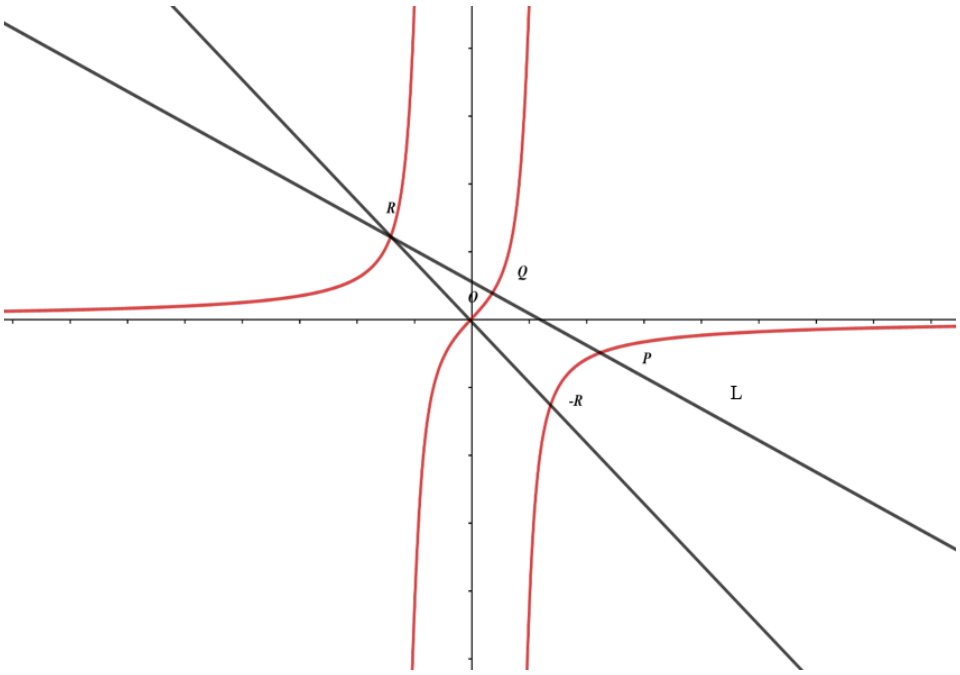


Figure 3.3: The addition $BRH_{\alpha,\beta}$ curve over R.

The most natural way to describe the "addition law" on $BRH_{\alpha,\beta}$ curves is to use geometry.

Let P and Q be two points on an $BRH_{\alpha,\beta}$ curve , as illustrated in Figure (3.3). start by drawing the line L through P and Q. This line L intersects $BRH_{\alpha,\beta}$ at three points, namely P, Q, and one other point $R = -(P + Q)$.The reflection of this intersection point about(0,0).gives the sum of$P$ and Q. This rule is verified by The point $-R$ by the Formula (3.4) gives $P + Q$. this process is nothing like ordinary addition.

Definition 3.2.2. (Adding points on $BRH_{\alpha,\beta}$ Curve): Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are two point lie on $BRH_{\alpha,\beta}$ Curve defined over a prime field F_p.Then the sum point $P + Q = R$ is defined by

$$R = (x_3, y_3) = \left(\frac{(x_1 + x_2)(\alpha y_1 y_2 + 1)}{(x_1 x_2 + 1)(1 - \alpha y_1 y_2)}, \frac{(y_1 + y_2)(x_1 x_2 + 1)}{(1 - x_1 x_2)(\alpha y_1 y_2 + 1)} \right) \quad (3.4)$$

The sum point R is represented geometrically in Figure (3.3)

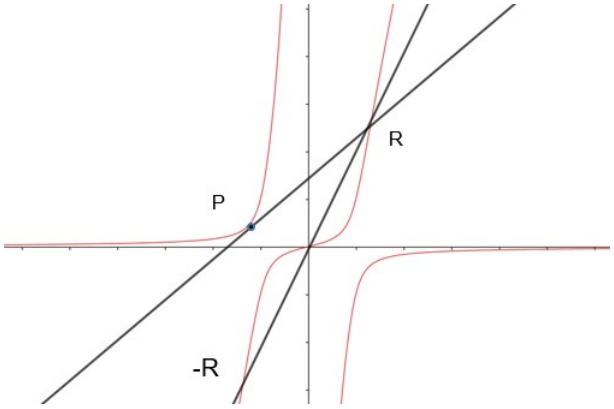


Figure 3.4: The doubling point $BRH_{\alpha,\beta}$ curve over R.

Definition 3.2.3. (Doubling point on $BRH_{\alpha,\beta}$ Curve): Let $P = (x_1, y_1)$ be a point lies on $BRH_{\alpha,\beta}$ Curve defined over F_p.Then a doubling point

$2P = (x_3, y_3)$ of P is defined by

$$2P = (x_3, y_3) = \left(\frac{2x_1 \left(\alpha y_1^2 + 1\right)}{\left(x_1^2 + 1\right)\left(1 - \alpha y_1^2\right)}, \frac{2y_1 \left(x_1^2 + 1\right)}{\left(1 - x_1^2\right)\left(\alpha y_1^2 + 1\right)} \right) \tag{3.5}$$

The doubling point also can be represented geometrically as shown in Figure 3.4

Remark: In the addition Formula that is Equation (3.4) $x_1 = x_2$ and $y_1 = y_2$ then ,it is easy to get the doubling Formula of any point lies on the $BRH_{\alpha,\beta}$ curve.So the Formulas in Equations (3.4) and (3.4) respecting to the $BRH_{\alpha,\beta}$ curve is an unified.

Proposition 3.2.2. Suppose $BRH_{\alpha,\beta}$ is a over a field F is defined as given in Equation (3.2) with $\beta\alpha\left(\beta^2\ \alpha\right) \neq 0$ and $x \neq 0, y \neq 0$.Let $P = (x, y)$ be a point lies on $BRH_{\alpha,\beta}$ (x, y) curve.Then $(-x, -y)$is also apoint lies on $BRH_{\alpha,\beta}$ curve.

Proof. Let $P = (x, y)$ be lies on $BRH_{\alpha,\beta}$ curve ,it needs to demonstrate that $-P$ is also lies on $BRH_{\alpha,\beta}$ curve. subsituting apoint $(-x, -y)$ in the $BRH_{\alpha,\beta}$ curve results

$$\begin{aligned} -x\left(\alpha(-y)^2 - 1\right) &= -\beta y\left((-x)^2 - 1\right) \\ -x\left(\alpha y^2 - 1\right) &= -\beta y\left(x^2 - 1\right) \\ x\left(\alpha y^2 - 1\right) &= \beta y\left(x^2 - 1\right) \end{aligned}$$

Thus, $(-x, -y)$ is also a point on $BRH_{\alpha,\beta}$ curve. □

Proposition 3.2.3. ($BRH_{\alpha,\beta}$ Curve Addition Algorithm). Let $BRH_{\alpha,\beta}$ be a curve as given in Equation (3.2) with $\beta\alpha\left(\beta^2 - \alpha\right) \neq 0$ and the points

P, Q and $O = (0, 0)$lie on $\mathrm{BRH}_{\alpha,\beta}$. The point O is a neutral point. Then, $\mathrm{BRH}_{\alpha,\beta}$ has the following qualities.

1. If $P = O$, then $P + Q = Q$.

2. Otherwise, if $Q = O$, then $P + Q = P$.

3. Otherwise,let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$.

4. If $-x_1 = x_2$ and $-y_1 = y_2$, then $P + Q = O$.

5. Otherwise, let
   $x_3 = -\dfrac{(x_1 + x_2)(\alpha y_1 y_2 + 1)}{(x_1 x_2 + 1)(1 - \alpha y_1 y_2)}$,
   and $y_3 = -\dfrac{(y_1 + y_2)(x_1 x_2 + 1)}{(1 - x_1 x_2)(\alpha y_1 y_2 + 1)}$, then $\mathrm{P} + Q = (-x_3, -y_3)$.

Proof. For (1), P is the neutral point (0,0), then the line through P and Q intersects $BRH_{\alpha,\beta}$ with the of 3, as P, Q and $-Q$. To obtain $P + Q$, one must take the inverse of the third point of the intersection. Thus, $-(-Q) = Q$. The similar proof follows for (2).

Part (4) is also obtained.

If $P = (x_1, y_1)$ and $Q = (x_2, y_2) = (-x_1, -y_1)$ then the third point of intersection of P and Q is O.The inverse of O is $-O = O$.

To prove (5), it is necessary to take algebraic step. If points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are two distinct points on $BRH_{\alpha,\beta}$ and neither of them equal to O, then the line through points P and Q has the slope λ. The line equation could be written as $y = \lambda x + \beta$, where $\beta = y_1 + \lambda x_1$. Substituting the line Equation in $BRH_{\alpha,\beta}$ gives us the Equation (3.3). It is clear that x_1 and x_2 are two roots of the above cubic Equation; thus, it could be written that

$$(x - x_1)(x - x_2)(x - x_3) = x^3 + (-x_1 - x_2 - x_3)x^2 + (xx_2 + x_3x_2 + x_1x_3)x - x_1x_2x_3$$

. □

Proposition 3.2.4. Let $BRH_{\alpha,\beta}$ be a curve defined over F_p. Then are three distinct points namely P, Q, and R lie on it.The line L as show in Figure(3.3) intersects the$BRH_{\alpha,\beta}$ curve in these points. Then, the associative law is satisfied on P, Q and R which is an equivalent to $O = (0, 0)$ point.

Proof. It requires to prove the points $P, Q$and R satisfy

$$P + (Q + R) = (P + Q) + R$$

on $BRH_{\alpha,\beta}$ curve defind over F_p.
Then starting will be with $Q + R$.

for x-coordinates first

$$Q + R = \frac{(x_2 + x_3)(\alpha y_2 y_3 + 1)}{(x_2 x_3 + 1)(1 - \alpha y_2 y_3)}$$

So,using the Formula in Equation (3.4) gives

$$P = \frac{(x_3 + x_2)(\alpha y_3 y_2 + 1)}{(x_3 x_2 + 1)(1 - \alpha y_3 y_2)}$$

Then

$$P + (Q + R) = -\frac{(x_3 + x_2)(\alpha y_3 y_2 + 1)}{(x_3 x_2 + 1)(1 - \alpha y_3 y_2)} + \frac{(x_3 + x_2)(\alpha y_3 y_2 + 1)}{(x_3 x_2 + 1)(1 - \alpha y_3 y_2)} = 0$$

It follows that,

$$(P + Q) + R = -\frac{(x_1 + x_2)(\alpha y_1 y_2 + 1)}{(x_1 x_2 + 1)(1 - \alpha y_1 y_2)} + \frac{(x_1 + x_2)(\alpha y_1 y_2 + 1)}{(x_1 x_2 + 1)(1 - \alpha y_1 y_2)} = 0$$

For y-coordinates, the following addition holds,

$$P + (Q + R) = -\frac{(y_2 + y_3)(x_2 x_3 + 1)}{(1 - x_2 x_3)(\alpha y_2 y_3 + 1)} + \frac{(y_2 + y_3)(x_2 x_3 + 1)}{(1 - x_2 x_3)(\alpha y_2 y_3 + 1)} = 0$$

And

$$(\mathrm{P} + \mathrm{Q}) + \mathrm{R} = -\frac{(y_1 + y_2)(x_1 x_2 + 1)}{(1 - x_1 x_2)(\alpha y_1 y_2 + 1)} + \frac{(y_1 + y_2)(x_1 x_2 + 1)}{(1 - x_1 x_2)(\alpha y_1 y_2 + 1)} = 0$$

In both sides the addition gives O.The x, y-coordinates respectively are equal to zero.Namely $(P+Q)+R = P+(Q+R) = O = (0,0)$.Therefore,the associative law is satisfied. □

Proposition 3.2.5. Suppose $BRH_{\alpha,\beta}$ is curve defined over afield F such that $char(F) \neq 2$.Then the addition law following conditions are hold.

1. Identity. $P + O = O + P = P$, where $O = (0,0)$ for all O and $P \in BRH_{\alpha,\beta}(F)$.

2. Negatives. If $P = (x,y) \in BRH_{\alpha,\beta}(F)$, then $(x,y) + (-x,-y) = O$. The point $(-x,-y)$ is represented by $-P$ and is known as the negative of P; notice that $-P$ is indeed point in $BRH_{\alpha,\beta}(F)$.

3. Associative. $(P+Q)+R = P+(Q+R)$,where P, Q and$R \in BRH_{\alpha,\beta}(F)$.

4. Commutative.$P+Q = Q+P$ for all Q and $P \in BRH_{\alpha,\beta}(F)$.

5. Point addition. Let $P = (x_1, y_1) \in BRH_{\alpha,\beta}(F)$ and $Q = (x_2, y_2) \in BRH_{\alpha,\beta}$ (F). Then $P+Q = (x_3, y_3)$, where

$$(x_3, y_3) = \left(\frac{(x_1+x_2)(\alpha y_1 y_2+1)}{(x_1x_2+1)(1-\alpha y_1 y_2)}, \frac{(y_1+y_2)(x_1x_2+1)}{(1-x_1x_2)(\alpha y_1 y_2+1)} \right).$$

Remark. If (x, y) is a point lies on $BRH_{\alpha,\beta}$ curve and $x, y \neq 0$, then $(-x, -y)$ and $(\pm 1/x, \mp y)$, points lie on the $BRH_{\alpha,\beta}$ curve.

Example 3.2.1. Suppose $\mathrm{BRH}_{\alpha,\beta}$ is a Huff curve defined by

$$\mathrm{BRH}_{7,11} : x\left(7y^2-1\right) = 11y\left(x^2-1\right)$$

over the prime field F_{223} .A set $BRH_{\alpha,\beta}(F_{223})$of all points lying on $\mathrm{BRH}_{\alpha,\beta}$ curve is computed by

$BRH_{\alpha,\beta}(F_{223}) = \{(0,0),(1,163),(1,60),(2,85),(2,220),\ldots\}$.

This set forms an abelian group under addition $BRH_{\alpha,\beta}$. This group is called $BRH_{\alpha,\beta}$ group and it has order $n = 229$.

Let $P = (2,220)$ and $Q = (22,70)$ points in $BRH_{\alpha,\beta}(F_{223})$

$P+Q = (2,220)+(22,70) = (13,2) \in BRH_{7,11}(F_{223})$

Also,computing 3P is done by

$3P = 2P+P = (15,35)+(2,220) = (32,171) \in BRH_{7,11}(F_{223})$

-$P = (221,3) \in BRH_{7,11}(\ F_{223})$

3.3. Comparison Between The Costs of The Huff Curve and $BRH_{\alpha,\beta}$

To analyze the efficiency of the arithmetic on the Huff curve models,it requires to determine computational cost.This cost depending on the cost to compute the points addition and doubling on selected curve.Suppose m is amultiplication operation ,s is a squaring operation ,a is an addition or subtraction and i is an inverse operation.On the Huff curve , the cost to compute the addition points $P + Q$ using Definition (2.6.2) is

$$6a + 6m + 2i$$

and the cost of computing the doubling point $2P$ on Huff curve using Definition (2.6.3) is determined by

$$4a + 2s + 6m + 2i$$

Where as, using the proposed $BRH_{\alpha,\beta}$ curve the costs of the addition and doubling are same since the addition law is as same as the doubling law, namely the formula of addition and doubling on the $BRH_{\alpha,\beta}$ curve is unfied in compare to the Huff curve which is not. Based on the unified law om the $BRH_{\alpha,\beta}$ curve ,the cost can be determined by

$$6a + 7m + 2i$$

for points addition and doubling.

3.4. The $BRH_{\alpha,\beta}$ curve Discrete Logarithm Problem

Suppose P and Q are two points on the $BRH_{\alpha,\beta}$ curve defined over the prime field F_p. Finding a positive integer η such that $Q = \eta P$is known by the $BRH_{\alpha,\beta}$ curve discrete logarithm problem $(BRH_{\alpha,\beta} - DLP)$. The$(BRH_{\alpha,\beta} - DLP)$ of Q with respect to a point P is denoted by $\eta = \log_P(Q)$ and it name, this number by the $BRH_{\alpha,\beta}$discrete logarithm of Q with respect to P.

3.4.1 $BRH_{\alpha,\beta}$ Curve Diffie-Hellman Key Exchange

Suppose both user Alice and Bob choose a certain $BRH_{\alpha,\beta}$ curve defined over F_p.Let $P \in BRH_{\alpha,\beta}$ .Both users choose their own secret integer η_A and η_B.They compute the associated multiples,Alice computes
$Q_A = \eta_A P \in BRH_{\alpha,\beta}$.
Bob computes $Q_B = \eta_B P \in BRH_{\alpha,\beta}$.They exchange the values of Q_A and Q_B.Alice then uses her secret multiplier to compute $\eta_A Q_B$ and Bob similarly computes $\eta_B Q_A$ .They now have the shered secret value

$$\eta_A Q_B = (\eta_A \eta_B) P = \eta_B Q_A$$

Example 3.4.1. Assume $BRH_{\alpha,\beta}$ is a curve as defined in Example 3.2.1 by

$$BRH_{7,11} : x\left(7y^2 - 1\right) = 11y\left(x^2 - 1\right) \pmod{223}$$

Let $P = (207, 45) \in BRH_{7,11}\left(F_{223}\right)$.
Alice and Bob Select their secret keys $\eta_A = 13$ and $\eta_B = 21$, Afterwards,

Alice computes $Q_A = 13P = (34, 38) \in BRH_{7,11}(\ F_{223})$ Bob computes $Q_B = 21P = (81, 78) \in BRH_{7,11}(\ F_{223})$ Bob sends Q_B to Alice, and Alice sends Q_A to Bob. Finally, Alice computes

$$\eta_A Q_B = 13(81, 78) = (43, 1) \in BRH_{7,11}(F_{223})$$

Bob computes

$$\eta_B Q_A = 21(34, 38) = (43, 1) \in BRH_{7,11}(\ F_{223})$$

The shared secret key between Alice and Bob is $(43, 1)$.

3.4.2 The $BRH_{\alpha,\beta}$ Curve ElGamal Public Key Cryptosystem

The ElGamal puplic key cryptosystem (EPKC) has been improved in this section using the $BRH_{\alpha,\beta}$ curve defined over a prime field F_p.The public parameters domain of the proposed $BRH_{\alpha,\beta} - EPKC$ is a prime number p,$BRH_{\alpha,\beta}$ curve defined over F_p and a point p lies on $BRH_{\alpha,\beta}$ curve. Both users are agreed about choosing these parsmeters.The proposed $BRH_{\alpha,\beta} - EPKC$ consists of three processes:key generation ,encryption and decryption. First user (Alice)chooses her secret key S and she computes her public key by $Q_A = AP$.The second user (Bob)selects his plaintext M such that $M \in BRH_{\alpha,\beta}(F_p)$.His an ephemeral Key K is chosen,where $K \in F_p$ .The ciphertext (C_1, C_2) is computed by

$$C_1 = KP(mod\, p) \text{ and } C_2 = M + KQ_A(mod\, p).$$

Bob sents the (C_1, C_2) to Alice.

she computes

$$C_2 - AC_1 \equiv ((M + KQ_A) - A(KP))(mod\, p)$$
$$\equiv (M + K\,(AP) - SA(KP))(mod\, p) \equiv M.$$

Example 3.4.2. Suppose $BRH_{\alpha,\beta}$ is a curve which is given as defined in Example (3.2.1),by

$$BRH_{7,11} : x\left(7y^2 - 1\right) = 11y\left(x^2 - 1\right)(\text{mod}223)$$

Let $P = (207, 45) \in BRH_{7,11}\,(F_{223})$.The procedure to encrypt Alice selected hir secret key the plaintext using the proposed $BRH_{\alpha,\beta} - EPKC$ is done as follows.

$A = 19$ and then Alice computes

$$Q_A = 19P = (128, 43) \in BRH_{7,11}\,(F_{223})$$

Bob chooses his plaintext M by

$$M = (34, 140) \in BRH_{7,11}\,(F_{223})\,.$$

His an ephemeral key $K = 15$. Bob computes

$C_1 = 15P = (143, 81)$

and $\quad C_2 = M + KQ_A = (34, 140) + 15(128, 43)$

$$= (144, 131)$$

Which are two points in $BRH_{\alpha,\beta}(F_{223})$. Bob sends the ciphtertext (C_1, C_2) to Alice ,Alice computes

$$C_2 - AC_1 = (144, 131) - 19(143, 81)(\text{mod } 223)$$
$$= (144, 131) + (210, 16)(\text{mod } 223)$$
$$= (34, 140) = M.$$

3.5. The $BRH_{\alpha,\beta}$ Curve in Digital Signature Scheme

Digital Signature Elliptic curve [13] means to sign through some cryptographic operations, a computerized or digitized document, instead of handwritten signature or seal. The receiver of signature can verify the signature and that the document is not altered after being signed, thus ensuring the authenticity of information and the integrity of the document. Digital signature as a technique, signature receiver can verify whether the received signature indeed is signed by a legitimate signer, only signer can generate his own signature, signature file contains the information not to be used as the signatures of other documents and signer cannot deny his signature at any time.

