Image Cryptosystems Using Elliptic Curve Cryptography

by

Ghulam Murtaza

Department of Mathematics
Quaid-i-Azam University
Islamabad, Pakistan

2023

Image Cryptosystems Using Elliptic Curve Cryptography

by

Ghulam Murtaza

Supervised by

Dr. Umar Hayat

Department of Mathematics
Quaid-i-Azam University
Islamabad, Pakistan

2023

Image Cryptosystems Using Elliptic Curve Cryptography

by

Ghulam Murtaza

A THESIS SUBMITTED IN THE PARTIAL FULFILLMENT FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

MATHEMATICS

Supervised by

Dr. Umar Hayat

Department of Mathematics
Quaid-i-Azam University
Islamabad, Pakistan

2023

Author's Declaration

I, **Ghulam Murtaza**, hereby state that my PhD thesis titled **Image Cryptosystems Using Elliptic Curve Cryptography** is my own work and has not been submitted previously by me for taking any degree from the Quaid-I-Azam University Islamabad, Pakistan, or anywhere else in the country/world.

At any time if my statement is found to be incorrect even after my graduation the university has the right to withdraw my PhD degree.

Murtaza

Name of Student: **Ghulam Murtaza**

Date: **12-June-2023**

Plagiarism Undertaking

I solemnly declare that the research work presented in the thesis titled "**Image Cryptosystems Using Elliptic Curve Cryptography**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and Quaid-i-Azam University towards plagiarism. Therefore, I as an Author of the above-titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above-titled thesis even afterward of PhD degree, the University reserves the right to withdraw/revoke my PhD degree and that HEC and the University have the right to publish my name on the HEC/University Website on which names of students are placed who submitted plagiarized thesis.

Murtaza

Student/Author Signature
Name: **Ghulam Murtaza**

Certificate of Approval

This is to certify that the research work presented in this thesis entitled **Image Cryptosystems Using Elliptic Curve Cryptography** was conducted by **Mr. Ghulam Murtaza** under the kind supervision of **Dr. Umar Hayat**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the Department of Mathematics, Quaid-i-Azam University, Islamabad, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the field of mathematics from the Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan.

Student Name: **Ghulam Murtaza** Signature: Murtaza

External committee:

a) **External Examiner 1**: Signature: Akbar Azam
Name: **Dr. Akbar Azam**
Designation: Professor
Office Address: Department of Mathematics,
COMSATS University, Park Road,
Chak Shahzad, Islamabad.

b) **External Examiner 2**: Signature:
Name: **Dr. Muhammad Ishaq**
Designation: Associate Professor
Office Address: School of Natural Sciences (SNS),
National University of Sciences & Technology (NUST),
Islamabad.

c) **Internal Examiner:** Signature:
Name: **Dr. Umar Hayat**
Designation: Associate Professor
Office Address: Department of Mathematics, QAU Islamabad.

Supervisor Name: Signature:
Dr. Umar Hayat

Name of Dean/ HOD Signature: Tariq Shah
Prof. Dr. Tariq Shah

Image Cryptosystems Using Elliptic Curve Cryptography

By

Ghulam Murtaza

CERTIFICATE

A THESIS SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF THE

DOCTOR OF PHILOSOPHY IN MATHEMATICS

We accept this thesis as conforming to the required standard

1. Tariq Shah
Prof. Dr. Tariq Shah
(Chairman)

2.
Dr. Umar Hayat
(Supervisor)

3. Akbar Azam
Prof. Dr. Akbar Azam
Department of Mathematics,
COMSATS University, Park Road,
Chak Shahzad, Islamabad.
(External Examiner)

4.
Dr. Muhammad Ishaq
School of Natural Sciences (SNS),
National University of Sciences &
Technology (NUST), Islamabad
(External Examiner)

Department of Mathematics
Quaid-I-Azam University
Islamabad, Pakistan
2023

To my family and my supervisor…

Acknowledgements

Alhamdolillah! It is indeed the blessing of **ALLAH Almighty**, Who gave me enough strength and consistency to complete this thesis. All regards and respects to **The Holy Prophet Hazrat Muhammad (SAW)**, Whose way of life has been a continuous guidance for life.

First and foremost, I want to pay gratitude to my supervisor **Dr. Umar Hayat**, who encourage me to accomplish this thesis. I thank you, Sir, for your role all the way, starting from the very beginning, for accepting me as a PhD student, and for training me as a research student. I especially thank **Dr. Naveed Ahmad Azam** and **Dr. Muhammad Ikram** for their guidance and help in the research work. I am also very thankful to all lab fellows– **Sumaira Azhar**, **Takreem Haidar**, and **Muhammad Bilal** for their worthwhile discussions and suggestions. My deepest gratitude and respect go to my parents, brothers, and sister for their prayers and sincerity. It would not be possible to write this thesis without their support.

I want to acknowledge and thank the support of my PhD research colleagues at Quaid-i-Azam University– **Mr. Muhammad Naeem**, **Mr. Muzammal Hanif**, and **Mr. Muhammad Waqar** for their cooperation and companionship. I am thankful to my seniors– **Dr. Yasir Naseer**, **Dr. Muhammad Bilal**, and **Dr. Mubashar Umar** for their hospitality and valuable discussion. I would also thank all the staff of the **Mathematics Department** for their cooperation and kind behavior during my studies.

May **ALLAH Almighty** shower the above-cited personalities with success and honor in their life.

Ghulam Murtaza
QAU, Islamabad
June, 2023

Preface

This thesis is about applications of elliptic curves in cryptography. Cryptography is a technique for protecting data and communications. It is a purposeful effort to diffuse and confuse data so that the attackers cannot obtain confidential information. More specifically, cryptosystems involve the creation, evaluation, and use of mathematical defenses against adversarial attacks.

An elliptic curve is a curve of the form $y^2 = f(x)$, where $f(x)$ is a cubic polynomial. The elliptic curve appears for the first time in the work of Diophantus in his text Arithmetica. To gain a general understanding of how operations on an elliptic curve actually work, we first prescribe the characteristics of an elliptic curve over the real numbers. Any field, including the complex, real, rational, or prime, can be used to compute the coordinates of an elliptic curve's points. However, elliptic curves over finite fields are desirable from an application perspective. The set of points on an elliptic curve over a finite field, together with a special point "O" ("point at infinity") form an additive abelian group. The group operation is a point addition that can be executed using arithmetic operations in the underlying finite field. Hasse's theorem [1] gives an approximate bound about the points of an elliptic curve.

Before the 1970's, encryption required two parties to physically meet to establish a shared secret key for secure communication, which made the procedure fairly difficult and time-consuming. Symmetric ciphers, which served as the foundation for private cryptosystems, were designed with this in mind. Miller [2] introduced the concept of Elliptic Curve Cryptography (ECC) and presented an elliptic curve-based cryptosystem that is 20 times faster than the Diffie-Hellmans algorithm. A cryptosystem based on an elliptic curve over a finite field is introduced by Koblitz in [3]. ECC-based algorithms are computationally efficient and provide greater security. For example, 256-bit ECC over a prime field and 2048-bit Rivest-Shamir-Adleman algorithm provide the same level of security [4]. In addition, ECC requires less memory on digital computers [5, 6].

Such features make ECC more suitable for devices with limited resources in power and network connectivity [7]. At present, ECC has currently gained commercial acceptance and has been embraced by numerous standard organizations, including NIST [8], ANSI [9], ISO [10] and IEEE [11].

Encryption techniques based on chaos maps are also used for image encryption. However, some chaotic systems with low dimensions have a short periodicity of orbits when implemented on digital computers that's why they degrade quickly [12]. Nowadays, spatiotemporal chaotic systems such as coupled map lattices have been widely used in cryptography. The security of the coupled map lattices-based image encryption schemes is greatly improved because the coupled map lattices have larger keyspace, better randomness, longer cycle, and more parameters [13]. Amara [7] analyzed that ECC has high security than the Rivest-Shamir-Adleman algorithm. In [14] an elliptic curve-based random-number generator is used for diffusion while dynamic substitution boxes (S-box) for confusion. Hayat et al. proposed an S-box generator and an image cryptosystem based on elliptic curves over finite rings [15]. Azam et al., [16] designed a secure elliptic curve-based image cryptosystem. In [17], an asymmetric multiple-images encryption method based on an elliptic curve and a quick response code is given.

Many researchers designed different schemes for image encryption and the construction of dynamic S-boxes. In [18], authors proposed an image encryption scheme using asymmetric key encryption. A genetic algorithm is used to generate a special key and then elliptic curves are utilized to encipher all pixels one by one. However, it increases the computational cost to encrypt all pixel one by one and search for the ideal keys. The scheme in [15] is highly secure but it is not possible to implement the scheme for elliptic curves of large size. A hybrid algorithm based on both Advanced Encryption Standard and ECC is introduced in [19].

The main motivation behind this thesis is to design a dynamic S-boxes generator and highly secure encryption schemes with lesser encryption time. The main objectives are listed below:

1- Defining new mathematical algorithms on elliptic curves to design new cryptosystems.

2- Construction of new structures on elliptic curves with the help of existing structures.

3- Developing encryption schemes that can provide both confidentiality and integrity.

4- Designing new structures and methods to construct a random-number generator based on elliptic curves.

5- Stating and proving theoretical results.

The thesis is organized as follows: The used concepts and corresponding notations are described in Chapter 1.

In Chapter 2, we introduced an S-box generator that is suitable for lightweight cryptography and outperforms previously designed S-box generators in terms of computation time and security analysis. We produce particular sequences of integers using ordered elliptic curves of short size and binary sequences, which are subsequently utilized to generate S-boxes. We conducted numerous conventional tests to find out the efficiency of the proposed generator. Comparisons indicate that the new generator requires less operating time and has more security against modern attacks than numerous existing well-known generators.

In Chapter 3, a new parametrization of resonant discrete triads to develop a new algorithm for the generation of all resonant triads in a grid of size L. We define a new transformation to map the resonant triads on a conic. We provide a full list of discrete Rossby wave triads in a given grid by solving Diophantine equations which appear in the context of the Charney-Hasegawa-Mima equation. Further, we extend the algorithm for the enumeration of quasi-resonant triads and experimentally demonstrate the robustness the algorithm to design the network of quasi-resonant triads. As an application, we apply a total order on generated triads to design an S-box generator. Finally, via extensive analysis, we show that the newly developed S-box outperforms the S-boxes generated by some of the existing algorithms.

In Chapter 4, a novel and secure cryptosystem for the real-time transmission of digital images is presented. The proposed work is based on elliptic curves and couple map lattices. The encryption technique is divided into two steps. The plain-image is initially diffused using an elliptic curve-based pseudo-random-number generator (PRNG). Then an S-box generator based on elliptic curves and couple map lattices is designed. The proposed encryption scheme has a larger keyspace and is robust against modern attacks. The use of couple map lattice-system makes the proposed cryptosystem secure against the known-plaintext attack.

In Chapter 5, image encryption is used to transform digital images into an unreadable form on open Networks. We have designed an image cryptosystem to tackle the issues related to small key sizes and plaintext attacks. The proposed cryptosystem is divided into three parts. In the first part, we have developed a PRNG to diffuse the pixels of the plain image and in the second step, an S-box generator is constructed to generate permutation S-boxes with high nonlinearity. In the final step, an image encryption technique is presented to encrypt grayscale images. Furthermore, the cryptographic properties of our pseudo-random-number generator are tested using NIST tests. Furthermore, pseudo-random-numbers are generated using the elliptic curves to create diffusion in the data of plain-images. An S-box is used as a permutation to scramble the diffused image.

In Chapter 6, the research results and suggested future directions are discussed. The thesis concludes with a list of references.

List of Publications from the Thesis

As a publication is one of the requirements of the Higher Education Commission of Pakistan, we give here a list of publications from the thesis:

1. Murtaza, G., Azam, N. A., and Hayat, U. Designing an Efficient and Highly Dynamic Substitution-Box Generator for Block Ciphers Based on Finite Elliptic Curves. *Security and Communication Networks*, **2021**.

2. Hayat, U., Ullah, I., Murtaza, G., Azam, N.A., and Bustamante, M.D. Enumerating Discrete Resonant Rossby/Drift Wave Triads and Their Application in Information Security. *Mathematics*, **2022**, *10(23)*, p.4395.

3. Azam, N. A., Murtaza, G., and Hayat, U. A novel image encryption scheme based on elliptic curves and coupled map lattices. *Optik*, **2023**, p.170517.

Contents

List of Tables

List of Figures

Chapter1

Introduction

In the beginning, the introduction of the elliptic curve is presented in Sec. 1.1. Preliminaries related to couple map lattice-systems are given in Sec. 1.2. A brief introduction of the Charney-Hasegawa-Mima equation is provided in Sec. 1.3. Finally, details about the fundamental security analysis used in cryptography are discussed in Sec. 1.4.

1.1 An Overview of the Elliptic Curve

Elliptic curves (ECs) are algebraic structures, used in many fields such as mathematical physics, number theory, and cryptography. In cryptography ECs provide high security with less memory consumption on digital computers. Basic definitions and algebraic operations on ECs are defined in this section.

Definition 1.1. Let F be a field whose characteristic Char(F) is not equal to 2 and 3. An elliptic curve (EC) over F is written as

$$E(F) : y^2 = x^3 + ax + b \text{ for } a, b \in F. \tag{1.1}$$

The EC in Eq. 1.1 is in a short Weierstrass form. The collection of points on an EC over the field F is given as

$$E(F) = \{(x, y) \mid (x, y) \in F^2, y^2 = x^3 + ax + b\} \cup \{O\}, \tag{1.2}$$

here "O" is a point at infinity.

Definition 1.2. The discriminant for the EC in Eq. 1.1 is defined as:

$$\Delta = -16(4a^3 + 27b^2). \tag{1.3}$$

The Weierstrass form of an EC is singular (means that the polynomial $x^3 + ax + b$ has repeated roots) if and only if $\Delta = 0$. The non-singularity of the curve is guaranteed by the condition that the discriminant $\Delta \neq 0$. The EC is smooth if and only if the discriminant $\Delta \neq 0$.

Definition 1.3. A point P of an EC $E(F)$ is called a singular point if $\Delta(E(p)) = 0$, where $\Delta(E(P))$ is the discriminant of an EC at a given point P.

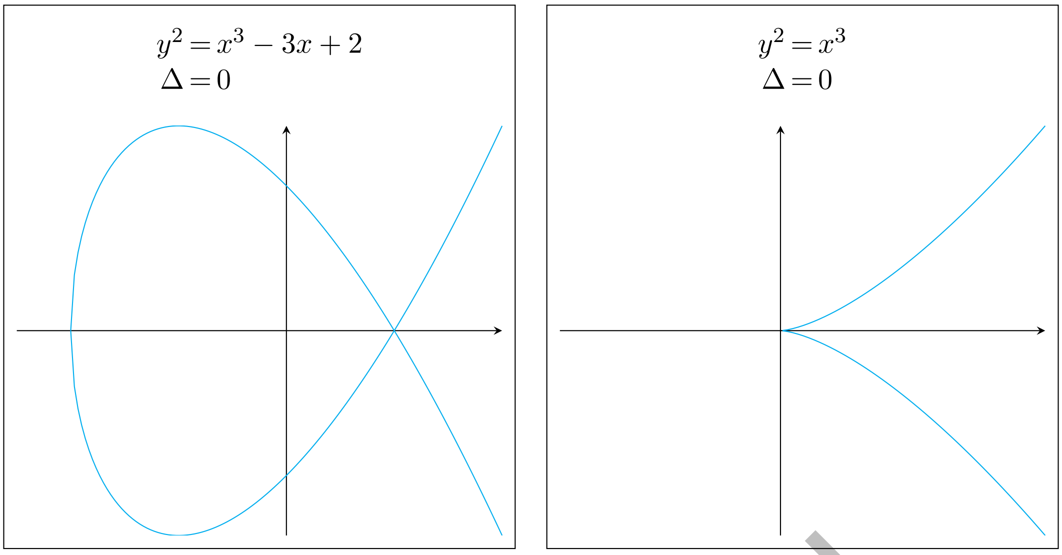
Two ECs in Fig. 1.1 are singular ($\Delta = 0$).

- The curve in Fig. 1.1(a) has a node (*i.e.*, $x^3 + ax + b$ has a double root) at $(0, 1)$.
- The curve in Fig. 1.1(b) has a cusp (*i.e.*, $x^3 + ax + b$ has a triple root) at $(0, 0)$.

In general, ECs with singularities have unusual behavior relative to non-singular ECs. From now on we exclude singular ECs and speak only of non-singular ECs.

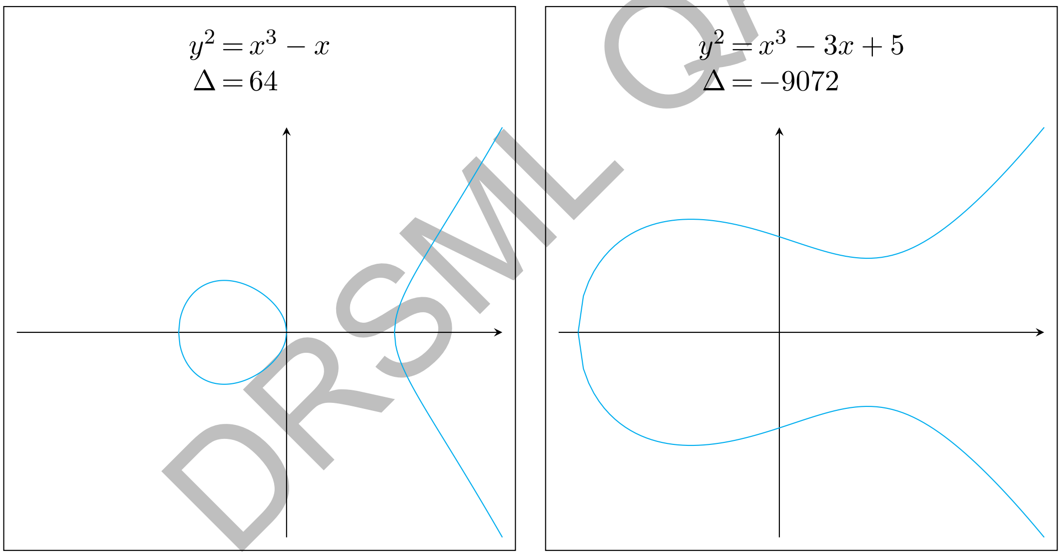
We first define ECs over real numbers $\mathbb{R}$ in order to have a general understanding of how operations over ECs work. If the discriminant Δ of an EC is positive, then the graph of the EC over $\mathbb{R}$ has two parts and if the discriminant Δ is negative, then the graph of the EC over $\mathbb{R}$ has only one component. In Fig. 1.2, the effect of the discriminant Δ on an EC shape is presented.

- The EC $y^2 = x^3 - x$ in Fig. 1.2(a) has $\Delta = 64$, therefore, has two components.
- In Fig. 1.2(b), the curve $y^2 = x^3 - 3x + 5$ has one component due to negative value of $\Delta = -9072$.



(a) (b)

Figure 1.1: Singular ECs, *i.e.*, $\Delta = 0$.



(a) (b)

Figure 1.2: Effect of Δ on the graph of ECs.

1.1.1 Group Law of the Elliptic Curve Points

The set of all points on an EC together with O (the point at infinity) form an abelian group under a binary operation " + ". Consider $P(x_P, y_P)$ and $Q(x_Q, y_Q)$ are any two points on an EC $E(F) : y^2 = x^3 + ax + b$. According to Bézout Theorem [26], the line PQ passing through the points P and Q will intersect the EC at the third point R (may coincide with P or Q). In such a case, $P + Q$ is the reflection of R across the x-coordinate. The point addition over an EC is described in detail as follows:

First Case: Let $P(x_P, y_P) \neq Q(x_Q, y_Q)$ and further we assume that $x_P \neq x_Q$. Also, neither P nor Q is a point at infinity. The slope (m) of the line (PQ) which passes through P and Q can be define as:

$$m = \frac{y_Q - y_P}{x_Q - x_P}. \tag{1.4}$$

The line PQ is defined as

$$PQ : y = m(x - x_P) + y_P. \tag{1.5}$$

By combining Eq. 1.5 with the EC in Eq 1.1 we have

$$(m(x - x_P) + y_P)^2 = x^3 + ax + b \tag{1.6}$$

$$0 = x^3 - m^2x^2 + \cdots . \tag{1.7}$$

Now, if $x_P, x_Q,$ and x_R are the roots of a cube, then

$$0 = (x - x_P)(x - x_Q)(x - x_R) \tag{1.8}$$

$$= x^3 - (x_P + x_Q + x_R)x^2 + \cdots . \tag{1.9}$$

Setting the coefficient of the x^2 in Eqs. 1.7 and 1.9 equal, we have $m^2 = x_P + x_Q + x_R$. Simplification gives that $x_R = m^2 - x_P - x_Q$. From Eq. 1.5, we have $y_R = m(x_R - x_P) + y_P$. Thus, the third point is $R(x_R, y_R) = (m^2 - x_P - x_Q, m(x_R - x_P) + y_P)$. Since $P + Q = (x_R, -y_R)$, because $P + Q$ is reflection of R. Thus

$$P + Q = (m^2 - x_P - x_Q, m(x_P - x_R) - y_P). \tag{1.10}$$

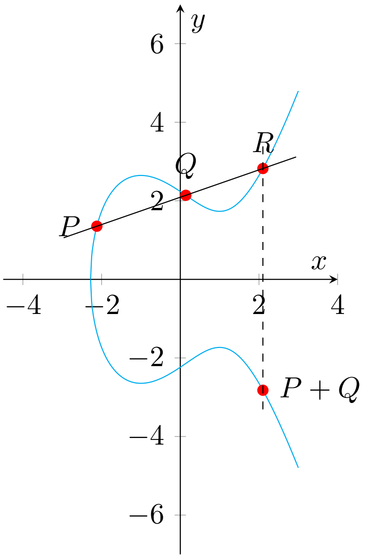


Figure 1.3: Addition $P + Q$.

Graphically, the first case is illustrated in Fig. 1.3.

Suppose P and Q are not equal but $x_P = x_Q$. Then PQ is a vertical line, and $R = O$ implies that its reflection is also a point at infinity, which means $P + Q = O$.

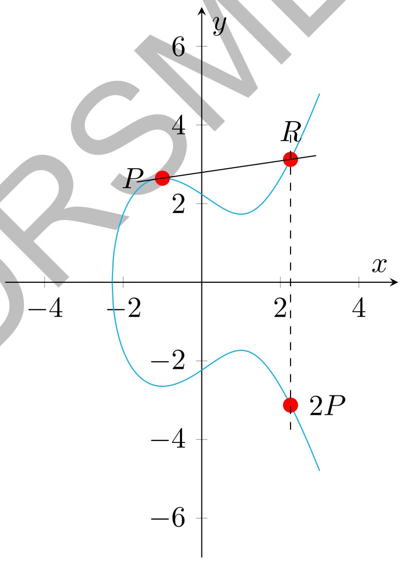


Figure 1.4: Doubling $2P$.

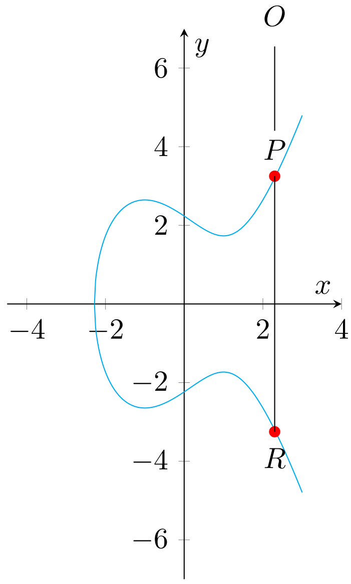


Figure 1.5: Addition of O and P.

Second Case: Let $P(x_P, y_P) = Q(x_Q, y_Q)$. The slop of the EC in Eq. 1.1 is computed as

$$2ydy = (3x^2 + a)dx \tag{1.11}$$

$$\frac{dy}{dx} = \frac{3x^2 + a}{2y}. \tag{1.12}$$

The slop m at $P(x_P, y_P)$ is

$$m = \frac{3x_P^2 + a}{2y_P}. \tag{1.13}$$

When y_P is zero and provided that $3x^2 + a \neq 0$, which only happen when $x^3 + ax + b$ has a double root. More generally, if $y_P = 0$, then there is a vertical tangent.