3.5.1 The $BRH_{\alpha,\beta}$ Curve Digital Signature Scheme

Using the $BRH_{\alpha,\beta}$ curve, the digital signature scheme can be implemented. As with the original technique [13], this scheme comprises of three phases: key generation, signature creation, and signature verification. These phases are discussed as follows.

1. **Key Generation Process**

   Let P be a prime number, $BRH_{\alpha,\beta}$ curve defined over a prime field

F_p and G be a point lies on $BRH_{\alpha,\beta}$ curve which has order n. To compute a public key of the first user, he| she selects a random integer x such that $x \in [2, n-1]$. Then computing of $Q = xG$ is done as a scalar multiplication on $BRH_{\alpha,\beta}$ curve. So the keys of the first user signer x and Q, where x is a private key and Q is a public key.Algorithm is used to generate several numerical resuits of key with different input.

Algorithm 1: Key Generation Process:

Input: p is a prime number, $G = (x, y) \in BRH_{\alpha,\beta}$ curve with an order $n, \alpha, \beta \in (F_p)$.
Output: A public key Q for signer.
(a) Select randomly secret integer x such that,$x \in [2, n-1]$.
(b) Compute $Q = xG$.
(c) Return x as a private key and Q as a public key of a signer.

---

2. **Signature Generation Process**

A signer (first user) wants to sign his|her plaintext, so, he|she selects an integerλ secretly such that $\lambda \in [2, n-1]$.He|she computes $\lambda G = (x_1, y_1)$ which is a point lies on $BRH_{\alpha,\beta}$ curve defined over F_p. Suppose $x_1(\mathrm{mod}p) \equiv r$. If $r = 0$ then it requir to choose another value of λ.Otherwise,the calculation of $\lambda^{-1}(\mathrm{mod}\ )$ is done.Another parameter s is computed by signer through $s = \lambda^{-1}(m + xr)$.If s=0,then it requires also to choose another value of λ.Otherwise,the signature of a plaintext m is (r, s).

Algorithm 2: Signature generation process:

(a) Choose an integer λ such that, $\lambda \in [2, n-1]$

(b) Compute $\lambda G = (x_1, y_1)$.

(c) Put $r = x_1 (\mathrm{mod} p)$.

(d) If $r = 0$ then go to step (1) to select new λ.

(e) Else calculate $\lambda^{-1} (\mathrm{mod} p)$.

(f) End if

(g) Calculate $s = \lambda^{-1}(m + xr)$.

(h) If $s = 0$ then go to step (1) to select new λ.

(i) Else return a signature (r, s).

(j) End if

3. **Signature Verification Process**

To verify the signature (r, s) of a plaintext m, second user (verifier) first checks the parameters values r and s such that they lie in $[1, n-1]$, so a signature is valid, otherwise, is invalid. The validity of signature requires computing the $s^{-1} (\mathrm{mod} p)$. Follows that the computations of u and v are done, where $u \equiv ms^{-1} (\mathrm{mod} p)$ and $v \equiv rs^{-1} (\mathrm{mod}\ p)$. Next, computing $uG + vQ = (x_2, y_2) = w$ is done,if $w{=}r$ then the signature is invalid,otherwise putting $t = x_2 (\mathrm{mod} p)$. With the case $t = r$ and $w = \lambda G$ the signature is valid. Algorithm(3) can be used for different implementation results.

Algorithm 3: Signature Verification Process:

(a) If $r \in [2, n-1]$ then

(b) If $s \in [2, n-1]$ then

(c) Compute s^{-1}.

(d) Calculate $u \equiv ms^{-1}(\mathrm{mod}p)$ and $v = rs^{-1}(\mathrm{mod}p)$.

(e) Calculate $w \equiv (x_2, y_2) = uG + vQ$.

(f) If $w = 0$ then stop

(g) Else calculate $t \equiv x_2(\mathrm{mod}p)$.

(h) End if

(i) If $t = r$ then the signature is valid.

(j) Else

(k) a signature is invalid

(l) End if

(m) End if

Proposition 3.5.1. The signature is a valid one if and only if $w \equiv \lambda G$

Proof.

$$\begin{aligned} w \equiv uG + vQ(\mathrm{mod}p) &= s^{-1}mG + s^{-1}rxG(\mathrm{mod}p) \\ &\equiv s^{-1}(m + xr)G(\mathrm{mod}p) \\ &\equiv \lambda G \end{aligned}$$

□

Example 3.5.1. Let p=223 be a prime number.
Suppose $BRH_{\alpha,\beta}$:$x\left(7y^2-1\right) = 11y\left(x^2-1\right)(\mathrm{mod}\ 223)$ is the $BRH_{\alpha,\beta}$ curve defined over a prime field F_{223}. Let $G = (221, 3)$ lies on $BRH_{7,11}$. The first user selects his or her private key $x = 50$. He | she computes his| her public key

$$Q = xG(\mathrm{mod}223) = (30, 24)$$

First user went to sign his/her digital document $D = 23$ so, he|she chooses a parameter $\lambda = 15$ and computes

$$\lambda G = (152, 76)$$

Putting $r = 152$ and computing $\lambda^{-1}(\mathrm{mod}223) = 119$. Therefore,

$$s = \lambda^{-1}(D + xr)(\mathrm{mod}223) = 196$$

. The signature of the digital document D is $(r, s) = (152, 196)$ which is a point lies on $BRH_{7,11}$.

To verify the signature $(r, s) = (152, 196)$, second user compute

$$s^{-1}(\text{mod}223) = 196^{-1}(\text{mod}223) = 33$$

The parameters u and v are computed respectively by

$$u = 33(23)(\text{mod}223) = 90$$

And

$$v = 33(152)(\text{mod}223) = 110.$$

So,

$$\begin{aligned} w &= 90(221, 3) + 110(30, 24) \\ &= (152, 76) \in \text{BRH}_{7,11} \end{aligned}$$

Thus, $w_x = 152 = r$. Hence, the signature is valid.

3.6. The Graphic $BRH_{\alpha,\beta}$ Curve Cryptosystem

In this section, anew graph which is called the $BRH_{\alpha,\beta}$ curve graph ($BRHg_{\alpha,\beta}$) is defined as a main point .The points on the$BRH_{\alpha,\beta}$ curve graph are the points on the $BRH_{\alpha,\beta}$ curves that form a group of $BRH_{\alpha,\beta}$ curves over a prime field. A subgraph of on the $BRH_{\alpha,\beta}$ curve graph is defined. Subgraph, fast computations on the proposed algorithm can be done through the matrices representation, which makes it easier to do. The $BRH_{\alpha,\beta}$ curve graphic cryptosystem is more secure than the

old asymmetric cryptosystem.So, the$BRH_{\alpha,\beta}$ curve graphic asymmetric cryptosystem is seen as a new way to use$BRH_{\alpha,\beta}$ curve cryptography. A new graph for the cryptographic application is proposed in this section which is called the $BRH_{\alpha,\beta}$curve graph($BRHg_{\alpha,\beta}$). This graph is defined using the $BRH_{\alpha,\beta}$ curve over a prime field. The $BRHg_{\alpha,\beta}$ is used to give another modified version of the ElGamal public key cryptosystem (EPKC) the proposed $BRH_{\alpha,\beta}$-ElGamal graphic cryptosystem enhances the secure communication in compare with original EPKC.

3.6.1 The $BRH_{\alpha,\beta}$ Curve Graph Over Prime Field

Let $BRH_{\alpha,\beta}(F_p)$ be a curve group. If $P + Q = (0, 0)$, where P and Q are two different points in $BRH_{\alpha,\beta}$ then P and Q are adjacent or can be joined by an edge in the graph.In addition, every $BRH_{\alpha,\beta}$ point is adjoined with the identity element. Then it is possible to form the $BRHg_{\alpha,\beta}$ graph $BRHg_{\alpha,\beta}(F_p)$ that is corresponding to $BRH_{\alpha,\beta}(F_p)$.This graph is similar to the graph defined by Karrar Aljamaly([20]) with Edward's curve with conditions the order of Edward's curve is even number.

For instance, if $BRH_{1,3} : x(y^2 - 1) = 3y(x^2 - 1)(\text{mod}7)$ is curve. The point on $BRH_{\alpha,\beta}$ form a set

$BRH_{1,3}(F_7) = \{(0, 0), (1, 1), (1, 6), (6, 1), (6, 6)\}$. The is $BRHg_{1,3}(F_7)$ of $BRH_{1,3}(F_7)$ shown in Figure (3.5)

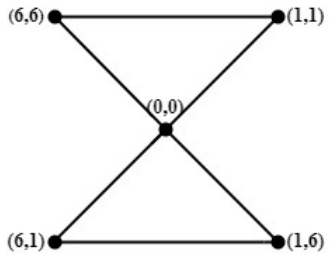


Figure 3.5: The $BRHg_{1,3}(F_7)$ of $BRH_{1,3}(F_7)$

3.6.2 The $BRH_{\alpha,\beta}$- EIGamal Graphic Cryptosystem

This section presents alternative version of EPKC using the $BRH_{\alpha,\beta}$ curve points and the $BRHg_{\alpha,\beta}$ graph. The public parameters are $BRH_{\alpha,\beta}$ curve group $BRH_{\alpha,\beta}(F_p)$ and the matrix $T \in GL_n(F_p)$. The proposed $BRH_{\alpha,\beta}$- ElGamal graphic cryptosystem $BRH_{\alpha,\beta}$-EGC is explained through the following algorithms.

Algorithm 4: Key Generation Process of $BRH_{\alpha,\beta}$-EGC

Input: A prime number p and the matrix $T \in GL_n(F_p)$.
Output: The public key A, where $A \in GL_n(F_p)$.

1. Alice selects the value a as her private key, where $a \in \{2, \ldots, p-1\}$.
2. She generates a public key $A = T^a (\mathrm{mod} p)$.
3. Alice keys are (A, a).

Algorithm 5: Encryption Process of The $BRH_{\alpha,\beta}$-EGC

Input: The matrices $T \in GL_n(F_p)$ and a public key A.
Output: The cipher text (C_1, C_2), where C_1 and C_2 are two matrices.

1. Bob selects his plaintext M which is the $BRH_{\alpha,\beta}$ curve subgroup of $BRH_{\alpha,\beta}(F_p)$ such that if $(x, y) \in M$ then $(-x, -y)$,(x, y_1) and$(-x, -y_1) \in M$.
2. He represents his plaintext M by $BRH_{\alpha,\beta}$ subgraph $BRHsg_{\alpha,\beta}(F_p)$
3. He converts the $BRHsg_{\alpha,\beta}(F_p)$ in to a weighted graph. This converting is done through computing the weights for all edges, if (x, y) and $(-x, -y)$ are two points between any edge then the weight of this edges is $w = \min\{x, -x\}$, if (x, y) and $(0, 0)$ two points between any edge then the weight of this edges is $w = 0$.
4. A weighted graph converted into another graph after deleting one of the triangles or not delet that have the same weights of edges which are join two vertices not equal to zero and each one of them is inverse to other and represents it by an adjacent matrix B.
5. He chooses his secret key $b \in \{2, 3, \ldots, p-1\}$.
6. The cipher text is computed by two square matrices $C_1 \equiv T^b (\mathrm{mod} p)$ and $C_2 \equiv A^b M (\mathrm{mod} p)$.
7. Bob sends the ciphertext pair (C_1, C_2) to Alice.

Algorithm 6: Decryption Process of the $BRH_{\alpha,\beta}$-EGC

Input: A prime p and a cipher text (C_1, C_2).
Output: The plaintext M which is the $BRH_{\alpha,\beta}$ curve subgroup of $BRH_{\alpha,\beta}(F_p)$.

1. Alice computes $(C_1^a)^{-1} (\mathrm{mod} p)$.
2. She computes the multiplication matrix $(C_1^a)^{-1} C_2 (\mathrm{mod} p)$.
3. She represents matrix B by a weighted graph.
4. She gets the $BRH_{\alpha,\beta}$ points by using the weights of all edges that give the original plaintext.

Example 3.6.1. Suppose $BRH_{\alpha,\beta}$ is curve defined by $BRH_{5,7} : x\left(7y^2-1\right) = 5y\left(x^2-1\right)$ over the prime field F_{103}. A set $BRH_{\alpha,\beta}\left(F_{103}\right)$ of all points lying on BRH is computed by

$$BRH_{7,5}\left(F_{103}\right) = \{(1,70),(1,33),(2,76),(2,6)\ldots(0,0)\}$$

This set an abelain group under addition and it has prime order $n = 117$. The public matrix

$$T = \begin{pmatrix} 31 & 41 & 101 & 55 & 23 \\ 23 & 67 & 78 & 90 & 3 \\ 56 & 89 & 18 & 23 & 7 \\ 76 & 33 & 101 & 13 & 23 \\ 12 & 34 & 67 & 17 & 99 \end{pmatrix} \in GL_5\left(F_{103}\right)$$

Alice chooses her private key $a = 4$ and compute her public key A by

$$A = T^4(\mathrm{mod}103) = \begin{pmatrix} 49 & 55 & 59 & 30 & 99 \\ 23 & 22 & 59 & 16 & 48 \\ 38 & 82 & 62 & 41 & 38 \\ 66 & 53 & 87 & 10 & 61 \\ 33 & 83 & 100 & 25 & 3 \end{pmatrix}$$

Bob chooses his plaintext

$M = \{(31, 4), (72, 99), (46, 88), (57, 15), (31, 11), (72, 92), (46, 52), (57, 51), (0, 0)\}$

M is a subset of $BRH_{7,5}(F_{103})$ curve set $BRHsg_{7,5}(F_{103})$ is a plaintext which is represented by $BRH_{7,5}$ curve weighted subgraph $BRHsg_{7,5}(F_{103})$ as shown in Figure 3.6.

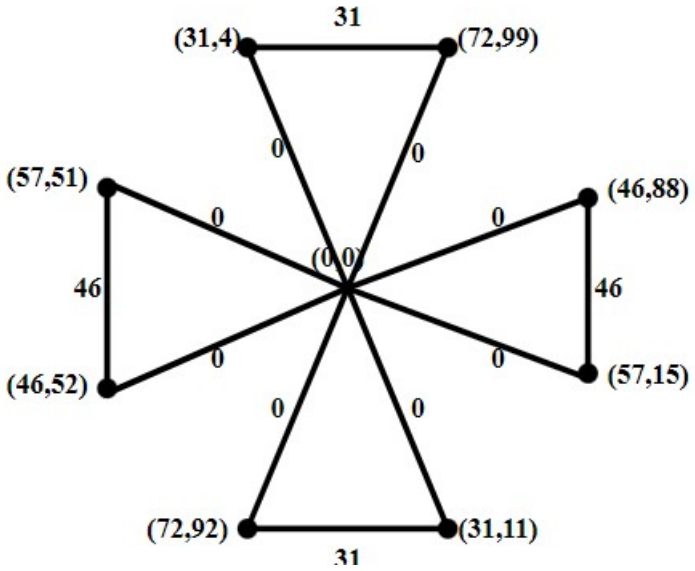


Figure 3.6: The weighted subgraph $BRHsg_{7,5}(F_{103})$ of M

Bob converts $BRH_{7,5}$ curve weighted subgraph $BRHsg_{7,5}(F_{103})$ into another $BRH_{\alpha,\beta}$ curve graph $BRHsg^{*}_{7,5}(F_{103})$ as seen in Figure 3.7.

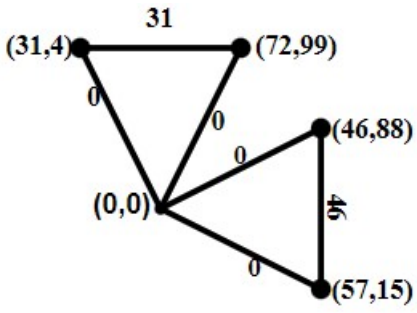


Figure 3.7: The $BRH_{\alpha,\beta}$ curve weighted subgraph $BRHsg^{*}_{7,5}(F_{103})$

An adjacent matrix B of $BRHsg^{*}_{7,5}(F_{103})$ is computed by

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 31 & 0 & 0 \\ 0 & 31 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 46 \\ 0 & 0 & 0 & 46 & 0 \end{pmatrix}$$

Bob chooses a secret key $b = 3$ and computes the ciphertext (C_1, C_2) by

$$C_1 = T^3(\mathrm{mod}103) = \begin{pmatrix} 67 & 35 & 52 & 65 & 55 \\ 50 & 38 & 69 & 63 & 66 \\ 51 & 38 & 99 & 96 & 59 \\ 3 & 23 & 37 & 90 & 35 \\ 17 & 67 & 61 & 11 & 34 \end{pmatrix}$$

And

$$C_2 = A^3 \times B(\mathrm{mod}103) = \begin{pmatrix} 0 & 9 & 95 & 54 & 33 \\ 0 & 13 & 21 & 37 & 4 \\ 0 & 93 & 28 & 86 & 4 \\ 0 & 42 & 22 & 50 & 29 \\ 0 & 53 & 84 & 38 & 99 \end{pmatrix}$$

He sends the cipher text pair (C_1, C_2) to Alice. After receiving the cipher text pair (C_1, C_2) to Alice, she wants to decrypt and recover the original plaintext. She computes first

$$\left(C_1^4\right)^{-1} (\text{mod} 103) = \begin{pmatrix} 6 & 51 & 8 & 27 & 21 \\ 28 & 15 & 27 & 51 & 102 \\ 77 & 32 & 6 & 14 & 28 \\ 85 & 42 & 0 & 18 & 64 \\ 63 & 40 & 0 & 97 & 27 \end{pmatrix}$$

$$\left(C_1^4\right)^{-1} \times C_2 (\text{mod} 103) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 31 & 0 & 0 \\ 0 & 31 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 46 \\ 0 & 0 & 0 & 46 & 0 \end{pmatrix} = B$$

Alice represents a matrix B by a weighted graph in Figure (3.7) and gets the original plaintext

$M = \{(31, 4), (72, 99), (46, 88), (57, 15), (31, 11), (72, 92), (46, 52), (57, 51), (0, 0)\}$.

By using the weighted of edges graph.

3.7. The $BRH_{\alpha,\beta}$ Matrix ElGamal Cryptosystem

Alternative version of ElGamal cryptosystem is proposed in this section.This version uses the scalar multiplication matrix .And its elements are the points which lie on $BRH_{\alpha,\beta}$ curve.

Definition 3.7.1. Let $BRH_{\alpha,\beta}(F_p)$ be curve group, let $P=(x,y)$, $Q=(1/x,-y)$, different points in $BRH_{\alpha,\beta}(F_p)$, where $(0,0)$ is the identity element then P and Q are adjacent or can be joined by an edge in the graph.In addition, every $BRH_{\alpha,\beta}$ point is adjoined with the identity element.Then it is possible to form the $BRHg_{\alpha,\beta}$ graph $BRHg_{\alpha,\beta}(F_p)$ that is corresponding to $BRH_{\alpha,\beta}(F_p)$.

3.7.1 The $BRH_{\alpha,\beta}$ Matrix ElGamal Cryptosystem Over Prime Field

This section presents alternative version of EPKC using the $BRH_{\alpha,\beta}$ curve points and the $BRHg_{\alpha,\beta}$ graph. The public parameters are a $BRH_{\alpha,\beta}$ curve group $BRH_{\alpha,\beta}(F_p)$ and the matrix T is a matrix of a point in $BRH_{\alpha,\beta}$, we represent it by $M(BRH_{\alpha,\beta}H(F_p))$. The proposed BRH -Matrix ElGamal graphic cryptosystem $(BRH_{\alpha,\beta}-MEGC)$ is explained by the following algorithms.

Algorithm 7: Key Generation of the $BRH_{\alpha,\beta} - MEGC$.

Input: A prime number p and the matrix $T \in M\left(BRH_{\alpha,\beta}\left(F_p\right)\right)$, Where $M\left(BRH_{\alpha,\beta}\left(F_p\right)\right)$ is a set of matrices their elements are point lie on $BRH_{\alpha,\beta}$ curve.

Output: The public key A, where $A \in M\left(BRH_{\alpha,\beta}\left(F_p\right)\right)$.

1. Alice selects the value a as her private key, where $a \in \{2, \ldots, p-1\}$.
2. She generates a public key $A = aT(\mathrm{mod}p)$
3. Alice keys are (A, a).

Algorithm 8: Encryption of the $BRH_{\alpha,\beta} - MEGC$.

Input: The matrices $T \in M\left(BRH_{\alpha,\beta}\left(F_p\right)\right)$ and a public key A.

Output: The cipher text (C_1, C_2), where C_1 and C_2 are two matrices.

1. Bob selects his plaintext M which is the $BRH_{\alpha,\beta}$ curve subgroup of $BRHsg_{\alpha,\beta}\left(F_P\right)$ such that if $(x, y) \in M$ then $(1/x, -y) \in M$.
2. He represents his plaintext M by subgraph $BRHsg_{\alpha,\beta}\left(F_P\right)$.
3. He converts the $BRHsg_{\alpha,\beta}\left(F_P\right)$ in to a weighted graph. This converting is done through computing the weights for all edges, if (x, y) and $(1/x, -y)$ two points between any edge then the weight of this edges is $w =$ the point of min $\{x, 1/x\}$, if (x, y) and $(0, 0)$ two points between any edge then the weight of this edges is $w = 0$
4. He chooses his secret key $b \in \{2, 3, \ldots, p-1\}$.
5. The cipher text is computed by two square matrices $C_1 \equiv bT(\mathrm{mod}p)$ and $C_2 \equiv bAM(\mathrm{mod}p)$.
6. Bob sends the cipher text pair (C_1, C_2) to Alice.

Algorithm 9: Decryption of the $BRH_{\alpha,\beta} - MEGC$

Input: A prime p and a cipher text (C_1, C_2).
Output: The plaintext M which is the $BRH_{\alpha,\beta}$ curve subgroup of $BRH\,(F_p)$.

1. Alice computes scalar multiplication matrix $aC_1(\mathrm{mod}p)$.
2. She computes the $-aC_1(\mathrm{mod}p)$.
3. She represents matrix B by a weighted graph.
4. She gets the $BRH_{\alpha,\beta}$ points by using the weights of all edges
5. that give the original plaintext.

Proposition 3.7.1. The decryption process is computed by

$C_2 - a \times C_1 = M$.

Proof. $C_2 - a \times C_1$

$=M + bA - a \times C_1,$ since $C_2 = M + bA$

$=M + bA - a \times bT$, since $C_1 = bT$

$= M + b \times a \times T - a \times b \times T$, since $A = a \times T$

$= M + b \times a \times T - b \times a \times T = M$ □

Example 3.7.1. Suppose $BRH_{\alpha,\beta}$ curve is defined in Example(3.6.1). The public matrix

$$T = \begin{pmatrix} (39,82) & (11,40) & (42,26) & (44,23) & (31,11) \\ (42,26) & (56,51) & (88,102) & (84,19) & (42,73) \\ (101,27) & (61,30) & (92,71) & (79,10) & (76,26) \\ (55,59) & (79,25) & (55,102) & (48,1) & (54,65) \\ (47,88) & (42,73) & (82,65) & (101,27) & (11,40) \end{pmatrix} \in M\,(BRH_{\alpha,\beta}\,(F_{103}))$$

Now, Alice performs the following steps:

1. She chooses her private key $a = 5$.

2. She computes her public key A

$$A = aT(\text{mod}103) = \begin{pmatrix} (33,85) & (72,92) & (7,87) & (27,30) & (28,32) \\ (7,87) & (24,93) & (21,12) & (51,76) & (7,23) \\ (38,19) & (96,80) & (31,4) & (57,15) & (44,87) \\ (49,38) & (57,51) & (49,12) & (54,91) & (88,102) \\ (79,25) & (7,23) & (55,102) & (38,19) & (72,92) \end{pmatrix}$$

So, her keys are $a = 5$ and A. Bob does the following steps:

1. He chooses a secret key $b = 4$.

2. He computes $C_1 = bT(modp)$.

$$C_1 = 4T(\text{ mod } 103) = \begin{pmatrix} (84,19) & (7,87) & (25,94) & (66,96) & (42,26) \\ (25,94) & (84,19) & (66,96) & (96,16) & (78,9) \\ (42,26) & (25,94) & (7,87) & (52,27) & (78,9) \\ (66,96) & (51,76) & (37,7) & (66,96) & (25,94) \\ (84,19) & (78,9) & (78,9) & (42,26) & (7,87) \end{pmatrix}$$

3. He computes $bA(\text{mod}p)$.

$$4T(\text{mod}103) = \begin{pmatrix} (51,76) & (61,77) & (66,96) & (78,9) & (7,87) \\ (66,96) & (51,76) & (78,9) & (42,26) & (52,27) \\ (7,87) & (66,96) & (61,77) & (84,19) & (37,7) \\ (78,9) & (84,19) & (25,94) & (78,9) & (66,96) \\ (51,76) & (37,7) & (37,7) & (7,87) & (61,77) \end{pmatrix}$$

4. He chooses his plaintext

   $M = \{(2,76),(52,27),(10,99),(31,4),(0,0)\}$

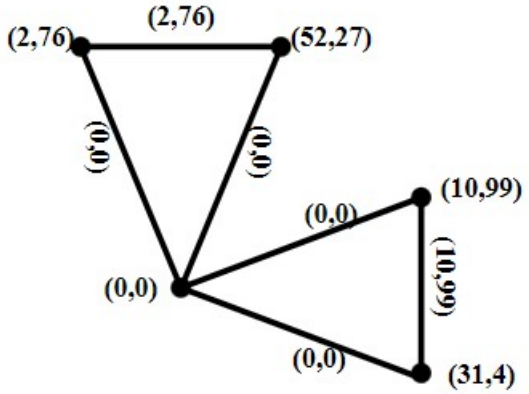


Figure 3.8: The $BRH_{\alpha,\beta}$ curve weighted subgraph $BRHsg_{\alpha,\beta}(F_{103})$

$$B = \begin{pmatrix} (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (2,76) & (0,0) & (0,0) \\ (0,0) & (2,76) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (10,99) \\ (0,0) & (0,0) & (0,0) & (10,99) & (0,0) \end{pmatrix}$$

5. He computes $C_2 = (bA + B)(\mathrm{mod} p)$

   $C_2 = (4T + B)(\mathrm{mod} 103) =$

$$\begin{pmatrix} (51,76) & (61,77) & (66,96) & (78,9) & (7,87) \\ (66,96) & (51,76) & (76,26) & (42,26) & (52,27) \\ (7,87) & (101,27) & (61,77) & (84,19) & (37,7) \\ (78,9) & (84,19) & (25,94) & (78,9) & (31,11) \\ (51,76) & (37,7) & (37,7) & (15,1) & (61,77) \end{pmatrix}$$

6. Bob send (C_1, C_2) to Alice

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

(a) She computes a C_1.

$$5C_1(\mathrm{mod}103) = \begin{pmatrix} (51,76) & (61,77) & (66,96) & (78,9) & (7,87) \\ (66,96) & (51,76) & (78,9) & (42,26) & (52,27) \\ (7,87) & (66,96) & (61,77) & (84,19) & (37,7) \\ (78,9) & (84,19) & (25,94) & (78,9) & (66,96) \\ (51,76) & (37,7) & (37,7) & (7,87) & (61,77) \end{pmatrix}$$

(b) She computes $(C_2 - aC_1)\,(\mathrm{mod}p)$

$$(C_2 - aC_1)\,(\mathrm{mod}103) =$$

$$\begin{pmatrix} (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (2,76) & (0,0) & (0,0) \\ (0,0) & (2,76) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (10,99) \\ (0,0) & (0,0) & (0,0) & (10,99) & (0,0) \end{pmatrix} = B$$

Alice represents a matrix B by a weighted graph in Figure (3.8) and gets the original plaintext
$M = \{(2,76),(52,27),(10,99),(31,4),(0,0)\}$. By using the weighted of edges graph.

3.8. Generated Graph for Text Encryption Scheme Based on $BRH_{\alpha,\beta}$ Curve

In this section,an encryption algorithm for text is proposed using the $BRH_{\alpha,\beta}$ curve. It is also employed the matrices ,number theory and graph theory this algorithm is discussed as follows.

With aprime number p, the $BRH_{\alpha,\beta}$ curve can be defined over a prime F_p .Let $Q = (x, y)$ is a point lies on $BRH_{\alpha,\beta}$ curve. An encoding table is created for the plointext letters through computing the doubling and addition points on Q as shown in Table(3.1) .To generate the Keys ,a public matrix $\gamma \in GL_n(F_p)$.First user (Alice)selects her private key a and she computes her public key A by

$$A = \gamma^a (mod P) \in GL_n(F_p).$$

so,the private and public keys are a and A respectively.Algorithm (10) can beused to the keys with the different input.

Algorithm 10: Key Generation process:

Input: p is prime number, $Q = (x, y) \in BRH_{\alpha,\beta}$ curve.
output: The public key A,where $A \in GL_n(F_p)$.

1. Create a table to encode each alphabet using δ. See Table 3.1
2. Select public a matrix $\gamma \in GL_n(F_p)$.

Key creation

1. Alice choose a private key a.
2. She computes her public key $A = \gamma^a (\mathrm{mod} p) \in GLn(Fp)$.

Table 3.1: Encode each alphabet

A	B	C	$\ldots$	Y	Z
Q	$2Q$	$3Q$	$\ldots$	$25Q$	$26Q$

Now,Second user (Bob)selects his private key b and computes C_1 by

$$C_1 = \gamma^b (mod p)$$

Bob chooses his plaintext m and converts each letter of it in to vertex. Acycle graph on these vertices is formed through connecting the consecutive vertices by edeg.The weight of each edge is determined through computing by the relation

$$w_i = v_{i+1} - v_i.$$

The addition of edges is drown to complete the graph . A special letter A is added for denoting to start of the string.The spaning tree (TS) is computed and its adjaceny matrix M_1 is created.The position order of the plaintext that is stored in the diagonal of a matrix M_1 to form another matrix M_2.Bob computes C_2 by

$$C_2 = A^b M_2 (mod p)$$

The ciphertext (C_1, C_2) has been sent to Alice .Algorithm (11)is used with various inputs to compute different results of the ciphertext.

Algorithm 11: Encryption process

Input: The matrices γ and A ,where $\gamma, A \in GL_n(F_p)$.
Output: The ciphertext (C_1, C_2),where C_1 andC_2 .

1. Bob choose a private key b. Compute $C_1 = \gamma^b (\mathrm{mod} p)$
2. He selects his message m.
3. Convert each letter of m to vertex.
4. Connect each pair of consecutive letters to form a cycle graph.
5. Compute the weights of edges by $w_i = v_{i+1} - v_i$.
6. Add the additional edges to complete the graph
7. Add a special letter to denote the beginning of the string (Let A).
8. Compute ST and its adjacency-matrix M_1.
9. Store the posstion order of the vertices in the M_2 matrix in the diagonal places.
10. Compute $C_2 = \gamma^b M_2$
11. Bob sends (C_1, C_2) to Alice

Alice receives the (C_1, C_2).So, she first computes

$$C_1^{-a} C_2 (mod p) = M_2$$

and she uses the encoding table to determine the original plaintext. Algorithm (12)is implemented to get the recovering the plaintext.

Algorithm 12: Decryption process.

Input:A cipher text (C_1, C_2).
Output: The plaintext M .

1. Alice compute $(C_1^a)^{-1} (mod\, p)$.
2. She cumpute $C_1^{-a} C_2 (mod\, p) = M_2$.
3. She knew that node (0,0) is A ,So by using encoding table she compute the original text

3.8.1 Cases on the Proposed Text Encryption

There are three cases followed in text encryption scheme that is proposed in section(3.8) according to the dimensions of the public key matrix, which are as follows:

1. Case I

   If the public matrix size equal to plaintext letter number then the text encryption schame (algorithm)in section (3.8)is directly implemented.

2. Case II

   If the public matrix size is greater than the plaintext letters number

then,it requires to add rows and columns of zero points to the matrix M_1 and after then applying thetext encryption algorithm is done.

3. case III

   If the public matrix size is smaller than the plaintext letters number then the plaintext letters are divided in to two (or more than two)parts.The matrices of these parts are formed.

3.8.2 The Computational Results on the Cases :I,II and III.

Example 3.8.1. (Case I)

Let $p = 103$ be a prime number.

Suppose $BRH_{5,11} : x\left(5y^2 - 1\right) = 11y\left(x^2 - 1\right) \pmod{103}$ is the $BRH_{\alpha,\beta}$ curve defined over a prime field F_{103}. Choose $Q = (101, 65)$ lies on $BRH_{7,11}$. Then a character encoding table is generated in the Table 3.2.

Table 3.2: Encoding table

A	B	C	$\cdots$	H	$\cdots$	X	Y	Z
Q	$2Q$	$3Q$	$\cdots$	$8Q$	$\cdots$	$24Q$	$25Q$	$26Q$

The public matrix is chosen by

$$\gamma = \begin{pmatrix} 2 & 3 & 5 & 6 & 3 \\ 7 & 0 & 1 & 1 & 10 \\ 9 & 7 & 2 & 2 & 11 \\ 12 & 11 & 2 & 1 & 7 \\ 13 & 4 & 8 & 2 & 1 \end{pmatrix} \in GL_5\left(F_{103}\right)$$

Now, Alice performs the following steps:

1. She chooses her private key $a = 2$.

2. She computes her public key A by

$$A = \gamma^2 (\mathrm{mod} 103) = \begin{pmatrix} 78 & 16 & 59 & 37 & 33 \\ 62 & 79 & 16 & 65 & 49 \\ 46 & 4 & 45 & 89 & 41 \\ 16 & 89 & 30 & 102 & 79 \\ 60 & 18 & 97 & 102 & 79 \end{pmatrix}$$

Alice's keys are $a = 2$ and a matrix A,

Bob does the following steps:

1. He chooses her private key $b = 3$.

2. He compute $C_1 = \gamma^b$

$$C_1 = \gamma^3 (\text{mod} 103) = \begin{pmatrix} 24 & 53 & 38 & 87 & 99 \\ 75 & 76 & 16 & 28 & 8 \\ 66 & 51 & 6 & 26 & 101 \\ 86 & 48 & 35 & 93 & 1 \\ 74 & 31 & 9 & 8 & 57 \end{pmatrix}$$

3. He chooses his massage (BRHC).

4. He convert each character to vertex.

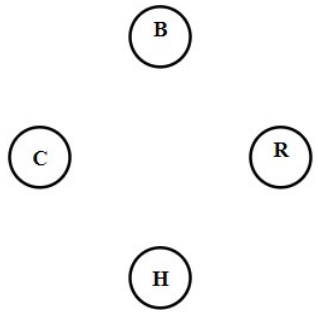


Figure 3.9: vertices ofthe Bob's plaintext letters.

5. Using the encoding table, the message is written in a form M $= \{(17, 24), (23, 51), (7, 90), (65, 43)\} \in BRH_{5,11}$ And compute the weight of each edge

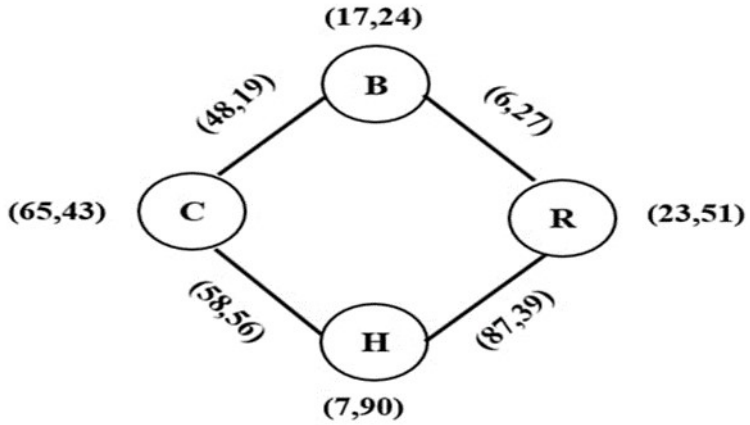


Figure 3.10: Weighted graph contains plaintext letters

6. He continue to add edges to the graph until it is full.

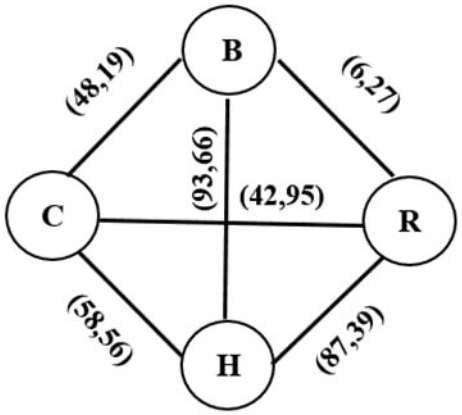


Figure 3.11: Complete plain graph

7. He put a special character before the first letter to indicate to the first characte

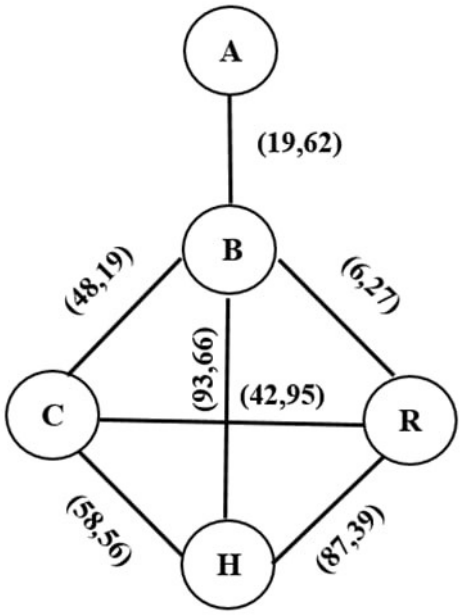


Figure 3.12: Complete plain graph with a special letter

8. He find the spanning tree in Figure(3.13)

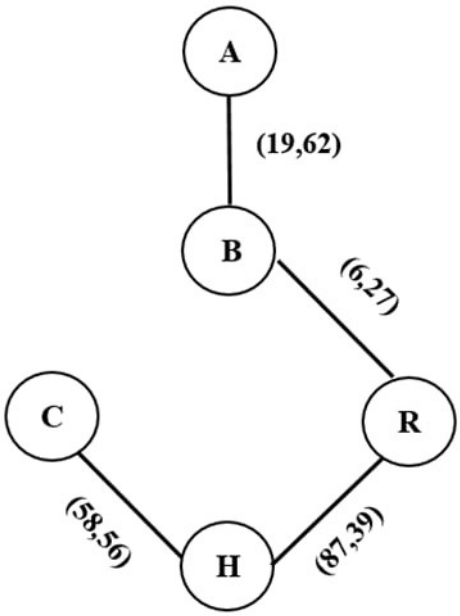


Figure 3.13: A spanning tree graph on the plaintext m

9. He represent as a matrix M_1

$$M_1 = \begin{bmatrix} (0,0) & (19,62) & (0,0) & (0,0) & (0,0) \\ (19,62) & (0,0) & (6,27) & (0,0) & (0,0) \\ (0,0) & (6,27) & (0,0) & (87,39) & (0,0) \\ (0,0) & (0,0) & (87,39) & (0,0) & (58,56) \\ (0,0) & (0,0) & (0,0) & (58,56) & (0,0) \end{bmatrix}$$

10. He store the letters order in the diagonal instead of $(0,0)$.

Table 3.3: Characters order

A	B	R	H	C
$(0,0)$	$(1,1)$	$(2,2)$	$(3,3)$	$(4,4)$

$$M_2 = \begin{bmatrix} (0,0) & (19,62) & (0,0) & (0,0) & (0,0) \\ (19,62) & (1,1) & (6,27) & (0,0) & (0,0) \\ (0,0) & (6,27) & (2,2) & (87,39) & (0,0) \\ (0,0) & (0,0) & (87,39) & (3,3) & (58,56) \\ (0,0) & (0,0) & (0,0) & (58,56) & (4,4) \end{bmatrix}$$

11. He compute $C_2 = A^b M_2$

$$A^b(\text{mod}103) = \begin{bmatrix} 24 & 34 & 16 & 10 & 71 \\ 72 & 16 & 46 & 39 & 17 \\ 24 & 74 & 4 & 87 & 21 \\ 70 & 7 & 4 & 53 & 43 \\ 17 & 51 & 15 & 30 & 92 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} (75,55) & (86,8) & (76,1) & (81,8) & (40,20) \\ (5,38) & (57,90) & (79,88) & (58,82) & (64,89) \\ (36,47) & (54,33) & (90,43) & (76,48) & (83,12) \\ (73,81) & (40,101) & (26,101) & (14,45) & (53,50) \\ (61,31) & (24,20) & (62,2) & (36,59) & (48,9) \end{bmatrix}$$

12. Bob send (C_1, C_2) to Alice

Decryption process

1. Alice compute C_1^{-a}

$$C_1^{-2} = \begin{pmatrix} 88 & 19 & 98 & 49 & 49 \\ 41 & 9 & 53 & 20 & 74 \\ 73 & 14 & 40 & 46 & 28 \\ 85 & 44 & 88 & 3 & 66 \\ 2 & 51 & 55 & 24 & 47 \end{pmatrix}$$

2. compute ${C_1}^{-a}\mathrm{C}_2$

$$C_1^{-2}C_2 = \begin{bmatrix} (0,0) & (19,62) & (0,0) & (0,0) & (0,0) \\ (19,62) & (1,1) & (6,27) & (0,0) & (0,0) \\ (0,0) & (6,27) & (2,2) & (87,39) & (0,0) \\ (0,0) & (0,0) & (87,39) & (3,3) & (58,56) \\ (0,0) & (0,0) & (0,0) & (58,56) & (4,4) \end{bmatrix} = M_2$$

3. She suppose that the node $(0,0)$ is A and using encoding table 3.3 she compute:

   $(0,0) = \text{code } A = (101,65)$

   Node $(1,1) = \text{code } A + w_1$

   $= ((101,65) + (19,62))(\mathrm{mod}103)$

   $= (17,24) = B$

   Node $(2,2) = \text{code } B + w_2$

   $= (23,51) = R$

   Node $(3,3) = \text{code } R + w_3$

   $= (7,90) = H$

   Node $(4,4) = \text{code } H + w_4$

   $= (65,34) = C$

Example 3.8.2. (Case II:)

with the same parameters that is defind in case I ,it is possible to choose

a matrix $\gamma$by

$$\gamma = \begin{pmatrix} 3 & 11 & 1 & 0 & 76 & 5 & 3 \\ 45 & 9 & 1 & 3 & 6 & 7 & 32 \\ 2 & 8 & 9 & 13 & 0 & 3 & 1 \\ 11 & 13 & 65 & 7 & 8 & 19 & 20 \\ 54 & 1 & 5 & 15 & 2 & 23 & 70 \\ 45 & 5 & 6 & 17 & 21 & 100 & 19 \\ 101 & 0 & 5 & 11 & 0 & 9 & 3 \end{pmatrix}$$

Now, Alice performs the following steps:

1. She chooses her private key $a = 4$.

2. She computes her public key A by

$$A = \gamma^4 (\mathrm{mod} 103) = \begin{pmatrix} 60 & 72 & 24 & 59 & 46 & 9 & 19 \\ 11 & 73 & 87 & 7 & 50 & 70 & 35 \\ 78 & 9 & 7 & 51 & 37 & 76 & 20 \\ 87 & 36 & 32 & 0 & 64 & 54 & 98 \\ 30 & 50 & 31 & 6 & 39 & 19 & 21 \\ 57 & 42 & 10 & 68 & 71 & 13 & 24 \\ 69 & 12 & 98 & 6 & 87 & 15 & 14 \end{pmatrix}$$

Alice's keys are $a = 4$ and a matrix A.