Suppose $y_P \neq 0$, since $x_P = x_Q$ then from first case, we get $x_R = m - 2x_P$ and hence Eq. 1.10 becomes

$$P + Q = 2P = (m - 2x_P, m(x_P - x_R) - y_P). \tag{1.14}$$

Graphically, the point doubling is given in Fig. 1.4.

Third Case: If $Q = O$, then $P + Q = P + O = P$. Graphically, the third case is displayed in

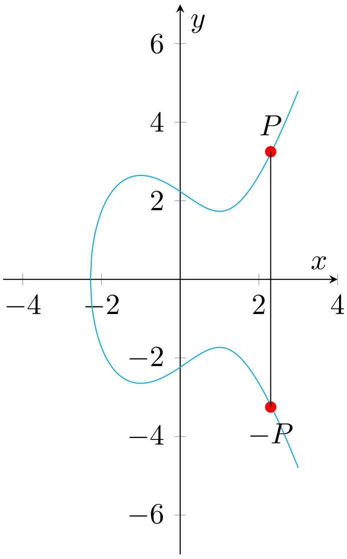


Figure 1.6: Inverse of a point on an EC.

Fig. 1.5. Furthermore, if $P = (x, y)$ then $-P = (x, -y)$ is inverse of P and $P + (-P) = O$. In Fig. 1.6, the relationship between a point P and its inverses on an EC is given.

1.1.2 Characteristics of an Elliptic Curve Over Finite Fields

An EC $E_{p,a,b}$ over a finite field F_p for a prime p can be written as:

$$E_{p,a,b} = \{(x, y) \in \mathrm{F}_p^2 \mid \; y^2 \equiv x^3 + ax + b \pmod{p}\} \cup \{O\}, \tag{1.15}$$

where $a, b \in F_p \setminus \{0\}$, "$O$" is a point at infinity and $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$. The points of an

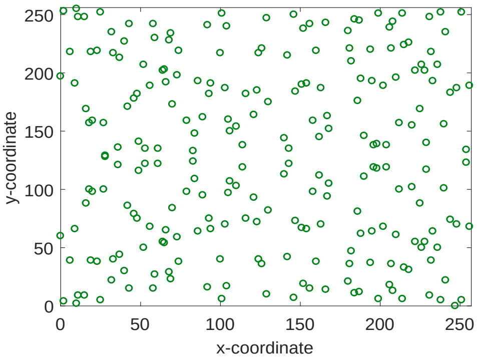


Figure 1.7: Plot of of the EC $E_{257,3,2} : y^2 \equiv x^3 + 3x + 2 \pmod{257}$.

EC $E_{257,3,2}$ are plotted in Fig. 1.7. For an EC $E_{p,a,b}$ the group structure is defined in the steps below.

- Point at infinity O is identity element. Therefore, $P+O=O+P=P$ for all $P\in E_{p,a,b}$.
- If $P=(x_P,y_P)\in E_{p,a,b}$, then negation of P is $-P=(x_P,p-y_P)$ and $P+(-P)=O$. Note that $-O=O$.
- For any two points P and Q over an $E_{p,a,b}$, addition $P+Q$ is provided as,

$$P+Q=\begin{cases} P \text{ if } Q=O \\ Q \text{ if } P=O \\ O \text{ if } P=-Q \\ R(x_R,y_R) \text{ otherwise.} \end{cases} \tag{1.16}$$

Here $x_R=m^2-x_P-x_Q \pmod p$, $y_R=m(x_P-x_R)-y_P \pmod p$, where m is

$$m=\begin{cases} \frac{y_Q-y_P}{x_Q-x_P} & \text{if } P\neq Q \\ \frac{3x_P^2+a}{2y_P} & \text{if } P=Q \text{ and } y_P\neq 0. \end{cases} \tag{1.17}$$

Example 1.1. *Suppose p=23 and* $E_{23,1,1}: y^2=x^3+x+1$. *So, we have* $a=b=1$ *this implies that* $\Delta=4(1)^3+27(1)^2=31\neq 0$.

(1) *Let* $P=(3,10)$ *and* $Q=(9,7)$. *We want* $P+Q=(x_R,y_R)$. *Since* $m=\frac{y_Q-y_P}{x_Q-x_P}=\frac{7-10}{9-3}=11$ $x_R=m^2-x_P-x_Q=-6 \pmod{23}=17$, $y_R=m(x_P-x_R)-y_P=89 \pmod{23}=20$. *Hence* $P+Q=(17,20)$.

(2) *Let* $P=(3,10)$ *then* $P+P=2P$. *Now* $m=\frac{3(3)^2+1}{20}=\frac{5}{20}=\frac{1}{4} \pmod{23}=6$. *Now,* $x_3=m^2-2x_1=30 \pmod{23}=7$ *and* $y_3=m(x_1-x_3)-y_1=-11 \pmod{23}=12$ *Hence* $2P=(7,12)$.

(3) If $P = (3, 10)$ *then* $-P = (3, 23 - 10) = (3, 13)$

(4) Select $P = (3, 10)$ *and* $Q = (3, 13)$, *since* $Q = -P$, *then* $P + Q = O$.

For ECs over an F_p, there are p possibilities for every x and at most 2 choices of y for each x. So $E_{p,a,b}$ has at most $2p + 1$ points including O.

Definition 1.4. For an EC $E_{p,a,b}$, the total number of points on the EC $E_{p,a,b}$ is denoted by $\#E_{p,a,b}$.

Definition 1.5. An estimation of the total number of points on an $E_{p,a,b}$ is given by Hasse's theorem [1]

$$p + 1 - 2\sqrt{p} \leq \#E_{p,a,b} \leq p + 1 + 2\sqrt{p}. \tag{1.18}$$

This inequality gives an upper bound and a lower bound for $\#E_{p,a,b}$.

Definition 1.6. For F_p, an EC

$$E_{p,0,b} = \{(x, y) \in F_p^2 \mid y^2 \equiv x^3 + b \pmod{p}\} \cup \{O\} \tag{1.19}$$

is called a Mordell-Elliptic-Curve (MEC), where $0 \neq b \in F_p$.

Theorem 1.7. *For a prime* $p \equiv 2 \pmod 3$, *the total number of points* ($\#E_{p,0,b}$) *on an* $E_{p,0,b}$ *are exactly* $p + 1$ *[1].*

Definition 1.8. Two ECs $E_{p,a,b} : y^2 \equiv x^3 + ax + b \pmod{p}$ and $E_{p,\tilde{a},\tilde{b}} : y^2 \equiv x^3 + \tilde{a}x + \tilde{b} \pmod{p}$ are isomorphic if there is a parameter $\xi \in F_p$ such that $a\xi^4 \equiv \tilde{a} \pmod{p}$ and $b\xi^6 \equiv \tilde{b} \pmod{p}$. The integer ξ is called isomorphism parameter between the $E_{p,a,b}$ and $E_{p,\tilde{a},\tilde{b}}$ over F_p.

Definition 1.9. Another comparatively basic idea that is notable at this stage is the idea of an EC point multiplication. It is crucial to understand that in this case, we are scaling points by integer values rather than multiplying two points in the traditional sense. For a point P of $E_{p,a,b}$ and a scalar $t \in F_p$, the scalar multiplication is defined as

$$tP = \begin{cases} O & \text{if } \ t = 0 \\ P + (t-1)P & \text{otherwise.} \end{cases} \tag{1.20}$$

In practice, the simplest scalar multiplication algorithm is the double-and-add algorithm.

Example 1.2. *Let $E_{17,3,2} : y^2 = x^3 + 3x + 2 \pmod{17}$ be an EC. Table 1.1 shows number of points for each value of x and scalar multiplication of $P = (12, 7)$.*

$$P = (12, 7) \xrightarrow{2P} (6, 7) \xrightarrow{3P} (16, 10) \xrightarrow{4P} (4, 0) \xrightarrow{5P} (16, 7) \xrightarrow{6P} (6, 10) \xrightarrow{7P} (12, 10) \xrightarrow{8P} (O, O).$$

Table 1.1: $E_{17,3,2} : y^2 = x^3 + 3x + 2 \pmod{17}$.

x	y	Points
0	6,11	(0,6), (0,11)
2	4,13	(2,4), (2,13)
3	2,15	(3,2), (3,15)
6	7,10	(6,7), (6, 10)
7	3,14	(7,3), (7, 14)
12	7,10	(12,7), (12, 10)
14	0	(14,0)
16	7,10	(16,7), (16,10)

Definition 1.10. Let $E_{p,a,b}$ be an EC over F_p. If G is a generator point of $E_{p,a,b}$ and let $G_1 =< G >$, a subgroup generated by G. Then finding an integer $t \in F_p$ such that $tG = R$ in $E_{p,a,b}$, where $R \in G_1$ is called is the discrete logarithm problem on $E_{p,a,b}$.

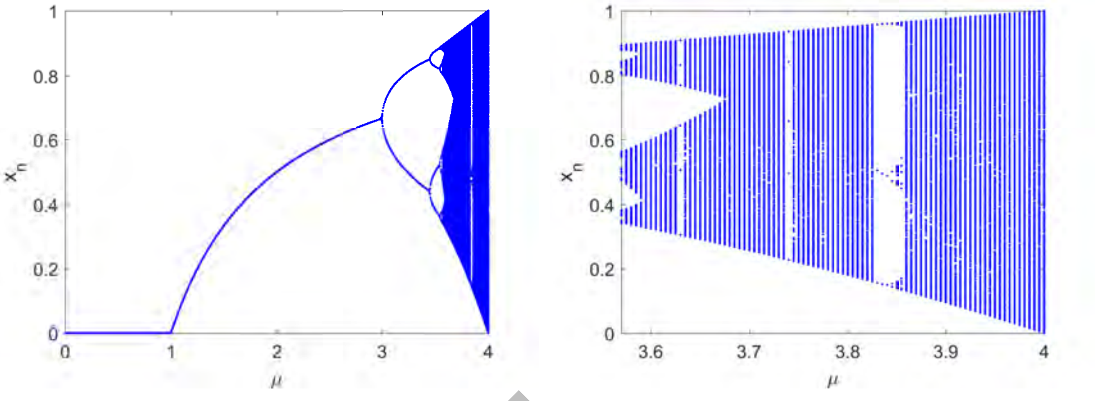
1.2 The Couple Map Lattice

Chaotic maps are widely used in cryptosystems due to their astonishing nature, such as unpredictability and sensitivity to initial parameters. Nowadays, a class of chaotic maps known as coupled map lattice (CML) is defined to overcome the shortcomings of traditional chaotic maps. The CML-systems have a large range and chaotic behavior over a large domain.

Definition 1.11. A special chaotic map known as a logistic map is define below

$$x_{n+1} = ux_n(1 - x_n). \tag{1.21}$$

For $x_n \in (0, 1)$ and $u \in (0, 4]$. In Fig. 1.8(a), the bifurcation of the logistic map is presented for $\mu \in (0, 4]$. The bifurcation plot is given in Fig. 1.8(b) for $\mu \in (3.57, 4]$.



(a)

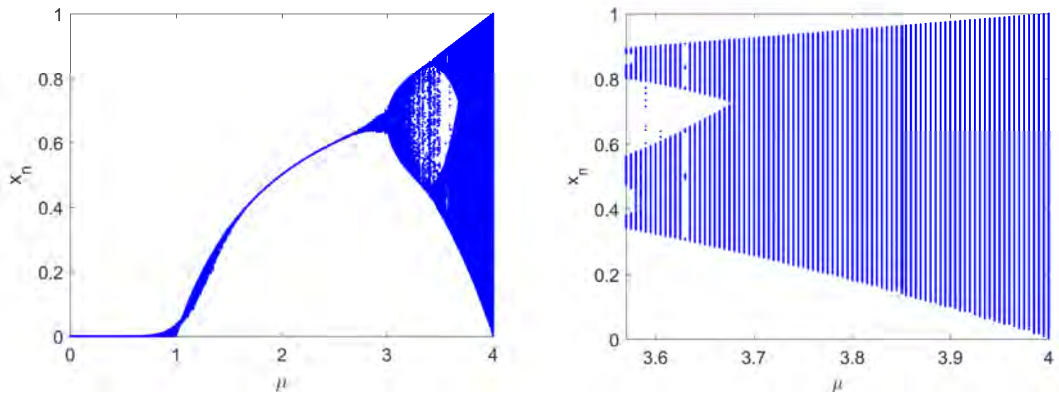
(b)

Figure 1.8: Chaotic behavior of the logistic map for different μ. (a) Bifurcation of the logistic map for $\mu \in (0, 4]$; (b) Bifurcation of the logistic map for $\mu \in (3.57, 4]$.

Definition 1.12. A CML-system is defined as

$$X_{n+1}(j) = (1 - \xi)f(X_n(j)) + \frac{\xi}{2}[f(X_n(j + 1)) + f(X_n(j - 1))], \tag{1.22}$$

where $\xi \in [0\ 1]$ is a coupling constant, j $(1, 2, \ldots, \tau)$ is the lattice site index, τ is the total number of lattices to be generated, n is the time variable and f is a real mapping. The periodic boundary condition $X_n(0) = X_n(\tau)$ is assumed. Fig. 1.9 show bifurcation plots of CML-systems. It is clear from Fig. 1.9 that the CML-system shows better chaotic behavior for different values of μ.



(a) (b)

Figure 1.9: Chaotic behavior of the CML-system for different μ. (a) Bifurcation of $9th$ CML for $\mu \in (0, 4]$; (b) Bifurcation of $9th$ CML for $\mu \in (3.57, 4]$.

1.3 The Charney-Hasegawa-Mima Equation

The partial differential equation known as the Charney-Hasegawa-Mima equation (CHME) is written as

$$\frac{\partial}{\partial t}(\triangle\psi - F\psi) + \beta\frac{\partial\psi}{\partial x_1} + [\psi, \triangle\psi] = 0, \tag{1.23}$$

where in the atmospheric context, $\psi : \mathbb{R}^2 \times [0, T) \to \mathbb{R}$ is the stream-function, $\beta > 0, F \geq 0$ are constants, and

$$\triangle\psi = \frac{\partial^2\psi}{\partial x_1^2} + \frac{\partial^2\psi}{\partial x_2^2}, \qquad [A, B] = \frac{\partial A}{\partial x_1}\frac{\partial B}{\partial x_2} - \frac{\partial A}{\partial x_2}\frac{\partial B}{\partial x_1}.$$

The sum of the first and second term represents the linear part and the last term $[\psi, \triangle\psi]$ is the nonlinear part of Eq. 1.23. The general solution of Eq. 1.23 is not known as it generically displays spatio-temporal chaos and turbulence. However, a number of particular exact solutions are available in the form of a family of cosine functions

$$\psi = \psi_{k,\ell}(x_1, x_2, t) = \cos(kx_1 + \ell x_2 - \omega t), \tag{1.24}$$

for the angular frequency

$$\omega = \omega(k,\ell) = \frac{-\beta k}{k^2+\ell^2+F}\,. \tag{1.25}$$

These solution are known as Rossby waves. Notice that for these solutions both the linear part and the nonlinear part of Eq. 1.23 vanish independently. The vector (k,ℓ) is a wavevector, whereas the integers k and ℓ are known as the zonal and meridional wavenumbers, respectively.

Definition 1.13. A resonant triad is a triple $(k_1,\ell_1),(k_2,\ell_2),(k_3,\ell_3)$ of Rossby waves satisfying the set of equations

$$\begin{cases} k_2 = k_3 - k_1, \\ \ell_2 = \ell_3 - \ell_1, \\ \omega_2 = \omega_3 - \omega_1, \end{cases} \tag{1.26}$$

where $\omega_i = \omega(k_i,\ell_i), i = 1,2,3$.

Definition 1.14. If for a small positive number δ, the condition $\omega_2 = \omega_3 - \omega_1$ on angular frequencies in the above set of equations is replaced by

$$|\omega_1 + \omega_2 - \omega_3| \leq \delta, \tag{1.27}$$

then the aforesaid triple is a quasi-resonant triad and the positive number δ is called a detuning level.

In the special case $F = 0$, the parameterized solutions for the resonant triads were firstly obtained in [27] and later on developed in [28], giving

$$\frac{k_1}{k_3} = \frac{(u^2+t^2)(u^2+t^2-2u)}{(1-2u)}, \tag{1.28}$$

$$\frac{\ell_3}{k_3} = \frac{(2u-1)u+(u^2+t^2)(u^2+t^2-2u)}{(1-2u)t}, \tag{1.29}$$

$$\frac{\ell_1}{k_3} = \frac{(u^2+t^2)\big((2u-1)+(u^2+t^2-2u)u\big)}{(1-2u)t}. \tag{1.30}$$

An independent and particular case is discussed in [29] as well. In [27], the resonant triad is explicitly transformed to a triple (x, y, d) with rational components as

$$\frac{k_1}{k_3} = \frac{x}{y^2 + d^2}, \quad \frac{\ell_1}{k_3} = \left(\frac{x}{y}\right)\left(1 - \frac{d}{y^2 + d^2}\right), \quad \frac{\ell_3}{k_3} = \frac{d-1}{y}. \tag{1.31}$$

Then $(x, y, d) \in \mathbb{Q}^3$ is inversely mapped to a triad via

$$x = \frac{k_3(k_1^2 + \ell_1^2)}{k_1(k_3^2 + \ell_3^2)}, \quad y = \frac{k_3(k_3\ell_1 - k_1\ell_3)}{k_1(k_3^2 + \ell_3^2)}, \quad d = \frac{k_3(k_3k_1 + \ell_1\ell_3)}{k_1(k_3^2 + \ell_3^2)}. \tag{1.32}$$

Let us consider now the case of general aspect ratio f. It is enough to consider aspect ratios whose squares are rational, so we can write $f = f_1\sqrt{f_2}$ for rational f_1 and square-free integer f_2. As shown in [27], in this case Eqs. 1.23–1.27 hold true except for Eq. 1.25, which must be replaced with

$$\omega = \omega(k, \ell) = \frac{-\beta k}{k^2 + f^2\ell^2}\,. \tag{1.33}$$

For such a choice of f, the mappings analogous to Eq. 1.31 take the form

$$\frac{k_1}{k_3} = \frac{x}{f^2y^2 + d^2}, \quad \frac{\ell_1}{k_3} = \left(\frac{x}{f^2y}\right)\left(1 - \frac{d}{f^2y^2 + d^2}\right), \quad \frac{\ell_3}{k_3} = \frac{d-1}{f^2y}. \tag{1.34}$$

Having the inverse mappings as follows

$$x = \frac{k_3(k_1^2 + f^2\ell_1^2)}{k_1(k_3^2 + f^2\ell_3^2)}, \quad y = \frac{k_3(k_3\ell_1 - k_1\ell_3)}{k_1(k_3^2 + f^2\ell_3^2)}, \quad d = \frac{k_3(k_3k_1 + f^2\ell_1\ell_3)}{k_1(k_3^2 + f^2\ell_3^2)}. \tag{1.35}$$

1.4 Preliminaries Related to Cryptography

Nowadays, due to the sharing and openness of networks, it is a great threat to send information such as audio, videos, and digital images through the internet. These images are very important information carriers and play a vital role in multimedia communications. In addition, many images contain very sensitive data such as state secrets or personal privacy. Furthermore, transmission through unprotected public communication networks leads to serious security threats because intruders can easily access and tempered confidential information. Thus, it becomes quite important to prevent illegal access to the contents of secret images from adversaries. With the use of image encryption technique, the private information contained in plain-images can be efficiently protected. Cryptography is the science of hiding information and is used to protect the information from a third party. Following terminologies are frequently used in today cryptography:

- **Plaintext:** An original message or data in its readable form is called a plaintext.
- **Encryption:** The process of converting confidential data into secret codes or an unreadable form by using cryptographic techniques is called encryption.
- **Ciphertext:** An unreadable or encrypted form of the confidential data is called a ciphertext.
- **Decryption:** The method of converting encrypted data back to its readable form (plaintext) by using an algorithm is called decryption.
- **Encryption key:** An encryption key is an input to the encryption algorithm used to transmit a plaintext into a ciphertext, and without the key, one cannot obtain a plaintext from its ciphertext.
- **Cryptosystem:** A cryptosystem consisting of a set of algorithms that transforms plaintext into ciphertext by using the encryption keys. The keys, encryption and decryption methods together form a cryptosystem.

Shannon [30] developed two principles, confusion and diffusion, to improve the security of a cryptosystem.

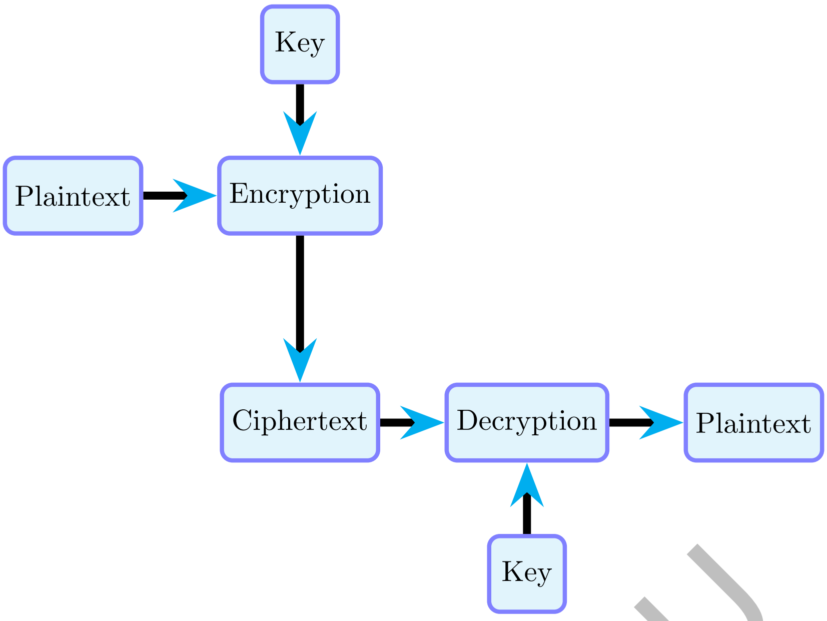


Figure 1.10: Encryption and decryption channel.

- **Confusion:** Confusion is the process of making the relationship between the ciphertext and the key as difficult as possible so that no one, even if they know the ciphertext, can figure out the key.
- **Diffusion:** Diffusion is the technique of spreading out the influence of one plaintext bit on many ciphertext bits such that statistical redundancy in the plaintext cannot be recognized.

Fig. 1.10 shows a general description of the encryption process.

Mainly cryptography has two categories:

(i) Private-key (or symmetric-key) cryptography;

(ii) Public-key (or asymmetric-key) cryptography.

Private-key cryptography is further divided into block cipher and stream cipher. A block cipher encrypts data in blocks of fixed length, while in a stream cipher one bit is encrypted in one go. When using private-key algorithms, all the communication parties share the same secret key that is used to carry out data encryption and decryption. If a sender wants to send a plaintext m

to a receiver in a form of a ciphertext c through private-key cryptography using encryption key k, then he will apply an encryption function E to get $c = E(m, k)$. Now, if the receiver want to decrypt the ciphertext c then he will apply the encryption key k and a decryption function D on the ciphertext c to recover the plaintext m such that $m = D(c, k)$. The private-key encryption protocol is presented in Fig. 1.11.

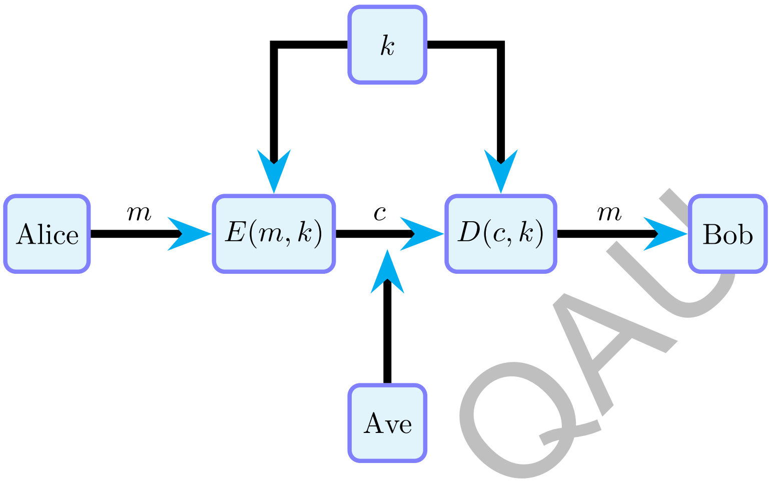


Figure 1.11: Symetric key cryptosystem.