Now, Bob does the following steps:

1. He chooses her private key $b = 5$.

2. He compute $C_1 = \gamma^{\mathrm{b}}$

$$C_1 = \begin{pmatrix} 61 & 27 & 28 & 89 & 82 & 58 & 57 \\ 54 & 48 & 2 & 74 & 84 & 1 & 63 \\ 15 & 76 & 12 & 19 & 0 & 61 & 4 \\ 18 & 102 & 43 & 56 & 97 & 22 & 52 \\ 23 & 66 & 89 & 88 & 47 & 43 & 85 \\ 86 & 67 & 52 & 45 & 18 & 96 & 52 \\ 93 & 93 & 2 & 46 & 49 & 88 & 59 \end{pmatrix}$$

3. Using the encoding Table 3.2, the message is written in a form $M = \{(17, 24), (23, 51), (7, 90), (65, 43)\} \in BRH_{5,11}$ The graph of message statement is the same steps as in the first case.

$$M_1 = \begin{pmatrix} (0,0) & (19,62) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (19,62) & (0,0) & (6,27) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (6,27) & (0,0) & (87,39) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (87,39) & (0,0) & (58,56) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (58,56) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \end{pmatrix}$$

4. He store the letters order in the diagonal instead of $(0,0)$ Table 3.3.

$$M_2 = \begin{pmatrix} (0,0) & (19,62) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (19,62) & (1,1) & (6,27) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (6,27) & (2,2) & (87,39) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (87,39) & (3,3) & (58,56) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (58,56) & (4,4) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \end{pmatrix}$$

5. He compute $C_2 = A^b M_2$

6. Bob send (C_1, C_2) to Alice.

$$C_2 = \begin{pmatrix} (54,19) & (12,74) & (17,48) & (89,66) & (97,10) & (0,0) & (0,0) \\ (95,1) & (102,78) & (11,74) & (18,102) & (97,18) & (0,0) & (0,0) \\ (2,77) & (57,75) & (64,95) & (6,78) & (25,7) & (0,0) & (0,0) \\ (20,49) & (18,22) & (39,10) & (25,30) & (34,33) & (0,0) & (0,0) \\ (54,19) & (29,25) & (2,90) & (94,47) & (73,15) & (0,0) & (0,0) \\ (40,98) & (9,37) & (40,15) & (57,4) & (47,85) & (0,0) & (0,0) \\ (62,18) & (83,17) & (10,81) & (34,94) & (92,72) & (0,0) & (0,0) \end{pmatrix}$$

Decryption process

1. Alice compute C_1^{-a}

$$C_1^{-4}(\mathrm{mod}103) = \begin{pmatrix} 51 & 95 & 62 & 95 & 33 & 40 & 80 \\ 67 & 68 & 92 & 98 & 57 & 92 & 58 \\ 19 & 39 & 56 & 82 & 64 & 21 & 77 \\ 30 & 77 & 36 & 41 & 91 & 40 & 77 \\ 7 & 30 & 88 & 15 & 5 & 58 & 48 \\ 73 & 22 & 85 & 93 & 55 & 74 & 80 \\ 78 & 30 & 74 & 77 & 6 & 85 & 48 \end{pmatrix}$$

2. compute $C_1^{-a}C_2$

   $C_1^{-4}C_2(\mathrm{mod}103) =$

$$\begin{pmatrix} (0,0) & (19,62) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (19,62) & (1,1) & (6,27) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (6,27) & (2,2) & (87,39) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (87,39) & (3,3) & (58,56) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (58,56) & (4,4) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \end{pmatrix} = M_2$$

3. She suppose that the node $(0,0)$ is A and using encoding Table 3.3 she compute:

   $(0,0) = \mathrm{code}A = (101,65)$

Node $(1,1)$=code$A + w_1 (\text{mod } 103)$

$= ((101, 65) + (19, 62))(\text{mod } 103)$

$= (17, 24) = B$

Node $(2,2) =$ code $B + w_2 = (23, 51) = R$

Node $(3,3) =$ code$R + w_3 = (7, 90) = H$

Node $(4,4) = codeH + w_4 = (65, 34) = C$

Since the number of rows containing $(0,0)$ is equal to the number of columns containing $(0,0)$, this does not represent a character code.

Example 3.8.3. (Case III)

with the same parameters that is defind in case I ,it is possible to choose a matrix γ by

$$\gamma = \begin{pmatrix} 4 & 100 & 77 \\ 67 & 35 & 101 \\ 37 & 41 & 23 \end{pmatrix}$$

Now, Alice performs the following steps:

1. She chooses her private key $a = 4$.
2. She computes her public key A by

$$A = \gamma^4 (\text{mod} 103) = \begin{pmatrix} 62 & 78 & 56 \\ 18 & 64 & 70 \\ 50 & 72 & 19 \end{pmatrix}$$

Alice's keys are $a = 4$ and a matrix A.

Now, Bob does the following steps:

1. He chooses her private key $b = 5$.

2. He compute $C_1 = \gamma^b$

$$C_1 = \gamma^5 (\mathrm{mod} 103) = \begin{pmatrix} 27 & 102 & 35 \\ 49 & 9 & 87 \\ 62 & 59 & 23 \end{pmatrix}$$

3. Using the encoding table, the message is written in a form

$$M = \{(17, 24), (23, 51), (7, 90), (65, 43)\} \in BRH_{5,11}$$

The graph of message statement is the same steps as in the first case

Since dimension of matrix public key less than dimension of M_1 in first case he divided M_1 for two matrix

$$M_1 = \begin{pmatrix} (0,0) & (19,62) & (0,0) \\ (19,62) & (0,0) & (6,27) \\ (0,0) & (6,27) & (0,0) \end{pmatrix}$$

and

$$M_1^* = \begin{pmatrix} (0,0) & (58,56) & (0,0) \\ (58,56) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) \end{pmatrix}$$

4. He store the characters order in the diagonal instead of $(0,0)$.

   And

$$M_2 = \begin{pmatrix} (0,0) & (19,62) & (0,0) \\ (19,62) & (1,1) & (6,27) \\ (0,0) & (6,27) & (2,2) \end{pmatrix}$$

$$M_2^* = \begin{pmatrix} (3,3) & (58,56) & (0,0) \\ (58,56) & (4,4) & (0,0) \\ (0,0) & (0,0) & (0,0) \end{pmatrix}$$

5. He compute $C_2 = A^b M_2$ and ${C_2}^* = A^b {M_2}^*$

$$A^5(\text{mod}103) = \begin{pmatrix} 36 & 66 & 58 \\ 15 & 83 & 35 \\ 19 & 74 & 62 \end{pmatrix}$$

$$C_2 = A^5 M_2(\text{mod}103) = \begin{pmatrix} (18,75) & (68,53) & (100,44) \\ (32,99) & (63,1) & (53,45) \\ (67,56) & (86,42) & (53,62) \end{pmatrix}$$

$$C_2{}^* = A^5 M_2{}^* (\text{mod} 103) = \begin{pmatrix} (22,96) & (86,14) & (0,0) \\ (18,58) & (69,39) & (0,0) \\ (23,81) & (59,21) & (0,0) \end{pmatrix}$$

6. Bob send (C_1, C_2, C_2^*) to Alice

Decryption process

1. Alice compute C_1^{-a}

$$C_1^{-4} = \begin{pmatrix} 22 & 72 & 75 \\ 90 & 36 & 45 \\ 42 & 28 & 23 \end{pmatrix}$$

2. Compute ${C_1}^{-a} C_2$ and ${C_1}^{-a} {C_2}^*$ And

$$C_1^{-4} C_2 (\text{mod} 103) = \begin{pmatrix} (0,0) & (19,62) & (0,0) \\ (19,62) & (1,1) & (6,27) \\ (0,0) & (6,27) & (2,2) \end{pmatrix} = M_2$$

$$C_1^{-a} C_2^* (\text{mod} 103) = \begin{pmatrix} (3,3) & (58,56) & (0,0) \\ (58,56) & (4,4) & (0,0) \\ (0,0) & (0,0) & (0,0) \end{pmatrix} = M_2^*$$

3. She suppose that the first node $(0,0)$ is A and using encoding table she compute:

   $(0,0)$ = code $A = (101, 65)$

   Node $(1,1)$ = code $A + w_1$

   $= ((101, 65) + (19, 62))(\text{mod}103)$

   $= (17, 24) = B$

   Node $(2,2)$ = code $B + w_2$=(23,51)R

   Node $(3,3)$ = code$R + w_3$=(7,90)=H

   Node $(4,4)$ = code $H + w_4 = (65, 34) = C$

   So on until she got the original text (BRHC).

3.9. The Security Considerations

In this section, discussed the security issue of the proposed cryptosystems over $BRH_{\alpha,\beta}$ is discussed as follows .

3.9.1 The Security with Using $BRH_{\alpha,\beta}$ Curve

Cryptography based on elliptic curves is the most commonly used type of asymmetric method. However, it is vulnerable to the power analysis assaults because the point doubling and addition are not united.While $BRH_{\alpha,\beta}$ curve is more resistant to power analysis attacks than Elliptic curve [37] [7], since usees a unified formula for computing the points addition and doubling.

3.9.2 The Security $BRH_{\alpha,\beta}$ -Digital Signature Scheme

The $BRH_{\alpha,\beta}$-DSS is depended on the hardness of $BRH_{\alpha,\beta}$-DLP.so, it is challenging for any attacker to create a fake signature with a good choice of the parameters to implement the $BRH_{\alpha,\beta}-DSA$,since there is no proper method to solve $BRH_{\alpha,\beta}-DLP$. On the proposed $BRH_{\alpha,\beta}-DSS$, two random integers α and x are chosen to create the signature (r, s), as a result, it is impossible to derive a private key from this relationship that allows an opponent to retrieve the signature's parameters r and s.

3.9.3 The Security of $BRH_{\alpha,\beta}$-EGC

The security of the proposed $BRH_{\alpha,\beta}$-EGC depends on the choice of the $BRH_{\alpha,\beta}$ subgraph that corresponds to the $BRH_{\alpha,\beta}$ subgroup of an $BRH_{\alpha,\beta}$ group $BRH_{\alpha,\beta}$ defined over F_p . With a large prime number p, the $BRH_{\alpha,\beta}$ graph has a large number of the vertices. Thus, it is possible to form a large number of the $BRH_{\alpha,\beta}$ subgraph. Choosing one of these subgraph randomly to represent a plaintext gives more secure to communicate using the proposed $BRH_{\alpha,\beta}$ graphic cryptosystem. Eve needs to compute many cases to reach the correct choice of the $BRH_{\alpha,\beta}$ subgraph. Thus, the $BRH_{\alpha,\beta}$ -EGC is more secure and suitable for $BRH_{\alpha,\beta}$ cryptographic communication schemes.

3.9.4 The Security of the $BRH_{\alpha,\beta}$ Matrix EIGamal Graphic Cryptosystem

Some consideration can be determined has for the security of $BRH_{\alpha,\beta}$-matrix-EGC.One of them is the difficulty to compute the scalar multiplication matrix.As well as,the Hardness to compute the DLP over the matrices and the $BRH_{\alpha,\beta}$-DLP .

3.9.5 The Security Considerations of Generated Graph for Text Encryption Scheme Based on $BRH_{\alpha,\beta}$ Curve

The security of this method depending in the difficulty to solve DLP applied on the matrices as well as the use of graph theory to encrypt the plaintexts. The graph theory concepts here considered as an essential tool to increase security through the generation of the spanning tree graphs.

CHAPTER 4

APPLYING OPTIMIZATION ALGORITHMS TO GENERATED THE PRIVATE KEY AND EPHEMERAL KEY

4.1. Introduction

In this chapter, optimization algorithms were used to generated the secret key and ephemeral key which used in ElGamal public key cryptosystem in an optimal way through repeated searches across a large solution space and improving the selection of these keys to achieve certain goals.In section 4.2 , Particale Swarm Optimization Algorithm was explained and applied to find the private keys, and then the time required for encryption and decryption after using these keys was ElGamal Algorithm. In section 4.3, The Cuckoo Algorithm was explained and applied to find the secret keys and calculate the time required for encryption and decryption. In this chapter, Python has used for algorithms and drawing.

4.2. Particle Swarm Optimization Algorithm

The particle swarm ptimization PSO method is a technique of stochastic optimization based on swarm which was proposed by Kennedy and Eberhart 1995. PSO method simulates behavior of animal's social, with herds, insects, fishes and birds. It is a cooperative way to find food.On this method each member in the swarms saves varying the search pattern according to the learning experiences of its own and other members [17, 22]. PSO method is to select a spot with adequate food and to replicate the social behavior of real creatures such as bird flocking and fish schooling. Indeed, without any central supervision, a coordinated

behavior based on local motions arises in such swarms. PSO method was originally created to solve problems involving continuous optimization. In [29, 36] the first applications to optimization issues are presented.

4.2.1 The Used Concepts in PSO Algorithm

The fundamental concepts that are used in the PSO Algorithm can be determined by:

Fitness Function. The Fitness function is the method for determining the best solution. It is usually considered as an objective function.

P_{best}**.** It's the particle's best location out of all the ones it's been to thus far.

G_{best} **.** The position in which all of the particles visited so far have the best fitness.

Velocity Update . Velocity is a vector that determines the particle's speed and direction.

Position Update . Every particle tries to get into the ideal place for maximum fitness. To discover the global optimum, each particle in PSO changes its location.

Some features on the PSO are also determined by

- Avoid collisions with neighboring.
- Stay near neighboring.
- Match the velocity of neighboring.

4.2.2 Mathematical Inter pretution of the PSO Method

Let $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in}) \in R^n$ be a swarem which is a set Xij with i,j=1,2,...,n of particles where a particle is a potential solution. The parameters U_i and V_i are position and volocity of particlesin swaram respetiveiy which are given by

$$V_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{in}) \in R^n$$
$$U_i = (u_{i1}, u_{i2}, u_{i3}, \ldots, u_{in}) \in R^n$$

In each iteration, the particle swarm retains the global optimal position and the current optimal position, and each particle is affected by these two optimal particles and approaches these two particles. The iterative Formulas for updating the velocity and position of the next generation of particles are given by

Where $P_{besti} = U_i$, $G_{best} = P_{besti}$

$$V_i^{k+1} = WV_i^k + c_1 \times r_1 \times \left(P_{\text{besti}} - U_i^k\right) + c_2 \times r_2 \times \left(G_{\text{best}} - U_i^k\right) \quad (4.1)$$

$$U_i^{k+1} = U_i^k + V_i^{k+1} \quad (4.2)$$

$$W = W_{\max} - \left(\frac{(W_{\max} - W_{\min})}{\text{iter}_{\max}}\right) \times iter$$

Where c_1 and $c_2$are two learning factors, r_1,r_2 is a random number between (0,1) and W is the inertia weight is between 0.4 and 0.9.iterative update

of P_{besti} and G_{best} are performed through the fitness value comparison.

$$P_{besti} = U_i \quad if \quad f_{U_i} \quad better \quad f_{P_{besti}}$$

and

$$G_{best} = P_{besti} \quad if \quad f_{P_{besti}} \quad better \quad f_{G_{best}}$$

Here V_1^k represents the rate of agent at $k-th$ iteration and V_1^{K+1} at next iteration U_1^k being current position of the k^{th} iteration agent which is also the position at next iteration. Maximum iteration is denoted by $iter_{max}$ and current iteration by $iter$. Position of the particle is denoted as P_{best} and position of its neighbor by G_{best}. Wherever best is outlined by some perform of the swarm.

The Total Functional Steps are as Shown Below.

Step 1: Initialize the swarm X_i from solution space

Step 2 : For $iter = 1$ to $iter_{max}$ or termination criteria

Step 3 : Evaluate the fitness of each particle

Step 4 : Update personal/global best fitness

Step 5 : Adapt velocity of the particle using equation 4.1.

Step 6 : Update the position of the particle

Step 7 : Publish global best fitness value.

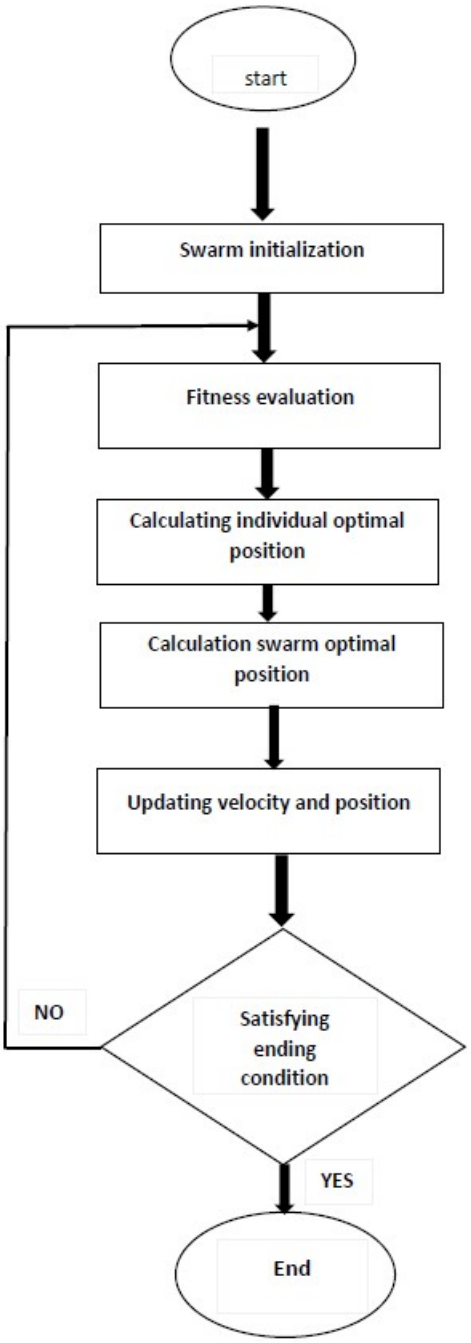


Figure 4.1: The PSO method

4.2.3 The PSO Algorithm to Generated Private Key and Ephememeral Key

The goal of using the algorithm is to generated a private key of ElGamal cryptosystem such that the product of its multiplication with any number in the field is small , provided that it does not equal one. In addition, public key for first users and public key for second user is calculated in a short time .The security of choice is maintained by choosing the algorithm's parameters.These phases are discussed as follows.PSO algorithm employed to find ephemeral key of ElGamal cryptosystem on $BRH_{\alpha,\beta}$ curve to lessening time of encryption and decryption .The PSO algorithm initialized a swarm of particles, where each particle represented a potential solution with its own position and velocity.The position of a particle corresponded to a unique combination of private key value. Initially, the particles are randomly distributed across the search space.In each iteration, the particles updated their velocities and positions based on their own best-known position and the best-known position among all particles in the swarm.This process aimed to balance exploration and exploitation by incorporating both personal and global information.The velocities controlled the movement of the particles, allowing them to explore different regions of the search space. During the updated process.The algorithm continues for a predefined number of iterations or until a termination condition is met.The private key values are determined based on the best position which is corresponded to the solution with the lowest multipling a private key.

Algorithm 13: Using PSO to Generated Private Key and Ephemeral Key

Input: A prime number P.

Output: Private key

1. Initialize the population of particles for the Particle Swarm Optimization (PSO) algorithm(for each particle i in a swarm population size p).
2. Define fitnes function $f(a) = \alpha a$, where $\alpha \in [1, P-1]$and $\alpha a \neq 1$
3. Set the iterations number as $t = 0 + 1$ randomly initialize.
4. Initialize a_{besti} where a_{besti}=a_i
5. Initialize G_{best} where G_{best}=a_i
6. For $i = 1, ..., p$
7. $V_i^{t+1} = wV_i^t + c_1 \times r_1 \times \left(a_{best_i} - a_i^t\right) + c_2 \times r_2 \times \left(G_{best} - a_i^t\right)$
8. $a_i^{t+1} = a_i^t + V_i^{t+1}$
9. If $f(a_i^t) \leq f(a_{besti})$ then a_{besti}=a_i
10. If $f(a_i^t) \leq f(G_{best})$ then G_{besti}=a_i
11. Stop until theshold iteration.
12. Update the global solution if a better solution is found among all particles, then $a_{best} = G_{best}$

Where

v_i is Velocity of particle i.

a_i is Position of private key of particle i

w is Inertia weight

c_1, c_2 is Acceleration coefficients

r is Random number between 0 and 1

Algorithm 14: The $BRH_{\alpha,\beta}$ curve- EIGamal PKC Based PSO Encryption process

---

Input: The cofficiants α and β of $BRH_{\alpha,\beta}$ curve , Q point $\in BRH_{\alpha,\beta}$ curve .

Output: The ciphertext (C_1, C_2).

1. Use Algorithm13 to cumpute private key a,compute $A = aQ$.
2. Use Algorithm13 to cumpute ephemeral key b.
3. Choose aplaintext M $\in BRH$ curve.
4. Compute $C_1 = bQ$ and $C_2 = M + bA$,where $C_1, C_2 \in BRH_{\alpha,\beta}$ curve.
5. Return (C_1, C_2).

Algorithm 15: Decryption Proces

Input: Aprivet key a, $C_1, C_2 \in BRH_{\alpha,\beta}$ curve which are the computonents of the ciphertext.

Output: Original plaintext M.

1. Compute ascalar multiplication aC_1 ,where $aC_1 \in BRH_{\alpha,\beta}$ curve .
2. Compute an inverse point of aC_1 , which is $-aC_1 \in BRH_{\alpha,\beta}$ curve.
3. Compute $M = C_2 - aC_1$.
4. Returen M.

Example 4.2.1. Let $p = 313$ be a prime number.

$x(y^2-1) = 2y(x^2-1) (\text{mod } 313)$ is the $BRH_{\alpha,\beta}$ curve and $Q = (202, 120)$ $\in BRH_{1,2}$ curve.

Alice uses algorithm 13 to compute her private key a

a = 63 and compute her public key $A = (52, 214) \in BRH_{1,2}$ curve by Algorithem 14

Bob choose his plaintext $M = (14, 218) \in BRH_{1,2}$ curve

The best ephemeral b= 228 by algorithm 13

Bob computes $C_1 = (86, 223)$ and $C_2 = (288, 147)$ and sends it to Alice.

After Alice receiving (C_1, C_2)

She compute $C_2 - aC_1 = (288, 147) - 63(86, 223)$

$$= (288, 147) - (78, 200)$$
$$= (288, 147) + (235, 113)$$
$$= (14, 218)$$
$$= \text{M}$$

The computations are done using Algorithm14with time is equal to 1.0008585 sec. While ,the time to decryption the ciphertext and recover M by Algorithm 15 is 1.001426 sec.and the time to compute best privet key a and ephemeral b using Algorithm 13 is 1.2857267 sec.

Figure shows all points on $BRH_{1,2}$ curve especially the points Q, A, C_1, C_2.

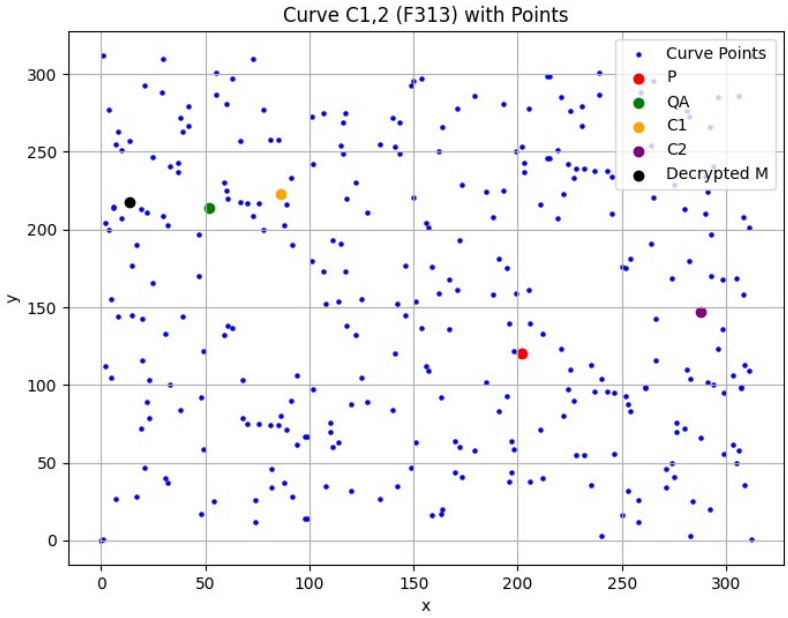


Figure 4.2: Shows the public point Q,the ciphertext point (C_1, C_2) and the plaintext point on $BRH_{1,2}$ curve over F_{313}

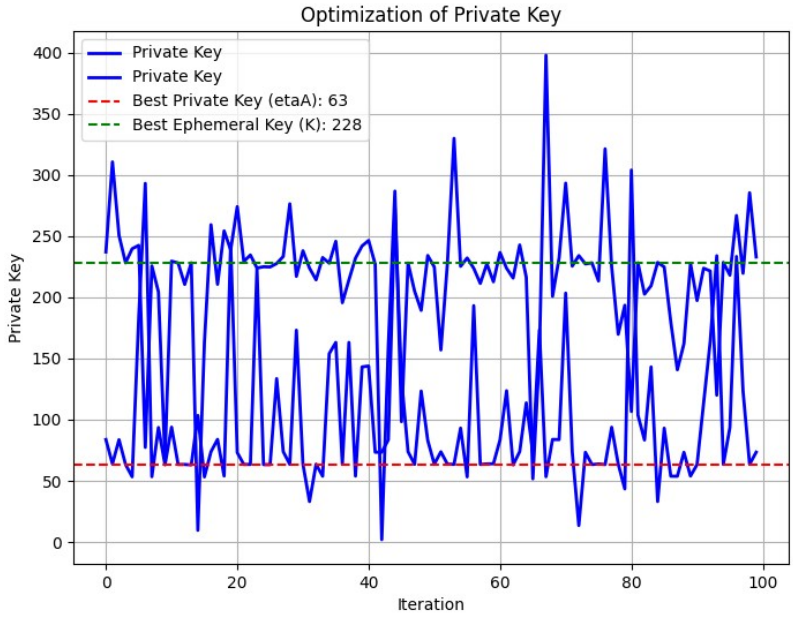


Figure 4.3: 0ptimization of private key and ephemeral key b

4.3. Cuckoo Optimization Algorithm

The Cuckoo Optimization Algorithm was inspired by the life of the 'cuckoo' [61].This optimization approach is based on this bird's specialized breeding and egg laying. Adult cuckoos and eggs were used to make this model. Adult cuckoos lay eggs in the nests of other birds. If the host bird does not find and remove the eggs, they develop into a full cuckoo. The migration of cuckoo groups and environmental conditions cause them to converge and select the best area for breeding . This is the best position for the goal function. After being inspired by nature, Yang and Deb founded Cuckoo Optimization in 2009 [61]. In 2011, Rajabioun developed the Cuckoo Optimization Algorithm [47].Cuckoo Optimization Algorithm (COA) is a novel continuous all-aware search algorithm based on the life of a cuckoo bird. COA, like other Meta heuristics, starts with a main population, a group of cuckoos. Cuckoos deposit eggs in their environment .of other hosts. A random set of possible solutions representing the habitat in COA is produced.

4.3.1 Cuckoo Breeding Behavior Strategy

The COA algorithm was inspired by some cuckoo species' obligatory brood parasitism, in which they lay their eggs in the nests of host birds. Some cuckoos have been implicated in such a manner that female parasitic cuckoos may resemble the colors and patterns of a few selected host species' eggs.This minimizes the likelihood of the eggs being abandoned, increasing re-productivity. It is worth noting that numerous

host birds are in direct combat with invading cuckoos. If the eggs are not their own, the host birds will either toss them away or abandon their nests and construct new ones. Parasitic cuckoos frequently seek nests where the host bird has recently placed its own eggs. In general, Cuckoo eggs hatch somewhat earlier than host eggs in general. When the first cuckoo chick hatches, his first reaction is to evict the host eggs by hurling them out of the nest. This behavior increases the cuckoo chick's proportion of the food offered by its host bird . Furthermore, research [21, 62] shows that a cuckoo chick may replicate the call of host chicks in order to obtain access to additional feeding opportunities. Cuckoo breeding behavior may be used to a variety of optimization challenges.

4.3.1.1 Levy Flights mechanism

In nature, animals look for food in a random or quasirandom manner. An animal's foraging path is essentially a random walk since the next move is determined by both the present location/state and the transition probability to the next site. The likelihood of the chosen directions is quantitatively modelled. Several investigations have indicated that the flying behavior of numerous animals and insects resembles that of Levy flights. A Levy flight is a random walk with step lengths determined by a heavy-tailed probability distribution. After a long number of steps, the distance from the origin of the random walk tends to a stable distribution.

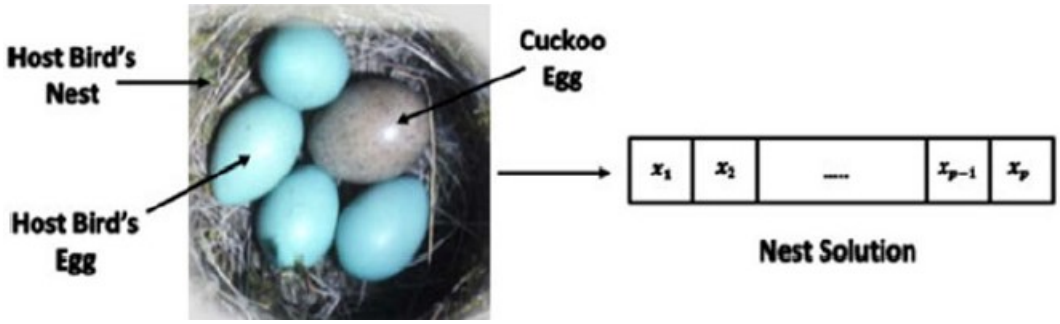


Figure 4.4: Representation of a nest solution in the Cuckoo search algorithm [47].

4.3.1.2 Mathematical COA Method

The algorithm [61] is based on the obligate brood parasitic behavior found in some cuckoo nests by combining a model of this behavior with the principles of Lévy flights, which discard worst solutions and generate new ones after some certain iteration.According to the mentioned characteristics, COA can be expressed as three idealized rules:

1. Each cuckoo lays one egg at a time, and places it in a randomly chosen nest.

2. The best nests with the highest-quality eggs (solutions) will be carried over to the next generations.

3. The number of available host nests is fixed, and the alien egg is discovered by the host bird with the probability $pa \in [0, 1]$. If the alien egg is discovered, the nest is abandoned and a new nest is built in a new location.

   The COA is equiponderant to the integration of Levy flights. The position of the i-th nest is indicated by using D-dimensional vector

$X_i = \{x_{i1}, x_{i2}, x_{i3}...x_{id}\}$

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus Levy(\lambda) \tag{4.3}$$

where $\alpha > 0$ is the step size that is used to control the range of the random search, which should be related to the scales of the problem of interests, and step size information is more useful can be computed by Equation $\alpha = \alpha_0 \oplus (x_i^t - x_j^t)$The product means entry-wise multiplications.x_i^t and x_j^t are two different solutions selected randomly. A new solution with the same number of cuckoos is generated after partial solutions are discarded. levy(λ) with the random walk can be expressed in terms of a simple power-law equation 4.3.

where u and t are two random numbers following the normal distribution and λ often takes a fixed value of 1.5.

$$levy(\lambda) \sim u = t^{-\lambda}, 1 < \lambda \leq 3 \tag{4.4}$$

$$levy(\lambda) \sim \frac{\phi \times u}{|v|^{1\backslash\lambda}} \tag{4.5}$$

$$\phi = \left[\frac{\Gamma(1+\lambda) \times sin(\frac{\pi \times \lambda}{2})}{\Gamma(\frac{\pi \times \lambda}{2}) \times \lambda \times 2^{(\frac{\pi \times \lambda}{2})}}\right]^{1\backslash\lambda} \tag{4.6}$$

where Γ is gamma function.u and v are random numbers drawn from a normal distribution with mean of 0 and standard deviation of 1, which have an infinite variance with an infinite mean. Here, the consecutive jumps/steps of a cuckoo essentially form a random walk

process that obeys a power-law step length distribution with a heavy tail. In Levy flights random walk component, the new solution X_i is generated through Equation 4.7

$$X_i^{t+1} = X_i^t + \alpha_0 \frac{\phi \times u}{|v|^{1\backslash\lambda}}(X_i^t - X_{best}^t) \tag{4.7}$$

Where X_{best}^T represents the best solution obtained thus far and α_0 is a scaling factor. The Levy distribution is a process of random walk; after a series of smaller steps, Levy flights can suddenly obtain a relatively larger step size. Levy distribution is implemented at the initial stage of algorithm, which helps to jump out of the local optimum.

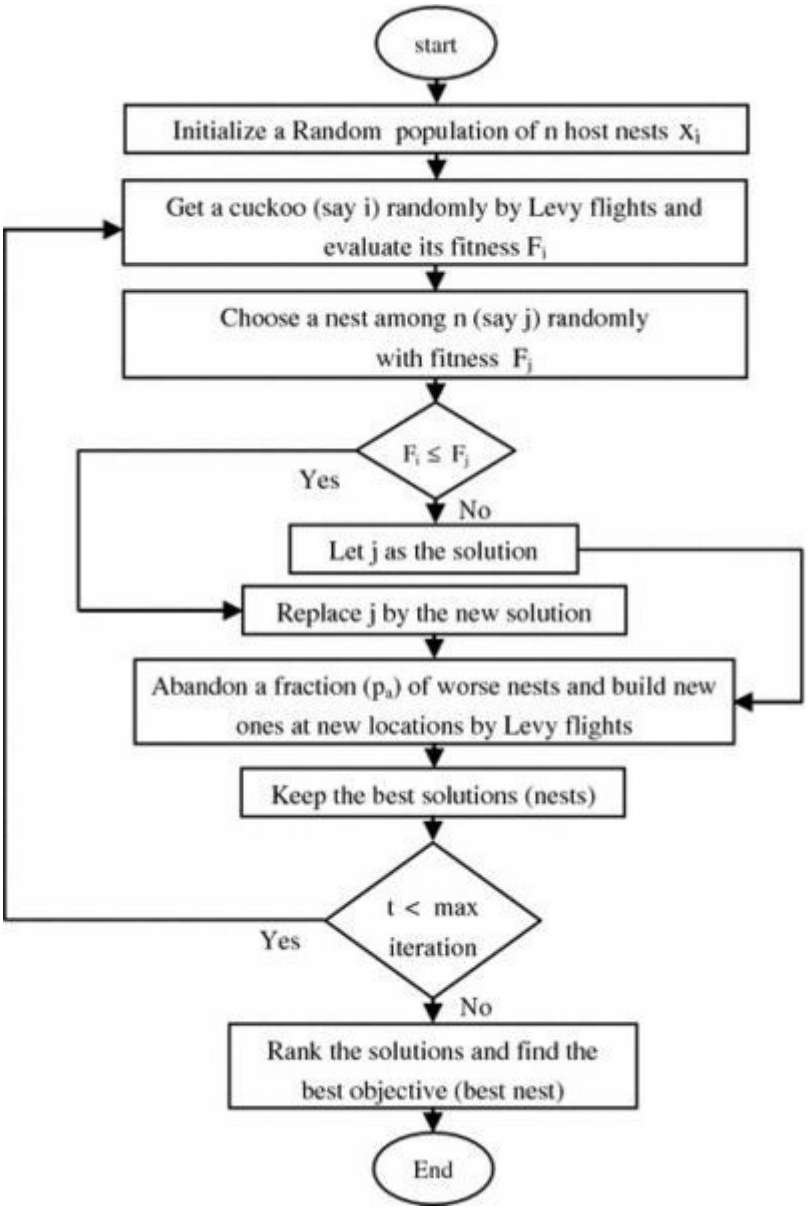


Figure 4.5: Using COA method

Algorithm 16: COA Algorithm

1. randomly initialize population of n host nests$X_i = \{x_1, x_2, x_3, ...x_n\}$ where i=1,2,3,...n .
2. calculate fitness value for each solution in each nest.
3. Generate x_i^{t+1} as new solution by using Levy flights 4.3.
4. Choose candidate solution x_i^t
5. If $f(x_i^t) > f(x^{t+1}{}_i)$
6. Replace x_i^t with new solution x_i^{t+1} .
7. end if
8. Throw out a fraction (pa) of worst nests.
9. keep best solution
10. If $f(x_i^t) > f(x_i^{t+1})$
11. Replace x_i^t with new solution x_i^{t+1} .
12. end if
13. Rank the solution and find the current best.
14. end while

4.3.2 Applying Cuckoo Algorithms to Generated Private key and Ephemeral Key

The Cuckoo Search algorithm was utilized to generated private key and ephemeral key for a ElGamal cryptographic scheme , so that the product of multiplying this key with any number in the field is as small as possible. provided that it does not equal one. In addition, public key for first users and public key for second user is calculated in a short time .The security of choice is maintained by choosing the algorithm's parameters. for the purpose of speeding up the encryption and decryption process while maintaining the security of information and the difficulty of decoding the encrypted point and the public key. The algorithm started by randomly generating a population of candidate solutions, represented as nests. Each nest corresponds to a unique combination of private key and ephemeral key values. In each iteration.The algorithm continued for a predefined number of iterations or until a termination condition was met. The private key and ephemeral key values were determined based on the nest with the lowest multipling a private key a and ephemeral key b.

Algorithm 17: Using COA Method to Generated Private Key and Ephemeral Key.

Input: A prime P.

output: Private key a.

1. randomly initialize population of P host nests $X_i = \{x_1, x_2, x_3, ...x_P\}$ where $i = 1, 2, 3, ...P$.
2. Choose candidate solution x_i^t,where $x_i^t = a$
3. Generate x_i^{t+1} as new solution by using Levy flights (4.3).
4. calculate fitness $f(a) = \alpha a$ value for each solution in each nest, where $\alpha \in$[1,P-1].
5. If $f(x_i^t) > f(x^{t+1}{}_i)$
6. Replace x_i^t with new solution x_i^{t+1}.
7. end if
8. Throw out a fraction (pa) of worst nests.
9. Keep best solution
10. If $f(x_i^t) > f(x_i^{t+1})$
11. Replace x_i^t with new solution x_i^{t+1} .
12. end if
13. Rank the solution and find the current best.
14. end if.
15. x_i^{t+1}= private key a
16. end

Example 4.3.1. Used the same curve in example (4.2.1)

$x(y^2-1) = 2y(x^2-1)(mod\,313)$ is the $BRH_{\alpha,\beta}$ curve and $Q = (202, 120)$ $\in$ $BRH_{1,2}$ curve. Alice uses Algorithm 17 to compute her private key a=9 and compute her public key $A = (212, 133)$ by use the same way in Algorithm 14.

Bob choose his plaintext (14,218).

The best ephemeral b =7 that is compute also by using Algorithm 17

Bob computes $C_1 = (221, 285), C_2 = (167, 168)$ and sendes to Alice .

After Alice receiving (C_1, C_2)

She compute $C_2 - aC_1 = (167, 168) - 9(221, 285)$

=(14,218)

=M

The computations are done using same way in Algorithm14with time is equal to 0.140871763 sec. While ,the time to decryption the ciphertext and recover M b is 1.005391598 sec.and the time to compute private key a and ephemeral b using Algorithm 17 is1.00540209 sec.

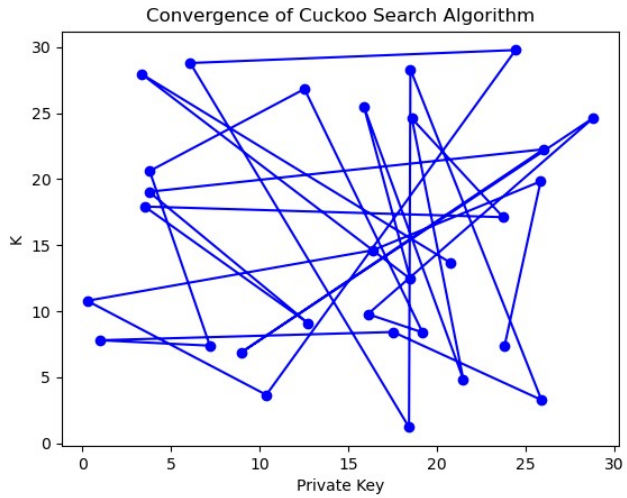


Figure 4.6: Convergence of Cuckoo search algorithm

4.4. The Security of Applying Optimization Algorithms to Generated Private Key and Ephemeral Key in ElGamal with $BRH_{\alpha,\beta}$

The use of optimization algorithms to generate private Key and ephemeral key in ElGamal with $BRH_{\alpha,\beta}$ offers several potential benefits. First, it can enhance the computational efficiency of key generation, encryption, and decryption processes, making them faster and more scalable. Second, optimization algorithms can help strengthen the security of the scheme by finding optimal key parameters that are less susceptible to attacks. Third, these algorithms provide a framework for exploring the trade-off between efficiency and security, allowing users to adjust parameters based on their specific requirements.