To use asymmetric algorithms each sender and receiver is required to have a public-key and a private-key. One of the earliest public-key cryptosystems was introduced by the Deffie-Hellman. Public-key cryptosystems, in contrast to conventional private-key cryptosystems, rely on trapdoor (oneway) functions, the inverse of which may be computed easily but exponentially more slowly without the decryption key. The keys needed to encrypt communication can be made public as it is computationally unfeasible to decipher a communicated message using enciphering keys alone. To transport secret messages, the hybrid cryptosystem known as the Deffie-Hellman key exchange algorithm combines the principles of symmetric and public-key encryption. A public-key cryptosystem called RSA was also developed shortly after the Diffie-Hellman key exchange. Although public-key cryptosystems are more practical but they are more likely to be inefficient because of challenging mathematical calculations.

Let two parties Alice and Bob want to communicate through public-key cryptosystem. Suppose

Alice has a public-key j and a private-key k. Bob uses the public-key j to encrypt the plaintext m and Alice uses the private-key k to decrypt the ciphertext c. The public-key encryption protocol is shown in Fig. 1.12. Elliptic Curve Cryptography (ECC) and Data Structure Algorithms (DSA) are also examples of asymmetric key cryptography.

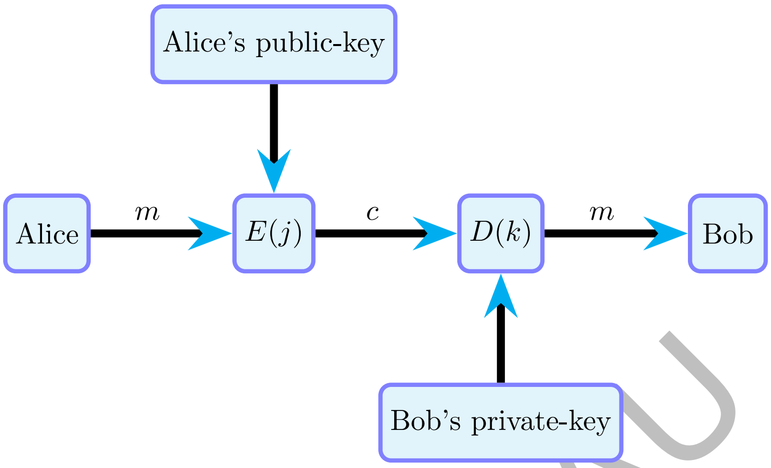


Figure 1.12: Asymetric key cryptosystem.

A substitution box (S-box) is a key component of various modern cryptosystems [31, 32] such as Advance Encryption Standard (AES), Skipjack and Data Encryption Standard (DES). A static S-box is used in AES to produce confusion in plaintexts. However, instead of a static S-box, cryptographers apply dynamic S-boxes in modern cryptosystems [33–35] to increase the security. Therefore, it urges researchers to design secure and fast S-box generators to construct dynamic and secure S-boxes. Followings are useful cryptographic notions and tests which will be used in upcoming chapters.

Definition 1.15. For a set $B = \{0, 1\}$ and a non-negative integer t, a function

$$\psi : B^t \to B \tag{1.36}$$

is known as a Boolean function, where B is a Boolean domain and t is arity of ψ. Since cardinalities of B^t and B are 2^t and 2 respectively, therefore we have 2^{2^t} distinct (t-array)

Boolean functions.

Definition 1.16. If the total number of inputs (x) mapped onto 1 is equal to the number of inputs (x) mapped onto 0 then the Boolean function in Eq. 1.36 becomes a balanced function. Mathematically, if

$$\#\{x : \psi(x) = 1\} = \#\{x : \psi(x) = 0\} \tag{1.37}$$

then ψ is balanced otherwise imbalanced function.

Definition 1.17. For $\alpha, x \in B^t$, a linear Boolean function $\mathscr{F} : B^t \to B$ is expressed as

$$\mathscr{F}_\alpha(x) = \alpha_1 \cdot x_1 \oplus \alpha_2 \cdot x_2 \oplus \cdots \oplus \alpha_t \cdot x_t, \tag{1.38}$$

where " $\cdot$ " is the AND operator and " $\oplus$ " is the XOR operator.

Definition 1.18. For $\alpha, x \in B^t$ and $\beta \in B$, then the Boolean function

$$\mathscr{F}_{\alpha,\beta}(x) = \alpha_1 \cdot x_1 \oplus \alpha_2 \cdot x_2 \oplus \cdots \oplus \alpha_t \cdot x_t \oplus \beta, \tag{1.39}$$

is an affine function.

Definition 1.19. An $m \times n$ S-box is a nonlinear Boolean function $\sigma : B^m \to B^n$ with m input and n output bits $u = (u_1, u_2, \ldots, u_m)$ and $v = (v_1, v_2, \ldots, v_n)$ such that $\sigma(u) = v$. Here m and n are two positive integers.

Unless otherwise specified, an $m \times n$ S-box will be denoted by σ.

The security strength of S-boxes is evaluated using a variety of standard tests, some of them are described as follows:

Definition 1.20. The idea of nonlinearity (NL) was first proposed in [36], which determines the ability of an S-box to create randomness in a plaintext. For an $n \times n$ S-box σ over $\mathrm{GF}(2^n)$, the NL is defined as

$$\mathrm{NL}(\sigma) = \min_{x,y,z}\{\alpha \in \mathrm{GF}(2^n) \mid x \cdot \sigma(\alpha) \neq y \cdot \alpha \oplus z\}\,,$$

where $x \in \mathrm{GF}(2^n) \setminus \{0\}$, $y \in \mathrm{GF}(2^n)$, $z \in \mathrm{GF}(2)$, and operation "." is an inner product over GF(2). If the NL is high, an S-box has a high resistance to linear attacks.

Definition 1.21. Matsui [37] was the first to introduce the Linear approximation probability (LAP) test for an S-box. The formula that computes the LAP is

$$\mathrm{LAP}(\sigma) = \frac{1}{2^n}\max_{x,y}\left|\#\{\alpha \in \mathrm{GF}(2^n) \mid x \cdot \alpha = y \cdot \sigma(\alpha)\} - 2^{n-1}\right|$$

where $x \in \mathrm{GF}(2^n)$, $y \in \mathrm{GF}(2^n) \setminus \{0\}$.

Definition 1.22. For S-boxes, the idea of a linear polynomial was first proposed in [38]. The number of non-zero terms in a linear polynomial of an S-box represents its algebraic complexity (AC).

Definition 1.23. Biham and Shamir [39] introduce the idea of differential approximation probability (DAP). For an S-box σ of size $n \times n$, the DAP is measured by

$$\mathrm{DAP}(\sigma) = \frac{1}{2^n}\max_{\triangle x, \triangle y} \#\{x \in \mathrm{GF}(2^n) \mid \sigma(x) \oplus \sigma(x + \triangle x) = \triangle y\},$$

where $\triangle x \in \mathrm{GF}(2^n) \setminus \{0\}, \triangle y \in \mathrm{GF}(2^n)$, and the operation $\oplus$ is bit-wise addition over $\mathrm{GF}(2^n)$. Cryptographically, an S-box has high security against differential attack if its DAP value is close to zero.

Definition 1.24. The capability of an S-box to create diffusion/confusion is measured by its Boolean function analysis. The SAC test [40] is a basic criterion to check the ability of an S-box to produce diffusion in a plaintext. The values for SAC of an $n \times n$ S-box σ is computed by matrix A(σ)= $[m_{ij}]$ for

$$m_{ij} = \frac{1}{2^n} \sum_{x \in \mathrm{GF}(2^n)} w\left(\sigma_i(x \oplus \alpha_j) \oplus \sigma_i(x)\right),$$

where $w(z)$ is the hamming weight of $z \in \mathrm{GF}(2^n)$, σ_i and σ_k are ith and kth Boolean functions of σ, respectively, and $1 \leq i, j, k \leq 8$. If the calculated value is closer to 0.5, then it means that an S-box fulfills the SAC criterion.

Definition 1.25. The BIC test [40] is used to determine how independent a pair of output bits is when one input bit is inverted. The diffusion-creating ability of an S-box is also determined by the BIC criterion. It is found by computing the dependence matrix $B(\sigma) = [d_{ij}]$, where d_{ij} is calculated by

$$d_{ij} = \frac{1}{2^n} \sum_{\substack{x \in \mathrm{GF}(2^n) \\ 1 \leq k \leq n, k \neq i}} w\big(\sigma_i(x \oplus \alpha_j) \oplus \sigma_i(x) \oplus \sigma_k(x \oplus \alpha_j) \oplus \sigma_k(x)\big).$$

The requirement of the BIC analysis is that the values of each element d_{ij} of the correlation matrix of $x_i \oplus x_j$ for all input $x_i \in \mathrm{GF}(2^n)$ (where $i, j = 1, 2, \ldots, n$ and $i \neq j$) of the given S-box should be approximately equal to 0.5.

In 2021, Azam et al. [22] proposed some necessary tests including complexity analysis, singularity analysis, variation analysis, sensitivity analysis, and confusion analysis to quantify the dynamic behavior of S-box generators. Each analysis is defined as.

Definition 1.26. The sensitivity of an S-box generator is the ability of an S-box generator to produce significantly different S-boxes when the inputs are slightly altered [22]. If an S-box generator has a high input sensitivity, it is said to be resistant to differential attacks.

Definition 1.27. Singularity of an S-box generator is defined as a legitimate input for which the generator is unable to produce an S-box [22]. Singularity-free S-box generators are regarded as suitable for encryption.

Definition 1.28. An S-box generation algorithm is regarded cryptographically effective if it can create a significant number of unique S-boxes [22]. An S-box generation algorithm has high resistance to brute-force attacks if it can create a large number of unique S-boxes.

Definition 1.29. An S-box generator is effective cryptographically if the output S-boxes include a small number of fixed points [22].

Definition 1.30. If a generator can produce S-boxes that are uncorrelated with one another, it is regarded as being efficient [22].

Definition 1.31. For the generation of S-boxes, diffusion and natural orderings are defined in [41] and for convenience these are denoted by T^1 and T^2 respectively. These orderings are used to arrange the points on a MEC. Let $(x_P, y_P), (x_Q, y_Q)$ be two points of the $E_{p,0,b}$

$$(x_P, y_P)\ T^1\ (x_Q, y_Q) \Leftrightarrow \begin{cases} \text{either } x_P + y_P < x_Q + y_Q, \text{ or} \\ x_P + y_P = x_Q + y_Q \text{ and } x_P < x_Q; \end{cases} \tag{1.40}$$

and

$$(x_P, y_P)\ T^2\ (x_Q, y_Q) \Leftrightarrow \begin{cases} \text{either } x_P < x_Q, \text{ or} \\ x_P = x_Q \text{ and } y_P < y_Q. \end{cases} \tag{1.41}$$

The two orderings $\{T^1, T^2\}$ are total orders.

1.4.1 Cryptanalysis of Encryption Algorithms

Security analysis plays a vital role in evaluating the efficiency of any cryptosystem. The process of deciphering ciphertexts without the use of a key is known as cryptanalysis, and the person who executes it is known as a cryptanalyst. Below is an explanation of some of the attacks used in cryptanalysis.

- **Brute-force attack:** A third party attempts all possible keys in this attack to break a cryptosystem to obtain the plaintext from the ciphertext. A cryptosystem's resistance to this attack is directly correlated with the size of the key.
- **Chosen-plaintext attack:** In this attack, an intruder choose arbitrary plaintexts to find their ciphertexts. The goal of this attack is to obtain information about the ciphertext in order to break the cryptosystem.
- **Chosen-ciphertext attack:** This attack is similar to the chosen-plaintext attack. An intruder chooses arbitrary ciphertext, decrypt it to find their plaintexts. The objective is to breach the cryptosystem by gathering information about plaintexts.
- **Known-plaintext attack:** In this attack, the cryptanalyst has access to some plaintext and the associated ciphertext. He tries to obtain further ciphertext or a secret key using this information.

Definition 1.32. For two data sets x and y of size M, the correlation is calculated as

$$\sigma_{xy} = \frac{\sum\limits_{i=1}^{M}(x_i - E(x))(y_i - E(y))}{\sqrt{\sum\limits_{i=1}^{M}(x_i - E(x))^2 \sum\limits_{i=1}^{M}(y_i - E(y))^2}}, \tag{1.42}$$

where

$$E(x) = \frac{1}{M}\sum_{i=1}^{M} x_i, \tag{1.43}$$

for $x_i \in x$ and $y_i \in y$.

Definition 1.33. Entropy is the key test to measure the randomness in an image data. It shows the intensity distribution of image pixels. Let I be a grayscale image then the entropy

(H) is

$$H = -\sum_{i=1}^{255} P(x_i)\log_2(P(x_i)). \tag{1.44}$$

In an image data high entropy implies high randomness.

Definition 1.34. The sensitivity of an encryption algorithms is measured using unified average change intensity (UACI) [42] and number of pixel change rate (NPCR) [42] analysis. In UACI-NPCR analysis, the ciphered images C_I and C'_I of $I_{N\times M}$ and $I'_{N\times M}$ plain-images are obtained by changing a single pixel. The NPCR value is calculated as

$$NPCR = \sum_{i,j} \frac{D(i,j)}{N \times M} \times 100, \tag{1.45}$$

where $D(i,j) = 1$ if $C_I(i,j) - C'_I(i,j) \neq 0$ and $D(i,j) = 0$ otherwise.

Mathematically, the UACI is given as

$$UACI = \sum_{i,j} \frac{|I_C(i,j) - I'_C(i,j)|}{N \times M \times 255} \times 100. \tag{1.46}$$

Definition 1.35. The histogram of a data collection $\mathcal{D}$ over a set of symbols Γ is a function $h_{\mathcal{D}}$ over Γ such that for each $v \in \Gamma$, $h_{\mathcal{D}}(v)$ is equal to the quantity of v in $\mathcal{D}$. The frequency of $v \in \mathcal{D}$ is denoted by the symbol $h_{\mathcal{D}}(v)$. If every component of the symbol set v in the data collection $\mathcal{D}$ has the same frequency then the data collection have a uniform histogram.

Definition 1.36. The randomness of binary sequences is tested using NIST-800-22 statistical tests [8]. NIST-800-22 consists of 15 different tests and the results are classified using the predefined p-value. If the p-value is 0.001, then each test's resultant p-values necessarily greater than or equal to 0.001 to pass the test.

Chapter2

Designing a Substitution-box Generator Based on Finite Elliptic Curves for Block Ciphers

2.1 Introduction

We develop a robust S-box generator that is suitable for lightweight cryptography and outperforms previously designed S-box generators in terms of computation time and security. We produce particular sequences of integers using ordered ECs of a short size and binary sequences, which are subsequently utilised to generate S-boxes. We conducted numerous conventional tests to find out the efficiency of the proposed generator. Computational results and comparisons indicate that the new generator requires less operating time and has more security against modern attacks than numerous existing well-known generators. The rest of the chapter is organized as: In Sec. 2.2, related work to the new S-box generator is given. A complete description of the generator is given in Sec. 2.3. In Sec. 2.4, a detailed analysis of the S-box generator is conducted. Conclusion is given in Sec. 2.5.

2.2 Background of the S-box Generator

An S-box is a basic component of many modern cryptosystems [31, 32], including DES and AES. The existing block ciphers, for example AES uses a static S-box to create confusion in a plaintext. However, several researchers have proposed to use a dynamic S-box in place of a static S-box to enhance the security of secret data against modern cryptanalysis [33–35]. Therefore, it is necessary to develop an S-box generation scheme that can generate cryptographically secure S-boxes.

Recently, several S-box generators have been proposed based on different mathematical structures. For example, in [43–50] chaotic maps are employed to generate S-boxes. Chaos based S-box generation methods such as [43–50] have small computation time but generate S-boxes with low NL. Graph theory and algebraic structures are used to design S-box generators in [51–56]. Although algebraic methods construct S-boxes with high NL, but they construct only a limited number of distinct dynamic S-boxes. For example, the generator in [57] generates 256 S-boxes, while the scheme in[20] generates total 16 strong S-boxes and the method in [21] constructs only one S-box. Recently, quantum walks and various optimization techniques have been used to construct cryptographically strong S-boxes [58–65]. But these techniques require large computation time to construct S-boxes with high NL. Linear fractional transformations are used to generate key-dependent S-boxes in [66, 67]. In [68], Musheer et al. used chaotic heuristic search and group action to construct and improve the cryptographic features of generated S-boxes. Furthermore, Wang et al. [69] successfully designed a secure user authentication scheme with low computation cost. In [70, 71] researchers suggested different authentication protocols for mobile devices and security of wireless sensor networks.

ECs are algebraic structures used in cryptography that provide excellent security with a short key size. Subsequently, some improved EC-based S-box generators have been proposed in [14, 16, 24, 25, 41, 72, 73]. These generators can output cryptographically strong S-boxes but they require to generate or store a precomputed EC. Furthermore, the number of distinct S-boxes obtained by these generators is directly proportional to the size of the underlying EC. Therefore, these generators cannot be used with ECs of large size and hence can generate a small number of S-boxes over an EC of small size. To address these limitations, Saleh and Abbas [74] designed an S-box generator over an EC of large size by generating some points over an EC which are then used for creating a permutation code. This scheme does not require a complete EC but still needs to do some calculations over a large prime. Therefore, the scheme may not be suitable for lightweight cryptography which has low storage capacity and limited computational power [75, 76].

2.3 Discription of the New Generator

The existing EC-based S-box generators [14, 23, 24, 41, 72, 74] require computation over large primes to generate S-boxes. Therefore such S-box generators are not suitable for lightweight cryptography. To overcome this problem, we proposed a new S-box generator to efficiently generate dynamic and cryptographically strong S-boxes based on an EC with small size. We use an ordered EC to create randomness in the integers in $[0, 2^n - 1]$ and a binary sequence to generate a $n \times n$ bijective S-box. The aim of the binary sequence is to generate plaintext-dependent S-boxes, for example, we can use SHA-256 hash function [77] to generate binary sequences for the plaintext which can be used in our generator to output plaintext-dependent S-boxes. Our generator has the following ten main steps to generate $n \times n$ bijective S-boxes, where we denote $2^n - 1$ by ℓ for notational convenience.

(1) Select two sequences $A^i = (a_0^i, a_1^i, \ldots, a_\ell^i)$ over the set of non-negative integers, $i = 1, 2$;

(2) Select an EC $E_{p,0,b}$ with $p \equiv 2 \pmod 3$, $p \geq 2^n$ and two orderings $\prec_{T^i}$, $i = 1, 2$;

(3) Select two sets $B^i = \{b_0^i, b_1^i, \ldots, b_\ell^i\}$, $i = 1, 2$ such that $|B^i| = 2^n$ and $B^i \pmod{2^n} = [0, \ell]$;

(4) Compute the sets $C^i = \{c_0^i, c_1^i, \ldots, c_\ell^i\}$ such that $c_j^i = (x_j^i, (b_j^i + a_j^i) \pmod p) \in E_{p,0,b}$;

(5) Now to create randomness in B^i, sort it w.r.t. C^i such that b_j^i is smaller than $b_{j'}^i$ if c_j^i is smaller than $c_{j'}^i$ w.r.t. the ordering $\prec_{T^i}$, $i = 1, 2$;

(6) Let $K^i = (k_0^i, k_1^i, \ldots, k_\ell^i)$ for $i = 1, 2$ denote the sequences obtained from ordered B^i after applying modulo 2^n, where $k_j^i \equiv (b_j^i) \pmod{2^n}$;

(7) Now, we generate an S-box $\sigma^{1,2}(A^i, T^i, B^i, p, b, n) : [0, \ell] \to [0, \ell]$ such that

$$\sigma^{1,2}(A^i, T^i, B^i, p, b, n)(k_j^1) = k_j^2, j \in [0, \ell];$$

(8) Generate a binary sequence L of size $n2^n$ and divide it from left to right into subsequences (s_j) each of length n. Now, convert these subsequences into decimal numbers n_j, $j \in [0, \ell]$. Let $N = (n_0, n_1, \ldots, n_\ell)$ be the sequence of integers obtained from the decimal form of the subsequences;

(9) Define a total order $\prec_T$ on the integers in $[0, \ell]$ based on N such that for $i, j \in [0, \ell]$ it holds that $i \prec_T j$ if "$n_i < n_j$" or "$n_i = n_j$ and $i < j$". Let $Q = (q_0, q_1, \ldots, q_\ell)$ denote the sequence obtained from the ordered set $[0, \ell]$ where the entries are listed from smallest to largest w.r.t. $\prec_T$;

(10) Finally, for $i = 1, 2$ we generate S-boxes $\sigma^i(A^i, T^i, B^i, L, p, b, n)$ such that

$$\sigma^i(A^i, T^i, B^i, L, p, b, n)(q_j) = k_j^i, j \in [0, \ell];$$

For given parameters $A^i, T^i, B^i, L, p, b,$ and n, $i = 1, 2$, the S-box generator generates three S-boxes $\sigma^i(A^i, T^i, B^i, L, p, b, n)$, $i = 1, 2$, and $\sigma^{1,2}(A^i, T^i, B^i, p, b, n)$. We observe that for two different binary sequences L and L', and fixed $A^i, T^i, B^i, p, b,$ and n, the corresponding S-boxes are different, i.e., it holds that

$$\sigma^i(A^i, T^i, B^i, L, p, b, n)(s) \neq \sigma^i(A^i, T^i, B^i, L', p, b, n)(s), \forall s \in [0, \ell]. \tag{2.1}$$

This proposition follows from the fact that the sequences N and N' that are obtained from L and L', respectively, in step (8) are different when $L \neq L'$.

An immediate application of our generator is to generate plaintext-dependent S-boxes which play an important role against plaintext attacks in image encryption schemes. For example, we can use the SHA-256 hash function to generate a binary sequence of length 256 which can be converted to a binary sequence of length $n2^n$ by simply replicating the SHA-256 hash sequence. Thus for each image, we can get a different binary sequence L and hence by Eq. 2.1, we can generate a different S-box for each image and can provide high security against cryptographic attacks.

A flowchart of the generator is given in Fig. 2.1 and an example of a 4 × 4 S-box generated by the new scheme is given in Fig. 2.2, where we use an EC $E_{17,0,1}$ and diffusion ordering T^1 given in Eq. 1.40 and natural ordering T^2 given in Eq. 1.41.

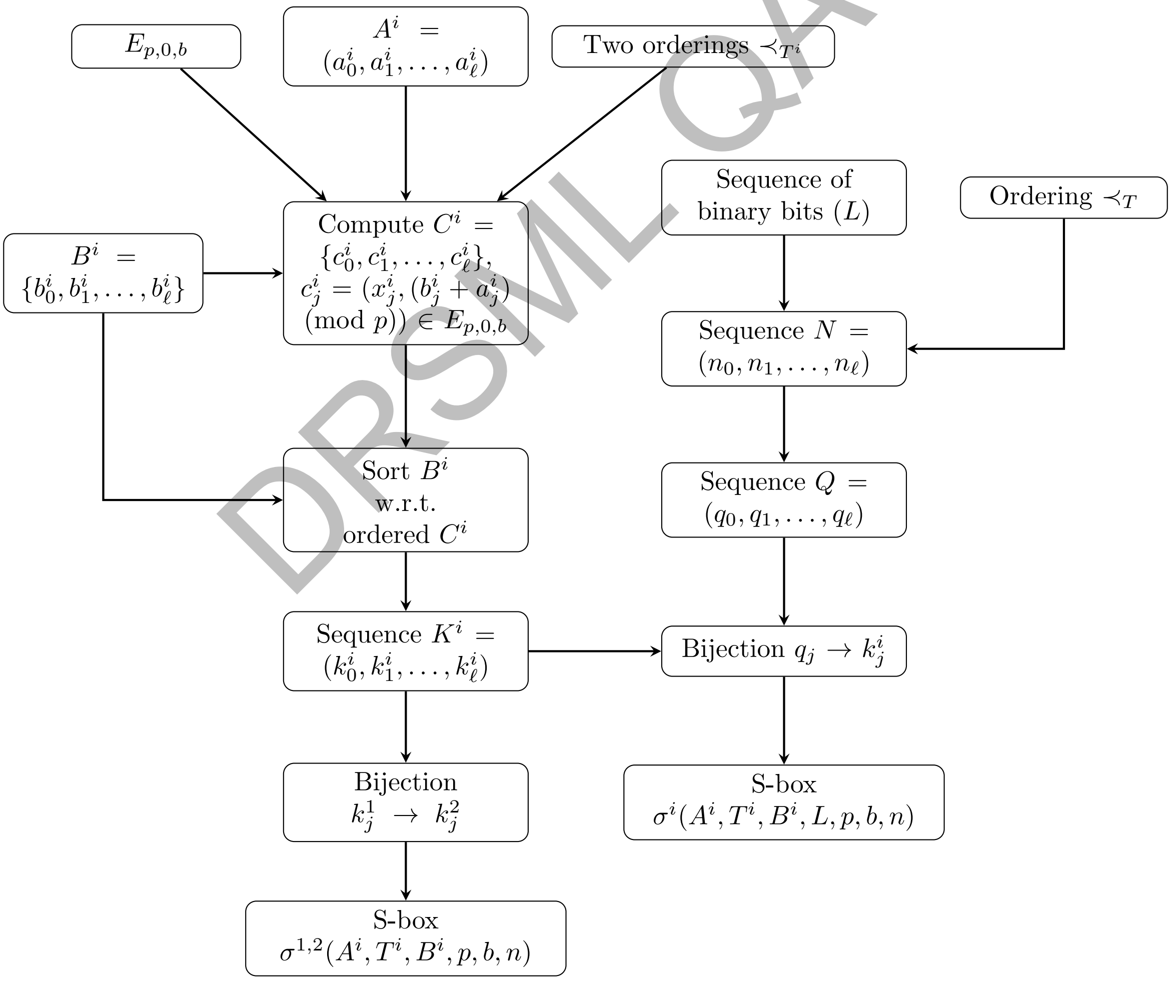

$E_{p,0,b}$
$A^i = (a^i_0, a^i_1, \ldots, a^i_\ell)$
Two orderings $\prec_{T^i}$
Sequence of binary bits (L)
Ordering $\prec_T$
$B^i = \{b^i_0, b^i_1, \ldots, b^i_\ell\}$
Compute $C^i = \{c^i_0, c^i_1, \ldots, c^i_\ell\}$, $c^i_j = (x^i_j, (b^i_j + a^i_j) \pmod p)) \in E_{p,0,b}$
Sequence $N = (n_0, n_1, \ldots, n_\ell)$
Sort B^i w.r.t. ordered C^i
Sequence $Q = (q_0, q_1, \ldots, q_\ell)$
Sequence $K^i = (k^i_0, k^i_1, \ldots, k^i_\ell)$
Bijection $q_j \to k^i_j$
Bijection $k^1_j \to k^2_j$
S-box $\sigma^i(A^i, T^i, B^i, L, p, b, n)$
S-box $\sigma^{1,2}(A^i, T^i, B^i, p, b, n)$

Figure 2.1: Flowchart of the generator.

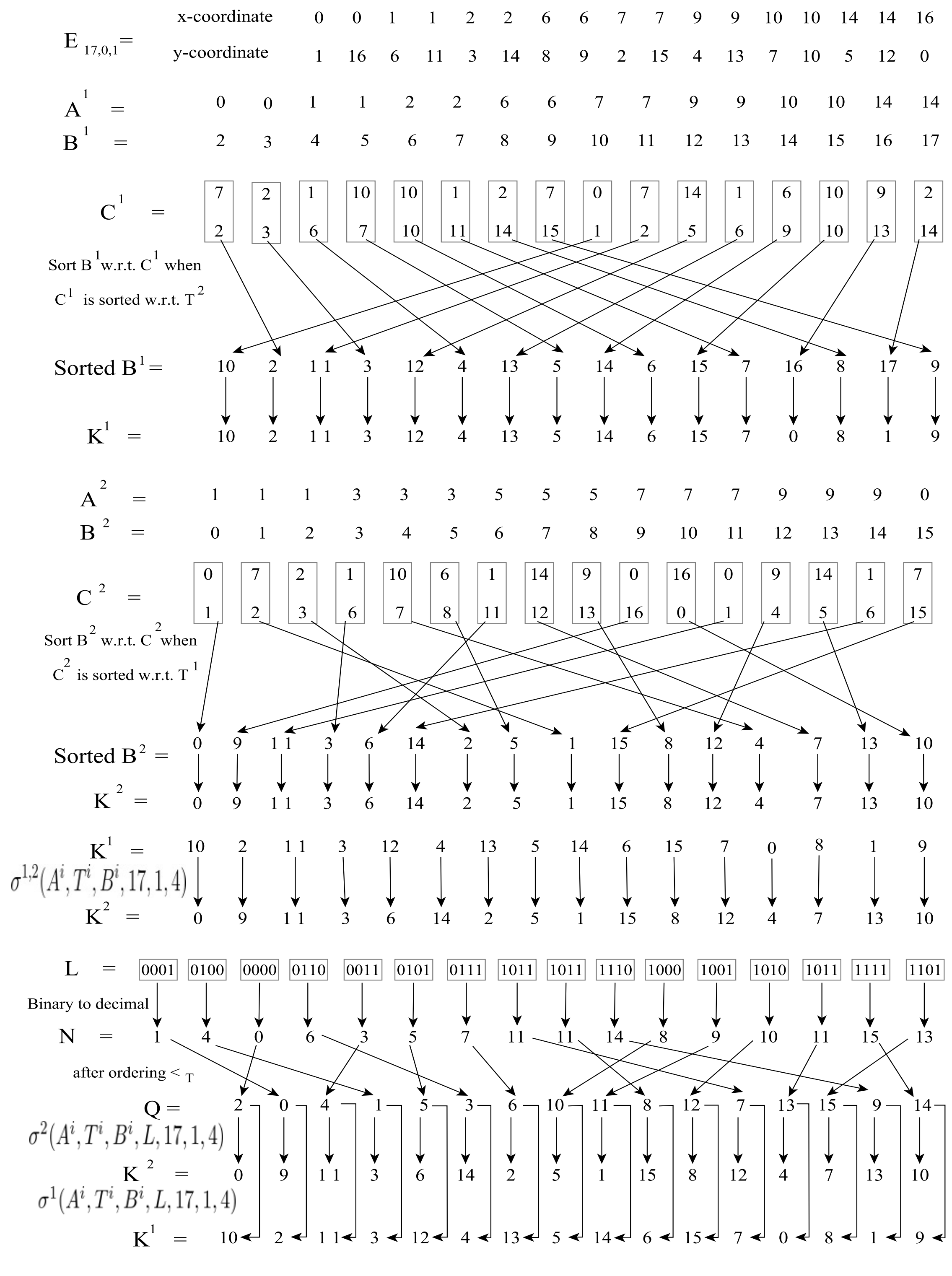

E 17,0,1 =
x-coordinate 0 0 1 1 2 2 6 6 7 7 9 9 10 10 14 14 16
y-coordinate 1 16 6 11 3 14 8 9 2 15 4 13 7 10 5 12 0
A 1 = 0 0 1 1 2 2 6 6 7 7 9 9 10 10 14 14
B 1 = 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
C 1 = 7 2, 2 3, 1 6, 10 7, 10 10, 1 11, 2 14, 7 15, 0 1, 7 2, 14 5, 1 6, 6 9, 10 10, 9 13, 2 14
Sort B 1 w.r.t. C 1 when C 1 is sorted w.r.t. T 2
Sorted B 1 = 10 2 11 3 12 4 13 5 14 6 15 7 16 8 17 9
K 1 = 10 2 11 3 12 4 13 5 14 6 15 7 0 8 1 9
A 2 = 1 1 1 3 3 3 5 5 5 7 7 7 9 9 9 0
B 2 = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
C 2 = 0 1, 7 2, 2 3, 1 6, 10 7, 6 8, 1 11, 14 12, 9 13, 0 16, 16 0, 0 1, 9 4, 14 5, 1 6, 7 15
Sort B 2 w.r.t. C 2 when C 2 is sorted w.r.t. T 1
Sorted B 2 = 0 9 11 3 6 14 2 5 1 15 8 12 4 7 13 10
K 2 = 0 9 11 3 6 14 2 5 1 15 8 12 4 7 13 10
K 1 = 10 2 11 3 12 4 13 5 14 6 15 7 0 8 1 9
σ1,2(Ai,Ti,Bi,17,1,4)
K 2 = 0 9 11 3 6 14 2 5 1 15 8 12 4 7 13 10
L = 0001 0100 0000 0110 0011 0101 0111 1011 1011 1110 1000 1001 1010 1011 1111 1101
Binary to decimal
N = 1 4 0 6 3 5 7 11 11 14 8 9 10 11 15 13
after ordering < T
Q = 2 0 4 1 5 3 6 10 11 8 12 7 13 15 9 14
σ2(Ai,Ti,Bi,L,17,1,4)
K 2 = 0 9 11 3 6 14 2 5 1 15 8 12 4 7 13 10
σ1(Ai,Ti,Bi,L,17,1,4)
K 1 = 10 2 11 3 12 4 13 5 14 6 15 7 0 8 1 9

Figure 2.2: An example of 4×4 S-boxes $\sigma^{1,2}(A^i, T^i, B^i, 17, 1, 4)$ and $\sigma^i(A^i, T^i, B^i, L, 17, 1, 4)$, $i = 1, 2$ generated by the S-box generator over the EC $E_{17,0,1}$ with diffusion ordering T^1 and natural ordering T^2.

2.4 Security Analysis and Comparison

We conduct rigorous analysis to test the dynamic behavior and cryptographic properties of the proposed S-box generator in Secs. 2.4.1 and 2.4.2, respectively.

2.4.1 Analysis of Dynamic Behavior of the Proposed Generator

For the comparison, we randomly generated two sets Ω_E and Ω_L each of size 10,000 of bijective S-boxes $\sigma^2(A^i, T^i, B^i, L, p, b, 8)$. Here the set Ω_E is generated by fixing an $E_{p,0,b}$ and randomly generated 10,000 sequences L, and Ω_L is generated by fixing a sequence L and randomly generated 10,000 ECs $E_{p,0,b}$ as follows:

- Fix $A^i = (0, 0, \ldots, 0)$ and $B^i = \{0, 1, \ldots, 255\}$, $i = 1, 2$;
- Fix the EC $E_{1031,0,2}$, two orderings diffusion ordering T^1 [41] and natural ordering T^2 [41], and randomly generate 10,000 binary sequences each of length 2^8. Each binary sequence is replicated 8-times to construct a binary sequence L of length $2,048$. Thus, we get 10,000 sequences L of length $2,048$, and generate a set Ω_E of S-boxes of size 10,000.
- We generate another set Ω_L of S-boxes of size 10,000, by fixing $L =$ 011001001110010010001000100011101011011100001001010110001011000011000111100000011001110011110000101111110100110100110000010101010110011111001111011011101001001110011000011000111000000100010100100111111011010100000001110101001001110110101000110110010100000000100, and randomly generate an EC $E_{p,0,b}$ by selecting parameters $p \in \{1031, 1049, 1061, 1091, 1097, 1103, 1109, 1151, 11159, 11939\}$ and $b \in [1, 1000]$, where orderings are diffusion ordering T^1 and natural ordering T^2.

2.4.1.1 Sensitivity Analysis

The minimum, maximum and average sensitivity of S-boxes in the sets Ω_L and Ω_E are recorded in Table 2.1. The average sensitivity of the S-box generator is 255 and 250 for the sets Ω_L and

Ω_E respectively, which is near the ideal value of 256. Furthermore, we show the effect of the parameters p, b and L on the resultant S-boxes in Fig. 2.3, which also implies that our algorithm is highly sensitive to the inputs.

Table 2.1: Sensitivity analysis.

S-box set	Sensitivity		
	minimum	maximum	average
Ω_L	245	256	255
Ω_E	176	256	250

2.4.1.2 Singularity Analysis

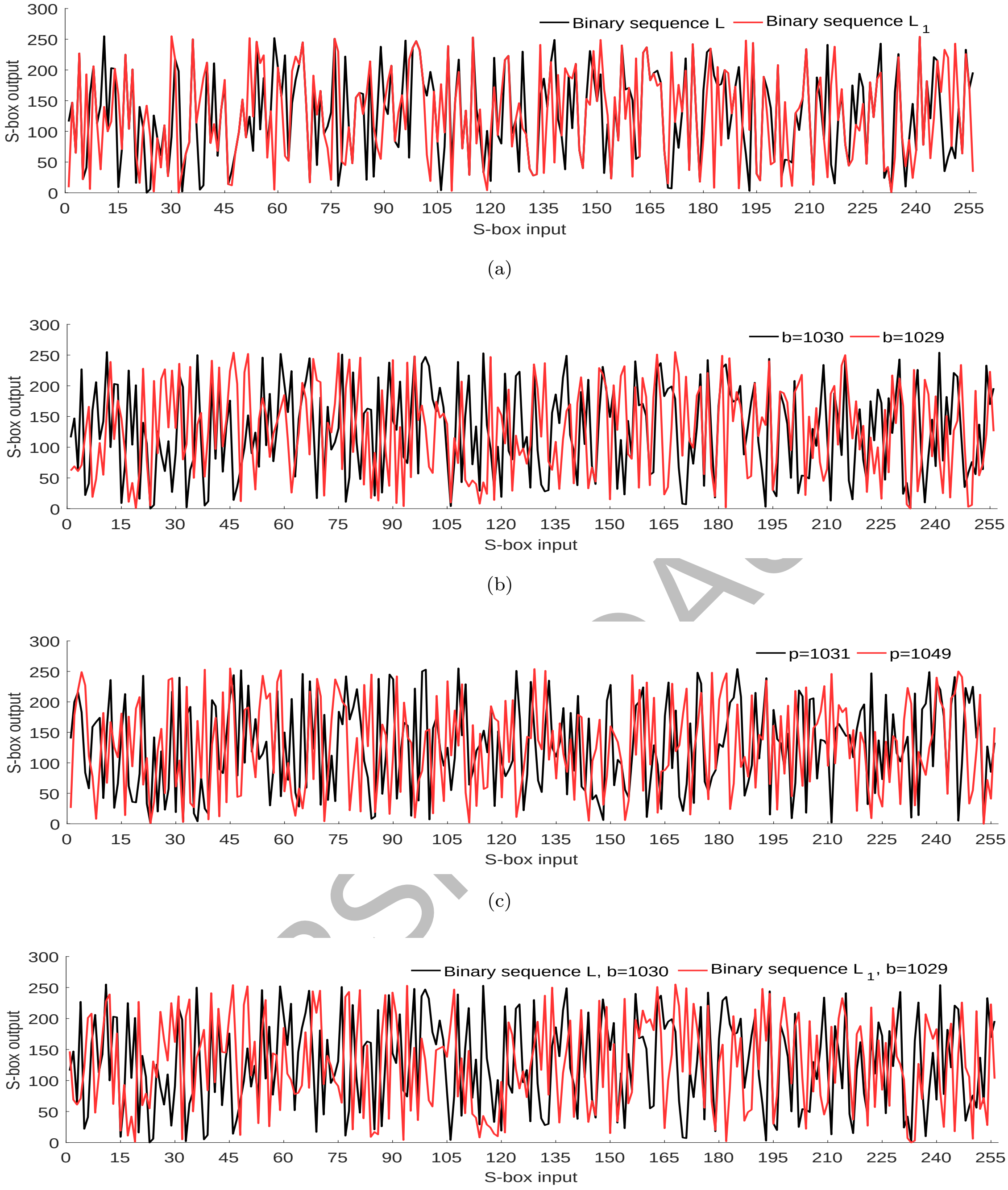
The proposed S-box generator has no singularity and outputs S-boxes for each set of valid inputs. The comparison of the singularity analysis of the new S-box generator with the generators in [14, 21, 23, 25] is given in Table 2.2. Our generator and the generator in [25] have no singularities, however, the generators in [14, 21, 23] have singularities. Hence the proposed generator is better than the generators in [14, 21, 23].

Table 2.2: Singularity comparison.

S-box generator	Proposed	[21]	[23]	[14]	[25]
Singularity	No	Yes	Yes	Yes	No

2.4.1.3 Variation Analysis

We have generated two sets Ω_L and Ω_E of S-boxes each of size 10,000 by fixing a binary sequence L and the EC $E_{1031,0,2}$, respectively. It has been observed that for distinct 10,000 binary sequences L, we have distinct 10,000 S-boxes and also any change in parameters of an EC $E_{p,0,b}$ gives a new distinct S-box. We compare the results with the generators in [14, 20–25]. The results of the distinct S-boxes analysis are plotted in Fig. 2.4(a), from which it is evident that the S-box generation scheme constructs a large number of distinct S-boxes when compared with existing S-box generators in [14, 20–25], where for convenience we denote the total number of distinct S-boxes by # S-boxes.



(a)

(b)

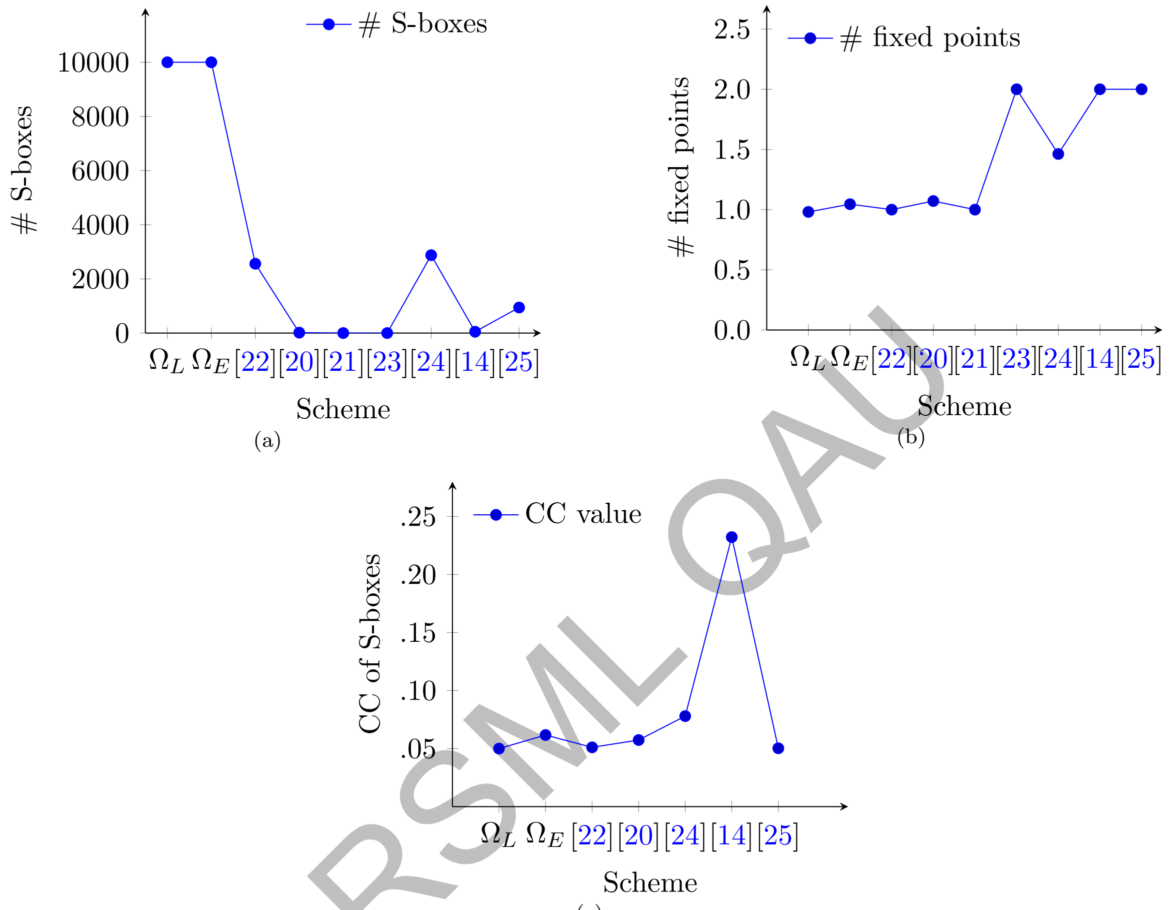
(c)

(d)

Figure 2.3: Sensitivity analysis. (a) Effect of binary sequence on the generator's output for the parameters $(A^i, T^i, B^i, L, 1031, 1030, 8)$, where L_1 is obtained by changing the 185-th bit of L. (b) Effect of b on the generator's output for the parameters $(A^i, T^i, B^i, L, 1031, b, 8)$ for $b \in \{1030, 1029\}$. (c) Effect of p on the S-box generator's output for the parameters $(A^i, T^i, B^i, L, p, 1, 8)$ for $p \in \{1031, 1049\}$. (e) Effect of binary sequence and b on the S-box generator's output for the parameters $(A^i, T^i, B^i, L, 1031, b, 8)$ for $b \in \{1030, 1029\}$.

2.4.1.4 Fixed Point Analysis

We determined the average number of fixed points of S-boxes in the sets Ω_L, Ω_E and S-boxes generated by the generators in [14, 20–25]. These results are given in Fig. 2.4(b) from which it is clear that our generator has a smaller number of fixed points than the generators in [14, 23–25] and is comparable with the generator in [20–22], for convenience we denote the total number of fixed points by # fixed points.



(a)

(b)

(c)

Figure 2.4: Analysis of the new algorithm with generators in [14, 20–25]. (a) Variation analysis. (b) Fixed point analysis. (c) Comparison of average correlation coefficient.

2.4.1.5 Correlation Analysis

The correlation coefficient (CC) of S-boxes in Ω_L, Ω_E and S-boxes generated by the generators in [14, 20, 22, 24, 25] is given in Fig. 2.4(c). The results in Fig. 2.4(c) show that the proposed generator has smaller CC than the generators in [14, 24] and is comparable with the generators in [20, 22, 25].

2.4.1.6 Computational Speed Analysis

An S-box generator can be used for real-time encryption if it has a low computational cost. We analyze the computation time of the proposed S-box generator over ECs of different sizes and compare it with other S-box generators in [14, 23, 25]. For experimental purpose, we use MATLAB R2016a on a system, Intel(R) Core(TM) i3-2370M CPU @ 2.40GHz with 6 GB of RAM. For computational analysis, two S-boxes are generated by each of the generators in [14, 23, 25] and S-boxes $\sigma^2(A^i, T^i, B^i, L, 2111, 9, 8)$ and $\sigma^2(A^i, T^i, B^i, L, 10247, 1, 8)$ by the proposed generator, using the same aforementioned setup. For experimental setup, we have fixed the uncommon parameters ($n = 8$, $m = 8$, $\ell = 1$, natural ordering(N)) and kept the overlapping parameters same for the each S-box generator. The computation time in seconds for these generators is presented in Table 2.3. Our generator has the lowest running time. Therefore, the S-box generator is suitable for lightweight cryptography and encryption purposes as compared to the generators in [14, 23, 25].

Table 2.3: Computation time comparison with existing S-box generators.