The security considerations for Particle Swarm Optimization (PSO) Algorithms and Cuckoo Search Algorithm (COA) in the context of optimizing the private keys for the ElGamal encryption scheme with $BRH_{\alpha,\beta}$ curve are as follows. Similar to the Particle Swarm Optimization (PSO) algorithm, The Cuckoo Search algorithm, inspired by the behavior of cuckoos in nature, provides notable benefits when applied to ElGamal BRH:

Enhanced Security: By leveraging the algorithms, the selection of private and ephemeral keys is optimized, resulting in keys that exhibit enhanced security properties. The algorithm considers constraints and

incorporates a cost function to guide the key selection process. These constraints, such as avoiding keys that are multiples of certain values or following specific patterns, contribute to the overall security of the encryption scheme. The optimization process ensures that the keys are more difficult to guess or compute, significantly raising the bar for potential code breakers.

Efficient Key Selection: The algorithms efficiently explores the search space to ge the private and ephemeral keys, satisfying the specified constraints and optimizing the cost function. Through the iterative application of mutation, crossover, and selection operations, the algorithm progressively improves the population of candidate solutions. This results in efficient key selection, reducing the time and computational complexity required to identify optimal keys. The exploration capabilities of the algorithm, combined with its ability to converge towards the optimal solution, contribute to the overall efficiency of the encryption and decryption processes.

Trade-Offs and Complexity: It is important to note that the integration of (PSO)and (COA) algorithms introduces certain trade-offs and complexities. The optimization process requires careful parameter tuning, such as population size, mutation rates, and convergence criteria, to achieve optimal results. Additionally, the computational requirements associated with exploring large search spaces and performing numerous iterations should be considered. Balancing these factors to achieve the desired level of security while maintaining acceptable performance becomes crucial.

CHAPTER 5

MORE IMPLEMENTED RESULTS ABOUT SUGGESTED CRYPTOSYSTEMS

5.1. Introduction

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.

5.2. The Results $BRH_{\alpha,\beta}$ Diffie-Hellman Key Exchange

With different values of a small (or a large) prime p; the computations of the BRH- Diffe-Hellman encryption scheme have been done as shown in Tables 5.1 and 5.2 respectively.

Table 5.1: The experimental results of $BRH_{\alpha,\beta}$ Diffie-Hellman Key Exchange.(Part 1)

P	Shear key D	Alice's secret key a	$A = aD(\mathrm{mod}p)$
191	$(114, 71)$	7	$(135, 15)$
311	$(307, 162)$	4	$(302, 47)$
421	$(416, 363)$	19	$(396, 125)$
997	$(995, 221)$	16	$(481, 206)$
1009	$(1000, 470)$	23	$(901, 434)$

Table 5.2: The experimental results of $BRH_{\alpha,\beta}$ Diffie-Hellman Key Exchange.(Part 2)

Bob's secret key P	$B = bD(\mathrm{mod}p)$	$bA = aD(\mathrm{mod}p)$
6	$(48, 39)$	$(147, 52)$
8	$(176, 72)$	$(69, 113)$
5	$(407, 141)$	$(268, 82)$
8	$(410, 666)$	$(650, 187)$
5	$(565, 792)$	$(25, 792)$

5.3. The Results Public-Key Cryptosystem $BRH_{\alpha,\beta}$ - ElGamal

Some simple computations of the $BRH_{\alpha,\beta}$-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 5.3 5.4 and 5.5 respectively.

Table 5.3: The experimental results of $BRH_{\alpha,\beta}$-ElGamal public key cryptosystem: key generation process

P	Shear key D	The private key a	public key $A \equiv aD(\mathrm{mod}p)$
197	$(194, 74)$	6	$(74, 1)$
383	$(204, 144)$	14	$(208, 19)$
983	$(592, 871)$	70	$(922, 95)$
1009	$(1002, 18)$	194	$(733, 198)$
1061	$(685, 140)$	34	$(567, 465)$

Table 5.4: The experimental results of BRH-ElGamal public key cryptosystem:(encryption process).

P	Shear key D	The private key a	public key $A \equiv aD(\mathrm{mod}p)$
197	$(194, 74)$	6	$(74, 1)$
383	$(204, 144)$	14	$(208, 19)$
983	$(592, 871)$	70	$(922, 95)$
1009	$(1002, 18)$	194	$(733, 198)$
1061	$(685, 140)$	34	$(567, 465)$

Table 5.5: The experimental results of $BRH_{\alpha,\beta}$ -ElGamal public key cryptosystem:(encryption process) .

$aC_1(\mathrm{mod}p)$	$-aC_1(\mathrm{mod}p)$	$C_2 - aC_1M(\mathrm{mod}p)$
(52,181)	(145, 16)	(61,20)
(312,342)	(71, 41)	(303,184)
(378,460)	(605,523)	(102,803)
(885,987)	(124, 22)	(123,566)
(285,241)	(776,820)	(992,211)

5.4. The Results on the $BRH_{\alpha,\beta}$ Curve in Digital Signature Scheme.

With different values of a small (or a large) prime p; the computations of applying the $BRH_{\alpha,\beta}$ curve in digital signature scheme have been done as shown in Tables 5.6,5.7,5.8and 5.9respectively.

Table 5.6: The experimental results of $BRH_{\alpha,\beta}$ Curve in Digital Signature Scheme: key generation process.

P	A public key G	A private key x	A public key $Q \equiv G(\mathrm{mod}p)$
383	(97,46)	3	(289,45)
457	(454,240)	50	(297,251)
577	(572,440)	10	(387,215)
1013	(1006,363)	97	(284,830)
1061	(269,632)	98	(274,57)

Table 5.7: The experimental results of $BRH_{\alpha,\beta}$ Curve in Digital Signature Scheme: Signature generation process.(part 1)

A private key	$\alpha G(\mathrm{mod}p)$	$x_1 = r$
10	$(261, 45)$	261
25	$(43, 102)$	43
12	$(143, 433)$	143
88	$(957, 463)$	957
2	$(832, 488)$	832

Table 5.8: The experimental results of $BRH_{\alpha,\beta}$ Curve in Digital Signature Scheme: Signature generation process. (part 2)

A plaintext$m$	$S = \alpha^{-1}(m + rx)(\mathrm{mod}p)$	Signature of plaintext $m(r, s)$
7	79	$(261, 79)$
13	379	$(43, 379)$
20	217	$(143, 217)$
90	653	$(957, 653)$
99	1030	$(832, 1030)$

Table 5.9: The experimental results of $BRH_{\alpha,\beta}$ Curve in Digital Signature Scheme: Signature Verification Process.

$s^{-1}(\mathrm{mod}p)$	$U = ms^{-1}(\mathrm{mod}p)$	$v = rs^{-1}(\mathrm{mod}p)$	$w = UG + vQ(\mathrm{mod}p)$
160	354	13	(261,45)
41	76	392	(43,102)
117	32	575	(143,433)
892	253	698	(957,463)
308	784	555	(832,488)

5.5. Case Study of Use of Graphs for Some Cryptosystems

Suppose $BRH_{\alpha,\beta}$ curve is define by $BRH_{5,12} : x\left(5y^2 - 1\right) = 12y\left(x^2 - 1\right)$ over the prime field F_{191}.

The public matrix

$$T = \begin{pmatrix} 3 & 2 & 99 & 5 & 43 & 89 & 10 \\ 1 & 5 & 8 & 45 & 33 & 87 & 56 \\ 98 & 45 & 22 & 98 & 4 & 7 & 6 \\ 34 & 7 & 12 & 76 & 77 & 4 & 9 \\ 5 & 3 & 6 & 2 & 22 & 55 & 33 \\ 11 & 100 & 8 & 4 & 6 & 71 & 34 \\ 56 & 12 & 90 & 65 & 40 & 37 & 112 \end{pmatrix} \in GL_7\left(F_{191}\right)$$

Now, Alice performs the following steps:

1. She chooses her private key $a = 7$
2. he computes her public key A by

$$A = T^7(\mathrm{mod}191) = \begin{pmatrix} 3 & 168 & 129 & 80 & 29 & 115 & 123 \\ 75 & 187 & 142 & 137 & 107 & 49 & 3 \\ 80 & 187 & 103 & 68 & 19 & 71 & 70 \\ 188 & 46 & 70 & 185 & 60 & 44 & 107 \\ 30 & 116 & 127 & 54 & 8 & 127 & 43 \\ 41 & 62 & 165 & 178 & 83 & 176 & 111 \\ 53 & 27 & 170 & 141 & 188 & 131 & 74 \end{pmatrix}$$

Bob performs the following steps:

1. chooses his plaintext M

   $M = \{(0,0),(2,25),(189,166),(2,55),(189,163),(36,170),(155,21),$
   $(36,171),(155,20)\} \in BRH_{5,12}(F_{191})$

2. Represented M by weighted subgraph $BRHsg_{5,12,}(F_{191})$.

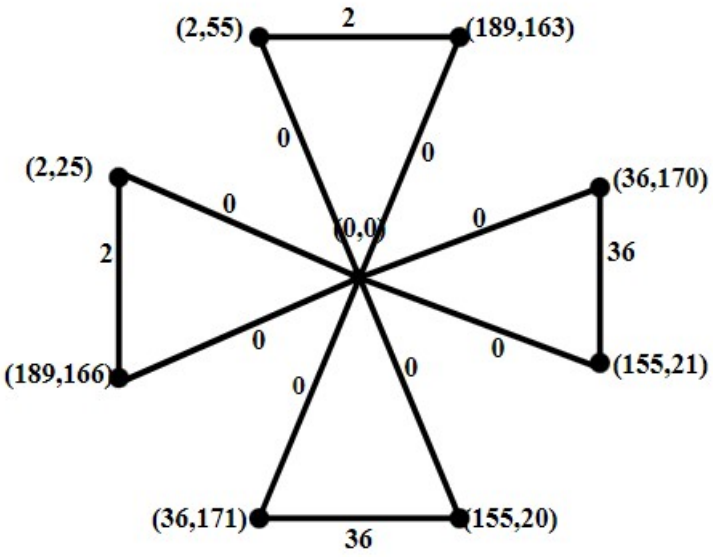


Figure 5.1: weighted subgraph $BRHsg_{5,12}(F_{191})$

3. converts $BRHg_{5,12}$ curve weighted subgraph $BRHsg_{5,12,}(F_{191})$ into another $BRH_{5,12}$ curve graph $BRHsg^{*}_{5,12,}(F_{191})$ as seen in Figure 5.2

.

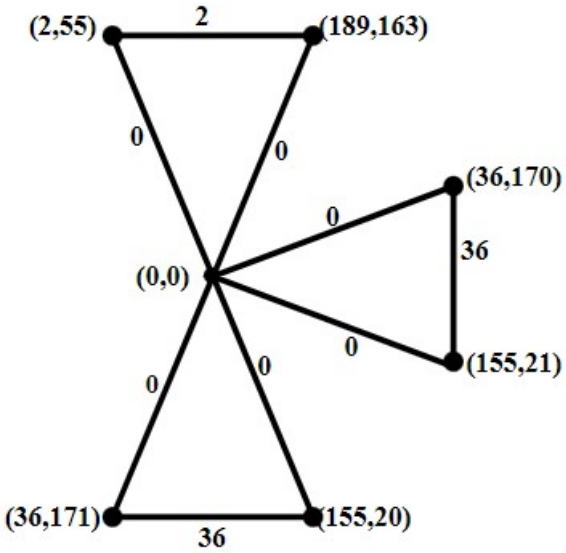


Figure 5.2: weighted subgraph $BRHsg^{*}_{5,12}(F_{191})$.

4. He computed an adjacent matrix B of $BRHsg^{*}_{5,12}(F_{191})$ by

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 36 & 0 & 0 \\ 0 & 0 & 0 & 36 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 36 \\ 0 & 0 & 0 & 0 & 0 & 36 & 0 \end{pmatrix}$$

5. He chooses a secret key $b = 4$ and computes C_1

$$C_1 = \begin{pmatrix} 35 & 102 & 140 & 163 & 155 & 23 & 118 \\ 175 & 161 & 30 & 143 & 115 & 7 & 140 \\ 3 & 140 & 19 & 166 & 187 & 82 & 146 \\ 167 & 61 & 163 & 59 & 134 & 100 & 36 \\ 171 & 26 & 154 & 175 & 169 & 180 & 186 \\ 55 & 2 & 176 & 11 & 167 & 23 & 5 \\ 181 & 127 & 151 & 14 & 105 & 68 & 167 \end{pmatrix}$$

And he computes C_2

$$C_2 = \begin{pmatrix} 0 & 10 & 24 & 157 & 86 & 163 & 14 \\ 0 & 76 & 117 & 47 & 124 & 72 & 50 \\ 0 & 176 & 58 & 77 & 113 & 98 & 99 \\ 0 & 142 & 129 & 146 & 93 & 50 & 27 \\ 0 & 167 & 47 & 110 & 31 & 167 & 41 \\ 0 & 102 & 135 & 26 & 111 & 43 & 56 \\ 0 & 140 & 104 & 92 & 184 & 13 & 120 \end{pmatrix}$$

6. He sends the cipher text pair $(\mathrm{C}_1, \mathrm{C}_2)$ to Alice.

Alice performs the following steps to decrypt and recover the original plaintext

1. She computes first

$$\left(C_1^7\right)^{-1} (\mathrm{mod} 191) = \begin{pmatrix} 50 & 26 & 177 & 22 & 146 & 75 & 12 \\ 46 & 14 & 123 & 83 & 175 & 125 & 172 \\ 135 & 22 & 85 & 68 & 22 & 63 & 60 \\ 4 & 94 & 159 & 47 & 161 & 88 & 160 \\ 145 & 147 & 62 & 171 & 16 & 147 & 36 \\ 30 & 26 & 117 & 10 & 68 & 183 & 175 \\ 144 & 169 & 79 & 28 & 54 & 35 & 189 \end{pmatrix}$$

2. She computes a matrix B by

$$\left(C_1^7\right)^{-1} \times C_2(\text{mod}191) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 36 & 0 & 0 \\ 0 & 0 & 0 & 36 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 36 \\ 0 & 0 & 0 & 0 & 0 & 36 & 0 \end{pmatrix} = B$$

3. She represents matrix B by weighted graph in Figure 5.2

4. She gets the original plaintext

   $M = \{(0,0),(2,25),(189,166),(2,55),(189,163),(36,170),(155,21),$
   $(36,171),(155,20\}$

   By using the weighted of edges graph.

5.6. The $BRH_{\alpha,\beta}$ Matrix-ElGamal Graphic Cryptosystem

Let P=1033 prime number.Suppose $BRH_{\alpha,\beta}$ curve defined by $BRH_{200,13}$=$x(200y^2-1) = 13y(x^2-1)$ over a prime F_{1033}. The public matrix

$$T = \begin{pmatrix} (117,885) & (171,909) & (229,743) & (918,676) & (229,545) \\ (535,870) & (312,237) & (957,121) & (395,434) & (118,30) \\ (1017,431) & (340,659) & (721,401) & (918,676) & (340,975) \\ (16,602) & (737,124) & (340,975) & (742,205) & (698,914) \\ (723,858) & (117,885) & (348,51) & (345,720) & (395,434) \end{pmatrix} \in M\left(BRH_{200,13}\left(F_{1033}\right)\right)$$

Now, Alice performs the following steps:

1. She chooses her private key $a = 30$
2. She computes her public key A

   So, her keys are $a = 30$ and A

$$A = 30T(\text{mod}1033) = \begin{pmatrix} (849,47) & (974,139) & (747,57) & (817,399) & (286,976) \\ (171,909) & (817,422) & (570,198) & (178,814) & (712,78) \\ (171,25) & (571,725) & (817,422) & (817,399) & (462,308) \\ (862,1008) & (974,139) & (462,308) & (199,815) & (679,783) \\ (838,809) & (849,47) & (284,974) & (971,798) & (178,814) \end{pmatrix}$$

Bob does the following steps:

1. He chooses a secret key $b = 20$.

2. He computes $C_1 = bT(modp)$.

$$C_1 = \begin{pmatrix} (462,218) & (846,624) & (354,814) & (547,286) & (679,219) \\ (694,666) & (486,747) & (525,242) & (321,955) & (644,10) \\ (339,367) & (35,327) & (486,747) & (547,286) & (998,706) \\ (694,666) & (846,624) & (998,706) & (35,327) & (712,78) \\ (862,124) & (462,218) & (855,783) & (923,611) & (321,955) \end{pmatrix}$$

3. He computes $20A(\mathrm{mod}1033)$.

$$20A(\mathrm{mod}1033) = \begin{pmatrix} (428,543) & (311,787) & (508,791) & (794,310) & (525,242) \\ (846,624) & (794,310) & (794,310) & (116,197) & (93,21) \\ (187,409) & (389,1023) & (794,310) & (239,723) & (644,10) \\ (846,624) & (311,787) & (644,10) & (389,1023) & (917,836) \\ (59,894) & (428,543) & (525,242) & (339,367) & (116,197) \end{pmatrix}$$

4. He chooses his plaintext M

   $M = \{(311,787),(940,246),(794,1023),(644,10),(0,0)\}$

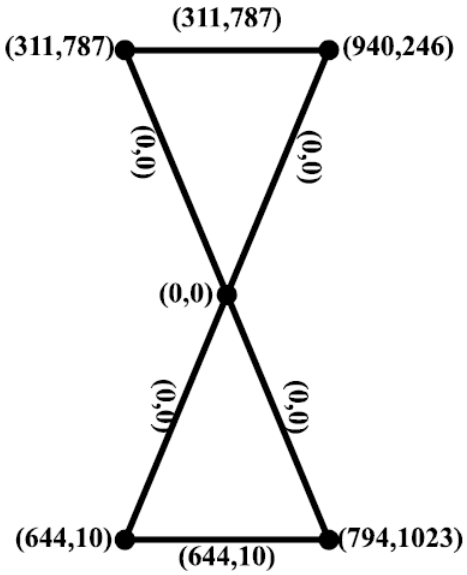


Figure 5.3: The $BRH_{\alpha,\beta}$ curve weighted subgraph $BRHsg_{\alpha,\beta}(F1033)$

$$B = \begin{pmatrix} (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (311,787) & (0,0) & (0,0) \\ (0,0) & (311,787) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (644,10) \\ (0,0) & (0,0) & (0,0) & (644,10) & (0,0) \end{pmatrix}$$

5. He computes $C_2 = (20A + B)(\mathrm{mod}1033)$

$$C_2 = \begin{pmatrix} (428,543) & (311,787) & (508,791) & (794,310) & (525,242) \\ (846,624) & (794,310) & (846,624) & (116,197) & (93,21) \\ (187,409) & (917,836) & (794,310) & (239,723) & (644,10) \\ (846,624) & (311,787) & (644,10) & (389,1023) & (311,787) \\ (59,894) & (428,543) & (525,242) & (547,286) & (116,197) \end{pmatrix}$$

6. Bob send (C_1, C_2) to Alice To decrypt and recover the original plaintext, Alice performs the following steps:

   - She computes $-aC_1$

$$-aC_1(\mathrm{mod}1033) = \begin{pmatrix} (605,490) & (722,246) & (525,242) & (239,723) & (508,791) \\ (187,409) & (239,723) & (239,723) & (917,836) & (940,1012) \\ (846,624) & (644,10) & (239,723) & (794,310) & (389,1023) \\ (187,409) & (722,246) & (389,1023) & (644,10) & (116,197) \\ (974,139) & (605,490) & (508,791) & (694,666) & (917,836) \end{pmatrix}$$

   - She computes $(C_2 - 20C_1)\,(\mathrm{mod}1033) =$

$$\begin{pmatrix} (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (311,787) & (0,0) & (0,0) \\ (0,0) & (311,787) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (644,10) \\ (0,0) & (0,0) & (0,0) & (644,10) & (0,0) \end{pmatrix} = \mathrm{B}$$

   Alice represents a matrix B by a weighted graph in Figure (5.3)

and gets the original plaintext

5.7. Generated Graph for Text Encryption Algorithm Based on $BRH_{\alpha,\beta}$ Curve

Let $P = 1093$ be a prime number. Suppose $BRH_{\alpha,\beta}$ curve defined over a prime field F_{1093}.

$$BRH_{5,7} : x(5y^2 - 1) = 7y(x^2 - 1)(\text{mod } 1093)$$

Choose $Q = (1091, 384)$ lies on $BRH_{5,7}$. Then a letter encoding table is generated.