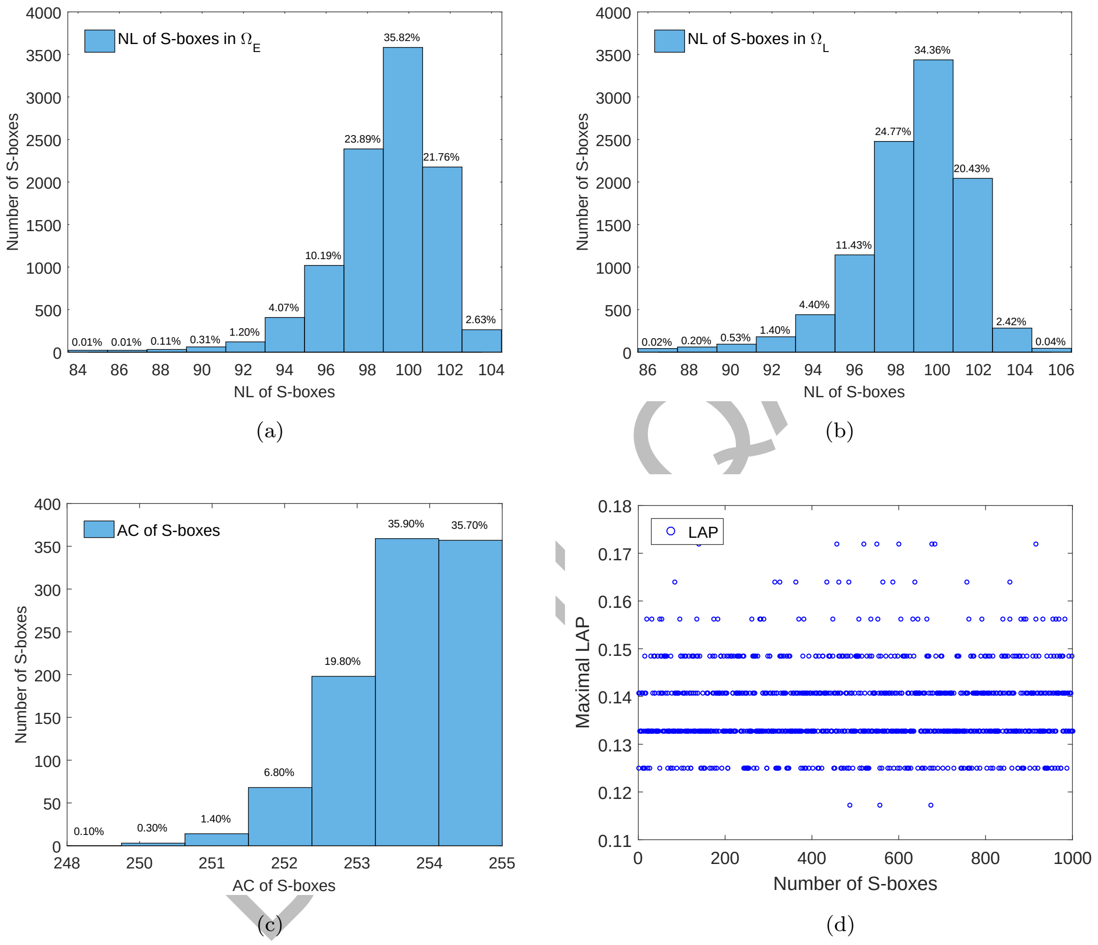
EC	Proposed	Ref. [23]	Ref. [14]	Ref. [25]
$E_{2111,0,9}$	0.00701	0.037278	0.040360	0.148023
$E_{10247,0,1}$	0.031848	0.704746	0.630767	0.659935

2.4.2 Cryptographic Properties

We compute and compare the cryptographic characteristics of S-boxes in the sets Ω_L and Ω_E and S-boxes constructed by the generators in [14, 33, 45, 46, 55, 74].

2.4.2.1 Linear Attacks

Nonlinearity (NL): We computed the NL of S-boxes in both sets Ω_E and Ω_L, and results are plotted in Fig.2.5(a)-(b). We observe that more than 93% S-boxes in these sets have the NL at least 96. We compared the minimum, maximum, and average NL of S-boxes in Ω_L and Ω_E, and 10,000 S-boxes generated by the generators in [14, 33, 45, 46, 55, 74], and the results are listed in Table 2.4. The minimum (resp., maximum and average) NL of S-boxes obtained by the generators in [14] (resp., [33], [46], [45] and [55]) is lower than the newly generated S-boxes. This implies that the S-box generation algorithm generates S-boxes with high NL as compared to the generators in [14, 33, 45, 46, 55, 74].



(a)

(b)

(c)

(d)

Figure 2.5: (a) Histogram of the NL of the S-boxes in Ω_E. (b) Histogram of the NL of the S-boxes in Ω_L.

Table 2.4: The NL analysis of 10,000 S-boxes.

Generator	Nonlinearity		
	minimum	maximum	average
Proposed Ω_L	86	106	99.10
Proposed Ω_E	84	104	99.27
Ref. [33]	52	104	84.64
Ref. [45]	0	106	90.20
Ref. [46]	82	104	99.05
Ref. [55]	82	104	97.45
Ref. [14]	64	102	92.05
Ref. [74]	84	106	99.07

Algebraic Complexity (AC): We computed the AC of 1000 S-boxes in Ω_L and Ω_E that have the best NL and find that more than 99% S-boxes have the AC at least 251. Histogram analysis of the AC is also illustrated in Fig. 2.5(c).

Linear approximation probability (LAP): We computed the LAP of 1000 S-boxes and the results are shown in Fig. 2.5(d). The LAP of S-boxes is in the range [0.117, 0.172]. This further justifies our claim that the proposed algorithm is highly secure against linear attacks.

2.4.2.2 Differential Analysis

Differential approximation probability (DAP): We computed the DAP of 1000 S-boxes generated by the S-box generator and the outcomes are given in Fig.2.6. All the S-boxes have the DAP value in the interval [0.039, 0.063] which are in the acceptable range. Therefore the proposed generator has high resistance against differential attacks.

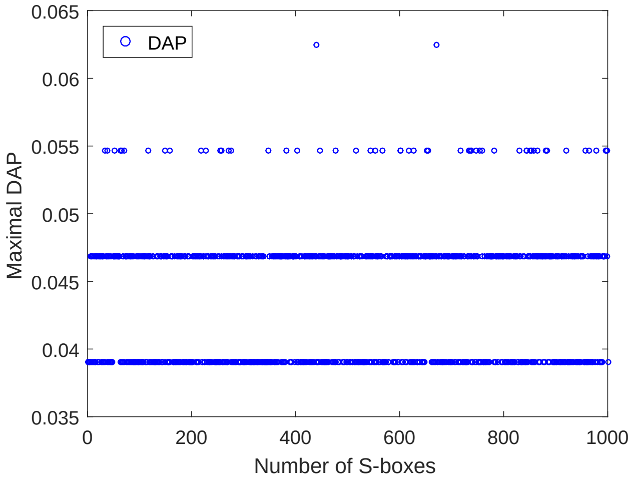


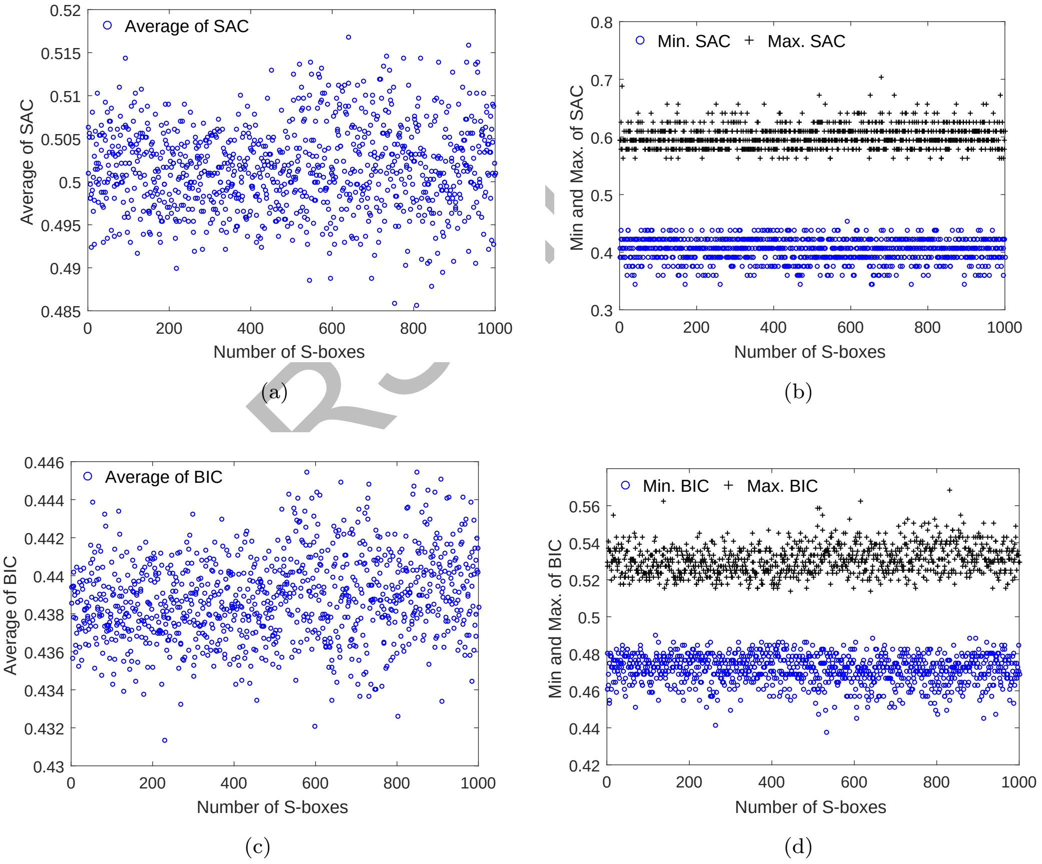
Figure 2.6: Distribution of the DAP for 1000 S-boxes constructed by the proposed generator.

2.4.2.3 Analysis of Boolean Function

Strict Avalanche Criterion (SAC): We applied the SAC on 1000 S-boxes and illustrated the average, minimum and maximum values for each S-box obtained by SAC criterion in Fig.2.7(a)-(b). The minimum, maximum and average values obtained by the SAC test are in the ranges

[0.344, 0.453], [0.563, 0.703], and [0.486, 0.517], respectively. These ranges are near to the optimal value of 0.5, and hence newly generated S-boxes pass the SAC test, and so the newly designed S-boxes have good resistance against Boolean functions attacks.

Bit Independence Criterion (BIC): We computed the BIC of 1000 S-boxes and illustrated the average, minimum and maximum values for each S-box obtained by this criterion in Fig. 2.7(c)-(d). The minimum, maximum and average values obtained by the BIC tests are in the ranges [0.438, 0.490], [0.514, 0.568], and [0.431, 0.445], respectively. These ranges are near to the optimal value of 0.5, and hence newly generated S-boxes pass the BIC test, and hence they have good resistance against Boolean functions attacks.



(a)

(b)

(c)

(d)

Figure 2.7: Distribution of the SAC and BIC of 1000.(a) Average value of the SAC. (b) Distribution of minimum and maximum of the SAC. (b) Average value of the BIC. (d) Distribution of minimum and maximum of the BIC.

Table 2.5: The S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ generated by proposed method.

23	198	150	201	232	149	175	140	29	16	199	229	224	12	87	62
84	92	68	2	116	133	30	172	38	254	247	195	134	194	139	222
176	33	160	48	162	3	230	179	205	43	125	49	131	85	178	19
25	77	97	47	233	203	72	55	46	9	158	209	196	208	53	107
181	76	120	152	79	51	165	73	220	60	27	246	164	66	94	163
170	174	145	50	54	31	121	114	250	221	122	171	153	231	180	81
190	40	245	200	193	234	1	104	215	148	244	156	241	237	155	144
52	161	136	35	42	8	252	249	236	217	95	182	188	17	146	166
226	157	218	117	21	59	34	154	58	207	255	111	248	26	83	11
102	67	99	108	204	214	228	240	103	142	88	112	41	28	15	225
69	202	129	78	135	61	184	137	126	211	186	4	177	20	89	147
173	101	96	65	100	109	242	185	169	45	143	197	14	110	80	98
124	5	105	10	37	113	63	235	168	183	90	128	24	93	251	239
13	56	243	115	132	39	223	36	86	212	18	7	70	227	71	91
253	32	167	22	106	213	206	57	189	159	0	138	219	64	74	141
192	216	6	130	75	187	119	210	82	118	127	191	123	238	44	151

Table 2.6: Comparisons of S-box analysis.

S-boxes	Method	NL	LAP	DAP	AC	SAC		BIC		
						min.	max.	min.	max.	NL
Ref. [43]	Chaos	96	0.023	0.050	254	0.391	0.625	0.477	0.531	92
Ref. [44]		103	0.132	0.039	255	0.398	0.570	0.472	0.535	96
Ref. [47]		100	0.129	0.039	255	0.422	0.594	0.477	0.525	98
Ref. [48]		100	0.152	0.039	255	0.391	0.586	0.468	0.537	100
Ref. [49]		96	0.125	0.047	255	0.422	0.609	0.471	0.547	96
Ref. [50]		104	0.094	0.023	255	0.391	0.578	0.476	0.529	103
Ref. [55]	Other	104	0.141	0.054	253	0.406	0.594	0.461	0.522	98
Ref. [56]		104	0.148	0.047	254	0.438	0.578	0.482	0.543	96
Ref. [21]		100	0.125	0.016	255	0.391	0.594	0.477	0.533	100
Ref. [78]		104	0.109	0.047	255	0.391	0.593	0.454	0.49	102
Ref. [58]		98	0.133	0.054	254	0.422	0.609	0.477	0.535	94
Ref. [59]		96	0.125	0.039	255	0.359	0.609	0.477	0.541	98
Ref. [60]	Optimized	102	0.148	0.039	254	0.375	0.609	0.470	0.521	100
Ref. [61]		104	0.133	0.039	255	0.359	0.609	0.457	0.535	96
Ref. [62]		104	0.133	0.039	254	0.438	0.641	0.475	0.547	98
Ref. [63]		102	0.133	0.039	254	0.359	0.562	0.467	0.535	98
Ref. [23]	ECs	104	0.145	0.039	255	0.391	0.625	0.471	0.531	98
Ref. [24]		106	0.148	0.047	255	0.406	0.625	0.471	0.539	96
Ref. [41]		106	0.148	0.039	255	0.406	0.641	0.471	0.537	98
Ref. [14]		106	0.148	0.039	255	0.438	0.594	0.465	0.545	98
Ref. [25]		106	0.188	0.039	253	0.406	0.609	0.465	0.527	98
Proposed		106	0.148	0.039	254	0.422	0.594	0.471	0.533	98

2.4.2.4 Further Comparison

We further compare the cryptographic properties of the S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ given in Table 2.5 and the S-boxes obtained by the generators in [14, 21, 23–25, 41, 43, 44, 47–50, 55, 56, 58–63, 78]. The cryptographic properties of these S-boxes are recorded in Table 2.6.

We list the key observations from these computational results as follows.

- Table 2.6 predicts that the S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ has better NL as compared to S-boxes in [23, 43, 44, 47–50, 55, 56, 58–63, 78]. So, the proposed S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ has high security as compared to S-boxes in [23, 43, 44, 47–50, 55, 56, 58–63, 78].
- On basis of the LAP listed in Table 2.6, it is concluded that the proposed S-box has a comparable LAP value with all existing schemes in Table 2.6. This finding indicates that the S-box generated by our scheme has high resistance against linear attacks.
- The DAP value of the S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ given in Table 2.6 shows that the proposed S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ is more secure against differential attacks than S-boxes in [24, 43, 49, 55, 56, 58, 78].
- The AC of the S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ is 254 which is very close to the optimal value. This is evident that the proposed S-box has high security against algebraic attacks.
- Table 2.6 shows that the minimum (resp., maximum) of the SAC for the S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ is 0.422 (resp., 0.594) which is near to 0.5, the optimal SAC value. We observe that the S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ has better SAC results as compared to S-boxes in [23–25, 41, 43, 44, 48, 50, 55, 58–61, 63, 74, 78]. Hence, the proposed S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ resists Boolean function cryptanalysis.
- The BIC of the S-box $\sigma^2(A^i, T^i, B^i, L, 1049, 600, 8)$ is almost comparable to S-boxes in Table 2.6.

2.5 Conclusion

To address the shortcomings of the traditional S-box generators, we presented a new generator based on ordered ECs and binary sequences. We tested and compared the efficiency of the proposed generator with several state-of-the-art existing generators. From the rigorous analysis we notice the following advantages of the proposed generator over the existing methods [14, 22–25, 33, 45, 46, 55, 74]:

- The proposed generator is better than the methods in [14, 21, 23] because they have singularities and are unable to generate an S-box for a valid set of inputs, on the other hand, our method has no singularity.
- From computational experiments, it is evident that the proposed generator generates a large number of highly uncorrelated S-boxes over an EC with a small size which is not possible in existing generators [14, 20–25].
- It is evident from Table 2.4 that our generator has better average NL when compared with generators in [14, 33, 45, 46, 55, 74].
- The most important feature of the proposed generator that qualifies it for lightweight cryptography is its less computation time and use of small size ECs than generators in [14, 23, 25].
- The detailed security analysis in Table 2.6 proves that the proposed generator can generate S-boxes with good cryptographic properties than the methods in [21, 23, 24, 43, 44, 47–49, 55, 56, 58–63, 78].

Due to the usage of binary sequences, a direct application of the proposed generator is to generate plaintext-dependent S-boxes to enhance the security of the existing cryptosystems against chosen-plaintext attacks.

Chapter3

Enumerating Resonant Discrete Rossby Wave Triads and Their Application in Information Security

3.1 Introduction

In this chapter, a new parametrization of the resonant discrete Rossby wave triads to design an algorithm for the generation of all triads in a given grid is proposed. To achieve this parametrization, we used techniques from arithmetic/algebraic geometry to project resonant triads on a certain class of conics. Further, we extend the algorithm to generate quasi-resonant triads. The new algorithm has a cubic time complexity and generates all triads in low computation time as compared to the existing methods. Furthermore, we expand the new technique to construct quasi-resonant triads. Further, this chapter is organized as follows: Sec. 3.2 contains of background literature. Sec. 3.3 discusses the new parametrization. Sec. 3.4 contains the proposed ordering, S-box generator, and their detailed analyses. Sec. 3.5 presents the conclusions.

3.2 Background and Motivation

Nonlinear wave resonances play an important role in a variety of systems, ranging from fusion reactors to weather prediction to water waves. The review by Horton and Hasegawa [79] presents, with historical and scientific perspectives, the analogy between Rossby waves (pertaining to atmospheric and oceanic physics) and drift waves (pertaining to plasma physics). For simplicity, we will restrict our discussion to the atmospheric side. Resonant and quasi-resonant Rossby/-drift wave triads play a vital role in atmospheric dynamics [80, 81]. Petoukhov et al. [82] linked

Rossby waves with several extreme weather phenomena via quasi-resonances. Coumou et al. [83] explored a connection between Rossby waves and global warming. The enumeration of all resonant Rossby wave triads is a practical problem and recently has gained great importance. In [84, 85], generic algorithms were used to enumerate resonant Rossby wave triads in a given box. Bustamante and Hayat used elliptic surfaces to classify resonant and quasi-resonant triads [27]. Kopp [28] used methods from projective geometry to obtain almost all resonant triads in a box of size 5000.

A Rossby wave [82, 83, 86–93] is defined in this chapter as a parametrized solution of the linearized form of the CHME [27], a partial differential equation that expresses the conservation of potential vorticity [94]. In other words, a Rossby wave represents a plane-wave solution of the linearized version of CHME. There are many such solutions to the linearized CHME, and these are important for the study of resonances. Mathematically, in the context of the CHME with periodic boundary conditions, a Rossby/drift wave is determined by a wavevector $\mathbf{k} \in \mathbb{Z}^2$. The first component of this vector represents the number of peaks of the wave along the zonal direction, and the second component represents the number of peaks along the meridional direction. In a special case, the nonlinear interaction of two waves of different wavevectors enumerates a third wave with a new wavevector, thus forming a 'triad' of wavevectors such that the interaction of any two waves in the triad produces the third one. In such a case, the nonlinear interaction is limited to only three waves under the constraint that no further waves may be produced. Such a triple of waves is known as a resonant triad.

The equations determining a resonant triad in CHME are Diophantine equations, namely, equations for integers. Solutions to these equations are of importance in reduced models of atmosphere and plasmas, as they give the wavevectors involved in three-wave interactions. The enumeration of all resonant triads is a practical problem. Over several centuries, classical methods were developed by Fermat, Euler, Lagrange, and Minkowski to classify solutions to some

Diophantine equations [95], but the CHME resonant triads lead to new Diophantine equations whose analytical and computational enumeration problem has received great attention. Bustamante and Hayat [27] proposed a new method to enumerate numerically all resonant triads. The newly developed method relies on the transformation of the wavevectors to a new set of variables, converting the Diophantine equations for CHME resonant triads into a simpler set of equations, solvable by Fermat's Xmas theorem. They extended the algorithm to include the enumeration of quasi-resonant triads, and the extended method was found practical. Kopp [28] provided a new parametrization to the resonant triads and found almost all the resonant triads in a box of wavenumber size 5000 using his parametrization. Subsequently, Hayat et al. [96] explicitly parametrized the resonant wavevectors by two rational parameters and proposed a new method of cubic complexity to enumerate all irreducible triads in a specific region.

Transfer of useful information via internet is usual in today's modern life. In cryptography, an S-box is a nonlinear vectorial Boolean function with m input and n output bits. Here, m and n are two positive integers. For data encryption, mainly two techniques are used: block cipher and stream cipher [97]. A block cipher encrypts data in blocks of fixed length, whereas in a stream cipher one bit is encrypted in one go. In a block cipher, an S-box is used as a fundamental nonlinear part to achieve a higher level of confusion in the data [98].

Using various mathematical methods, many S-box generators have been developed in recent years [24, 33, 67] for possible applications in image encryption algorithms using S-boxes as nonlinear components [22, 61]. Cryptographers are widely using chaotic maps to develop new algorithms for S-box generation [22, 43, 44, 47–49, 58, 62]. Optimization techniques are adopted to design highly nonlinear S-boxes in [50, 60, 63]. Other domains of mathematics, such as graph theory [56], cubic polynomial mappings [55], and linear trigonometric transformations [99], are also used to construct S-boxes. In [100], a new S-box construction algorithm is developed using chaotic maps, symmetric groups, and Mobius transformations to generate a highly secure S-box.

Another important structure is the EC to construct secure S-boxes [14, 23, 25]. In [41], Azam et al. proposed an efficient S-box generator over ECs to construct dynamic S-boxes. Saleh and Abbas [74] designed an S-box generator using the points over an EC to construct highly secure and key-dependent S-boxes. Ullah et al. [72] designed efficient S-boxes and pseudo-random-number generator (PRNG) over ECs. Hayat et al. [24] presented an S-box generation technique using an EC. The designed S-boxes are further evaluated for the encryption of images. Recently, Murtaza et al. [101] introduced a dynamic S-box generator using ECs and binary sequences to design robust and dynamic S-boxes for block ciphers.

3.3 New Parametrization of the Elliptic Surface

The discussion in this section is based on [27] and references therein. The following equation of an elliptic surface was derived in [27] for resonant Rossby wave triads, namely those triads satisfying Eq. 1.26:

$$f^2y^2 = x^3 - 2dx^2 + 2dx - d^2, \tag{3.1}$$

where x, y and d are defined in Eq. 1.35 and f is the aspect ratio of the system. We fix the variable x to equal some constant ξ. It is direct to see that the surface in Eq. 3.1 can be re-written as the following equation of an ellipse:

$$\frac{f^2y^2}{r^2} + \frac{(d+a)^2}{r^2} = 1, \tag{3.2}$$

where $a = \xi^2 - \xi$ and $r = \sqrt{\xi^4 - \xi^3 + \xi^2}$. Now transform the ellipse in Eq. 3.2 into a polar form by substituting

$$y = \frac{r}{f}\cos\theta, \quad d + a = r\sin\theta \tag{3.3}$$

for a parameter $\theta \in [0, 2\pi)$. In summary, we have

$$x = \xi, \quad y = \frac{\sqrt{\xi^4 - \xi^3 + \xi^2}}{f} \cos\theta, \quad d = \sqrt{\xi^4 - \xi^3 + \xi^2} \sin\theta - \xi^2 + \xi. \tag{3.4}$$

From Eqs. 3.3 and 3.4, the inverse relation is obtained as

$$\xi = x, \quad \theta = \tan^{-1}\left(\frac{d + x^2 - x}{fy}\right). \tag{3.5}$$

Now using the transformation in Eq. 3.4 and the fundamental identity $\cos^2\theta + \sin^2\theta = 1$ we get

$$f^2y^2 + d^2 = 2\xi^4 - 3\xi^3 + 2\xi^2 - 2(\xi^2 - \xi)\sqrt{\xi^4 - \xi^3 + \xi^2} \sin\theta. \tag{3.6}$$

Thus, using Eqs. 1.34 and 3.4, we have the following three equations defining the triad wavenumbers in terms of the new parameters ξ and θ:

$$\frac{k_1}{k_3} = \frac{1}{2\xi^3 - 3\xi^2 + 2\xi - 2(\xi - 1)\sqrt{\xi^4 - \xi^3 + \xi^2} \sin\theta}, \tag{3.7}$$

$$\frac{\ell_1}{k_3} = \frac{\xi}{f\sqrt{\xi^4 - \xi^3 + \xi^2} \cos\theta}\left(1 - \frac{\sqrt{\xi^4 - \xi^3 + \xi^2} \sin\theta - \xi^2 + \xi}{2\xi^4 - 3\xi^3 + 2\xi^2 - 2(\xi^2 - \xi)\sqrt{\xi^4 - \xi^3 + \xi^2} \sin\theta}\right), \tag{3.8}$$

$$\frac{\ell_3}{k_3} = \frac{\sqrt{\xi^4 - \xi^3 + \xi^2} \sin\theta - \xi^2 + \xi - 1}{f\sqrt{\xi^4 - \xi^3 + \xi^2} \cos\theta}. \tag{3.9}$$

Now from Eqs. 1.35 and 3.5, ξ and θ can be written as

$$\xi = \frac{k_3(k_1^2 + f^2\ell_1^2)}{k_1(k_3^2 + f^2\ell_3^2)}, \tag{3.10}$$

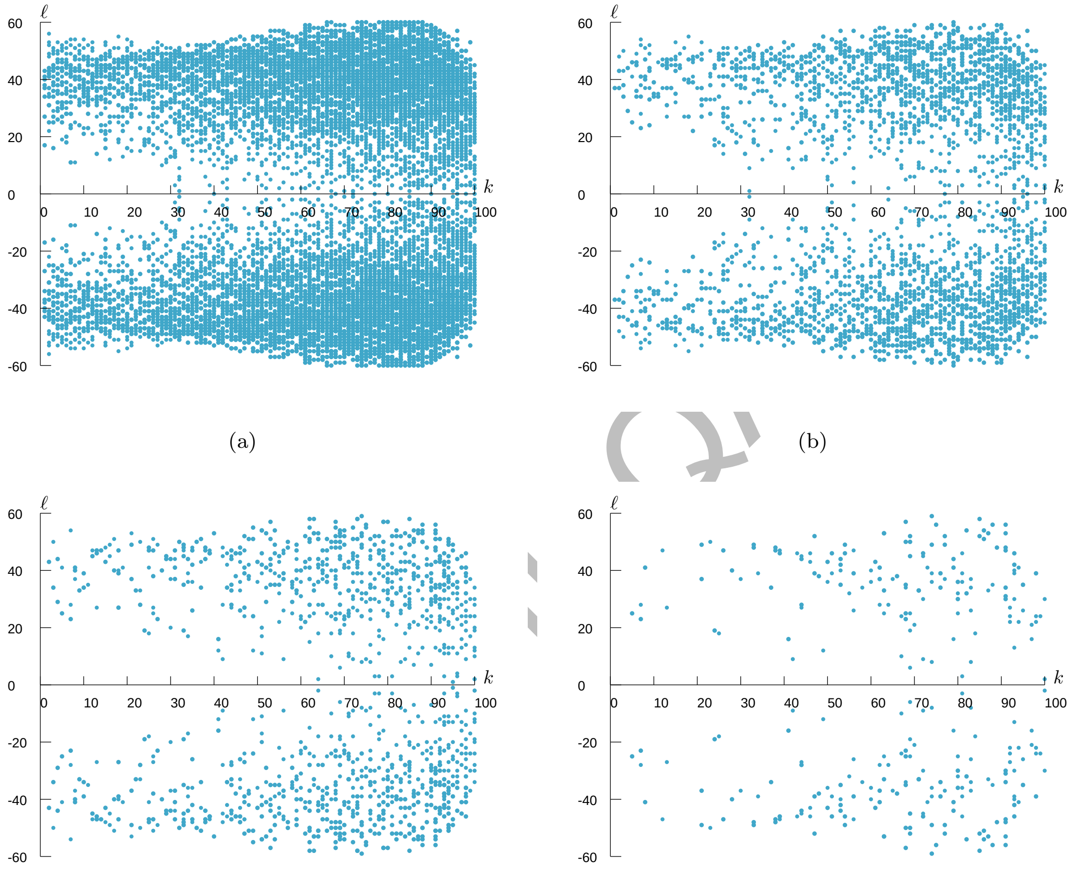
$$\theta = \tan^{-1}\left[\left((k_3 - k_1)k_1 + (\ell_3 - \ell_1)f^2\ell_1 + \left(\frac{k_3(k_1^2+f^2\ell_1^2)^2}{k_1(k_3^2+f^2\ell_3^2)}\right)\right)\left(\frac{1}{f(k_3\ell_1 - k_1\ell_3)}\right)\right]. \tag{3.11}$$

Hence Eqs. 3.7–3.9 and theirs inverse in Eqs. 3.10–3.11 represent the explicit parametrization of the resonant triad in terms of the parameters ξ and θ. Using this new parametrization, we generate all irreducible resonant triads in a specific box such that $|k|, |\ell| \leq \mathrm{L}$. For this, choose two rational numbers a', b' and an integer e to design a set of rational numbers ${}_{a'}\mathrm{A}_{b'}$ of size n with end points a', b' such that $a_1 = a', a_n = b'$ and $a_i = a_{i-1} + h$ for $h = \frac{b'-a'}{e}$ and $1 < i < n$. Further, select a subset ${}_{c'}\mathrm{B}_{d'} \subseteq [0°, 360°]$ such that $d' = c' + tg$ for some integer t and a rational number g. Now the step by step technique for the enumeration of resonant triads is given in Algorithm 1. We borrow an idea from physical considerations: to fully understand the dynamics of a system of Rossby waves, it is necessary to understand the behavior of quasi-resonant triads. Therefore, to further investigate the newly designed parametrization, we generate quasi-resonant triads using Eq. 1.26, where in Eq. 1.26, $\omega_2 = \omega_3 - \omega_1$ is replaced by the condition in Eq. 1.27. In practice, we apply the same parametrization given in Eqs 3.7–3.9 but we approximate the output wavenumbers to numbers within a box of size L. This leads to a mismatch in the frequency resonance condition. The value of the newly introduced "detuning parameter" δ from Eq. 1.27 determines the number of quasi-resonant triads obtained. Our proposed parametrization directly computes the quasi-resonant triads. For this purpose we

Algorithm 1 Enumeration of resonant triads.

```
Require: Two sets ${}_{a'}\mathrm{A}_{b'}, {}_{c'}\mathrm{B}_{d'}$, a box size L and an aspect ratio $f$.
Ensure: A set T of resonant triads.
    /* A set T of resonant triads and initialized as an empty set. Furthermore, $\triangle$
    represents an arbitrary triad and $k_1', \ell_3', \ell_1'$ are the right hand sides of Eqs. 3.7–3.9, re-
    spectively. Moreover, $\lfloor \cdot \rceil$ is an integer function.
    $\mathrm{T} := \emptyset$;
    for $\xi \in {}_{a'}\mathrm{A}_{b'}$ do
      for $\theta \in {}_{c'}\mathrm{B}_{d'}$ do
        Compute $k_1'$, $\ell_3'$ and $\ell_1'$ for $\xi, \theta$ by using Eqs. 3.7–3.9.
        for $k_3 \in [1, \mathrm{L}]$ do
          $k_1 = \lfloor k_1' \cdot k_3 \rceil, \ell_3 = \lfloor \ell_3' \cdot k_3 \rceil$ and $\ell_1 = \lfloor \ell_1' \cdot k_3 \rceil$;
          $k_2 = k_3 - k_1, \ell_2 = \ell_3 - \ell_1$ and $\omega_i = k_i/(k_i^2 + f^2\ell_i^2)$ for $i = 1, 2, 3$;
          if $\omega_3 = \omega_1 + \omega_2$ and $|k_i|, |\ell_i| < \mathrm{L}$ for $i = 1, 2, 3$; then
            $\mathrm{T} = \mathrm{T} \cup \{\triangle\}$;
          end if
        end for
      end for
    end for
    Output T as a set of triads.
```

select the parameters $f = 1,\ a' = 1.025,\ b' = 1.204,\ c' = 0,\ d' = 360,\ e = 716,\ g = 0.0125,\ \mathrm{L} = 100$ and several choices of δ: $2\times10^{-7}, 4\times10^{-7}, 8\times10^{-7}, 2\times10^{-6}, 4\times10^{-6}, 8\times10^{-6}, 2\times10^{-5}$. The wavevectors of the quasi-resonant triads generated for the chosen parameters are illustrated in Fig. 3.1. A bar graph analysis for all the computed unique and irreducible quasi-resonant triads



(a)

(b)

(c)

(d)

Figure 3.1: Illustration of the wavevectors (k, ℓ) such that $|k|, |\ell| \le 100$ and $f = 1$, where (a) $\delta = 8\times10^{-6}$, (b) $\delta = 2\times10^{-6}$, (c) $\delta = 8\times10^{-7}$, and (d) $\delta = 2\times10^{-7}$.

is given in Fig. 3.2. It is evident from Figs. 3.1 and 3.2 that the new method can generating a large number of triads by relaxing the condition on the angular frequencies. In Example 3.1, we explain how the designed algorithm maps a triad on the surface point and vice versa.

Example 3.1. *To map the triad on the surface and hence on the conic, let us choose the triad* $(1, -11)$, $(8, 34)$, $(9, 23)$, *then calculate* $\xi = \frac{9}{5}$ *and augmented angle* $\theta = 180^\circ + \frac{1807}{36}^\circ$ *by Eqs. 3.10*

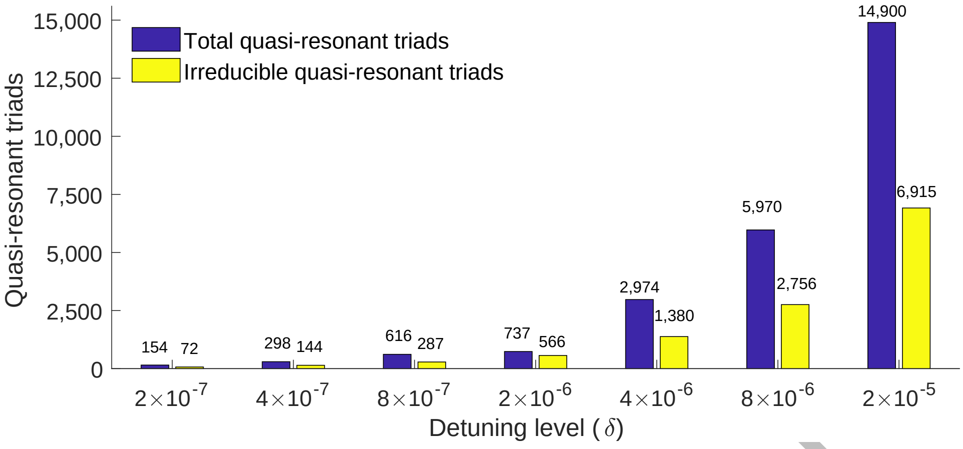


Figure 3.2: A bar graph of quasi-resonant triads in terms of detuning levels, for aspect ratio $f = 1$ and box size L = 100.

and 3.11, respectively. For $\xi = \frac{9}{5}$ *we have* $a = \frac{36}{25}$ *and* $r = \frac{\sqrt{4941}}{25}$*. Hence from Eq. 3.4 it follows that* $x = \frac{9}{5}, y = -\frac{9}{5}$ *and* $d = -\frac{18}{5}$*. Now to map the surface point back to triad, take the point* $(x, y, d) = (\frac{9}{5}, -\frac{9}{5}, -\frac{18}{5})$ *on the elliptic surface, and by Eq.3.5 we have* $\xi = \frac{9}{5}$ *and augmented angle* $\theta = 180^\circ + \frac{1807}{36}^\circ$*. So that Eq. 1.31 gives that* $\frac{k_1}{k_3} = \frac{1}{9}, \frac{\ell_1}{k_3} = -\frac{11}{9}, \frac{\ell_3}{k_3} = -\frac{23}{9}$*. From this we can write that* $k_3 = 9, k_1 = 1, \ell_1 = -11$ *and* $\ell_3 = 23$*.Using these values we compute* $k_2 = k_3 - k_1 = 8, \ell_2 = \ell_3 - \ell_1 = 34$*. Hence we find the same triad* $(1, -11)$*,* $(8, 34)$*,* $(9, 23)$*.*

It was verified in [96] using a brute-force method that there is a total of 472 irreducible resonant triads in a grid of size 5000. Previously, Kopp [28] introduced a fast parametrization to enumerate irreducible triads. Using this parametrization and a simple search, he could enumerate only 463 out of the 472 irreducible resonant triads in a grid of size 5000. For this reason, the authors in [96] developed another parametrization. This parametrization could enumerate all 472 irreducible resonant triads in a grid of size 5000, but it took 10.5 days to compute this on a 16-core-machine. To overcome the shortcomings of the aforementioned algorithms, in this chapter we introduced parametrization in Eqs. 3.7–3.9 to irreducible triads. A major advantage of our parametrization over the existing ones is that we can enumerate all 472 irreducible resonant triads in a given grid of size 5000 in only 2 days on a 16-core-machine. Table 3.1 indicates that, for aspect ratio $f = 1$, the new method to enumerate triads is more efficient than the existing algorithms.

Table 3.1: Number of irreducible triads for aspect ratio $f = 1$.

Methods	Time	No. of Triads		System Specifications
		Optimal	Obtained	
Proposed	2 days	472	472	MATLAB, 16-core-machine
Ref. [96]	10.5 days	472	472	Mathematica, 16-core-machine
Ref. [28]	-	472	463	Mathematica, MacBook 2.9 GHz dual-core i7

For different values of aspect ratio f including the standard case ($f = 1$), we have computed the number of irreducible resonant triads in a grid of size 100 using a brute-force method. The results are shown in Table 3.2. Notice that the case $f = \sqrt{3}$ gives an extremely large number of resonant triads.

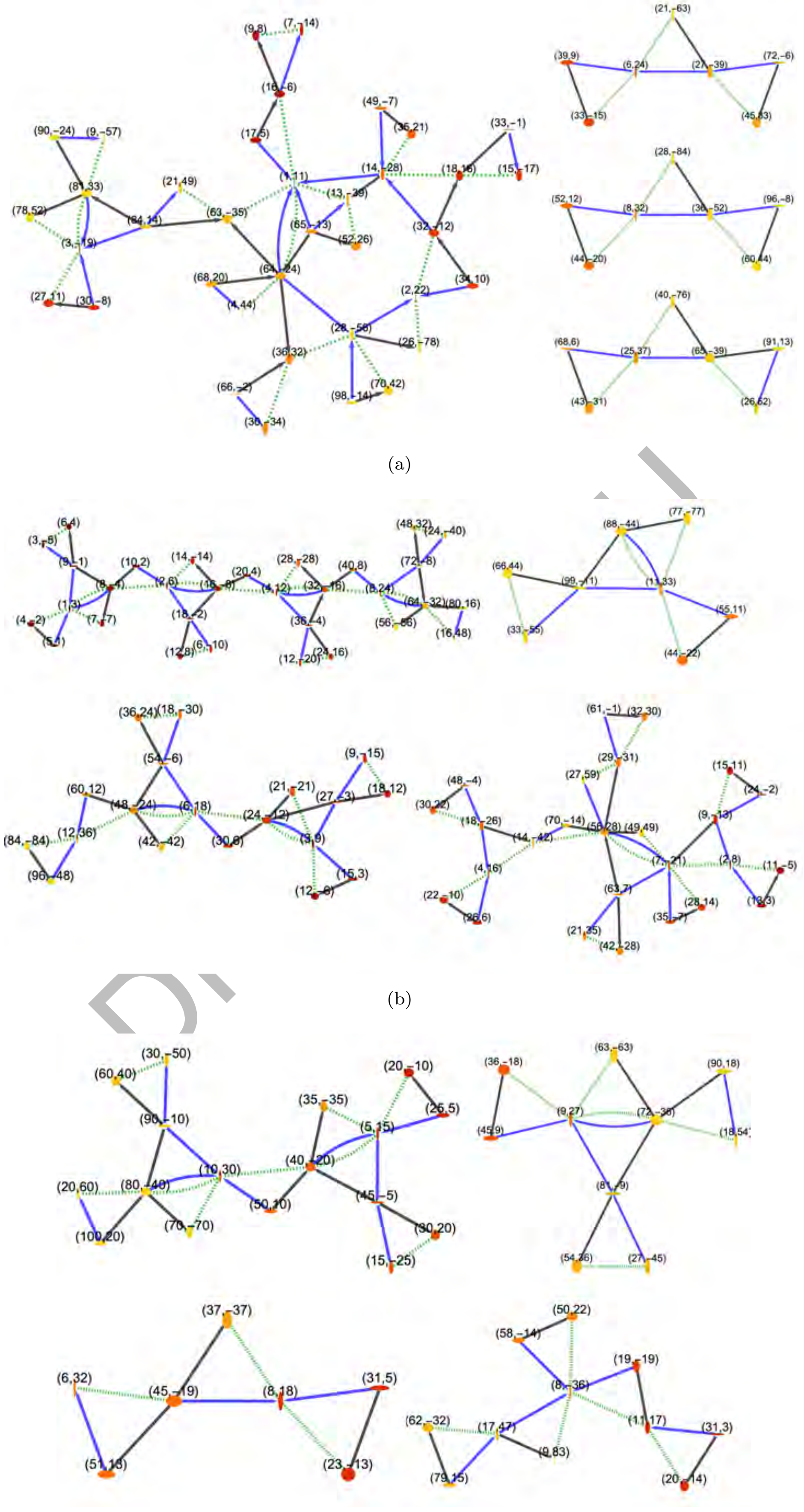
Table 3.2: Number of irreducible triads against different values of aspect ratio f.

f	1	$\frac{1}{2}$	$\sqrt{2}$	$\frac{\sqrt{2}}{3}$	$\sqrt{3}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{3}$	$\frac{\sqrt{3}}{4}$	$\frac{\sqrt{3}}{5}$	$2\sqrt{3}$	$3\sqrt{3}$	$4\sqrt{3}$	$\frac{\sqrt{6}}{2}$	$\frac{\sqrt{5}}{2}$	$\frac{\sqrt{7}}{2}$
Triads	12	7	6	3	337	231	281	65	63	97	64	30	4	3	7

The plot in Fig. 3.3 contains clusters (except for the repeated "mirrored" clusters) that occur within a box of size L = 100 and aspect ratio $f = \sqrt{3}$, with the exception of special triads with $k_3 = 0, |k_1| = |k_2|$, which we did not include because they are a kind of degenerate case from the viewpoint of the dynamical system. The isolated triads provide roughly 50% of the overall number of resonant triads, and this fact holds true for increasing box sizes. However, within a specific box size, the distribution of cluster sizes is nontrivial.

3.4 An Application in Cryptography

To get the desired security in an S-box based cryptosystem, an S-box should be capable of creating enough confusion and diffusion. Total orders play an important role in achieving the



(a)

(b)

(c)

Figure 3.3: Plot of resonant triads clusters for aspect ratio $f = \sqrt{3}$ and box size L = 100.

aforementioned purpose. For example, Azam et al. [41] defined three different orderings on the points of ECs to construct secure S-boxes. Similarly, quasi-resonant triads are ordered in [102] to develop an image encryption scheme. Motivated by [102], we develop a new ordering $\prec_\theta$ on resonant triads with respect to the new parametrization. Let $\triangle$ and $\triangle'$ be two arbitrary triads (*e.g.*, $\triangle = (k_i, \ell_i)$ and $\triangle' = (k'_i, \ell'_i)$ for $i = 1, 2, 3$), then the ordering $\prec_\theta$ is defined by

$$\triangle \prec_\theta \triangle' \Leftrightarrow \begin{cases} \text{either } \xi < \xi', \text{ or} \\ \\ \xi = \xi' \text{ and } \theta \leq \theta'. \end{cases}$$

We prove that $\prec_\theta$ is a total order.

Lemma 3.1. *If* T *represents the set of irreducible resonant triads, then* $\prec_\theta$ *is a total order on the set* T.

Proof. We need to show that $\prec_\theta$ is reflexive, antisymmetric, and transitive. The reflexive property follows from the fact $\xi = \xi$ and $\theta = \theta$ always.

Now, suppose that $\triangle \prec_\theta \triangle'$ and $\triangle' \prec_\theta \triangle$ then $\xi < \xi'$ and $\xi' < \xi$. Therefore, it can be concluded that $\theta \leq \theta'$ and $\theta' \leq \theta$. Hence $\theta = \theta'$. Consequently, from Eqs. 3.7–3.9, it follows that

$$\frac{k_1}{k_3} = \frac{k'_1}{k'_3}, \quad \frac{\ell_3}{k_3} = \frac{\ell'_3}{k'_3}, \quad \frac{\ell_1}{k_3} = \frac{\ell'_1}{k'_3}, \tag{3.12}$$

which is possible if $k_1 = c_1 k'_1, k_3 = c_1 k'_3, \ell_3 = c_2 \ell'_3, k_3 = c_2 k'_3$, and $\ell_1 = c_3 \ell'_1, k_3 = c_3 k'_3$ for some integers c_1, c_2, and c_3, but $k_3 = c_1 k'_3 = c_2 k'_3 = c_3 k'_3$ implies that $c_1 = c_2 = c_3 = c$ for some integer c. Moreover, $\ell_1 = c\ell'_1$ and $\ell_3 = c\ell'_3$. From Eq. 1.26, it is evident that $k_2 = k_3 - k_1$ and $\ell_2 = \ell_3 - \ell_1$, and it follows that $k_2 = ck'_2$ and $\ell_2 = c\ell'_2$. That is, we have $(k_i, \ell_i) = (ck'_i, c\ell'_i)$ for $i = 1, 2, 3$. Thus $\triangle = c\triangle'$, but all triads in T are irreducible. Thus $c = 1$, and $\prec_\theta$ is antisymmetric.

For transitivity, let $\triangle \prec_\theta \triangle'$ and $\triangle' \prec_\theta \triangle''$. Then one possibility is $\xi < \xi'$ and $\xi' \leq \xi''$, which implies that $\xi < \xi''$. The second possibility is $\xi = \xi'$ and $\xi' < \xi''$, which also implies that $\xi < \xi''$. The last case is $\xi = \xi' = \xi''$, which gives that $\theta \leq \theta' \leq \theta''$ and hence $\theta \leq \theta''$. Thus, in all possible cases, $\triangle \prec_\theta \triangle''$. Consequently, $\prec_\theta$ is transitive and hence a total order. □

Based on the ordering $\prec_\theta$, an S-box generator is introduced in the following section.

3.4.1 The New S-box Generator

Suppose we want to construct an S-box over the set $[0, 2^m - 1]$ for some positive integer m. Consider the following steps:

Step 1. Choose L as a grid size and generate two sets ${}_{a'}\mathrm{A}_{b'}$ and ${}_{c'}\mathrm{B}_{d'}$ as required by Algorithm 1 with the constraint that the parameters a', b', c', d', e, g, and t such that we can enumerate exactly $u = \lceil \frac{2^m}{6} \rceil$ triads.

Step 2. After enumerating all u triads, take the absolute of all wavevectors. Then, arrange the triads using the ordering $\prec_\theta$ to obtain a matrix $\mathrm{M}_{u\times 6}$, where the ith row represents the ith triad.

Step 3. Select an integer $\ell \leq |\Lambda_{\mathrm{M}}|$ to apply the modulo ℓ operator on the matrix M in order to obtain the matrix M_ℓ. Here, Λ_{M} denotes the greatest value in M.

Step 4. Neglect $\mathrm{M}_\ell(i)$ for $i > 2^m$, and define a mapping $\phi : [0, 2^m - 1] \to [0, 2^m - 1]$ such that $\phi(n) = i - 1$, where i represents the index of the nth least value of M_ℓ in linear ordering.

It is noted that if $\mathrm{M}_\ell(i)$ is the rth least value and $\mathrm{M}_\ell(i) = \mathrm{M}_\ell(j)$ for $i < j$ then $\mathrm{M}_\ell(j)$ is considered to be the $(r + 1)$th least value of M_ℓ. For parameters $a' = 1.0667, b' = 1.1248, c' = 14.92, d' = 254.70, e = 581, g = 23{,}978, f = 1, \mathrm{L} = 5000$, and $\ell = 859$, the S-box constructed is prescribed in Table 3.3. Our algorithm is important in the sense that an S-box is guaranteed for each value of ℓ. Consequently, the total number of S-boxes for the chosen parameters is equal to the number of values of ℓ. The time complexity of the generator to construct an S-box is given by the following:

Lemma 3.2. *Suppose that all the inputs* L, a', b', c', d', e, g, f, m, *and integer* $\ell \leq |\Lambda_{\mathrm{M}}|$ *are known and the size of the set* ${}_{c'}\mathrm{B}_{d'}$ *is* η. *The time complexity for the proposed S-box is* $\mathcal{O}\big(\max(n\eta\mathrm{L}, 2^{2m})\big)$.