Table 5.10: Encoding Table

A	B	C	$\cdots$	H	$\cdots$	X	Y	Z
Q	$2Q$	$3Q$	$\cdots$	$8Q$	$\cdots$	$24Q$	$25Q$	$26Q$

And choose public matrix

$$\Upsilon = \begin{pmatrix} 55 & 100 & 88 & 4 & 3 \\ 99 & 31 & 78 & 43 & 76 \\ 56 & 775 & 3 & 92 & 133 \\ 776 & 345 & 143 & 98 & 134 \\ 657 & 77 & 89 & 34 & 98 \end{pmatrix} \in GL_5\left(F_{1093}\right)$$

Now, Jen performs the following steps:

1. She chooses her private key a = 40.

2. She computes her public key Υ

$$A = \Upsilon^{40}(\text{mod}1093) = \begin{pmatrix} 58 & 900 & 935 & 550 & 714 \\ 201 & 967 & 674 & 555 & 1013 \\ 657 & 311 & 800 & 687 & 253 \\ 320 & 734 & 71 & 196 & 718 \\ 960 & 809 & 64 & 496 & 129 \end{pmatrix}$$

   Jen's keys are a =40 and a matrix A.

Now, Bob does the following steps:

1. He chooses her private key $b = 33$.

2. He compute $C_1 = \Upsilon^b$

$$C_1 = \Upsilon^{33}(\text{mod}1093) = \begin{pmatrix} 375 & 583 & 460 & 273 & 52 \\ 579 & 859 & 198 & 587 & 1012 \\ 261 & 18 & 548 & 1015 & 431 \\ 119 & 731 & 1011 & 158 & 696 \\ 740 & 289 & 338 & 139 & 80 \end{pmatrix}$$

3. He chooses his massage (HI JEN)

4. He convert each letters to vertex

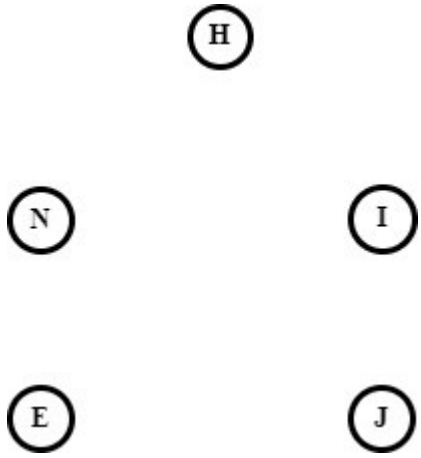


Figure 5.4: Vertices of the Bob plaintext letters.

5. Using the letter encoding table, the message is written in a form $M = \{(792, 453), (959, 320), (1063, 561), (308, 839), (190, 270)\}$. And compute the weight of each edge

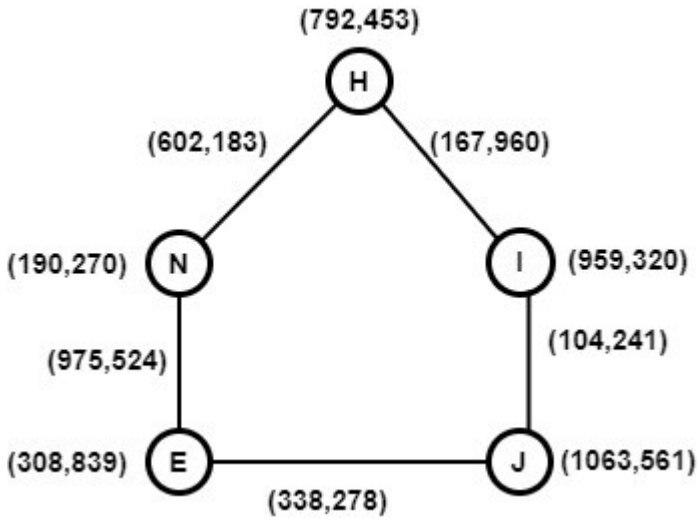


Figure 5.5: A weighted graph contains plaintext letters

6. He continue to add edges to the graph until it is full.

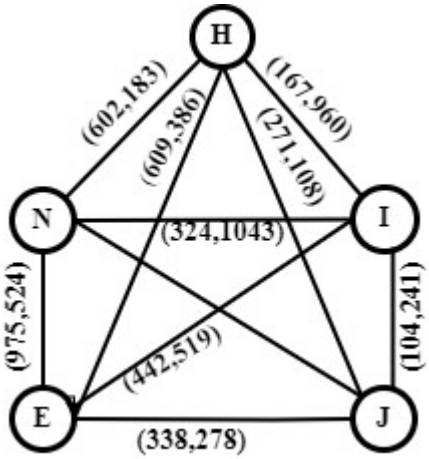


Figure 5.6: Complete graph of the plaintext.

7. He put a special letter before the first letter to indicate to the first character.

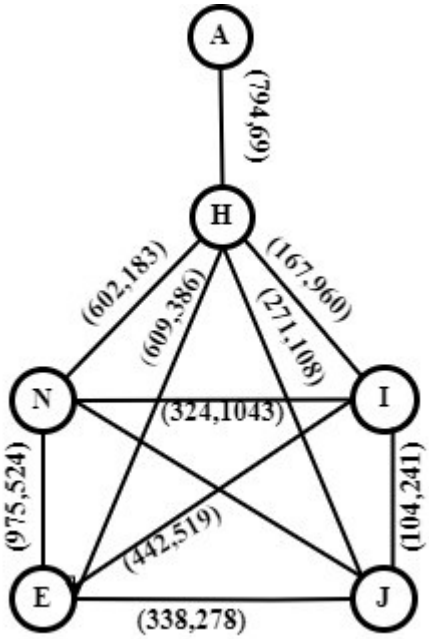


Figure 5.7: Complete graph of the plaintext with a special letter A

8. He find the spanning tree

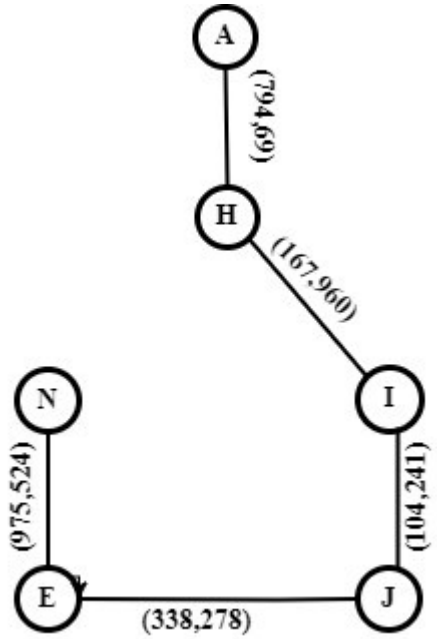


Figure 5.8: A Spanning Tree Graph on the plaintext.

9. He represent as a matrix M_1 and M_1^*

$$M_1 = \begin{pmatrix} (0,0) & (791,69) & (0,0) & (0,0) & (0,0) \\ (791,69) & (0,0) & (167,960) & (0,0) & (0,0) \\ (0,0) & (167,960) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \end{pmatrix}$$

$$M_1^* = \begin{pmatrix} (0,0) & (338,278) & (0,0) & (0,0) & (0,0) \\ (338,278) & (0,0) & (975,542) & (0,0) & (0,0) \\ (0,0) & (975,524) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \end{pmatrix}$$

10. He store the letters order in the diagonal instead of $(0,0)$

$$M_2 = \begin{pmatrix} (0,0) & (791,69) & (0,0) & (0,0) & (0,0) \\ (791,69) & (1,1) & (167,960) & (0,0) & (0,0) \\ (0,0) & (167,960) & (2,2) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \end{pmatrix}$$

$$M_2^* = \begin{pmatrix} (3,3) & (338,278) & (0,0) & (0,0) & (0,0) \\ (338,278) & (4,4) & (975,542) & (0,0) & (0,0) \\ (0,0) & (975,524) & (5,5) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \end{pmatrix}$$

11. He compute $C_2 = A^b M_2$ and ${C_2}^* = A^b {M_2}^*$

$$A^{33}(\text{mod}1093) = \begin{pmatrix} 84 & 747 & 94 & 540 & 218 \\ 136 & 714 & 850 & 1000 & 217 \\ 553 & 440 & 528 & 419 & 216 \\ 952 & 1038 & 297 & 280 & 384 \\ 893 & 434 & 960 & 868 & 634 \end{pmatrix}$$

$$C_2 = \begin{pmatrix} (657,172) & (914,599) & (335,300) & (0,0) & (0,0) \\ (786,81) & (1036,883) & (708,736) & (0,0) & (0,0) \\ (466,849) & (306,70) & (212,465) & (0,0) & (0,0) \\ (215,577) & (314,993) & (153,258) & (0,0) & (0,0) \\ (92,435) & (368,1044) & (74,1034) & (0,0) & (0,0) \end{pmatrix}$$

$$C_2^* = \begin{pmatrix} (255,248) & (614,179) & (857,604) & (0,0) & (0,0) \\ (187,1067) & (988,772) & (880,208) & (0,0) & (0,0) \\ (638,470) & (675,431) & (998,391) & (0,0) & (0,0) \\ (661,682) & (144,352) & (324,1083) & (0,0) & (0,0) \\ (723,915) & (108,1046) & (587,500) & (0,0) & (0,0) \end{pmatrix}$$

12. Bob send (C_1, C_2, C_2^*) to Jen Decryption process

(a) Jen compute C_1^{-a}

$$C_1^{-40}(\text{mod}1093) = \begin{pmatrix} 931 & 1004 & 1050 & 523 & 996 \\ 549 & 657 & 1032 & 414 & 889 \\ 232 & 859 & 259 & 771 & 876 \\ 756 & 332 & 575 & 1010 & 891 \\ 390 & 877 & 1021 & 800 & 478 \end{pmatrix}$$

(b) compute $C_1^{-a}C_2$

$$C_1^{-a}C_2 = \begin{pmatrix} (0,0) & (791,69) & (0,0) & (0,0) & (0,0) \\ (791,69) & (1,1) & (167,960) & (0,0) & (0,0) \\ (0,0) & (167,960) & (2,2) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \end{pmatrix}$$

$$C_1^{-a}C_2^* = \begin{pmatrix} (3,3) & (338,278) & (0,0) & (0,0) & (0,0) \\ (338,278) & (4,4) & (975,542) & (0,0) & (0,0) \\ (0,0) & (975,542) & (5,5) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \end{pmatrix} = M_2^*$$

(c) She suppose that the node $(0,0)$ is A and using encoding Table (5.10) she compute:

$(0,0) = \text{code}A = (1091, 384)$

$\text{Node}(0,0) = (\text{code}A + w_1) \bmod\ 1093)$

$= ((1091, 384) + (791, 69))(\text{mod } 1093)$

$= (792, 453) = H$

$\text{Node}(2,2) = (\text{code}H + w_2)(\text{mod } 1093)$

$= ((792, 453) + (167, 790))(\text{mod } 1093)$

$= (959, 320) = I$

Since the number of rows containing $(0,0)$ is equal to the number of columns containing $(0,0)$, this does not represent a character code and it means the end of the first word.

From M_1^* compute :

$\text{Node}(3,3) = (\text{code}I + w_3)(\text{mod } 1093)$

$= ((959, 320) + (104, 241))(\text{mod } 1093)$

$= (1063, 561) = J$

$\text{Node}(4,4) = (\text{code}J + w^4)(\text{mod } 1093)$

$= ((1063, 561) + (338, 278))(\text{mod } 1093)$

$= (308, 839) = E$

$\text{Node}(5,5) = (\text{code}E + w_5)(\text{mod } 1093)$

$= ((308, 839) + (975, 524))(\text{mod } 1093)$

$= (190, 270) = N$

The original text (HI JEN).

5.8. The Results of Applying Optimization Algorithms to Generate Private Key and Ephemeral Key in ElGamal with $BRH_{\alpha,\beta}$ Curve

Some simple computations of the applying optimization algorithms to find private key and ephemeral key in ElGamal with $BRH_{\alpha,\beta}$ Curve have been done. The experimental samples with different values of a prime p are chosen and compute the computational results to generate the keys, encryption, decryption processes and comparison time of the optimization algorithms.

5.8.1 The Results Applying Particle Swarm Optimization Algorithms to Generate Private Key and Ephemeral Key in ElGamal with $BRH_{\alpha,\beta}$ Curve

Table 5.11: The experimental results of PSO-ElGamal public key cryptosystem: key generation process

n	P	Shear key D	private key a	public key $A \equiv aD(\mathrm{mod}P)$
1	373	$(266, 301)$	42	$(40, 6)$
2	479	$(66, 403)$	366	$(305, 307)$
3	677	$(93, 170)$	643	$(575, 640)$
4	991	$(59, 63)$	506	$(41, 521)$
5	1013	$(275, 267)$	497	$(658, 233)$

Table 5.12: The experimental results of PSO-ElGamal public key cryptosystem: encryption process

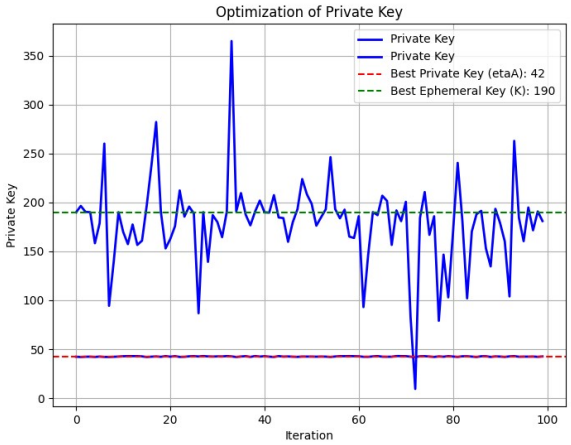
Ephemeral key b	$C_1 \equiv bD(modp)$	A plain text M	$C_2 \equiv bA + M(modp)$
190	$(348, 276)$	$(273, 358)$	$(125, 112)$
468	$(173, 405)$	$(476, 478)$	$(46, 368)$
555	$(296, 290)$	$(285, 402)$	$(456, 465)$
262	$(393, 606)$	$(11, 227)$	$(712, 558)$
504	$(743, 990)$	$(351, 935)$	$(484, 424)$

Table 5.13: The experimental results of PSO- ElGamal public key cryptosystem: decryption process

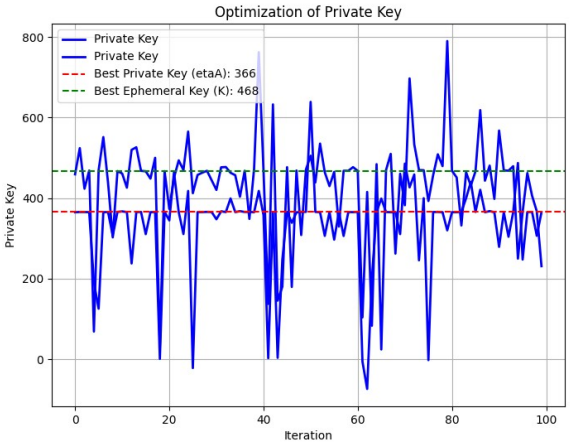
$aC_1(\text{mod } p)$	$-aC_1(\text{mod } p)$	$C_2 - aC_1 \equiv M(\text{mod } p)$
$(370, 242)$	$(3, 131)$	$(273, 358)$
$(439, 281)$	$(40, 198)$	$(476, 478)$
$(110, 496)$	$(567, 181)$	$(285, 402)$
$(6, 482)$	$(985, 509)$	$(11, 227)$
$(380, 528)$	$(633, 485)$	$(351, 935)$

Table 5.14: Time of PSO algorithm, encryption time and decryption time

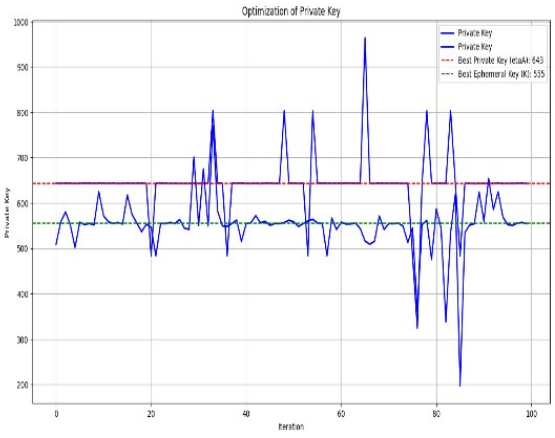
PSO Time(s)	Encryption Time(s)	Decryption Time(s)
1.707979	1.001500	1.001397
1.802138	1.005010	1.003153
54.498406	1.006409	1.003803
46.771115	1.004097	1.003516
71.71755	1.007454	1.003422



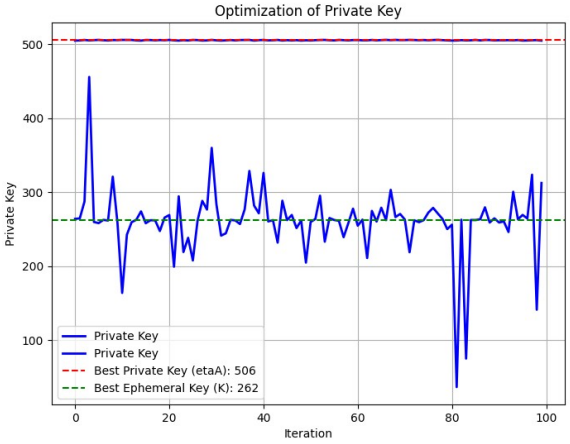
(a) ***P=373***



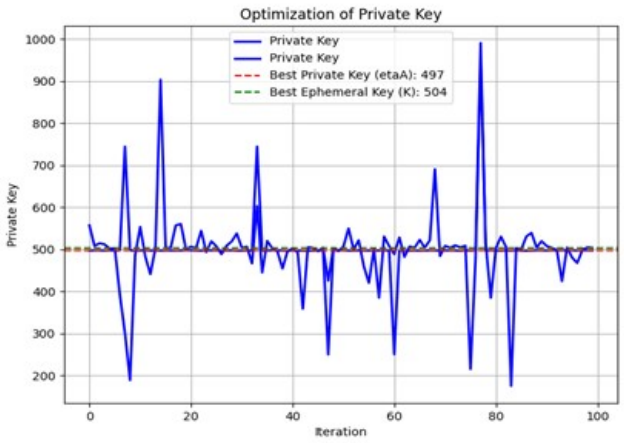
(b) ***P=479***



(c) ***P=677***



(d) ***P=991***



(e) ***P=1013***

Figure 5.9: PSO Optimization Private Key

5.8.2 The Results Applying Cuckoo Optimization Algorithms to Generate Private Key and Ephemeral Key in ElGamal with $BRH_{\alpha,\beta}$ Curve

Table 5.15: The experimental results of COA-ElGamal public key cryptosystem: key generation process

n	P	Shear key D	private key a	public key $A \equiv aD(\mathrm{mod}p)$
1	373	$(266, 301)$	245	$(215, 312)$
2	479	$(66, 403)$	423	$(141, 149)$
3	677	$(93, 170)$	23	$(321, 410)$
4	991	$(59, 63)$	73	$(562, 841)$
5	1013	$(275, 267)$	250	$(658, 242)$

Table 5.16: The experimental results of COA-ElGamal public key cryptosystem: encryption process

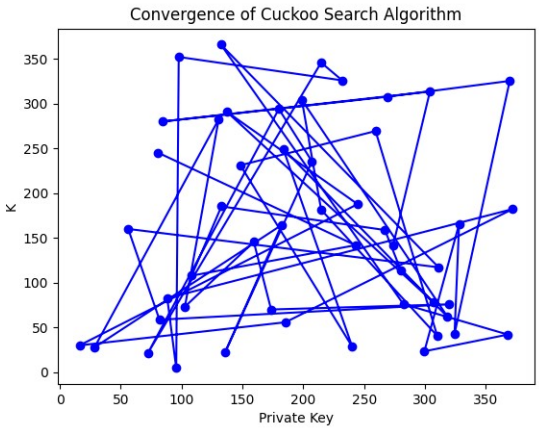
Ephemeral key b	$C_1 \equiv bD(modp)$	A plain text M	$C_2 \equiv bA + M(modp)$
188	$(316, 122)$	$(273, 358)$	$(243, 167)$
438	$(313, 422)$	$(476, 478)$	$(179, 282)$
534	$(8, 567)$	$(285, 402)$	$(318, 597)$
855	$(685, 319)$	$(11, 227)$	$(372, 902)$
140	$(953, 984)$	$(351, 935)$	$(358, 107)$

Table 5.17: The experimental results of COA- ElGamal public key cryptosystem: decryption process