Proof. For given inputs the enumeration of u triads the S-box generator needs $\mathcal{O}(n\eta\mathrm{L})$ time. The arrangements of triads according to the ordering $\prec_\theta$ takes $\mathcal{O}\big((u)\log(u)\big)$ time. The time taken by the execution of nested two loops is $\mathcal{O}(2^{2m})$. As $2^{2m} > (u)\log(u)$, therefore, the time complexity of the proposed S-box algorithm is $\mathcal{O}\big(\max\{n\eta\mathrm{L}, 2^{2m}\}\big)$. □

3.4.2 Security Analysis

The analysis of an S-box obtained by the new technique is given as follows.

Table 3.3: The proposed S-box.

2	73	41	58	10	65	118	55	83	166	249	127	103	185	115	150
206	186	149	145	134	221	174	81	167	111	244	57	49	66	239	232
80	200	173	1	156	50	87	177	162	29	155	88	52	72	252	141
209	230	190	95	184	241	236	9	181	168	60	46	94	117	189	163
5	98	86	14	212	169	38	159	89	172	48	223	121	53	135	119
248	225	13	45	122	74	91	67	215	197	120	35	30	148	201	139
128	56	131	176	147	140	32	219	105	22	234	183	179	21	195	203
154	62	54	204	17	15	198	47	213	24	64	79	78	70	247	37
211	40	157	6	224	133	61	59	229	69	129	144	82	96	27	237
170	188	158	152	110	160	161	93	12	28	126	142	137	23	97	39
26	218	116	194	71	242	202	0	63	132	151	222	243	199	34	77
92	44	164	208	114	255	20	178	231	193	84	124	238	113	85	143
182	104	101	180	235	18	123	192	251	138	130	109	107	100	108	19
253	207	3	187	11	146	25	4	254	16	42	125	226	99	31	153
8	233	102	220	217	210	165	214	196	227	245	175	246	43	228	112
68	205	216	106	7	171	191	240	51	33	136	90	250	36	76	75

3.4.2.1 Linear Attacks

Nonlinearity (NL): The NLs of all component functions of the S-box are given in Fig. 3.4. The minimum NL is 106, which is sufficient to resist powerful linear attacks.

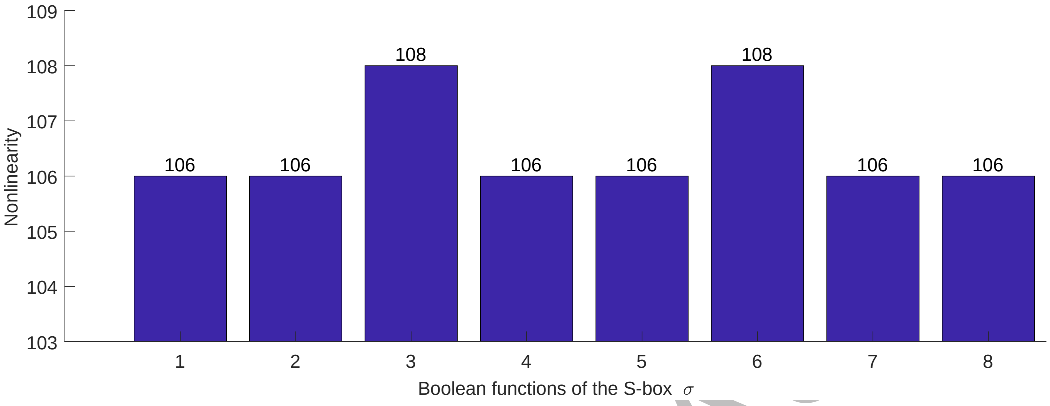


Figure 3.4: The NLs of each Boolean function of the newly designed S-box σ.

Algebraic complexity (AC): An S-box can have a maximum AC value of 255. For the proposed S-box, the AC is 254. This shows that the proposed S-box is very strong against algebraic attacks. In Table 3.4, the coefficients of the linear polynomial are shown.

Table 3.4: The coefficients of the linear polynomial for the S-box.

0	6	42	166	207	41	47	29	248	71	175	56	89	156	93	154
241	241	180	24	18	184	167	158	9	166	24	197	4	169	90	189
85	126	114	51	130	26	66	202	86	218	16	69	73	203	194	155
151	217	132	51	41	126	162	153	33	144	181	249	81	247	218	68
130	220	111	70	184	240	112	99	115	56	247	188	245	20	172	57
207	30	209	22	68	44	140	7	24	94	151	135	247	161	0	229
119	52	6	99	233	126	100	52	52	121	123	192	172	4	131	57
251	18	119	193	98	39	51	202	26	195	123	230	134	251	137	70
191	209	105	173	146	39	35	233	72	32	90	58	177	194	177	88
139	47	119	82	101	46	234	180	18	60	18	75	16	28	226	155
38	217	248	78	108	197	36	225	149	233	144	153	123	83	30	152
222	94	17	211	25	54	20	83	217	143	121	44	45	248	236	244
130	76	104	161	8	106	117	57	93	46	247	29	47	131	26	210
132	73	200	219	154	199	19	94	109	110	142	220	90	199	31	18
116	212	147	189	50	252	88	163	116	137	213	98	125	31	74	49
82	227	83	131	238	5	60	170	176	245	221	174	180	51	154	2

Linear approximation probability (LAP): The proposed S-box has the LAP value 0.133, which shows that the proposed S-box is secure against linear attacks.

3.4.2.2 Differential Attacks

Differential approximation probability (DAP): The DAP of the S-box is 0.039, hence the S-box has high security against approximation attacks.

3.4.2.3 Analysis of Boolean Functions

Strict avalanche criteria (SAC): If the calculated value is closer to 0.5, then it means that an S-box fulfills the SAC criterion. The dependence matrix of the SAC is shown in Table 3.5, where the minimum and maximum of the SAC is 0.422 and 0.578, respectively.

Table 3.5: Dependence matrix with entries m_{ij}.

0.424	0.563	0.578	0.531	0.500	0.516	0.500	0.531
0.484	0.578	0.469	0.469	0.500	0.5	0.516	0.453
0.422	0.547	0.500	0.484	0.563	0.547	0.516	0.422
0.469	0.547	0.500	0.563	0.500	0.469	0.500	0.516
0.500	0.547	0.500	0.531	0.563	0.484	0.500	0.500
0.500	0.531	0.516	0.531	0.578	0.547	0.578	0.531
0.563	0.500	0.438	0.500	0.547	0.469	0.469	0.500
0.516	0.453	0.531	0.453	0.453	0.563	0.531	0.531

Bit independence criterion (BIC): The requirement of the BIC analysis is that all values of d_{ij} should be approximately equal to 0.5. It can be observed that each d_{ij} ranges between 0.473 and 0.531. This means that the S-box satisfies the BIC criterion. Furthermore, Table 3.6 indicates that the values of each element d_{ij} of the correlation matrix of $x_i \oplus x_j$ for all input $x_i \in \mathrm{GF}(2^8)$ $(i \neq j$, where $i, j = 1, 2, \ldots, 8)$ of the given S-box are all close to 0.5, which shows that the S-box meets the BIC. A bar chart of BIC-NL for the proposed S-box is shown in Fig. 3.5, which reveals that the newly designed S-box satisfies the BIC criterion.

3.4.3 Alteration in S-boxes

To have a sufficient cryptographic strength, the S-box construction technique should be capable of constructing a number of variant S-boxes [41]. This is because many cryptosystems require more than one secure S-box. We took a fixed set of u triads enumerated by the parameters noted

Table 3.6: Dependence matrix with entries d_{ij}.

—	0.488	0.484	0.488	0.477	0.525	0.531	0.523
0.488	—	0.5	0.490	0.508	0.504	0.516	0.473
0.484	0.5	—	0.486	0.522	0.506	0.510	0.488
0.488	0.490	0.486	—	0.512	0.504	0.494	0.498
0.477	0.508	0.522	0.512	—	0.498	0.522	0.5
0.525	0.504	0.506	0.504	0.498	—	0.486	0.527
0.531	0.516	0.510	0.494	0.522	0.486	—	0.514
0.523	0.473	0.488	0.498	0.5	0.527	0.514	—

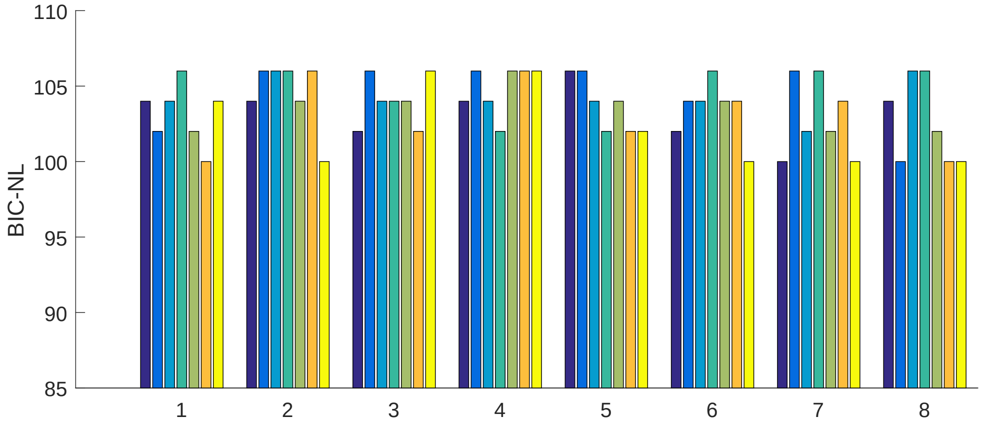


Figure 3.5: A bar chart of the BIC-NL of the S-box σ. The minimum BIC-NL of each component is 100 and maximum is 106 as it is shown in given bar chart.

in Sec. 3.4.1. Since for each value of ℓ, an S-box is guaranteed, we picked all the corresponding S-boxes for 1319 randomly chosen values of ℓ and computed the NL of each S-box. We found that the total number of distinct S-boxes is 1256, which is 95% of all the S-boxes. Further, the distribution of values of ℓ is shown in Fig. 3.6(a), and the behavior of NL of the constructed S-boxes is illustrated by Figure 3.6(b). More explicitly, the j-th value in Fig. 3.6(b) represents the minimum NL of the S-box generated by the randomly chosen jth value of the parameter ℓ in Fig. 3.6(a). The fluctuation in the NL values is evident from Fig. 3.6(b), which implies a variation in the associated S-boxes.

Furthermore, Fig. 3.6(b) shows that the minimum NL for most of the S-boxes oscillates between 90 and 100. However, there exists a large number of S-boxes with NL greater than 100. Thus, the above arguments explain that the proposed method is not only capable of constructing a number of distinct S- boxes but also has the capability to construct highly nonlinear S-boxes.

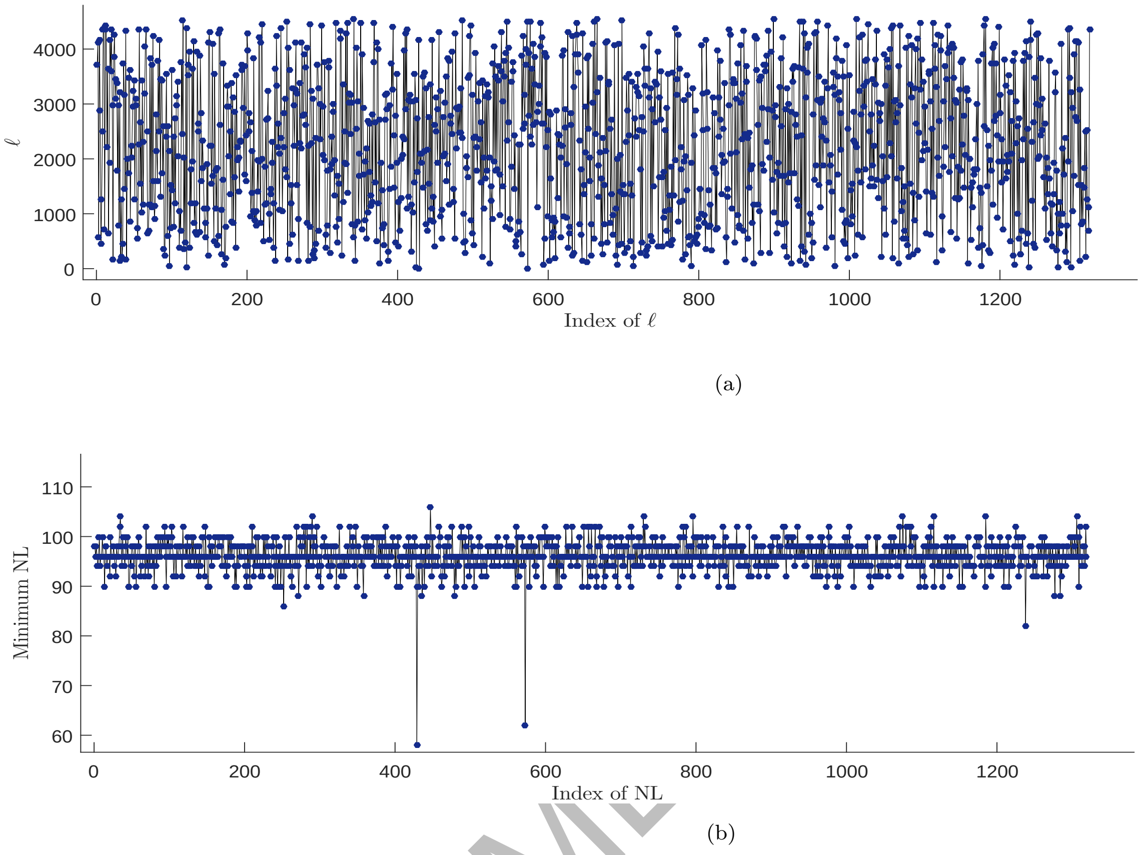


Figure 3.6: Illustration of distinct S-boxes (a) The distribution of parameter ℓ (b) The Oscillation of the minimum nonlinearity of the corresponding S-boxes.

3.4.4 Performance Comparison

We compare our S-box with the S-boxes constructed by different methods, including different chaotic maps and ECs [14, 23–25, 41, 43, 44, 47–50, 55, 56, 58–63, 78]. The NL comparison in Table 3.7 shows the S-box is better interm of NL than the S-boxes in [23, 43, 44, 47–50, 55, 56, 58–63, 78] and, consequently, it has better security than the S-box with lower NL. Our S-box has lower LAP than the S-boxes in [14, 23–25, 41, 48, 55, 56, 60]. Similarly, the DAP results of our S-box are better than the S-boxes in [24, 43, 49, 55, 56, 58, 78]. Hence, our S-box is more secure against approximation attacks. The SAC of the S-box is between 0.421 and 0.578, which is close to 0.5. The AC of the proposed S-box is higher than the AC of S-boxes in [25, 55] and comparable with other schemes. There are two methods that outperform our method in more than four indicators: Ref. [50], which outperforms our method in four indicators, while in two indicators (AC and max BIC), it outperforms our method by a very small amount. Ref. [100] outperforms our method in all indicators except algebraic complexity.

Table 3.7: Comparison of S-boxes across a range of methods (rows) against our method based on triads. The comparison is made over nine different indicators (columns). For each indicator (column), we present in boldface the methods that outperform our method.

S-boxes	**Method**				**SAC**		**BIC**			**DAP**
		NL	**AC**	**LAP**	**Min**	**Max**	**Min**	**Max**	**NL**	
Ref. [47]	Chaos	100	**255**	**0.129**	0.422	0.594	**0.477**	**0.525**	98	0.039
Ref. [43]		96	254	**0.023**	0.391	0.625	**0.477**	0.531	92	0.050
Ref. [44]		103	**255**	**0.132**	0.398	**0.570**	0.472	0.535	96	0.039
Ref. [48]		100	**255**	0.152	0.391	0.586	0.468	0.537	100	0.039
Ref. [58]		98	254	0.133	0.422	0.609	**0.477**	0.535	94	0.054
Ref. [61]		104	**255**	0.133	0.359	0.609	0.457	0.535	96	0.039
Ref. [62]		104	254	0.133	**0.438**	0.641	**0.475**	0.547	98	0.039
Ref. [49]		96	**255**	**0.125**	0.422	0.609	0.471	0.547	96	0.047
Ref. [100]		**112**	253	**0.063**	**0.453**	**0.563**	**0.480**	**0.525**	**112**	**0.016**
Ref. [55]	Other	104	253	0.141	0.406	0.594	0.461	**0.522**	98	0.054
Ref. [63]		102	254	0.133	0.359	**0.562**	0.467	0.535	98	0.039
Ref. [60]		102	254	0.148	0.375	0.609	0.470	**0.521**	100	0.039
Ref. [78]		104	**255**	**0.109**	0.391	0.593	0.454	**0.49**	**102**	0.047
Ref. [50]		104	**255**	**0.094**	0.391	0.578	**0.476**	**0.529**	**103**	**0.023**
Ref. [59]		96	**255**	**0.125**	0.359	0.609	**0.477**	0.541	98	0.039
Ref. [56]		104	254	0.148	**0.438**	0.578	**0.482**	0.543	96	0.047
Ref. [41]	Elliptic Curves	106	**255**	0.148	0.406	0.641	0.471	0.537	98	0.039
Ref. [25]		106	253	0.188	0.406	0.609	0.465	**0.527**	98	0.039
Ref. [14]		106	**255**	0.148	**0.438**	0.594	0.465	0.545	98	0.039
Ref. [23]		104	**255**	0.145	0.391	0.625	0.471	0.531	98	0.039
Ref. [24]		106	**255**	0.148	0.406	0.625	0.471	0.539	96	0.047
Our Method	Triads	106	254	0.133	0.422	0.578	0.473	0.531	100	0.039

This performance comparison of our proposed scheme based on triads against already established methods reveals that the proposed scheme has the capability to design highly secure S-boxes when compared with other schemes with different underlying mathematical structures such as chaotic maps, elliptic curves, and some other algebraic structures.

3.5 Conclusions

In this chapter, we have developed a new geometric method to enumerate all distinct resonant triads in a given box of wavenumbers. Considering aspect ratio $f = 1$, we computed all triads in a specific region and observed that the new method is very efficient for obtaining all triads in the grid of wavenumber size 5000 when compared with the schemes [28, 96]. The new method can also be used to obtain quasi-resonant triads. For aspect ratios $f \neq 1$, we proved numerically that the proposed method is capable of enumerating quasi-resonant triads in any region. As an application, we have defined a new total order on the set of triads to subsequently design an S-box generator via a novel algorithm whose time complexity we prove mathematically. We show that the S-box generator is useful in cryptosystems using a number of S-boxes. Analysis of the S-box generator shows that our generator is capable of enumerating a highly secure S- box: considering nine key indicators based on NL, AC, LAP, DAP, SAC, and BIC, we compared our method based on triads against 20 state-of-the-art methods based on elliptic curves, chaos, and other algebraic methods. We outrank 18 of these methods [14, 23–25, 41, 43, 44, 48, 49, 55, 56, 58–63, 78] in the majority of the nine key indicators, and we basically draw with the methods from Refs. [47, 50].

Chapter 4

An Image Cryptosystem Over Elliptic Curves and Coupled Map Lattices

4.1 Introduction

In this chapter, an image cryptosystem for the real-time transmission of these images is presented. The proposed work aims to address the shortcomings of traditional encryption algorithms. This work is based on CML-systems and ECs. We have used the CML-systems and ECs in the proposed work due to their long period and random behavior. The encryption procedure is divided into two steps. In the first step, an EC-based PRNG is utilized to create diffusion in the plain-image. In the second step, an S-box generator based on ECs and CMLs is designed to create confusion in the diffused image. The rest of the chapter is organized as follows: In Sec. 4.2, a background of the image encryption scheme is given. A PRNG is proposed in Sec. 4.3. An S-box generator is designed in Sec. 4.4. In Sec. 4.5, an image encryption technique is introduced. The security performance of the proposed method is evaluated in Sec. 4.6. In Sec. 4.7, conclusions are drawn.

4.2 Background

Why we need image encryption: With the rapid development of multimedia and telecommunication technologies, multimedia information such as videos and digital images with highly sensitive information is generated and transmitted by many smart devices and digital applications [103, 104]. In particular, these images are very important information carriers and play a vital role in multimedia communications. In addition, many images contain very sensitive data such as national secrets or personal privacy. Furthermore, transmission through unprotected public communication networks leads to serious security threats because intruders can easily access and tempered confidential information. Thus, it becomes quite important to prevent illegal access to the contents of secret images from adversaries. To insure the security of digital images, various cipher techniques are proposed by cryptographers including image encryption [105–107], data hiding [108] and water marking [109]. Among all these methods, image encryption is a dynamic and straightforward way to protect plain-images against potential adversaries when images data are transmitted over a communication network [110]. The original digital image can only be obtained by using the encryption keys. However, digital images have big data capacity, high redundancy and strong correlation [111, 112] which reduce the performance of conventional techniques such as IDEAS, DES, 3DES and AES. In a block cipher, there are two main principles, one is diffusion and the other is confusion. In the diffusion phase, many cipher bits are influenced by a single bit of a plain-image to change the plain-image's statistical properties.

Literature review: Various encryption algorithms based on chaotic maps, algebraic systems, and elliptic curves are proposed to secure the transfer of important data by converting it into an unreadable form. We give a short review of these schemes as follows.

Chaos-based algorithms: Numerous chaos-based encryption techniques have been proposed. A four-dimensional chaotic-system in [113] is proposed with hidden attractors, which is then used to develop an image encryption technique that can resist differential and statistical attacks. Gong et al. [114] proposed a new dynamical chaotic-system to generate random numbers. The chaotic-system showed a variety of chaotic traits, such as coexistence attractors and hidden attractors. Huang and Zhou [115] developed a unique chaos-based encryption method to encrypt digital images.

Chaotic-systems used for encryption purposes are primarily divided into two classes: (i) one-dimension (1D) chaotic maps and (ii) multi-dimension (MD) chaotic maps. These chaotic maps have shown some good results but have some shortcomings that limit their real-time applications. For example, 1D chaotic maps have (1) limited or discontinuous range; (2) orbit with small period; and (3) the nonuniform data distribution [12, 116]. To address these issues, many cryptographers used MD chaotic-systems for image encryption as an alternative because they have multiple parameters and complex structures [117]. But MD chaotic-systems have high computational cost, difficulty in implementations and also they may produce a strange attractor whose dimension is fractional [118]. Spatiotemporal chaotic-systems such as the CML-system [119] also are used in cryptography. Even with the dynamical degeneration, CML-systems has a lengthy period [120]. Furthermore, the complex dynamics ensure the unpredictability of the orbits of a CML-system. And also, a chaos-based CML-system has numerous lattice sites that can create different key streams at the same time. CML-systems outperform low and high-dimensional chaotic-systems because they have more fascinating qualities, such as a wider range of parameters. It boosts the security of cipher and makes CML-systems more suitable for cryptography. In [121], a highly secure cryptosystem based on dynamically random CML-systems is developed. An encryption algorithm based on a CML-system is presented in [122] for color images. In [123], the system of non-adjacent CML-systems is developed, which offers more dynamic cryptographic properties than the logistic map or simple CML-systems, and a new CML-based image encryption technique is also suggested.

Elliptic curve-based algorithm: Miller [2] introduced the concept of ECC and presented an EC-based cryptosystem that is 20 times faster than the Diffie-Hellmans algorithm. A cryptosystem based on EC over a finite field is introduced by Koblitz in [3]. ECC-based algorithms are computationally efficient and provide greater security. For example, 256-bit ECC over a prime field and 3072-bit RSA provide the same security level [4]. In addition, ECC requires less memory to execute on digital computers [5, 6]. Such features make ECC more suitable for devices with limited resources in power and network connectivity [7]. Amara [7] analyzed that ECC has high security than RSA. Recently, Ye et al. [124] presented an encryption technique for digital images based on compressed sensing and ECs. An ECs-based S-box scheme and an encryption technique are constructed in [15]. Azam et al. [16] designed a secure EC-based image cryptosystem.