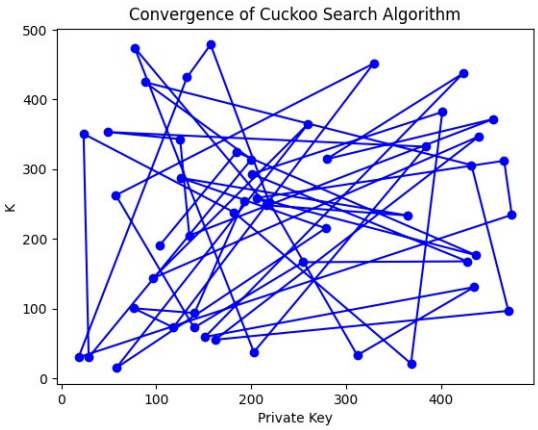
$aC_1(\text{mod } p)$	$-aC_1(\text{mod } p)$	$C_2 - aC_1 \equiv M(\text{mod } p)$
$(42, 342)$	$(331, 31)$	$(273, 358)$
$(40, 198)$	$(439, 281)$	$(476, 478)$
$(14, 337)$	$(663, 340)$	$(285, 402)$
$(939, 936)$	$(52, 55)$	$(11, 227)$
$(442, 641)$	$(571, 372)$	$(351, 935)$

Table 5.18: Time of PSO algorithm, encryption time and decryption time

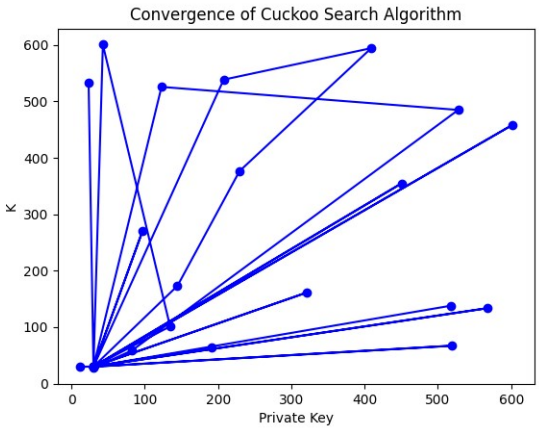
COA Time(s)	Encryption Time(s)	Decryption Time(s)
0.319996	1.001868	1.002009
0.233590	1.005529	1.000983
1.540367	1.005464	1.000535
1.873281	1.004748	1.000555
0.529075	1.003538	1.001772



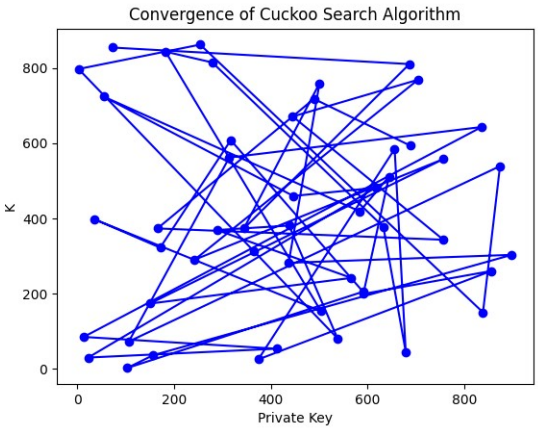
(a) ***P=373***



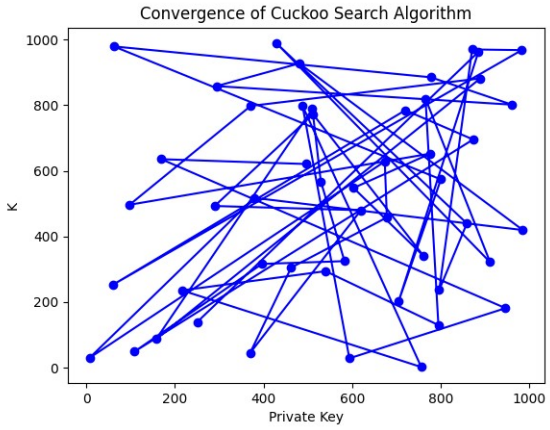
(b) ***P=479***



(c) ***P=677***



(d) ***P=991***



(e) ***P=1013***

Figure 5.10: Convergence of Cuckoo Search Algorithm

5.8.3 The Comparison of the Two Optimization Algorithms

In this section comparison between the time of apply optimization algorithms to generate private key and ephemeral key , encryption and decryption time and find the best optimization algorithm that has less time to apply.

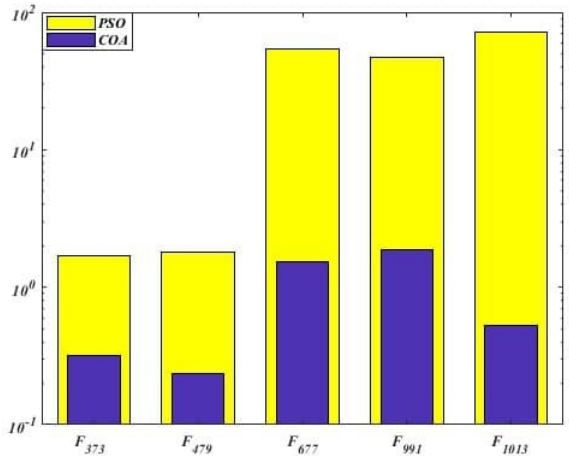


Figure 5.11: Comparison between POS, COA Times

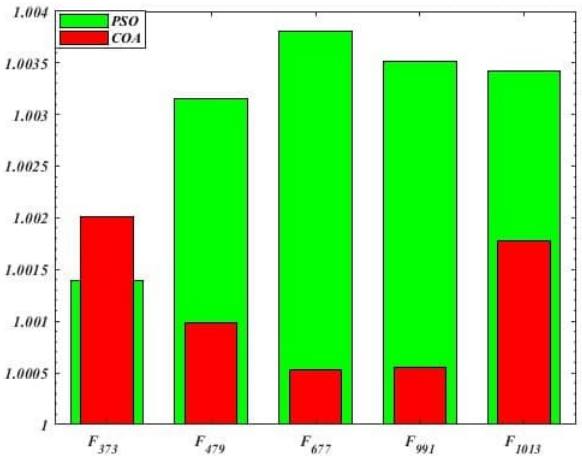


Figure 5.12: Comparison Encryption Time

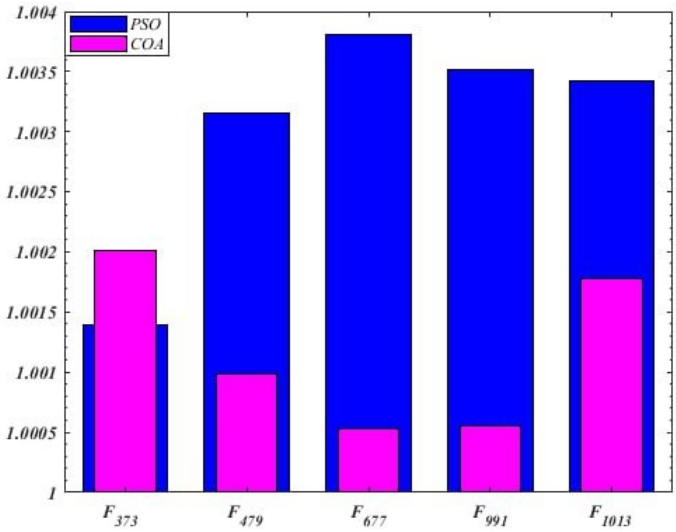


Figure 5.13: Comparison Decryption Time

From Figures 5.11, 5.12 , 5.13 COA exhibits a relatively faster execution time , indicating efficient exploration and convergence within a shorter time frame compared to POS. COA provides accurate private key and ephemeral key selections, resulting in successful encryption and decryption processes and faster.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

6.1. Conclusions

The conclusions of this work is summarized by

1. This work proposed and anlyzed a new of elliptic curve which is called the $BRH_{\alpha,\beta}$ curve defined over F_p.

2. The $BRH_{\alpha,\beta}$ curve is a smooth curve.

3. The arthimatic on the $BRH_{\alpha,\beta}$ curve is done through computing the doubling point and addition of two points using defined formulas that is same formula.So,the doubling and addition formulas are uniformed.

4. Using the $BRH_{\alpha,\beta}$ curve under the addition operation $(BRH_{\alpha,\beta}, +_{BRH_{\alpha,\beta}})$ is an abalian group.

5. The $BRH_{\alpha,\beta}$ curve has an affine form.

6. The $BRH_{\alpha,\beta}$ curve is applied on some encryption schemes to speed them and also used to design other encryption schemes.

7. The $BRH_{\alpha,\beta}$ curve is employed with the graph theory to design new versions of the cryptosystems and to increasing the security level.

8. The $BRH_{\alpha,\beta}$ curve is used with the some optimization algorithm for more efficient and secure cryptographic .

9. All proposed encryption algorithms based on the $BRH_{\alpha,\beta}$ curve defined over F_p are more secure in compare with original ones.

10. Several new experimenal resnlts of proposed algorithms are discussed based on the implemented programing using the Matlab and Python.

6.2. Future works

some future works can be suggested :

1. It can apply these results to image encryption and text encryption through using other types of graphs with other linds of cryptosystem.

2. Can also apply other kinds of graphs to modified the digital signatures schem.

3. It is possble to proposed a new optimization algorithms to generate privet key and compute the privet key for image and text encryption algorithm .

REFERENCES

[1] Ahmed A Abd El-Latif and Xiamu 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] Shubham Agarwal, Anand Singh Uniyal, et al. Prime weighted graph in cryptographic system for secure communication. *International journal of Pure and applied Mathematics*, 105(3):325–338, 2015.

[3] Ruma Kareem K Ajeena. The soft graphic integer sub-decomposition method for elliptic scalar multiplication. *Journal of Discrete Mathematical Sciences and Cryptography*, 24(6):1751–1765, 2021.

[4] Karrar Taher R Aljamaly and Ruma Kareem K Ajeena. The elliptic scalar multiplication graph and its application in elliptic curve cryptography. *Journal of Discrete Mathematical Sciences and Cryptography*, 24(6):1793–1807, 2021.

[5] Karrar Taher R Aljamaly and Ruma Kareem K Ajeena. Undirected complete graph to design new public key cryptosystem. In *Journal*

of Physics: Conference Series, volume 1897, page 012045. IOP Publishing, 2021.

[6] Fatima AMOUNAS. An innovative approach for enhancing the security of amazigh text using graph theory based ecc. *International Journal of Scientific Research in Science, Engineering and Technology*, 2:480–487, 2016.

[7] Muhammad Ashraf and Baris Kirlar. On the alternate models of elliptic curves. *International Journal of Information Security Science*, 1(2):49–66, 2012.

[8] Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media, 2012.

[9] Ashok D Belegundu and Tirupathi R Chandrupatla. *Optimization concepts and applications in engineering*. Cambridge University Press, 2019.

[10] Daniel J Bernstein, Chitchanok Chuengsatiansup, David Kohel, and Tanja Lange. Twisted hessian curves. In *International Conference on Cryptology and Information Security in Latin America*, pages 269–294. Springer, 2015.

[11] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[12] Omid Bozorg-Haddad. *Advanced optimization by nature-inspired algorithms*, volume 720. Springer, 2018.

[13] Manoj Kumar Chande and Cheng-Chi Lee. An improvement of a elliptic curve digital signature algorithm. *International Journal of Internet Technology and Secured Transactions*, 6(3):219–230, 2016.

[14] Abdoul Aziz Ciss and Djiby Sow. On a new generalization of huff curves. *Cryptology ePrint Archive*, 2011.

[15] S Pramela Devi and K Sindhuja. A public key cryptosystem using ecc and genetic algorithm‖. *International Journal of Engineering Research & Technology (IJERT) Vol*, 3, 2014.

[16] Julien Devigne and Marc Joye. Binary huff curves. In *Topics in Cryptology–CT-RSA 2011: The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, pages 340–355. Springer, 2011.

[17] Russell Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, pages 39–43. Ieee, 1995.

[18] Harold Edwards. A normal form for elliptic curves. *Bulletin of the American mathematical society*, 44(3):393–422, 2007.

[19] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[20] K. T. Radhi Ewad. *New Design Cryptosystems via Graphs Theory.* PhD thesis, University of Babylon. College of Education for Pure Sciences, 2022.

[21] Iztok Fister, Xin-She Yang, Dušan Fister, and Iztok Fister. Cuckoo search: a brief literature review. *Cuckoo search and firefly algorithm: Theory and applications*, pages 49–62, 2014.

[22] Simon Goss, Serge Aron, Jean-Louis Deneubourg, and Jacques Marie Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989.

[23] James A Green. The characters of the finite general linear groups. *Transactions of the American Mathematical Society*, 80(2):402–447, 1955.

[24] Haihua Gu, Dawu Gu, Wenlu Xie, and Ray CC Cheung. Efficient pairing computation on huff curves. *Cryptologia*, 39(3):270–275, 2015.

[25] D Menezes Hankerson. Aj and s. vanstone (2006)"guide to elliptic curve cryptography".

[26] Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[27] Jeffrey Hoffstein. An introduction to cryptography. In *An Introduction to Mathematical Cryptography*, pages 1–58. Springer, 2008.

[28] Gerald B Huff. Diophantine problems in geometry and elliptic ternary forms. 1948.

[29] Muhammad Imran, Rathiah Hashim, and Noor Elaiza Abd Khalid. An overview of particle swarm optimization variants. *Procedia Engineering*, 53:491–496, 2013.

[30] Nathan Jacobson. *Lectures in Abstract Algebra: III. Theory of Fields and Galois Theory*, volume 32. Springer Science & Business Media, 2012.

[31] Atif Raza Jafri, Muhammad Najam ul Islam, Malik Imran, and Muhammad Rashid. Towards an optimized architecture for unified binary huff curves. *Journal of Circuits, Systems and Computers*, 26(11):1750178, 2017.

[32] David Jao, Stephen D Miller, and Ramarathnam Venkatesan. Expander graphs based on grh with an application to elliptic curve cryptography. *Journal of Number Theory*, 129(6):1491–1504, 2009.

[33] Marc Joye, Mehdi Tibouchi, and Damien Vergnaud. Huff's model for elliptic curves. In *Algorithmic Number Theory: 9th International Symposium, ANTS-IX, Nancy, France, July 19-23, 2010. Proceedings 9*, pages 234–250. Springer, 2010.

[34] Thomas W Judson. *Abstract algebra: theory and applications*. Orthogonal Publishing, 2013.

[35] Balasubramanian Prabhu Kavin and Sannasi Ganapathy. A new digital signature algorithm for ensuring the data integrity in cloud using elliptic curves. *Int. Arab J. Inf. Technol.*, 18(2):180–190, 2021.

[36] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[37] Suhri Kim. Complete analysis of implementing isogeny-based cryptography using huff form of elliptic curves. *IEEE Access*, 9:154500–154512, 2021.

[38] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[39] Neal Koblitz, Alfred Menezes, and Scott Vanstone. The state of elliptic curve cryptography. *Designs, codes and cryptography*, 19:173–193, 2000.

[40] Serge Lang. *Elliptic curves: Diophantine analysis*, volume 231. Springer, 1978.

[41] An Liu and Peng 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.

[42] Ralph C Merkle. Protocols for public key cryptosystems. In *Secure communications and asymmetric cryptosystems*, pages 73–104. Routledge, 2019.

[43] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[44] Lothrop Mittenthal. Sequencings and directed graphs with applications to cryptography. In *Sequences, Subsequences, and Consequences: International Workshop, SSC 2007, Los Angeles, CA, USA, May 31-June 2, 2007, Revised Invited Papers*, pages 70–81. Springer, 2007.

[45] Peter L Montgomery. Speeding the pollard and elliptic curve methods of factorization. *Mathematics of computation*, 48(177):243–264, 1987.

[46] WD Peeples. Elliptic curves and rational distance sets. *Proceedings of the American Mathematical Society*, 5(1):29–33, 1954.

[47] Ramin Rajabioun. Cuckoo optimization algorithm. *Applied soft computing*, 11(8):5508–5518, 2011.

[48] Santanu Saha Ray. *Graph theory with algorithms and its applications: in applied science and technology*. Springer, 2013.

[49] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[50] Gideon Samid. Denial cryptography based on graph theory, November 23 2004. US Patent 6,823,068.

[51] R Selvakumar and Nishant Gupta. Fundamental circuits and cut-sets used in cryptography. *Journal of Discrete Mathematical Sciences and Cryptography*, 15(4-5):287–301, 2012.

[52] K Shankar and P Eswaran. An efficient image encryption technique based on optimized key generation in ecc using genetic algorithm. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems: Proceedings of ICAIECES 2015*, pages 705–714. Springer, 2016.

[53] Joseph H Silverman, Jill Pipher, and Jeffrey Hoffstein. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[54] Piotr Szczechowiak, Leonardo B Oliveira, Michael Scott, Martin Collier, and Ricardo Dahab. Nanoecc: Testing the limits of elliptic curve cryptography in sensor networks. In *Wireless Sensor Networks: 5th European Conference, EWSN 2008, Bologna, Italy, January 30-February 1, 2008. Proceedings*, pages 305–320. Springer, 2008.

[55] Dhanashree K Toradmalle, Jayabhaskar Muthukuru, and B Sathyanarayana. Cryptanalysis of an improved ecdsa. *International Journal of Engineering Research and Technology*, 11(4):615–619, 2018.

[56] Richard J Trudeau. *Introduction to graph theory*. Courier Corporation, 2013.

[57] Vasyl Ustimenko. Cryptim: Graphs as tools for symmetric encryption. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes: 14th International Symposium, AAECC-14 Melbourne, Australia, November 26–30, 2001 Proceedings 14*, pages 278–286. Springer, 2001.

[58] Philomena Waruhari and Lawrence Nderu. An elliptic curve digital

signature algorithm (ecdsa) for securing data: an exemplar of securing patient's data. 2017.

[59] M Yamuna, Meenal Gogia, Ashish Sikka, and Md Jazib Hayat Khan. Encryption using graph theory and linear algebra. *International Journal of Computer Application*, 5(2):102–107, 2012.

[60] Xin-She Yang. *Nature-inspired optimization algorithms*. Academic Press, 2020.

[61] Xin-She Yang and Suash Deb. Cuckoo search via lévy flights. In *2009 World congress on nature & biologically inspired computing (NaBIC)*, pages 210–214. Ieee, 2009.

[62] GA Zweers. Pathways and space for evolution of feeding mechanisms in birds. *The unity of evolutionary biology*, pages 1530–1547, 1991.

الملخص

تناقش هذه الأطروحة نماذج المنحني الإهليجي على حقل الأعداد الأولية وخاصة منحني Huffو $BRH_{\alpha,\beta}$ المقترح (Batool-Ruma-Huff) وتطبيقاتها في مخططات التشفير. تم إثبات أن منحني $BRH_{\alpha,\beta}$ المقترح هو منحنى ناعم بشرط معين. وتم اشتقاق صيغة Affine لمنحنى $BRH_{\alpha,\beta}$ لحساب صيغة مضاعفة النقطة والجمع التي تم تعريفها على منحني $BRH_{\alpha,\beta}$ وقد كانت صيغة موحدة . ان منحني $BRH_{\alpha,\beta}$ مع عملية الجمع قد تم اثباته كزمرة معرفة على الحقل الاولي F_P.

بعض انظمة التشفير تم اقتراحها باستخدام المنحني $BRH_{\alpha,\beta}$ المعرف على الحقل الاولي F_p. احد هذه التطبيقات نظام تبادل المفاتيح Diffie-Hellman ونظام تشفير آخر للمفتاح العام هوEl-Gamal. بالإضافة الى ذلك، المنحني $BRH_{\alpha,\beta}$ تم تطبيقه ليعطي نموذج جديد من نظام التوقيع الرقمي.

من ناحية اخرى، المنحني $BRH_{\alpha,\beta}$ تم توظيفه مع نظرية البيان لاقتراح وتطوير نماذج جديدة من انظمة التشفير التي تم تسميتها matrix ElGamal graphic ، $BRH_{\alpha,\beta}$ و البيان المولد لنظام تشفير النص بالاعتماد على المنحني$BRH_{\alpha,\beta}$ في ثلاث حالات.

هذا العمل يقترح ايضا طريقتين للتشفير من خلال استخدام خوارزميات الامثلية المطبقة على المنحني $BRH_{\alpha,\beta}$ المعرف على الحقل F_p. هذه الخوارزميات سميت نظام التشفيرالمفتاح المعلن ElGamal للمنحني $BRH_{\alpha,\beta}$ خوارزمية سرب الطيور (PSO) ونظام تشفير المعلن ElGamalباستخدام خوارزمية الواقواق (COA) .

نتائج تجريبية جديدة لكل الخوارزميات المقترحة تم مناقشتها. بعض الخوارزميات تم تنفيذها باستخدام لغة Matlab، والبعض الاخرتم تنفيذها باستخدام لغة Python.

الاعتبارات الامنية لكل خوارزميات التشفير المقترحة تم تحديدها بناءاً على المسائل الرياضية الصعبة المعتمدة عليها.

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

نموذج جديد على المنحنيات الاهليجية لانظمة تشفير المفتاح العام

اطروحة

مقدمة الى مجلس كلية التربية للعلوم الصرفة / جامعة بابل وهي جزء من

متطلبات نيل درجة الدكتوراه فلسفة في التربية / الرياضيات

من قبل الطالبة

بتول حاتم عكار حسين

بإشراف

ا.م.د. رومى كريم خضرعجينة

1445 هـ **2023 م**