Research problems: Many encryption schemes and S-box generators based on ECs and chaotic maps are developed for image security [14, 15, 17–19, 23, 47–49, 58, 59, 124–137], but some of them are either not secure against cryptographic attacks [128, 138] or computationally inefficient [14, 15, 18, 19, 132–136]. S-boxes are widely used as a nonlinear component in block ciphers such as AES and DES [139]. Therefore it is necessary to generate dynamic S-boxes with high nonlinearity and low computation time. The S-boxes constructed by the methods in [47–49, 58, 59, 130, 131] have nonlinearity atmost 102 and S-box generators in [33, 45, 55] generate S-boxes with nonlinearity upto 104. So, there is a need to refine the algorithms to generate S-boxes with better nonlinearity. The EC-based S-box generator in [23] has singularity, which causes a delay in the encryption process. In [14], a PRNG is developed over the points of an EC, the PRNG is further used for diffusion in a plain-image, while dynamic S-boxes are for confusion. Although the scheme is highly secure but its time complexity depends on the primes of ECs which makes it computationally infeasible for transmission of digital data. A novel S-box generator and an encryption scheme are suggested in [15] using ECs. The generator and the encryption scheme have good cryptographic properties but they are not computationally efficient over a set of large primes and may not be useful for real-time encryption. ECC and genetic algorithm are combined in [18] to propose a new asymmetric key encryption scheme. The genetic algorithm and pixel-level encryption make the encryption scheme computationally costly.

4.3 A Pseudo-Random-Numbers Generation Algorithm

For a prime p' such that $p' \equiv 2 \pmod 3$, three integers $m_{\mathrm{x}}, m_{\mathrm{y}}$ and b' such that $m_{\mathrm{x}}, m_{\mathrm{y}} \in [1, p']$ and $b' \in [1, p'-1]$ and a subset A of $\mathbb{Z} \times \mathbb{Z}$. Define a mapping $\gamma_{p',b',m_{\mathrm{x}},m_{\mathrm{y}},A}$ from A to $[0, m_{\mathrm{x}}-1]$ such that for each $(a_1, a_2) \in A$ and $(a, (a_1 + a_2) \pmod{m_{\mathrm{y}}}) \in E_{p',0,b'}$ it holds that

$$\gamma_{p',b',m_{\mathrm{x}},m_{\mathrm{y}},A}(a_1, a_2) = a \pmod{m_{\mathrm{x}}}. \tag{4.1}$$

To analyze the random behavior of the sequence $\gamma_{p',b',m_{\mathrm{x}},m_{\mathrm{y}},A}$ for images of different sizes, we have fixed parameters as follow: $p' = 1048847$, $b' = 1$, $m_y = p'$ and $m_x = 256$.

Analysis of entropy and periods of pseudo-random-numbers (PRNs) generated by our method are given in Table 4.1. Different plain-images are used as a set $A \subset \mathbb{Z} \times \mathbb{Z}$ in $\gamma_{p',b',m_{\mathrm{x}},m_{\mathrm{y}},A}$ to generate PRNs. Histograms and PRNs graphs for aforementioned images are shown in Fig. 4.1. For all-white and all-black images, histograms and PRNs graphs are shown in Fig. 4.2. It is evident from Table 4.1 that the entropy of the PRNs is approximately close to the upper bound. Histograms of PRNs are also uniform for each gray image as illustrated in Figs. 4.2 and 4.1. We can observe in Table 4.1 that the proposed PRNs generator generates PRNs sequences with high periods and entropy values. So, it can create high diffusion in a plain-image and thus suitable for image encryption.

Table 4.1: Entropy and periods of PRNs generated by the proposed algorithm for different input plain-images (P).

Images (256 × 256)	All white	All black	Lena	Mandrill	Pepper	Cameraman
Upper bound of entropy	8	8	8	8	8	8
Entropy	7.9972	7.9971	7.9930	7.9932	7.9936	7.9941
Period	65536	65536	65536	65536	65536	65536

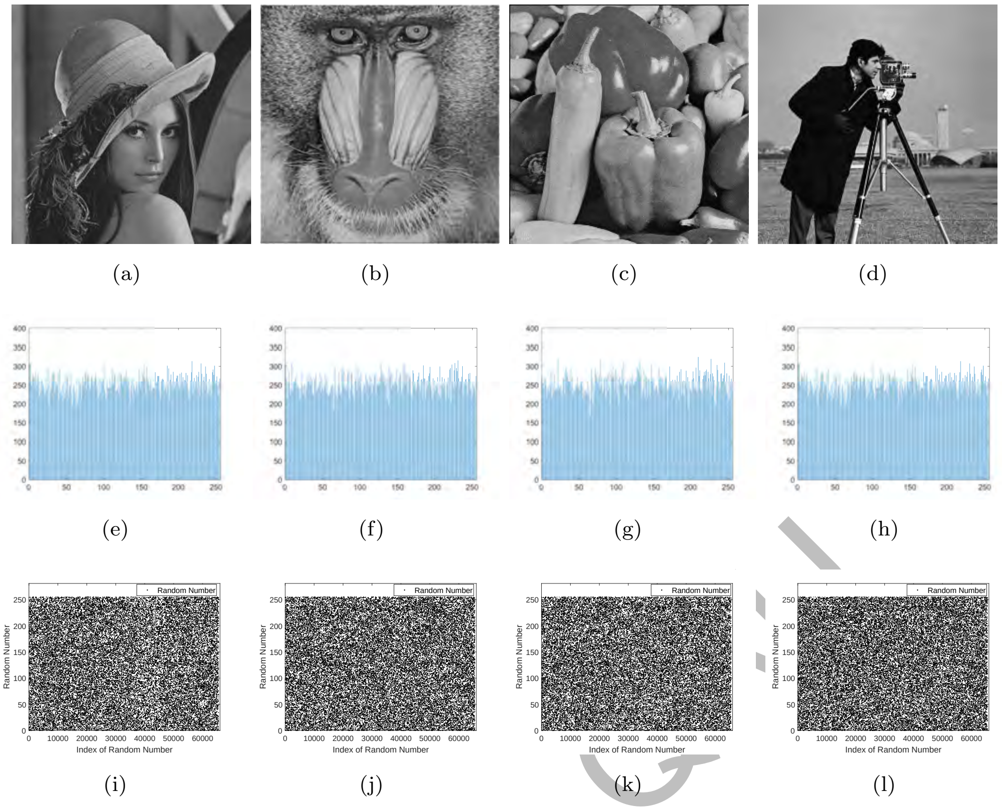
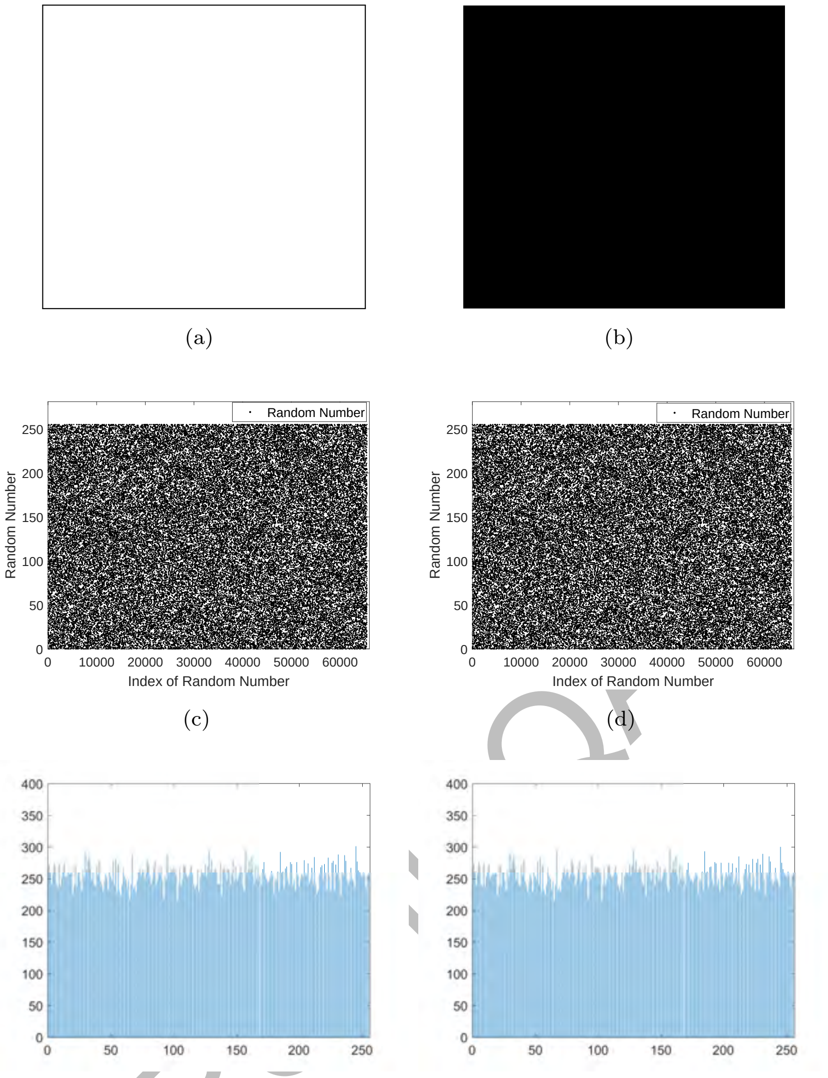


Figure 4.1: PRNs analysis for images Lena, Mandrill, Peppers, Cameraman of size 256×256 for parameters ($p' = 1048847, b' = 1, m_y = p', m_x = 256$); (a) Lena; (b) Mandrill; (c) Peppers; (d) Cameraman; (e)-(h) Histogram of PRNs for images in (a)-(d) generated by sequence $\gamma_{p',b',m_x,m_y,A}$; (i)-(l) PRNs obtained from sequence $\gamma_{p',b',m_x,m_y,A}$ for images in (a)-(d).

4.4 A CML and an EC-Based S-box Generator

We proposed an S-box generator to generate $m \times m$ dynamical S-boxes. The proposed generator use a CML-system and an EC to construct dynamical S-boxes. For an $m \times m$ S-box, choose parameters of $E_{p,a,b}$ and a generator G (or generator of a subgroup of the $E_{p,a,b}$ of large order if it is not cyclic EC). Generate a string S using G such that $|S| = 2^\ell$ for some $0 \leq \ell \leq m$ and a set $T \subset [0, 2^m - 1]$. Now using S and T generate a set X such that $|X| = 2^m$. Iterate the CML in Eq. 1.22 and construct a sequence of integers $U \subset [0, 2^m - 1]$. Generate candidate S-boxes by swapping entries of the initial S-box C_o. Main steps of our S-box generation scheme are followings:

(1) Setting parameters for CML-systems and an EC: Choose parameters p, a, b of an $E_{p,a,b}$



(a) (b)

(c) (d)

(e) (f)

Figure 4.2: PRNs analysis for images with all pixel values 255 (all-white) and all pixel values 0 (all-black) for parameters ($p' = 1048847, b' = 1, m_y = p', m_x = 256$); (a) All-white; (b) All-black; (c) PRNs generated from image in (a);(d) PRNs generated from image in (d); (e) Histogram of PRNs generated from image in (a); (f) Histogram of PRNs generated from image in (d).

and a generator G of the EC $E_{p,a,b}$. Compute the initial conditions x_i, μ, ξ and $x_j = \text{mod}(z + x_{j-1}, 1)$ $j = 1, \ldots, \tau$, where τ is number of lattices and z is used to compute initial parameter x_j for each lattice τ.

(2) Generation of points on an EC: Generate a string of points on the $E_{p,a,b}$ using G. Let $S = \langle G \rangle$ such that $|S| = 2^{\ell}$, where $0 \leq \ell \leq m$. Define $n = m - \ell$ and choose a set $T = \{t_1, \ldots, t_n\} \subset [0, 2^m - 1]$, $t_i \neq t_j$ for $i \neq j$, (i.e., $T = \{2^m \pmod{2^m - i}$ for $i = 1, \ldots, n\}$).

(3) Generation of an EC-based sequence of integers: Let $S_i = S_y \pmod{t_i}$, $i = 1, \ldots, n$, where

S_y is the y-coordinate of the EC points in S. Construct a set X by concatenating ($\|$) each S_i, i.e., $X = [S_1 \| S_2 \| \cdots \| S_n]$.

(4) Iterating the CML-system: The CML-system in Eq. 1.22 with real function as a logistic map in Eq. 1.21 is iterated $2^m + \alpha$ times for $\alpha \geq 10$ and discard the first α iterations to get a sequence not disturbed by the initial conditions. Convert the each iteration of the CML into an integer value as,

$$Y_i(j) = \lceil X_i(j) \times 10^{12} \rceil. \tag{4.2}$$

(5) Construction of a CML based sequence of integers: For an $m \times m$ S-box, define a sequence of integers $U(j) \subset [0, 2^m - 1]$ for each lattice j by applying modulo 2^m as

$$U(j) = [Y_i(j) \pmod{2^m}], i = 1, \ldots, 2^m, j = 1, \ldots, \tau.$$

(6) Generation of S-boxes: Generate candidate S-boxes B_i, $i = 1, \ldots, 2^m$ by swapping entries of the initial S-box $C_o = \{0, 1, \ldots, 2^m - 1\}$. For the S-boxes B_i, use sets U and X to perform a swap operation ($\longleftrightarrow$) on C_o iteratively, i.e.,

$$C_o(U(i) + 1) \longleftrightarrow C_o(X(i) + 1).$$

Final B_i is the required S-box. In Fig. 4.3 the flowchart of our S-box generator is given.

4.4.1 Evaluation of the S-box Generator

In this section, we discuss the experimental analysis of our S-box generator. For this we use parameters of 256-bits given in Table 4.2 for an $E_{p,a,b}$,

We generated a random set of 10000 S-boxes by using an $E_{p,a,b}$ with parameters given in Table 4.2 and a CML with different initial conditions $x_o \in (0, 1)$ and fixing other conditions as

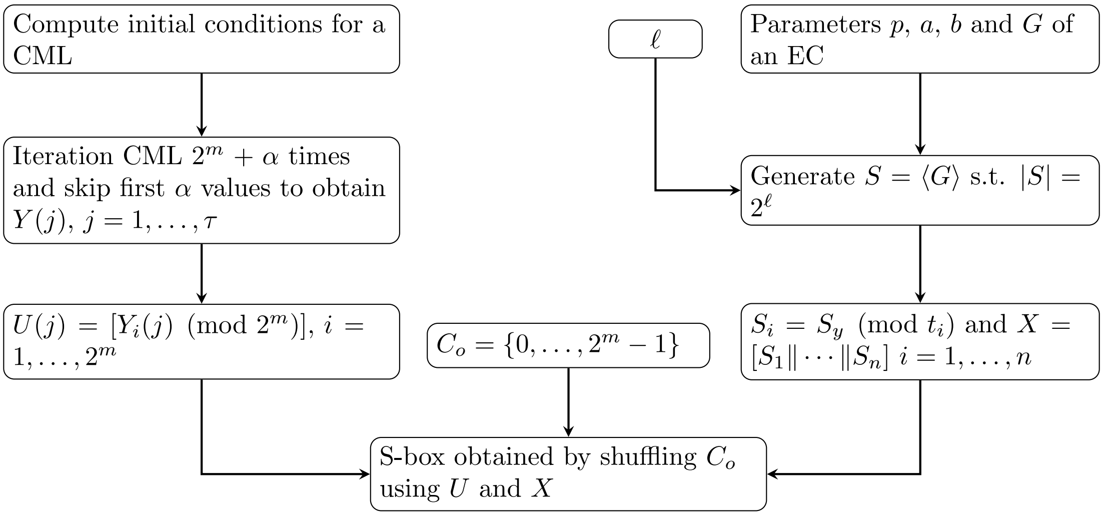

Compute initial conditions for a CML
Iteration CML $2^m + \alpha$ times and skip first α values to obtain $Y(j)$, $j = 1, \dots, \tau$
$U(j) = [Y_i(j) \pmod{2^m}]$, $i = 1, \dots, 2^m$
$C_o = \{0, \dots, 2^m - 1\}$
ℓ
Parameters p, a, b and G of an EC
Generate $S = \langle G \rangle$ s.t. $|S| = 2^\ell$
$S_i = S_y \pmod{t_i}$ and $X = [S_1 \| \cdots \| S_n]$ $i = 1, \dots, n$
S-box obtained by shuffling C_o using U and X

Figure 4.3: Flowchart of the S-box generator.

Table 4.2: The set of parameters for the EC.

p	=115792089210356248762697446949407573530086143415290314195533631308867097853951
a	=115792089210356248762697446949407573530086143415290314195533631308867097853948
b	=115792089210356248762697446949407573530086143415290314195533631308867097853951
G_x	=48439561293906451759052585252797914202762949526041747995844080717082404635286
G_y	=36134250956749795798585127919587881956611106672985015071877198253568414405109

$\mu = 3.9575$, $\xi = 0.4254$, $\alpha = 10$, $z = 0.7500$, $\ell = 7$ and $T = \{256, 255\}$. An S-box constructed by our generator is given in Table 4.3. Following tests are applied to check the performance of the generator for image encryption:

(1) Sensitivity: In image encryption an S-box generator with high sensitivity has great importance because it makes an encryption algorithm strong against differential attacks. The proposed generator is highly sensitive to the input parameters. Effect of the input over the output S-box of our generator is presented in Fig. 4.4. A slight change either in x_o or ξ effects the output of the S-box significantly. Thus, we can say that the generator is suitable for encryption purposes.

(2) Singularity: Our generator has no singularity and generates S-boxes for each valid set of parameters. This feature of the generator speedup the encryption process.

Table 4.3: The S-box for parameters $x_o = 0.7500$, $\mu = 3.9575$, $\xi = 0.4254$, $\alpha = 10$, $z = 0.7500$, $\ell = 7$ and $T = \{256, 255\}$, where p, a, b and G are same as given in Table 4.2.

53	85	207	89	213	173	66	80	162	132	142	93	232	9	105	59
151	17	33	166	23	64	199	81	171	189	95	217	233	20	245	148
209	231	11	92	156	6	178	114	45	146	253	19	241	160	228	78
158	188	120	73	67	195	83	116	41	239	186	130	227	119	123	236
192	51	36	161	57	140	205	94	220	170	60	118	152	62	40	99
5	237	167	4	226	179	121	55	110	149	187	22	169	76	229	70
46	240	176	39	63	27	234	117	164	90	144	182	30	102	97	242
112	65	68	122	155	180	72	211	135	196	200	183	103	141	150	247
107	125	3	98	28	230	104	204	218	16	197	214	185	249	101	1
208	7	24	246	193	82	91	250	201	153	71	133	86	108	49	216
221	26	21	58	168	255	106	42	29	75	154	0	202	136	18	111
244	165	69	198	87	177	113	181	61	243	52	2	203	25	235	8
77	157	137	225	88	163	223	212	96	210	147	109	134	159	175	252
38	139	48	138	12	124	115	174	251	128	222	184	44	84	191	143
32	74	131	10	129	215	34	248	79	219	145	190	206	54	238	254
15	172	47	100	43	126	35	127	224	56	37	13	14	194	31	50

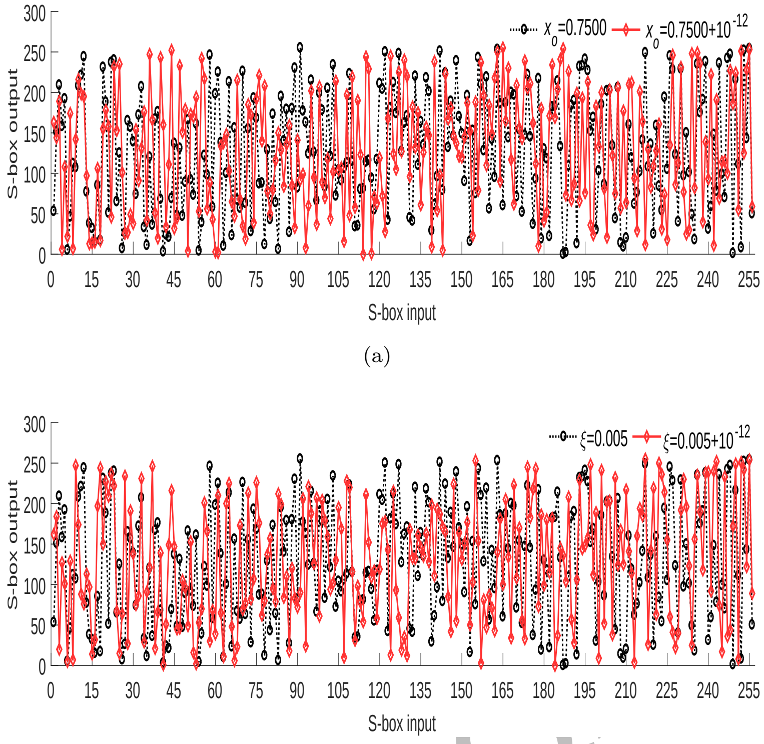
Table 4.4: The NL analysis of 10000 S-boxes.

Generator	NL		
	minimum	average	maximum
Proposed	80	97.70	106
Ref. [55]	82	97.45	104
Ref. [14]	64	92.05	102
Ref. [33]	52	84.64	104
Ref. [101]	84	99.27	104
Ref. [45]	0	90.20	106

Table 4.5: S-box analysis of the proposed -box with available schemes.

S-box	NL	LAP	DAP	AC	SAC		BIC		
					min.	max.	min.	max.	NL
Proposed	106	0.148	0.047	254	0.406	0.641	0.459	0.523	98
Ref. [55]	104	0.141	0.054	253	0.406	0.594	0.461	0.522	98
Ref. [48]	100	0.152	0.039	255	0.391	0.586	0.468	0.537	100
Ref. [49]	96	0.125	0.047	255	0.422	0.609	0.471	0.547	96
Ref. [58]	98	0.133	0.054	254	0.422	0.609	0.477	0.535	94
Ref. [59]	96	0.125	0.039	255	0.359	0.609	0.477	0.541	98
Ref. [61]	104	0.133	0.039	255	0.359	0.609	0.457	0.535	96
Ref. [62]	104	0.133	0.039	254	0.438	0.641	0.475	0.547	98
Ref. [23]	104	0.145	0.039	255	0.391	0.625	0.471	0.531	98
Ref. [56]	104	0.148	0.047	254	0.438	0.578	0.482	0.543	96
Ref. [50]	104	0.094	0.023	255	0.391	0.578	0.476	0.529	103

(3) Cryptographic Strength of output: NL (minimum, average, maximum) of our generator and available schemes for 10000 S-box is given in Table 4.4. Results in Table 4.4 show that the new generator can construct S-boxes with $80 \leq NL \leq 106$ and average NL is



(a)

(b)

Figure 4.4: Sensitivity analysis of the proposed S-box generator. (a) To observe effect of x_o on the S-box generator's output, fix parameters $\{p, a, b, G, n, \mu, \xi, z, \ell\}$ of an S-box and change $x_o = 0.7500$ with $x_o = 0.7500 + 10^{-12}$. (b) To observe the effect of ξ on the S-box generator's output, fix parameters $\{p, a, b, G, n, \mu, x_o, z, \ell\}$ of S-box and change $\xi = 0.005$ with $\xi = 0.005 + 10^{-12}$.

97.70. Thus, Table 4.4 reflects that the proposed generator is efficient than generator in [14, 33, 45, 55, 101]. NL results for 10000 S-boxes in [14, 33, 45, 55] are taken the paper [74]. The performance analysis with S-boxes in [23, 48–50, 55, 56, 58, 59, 61, 62] is given in Table 4.5. The NL of our S-box is batter than the S-boxes in [23, 48, 49, 55, 58, 59, 61, 62, 130, 131] and the BIC-NL is better than [49, 58, 61, 131]. Differential and linear approximation probability [37, 39] of the S-box are 0.047 and 0.148, respectively. Strict avalanche criterion (SAC) [40] and bit independence criterion (BIC) [40] of the proposed S-box are [0.406 0.641] and [0.459 0.523] respectively. Table 4.5 shows that the BIC and SAC are also comparable with S-boxes in [23, 48, 49, 55, 58, 59, 61, 62]. The comparison shows that the newly designed generator is efficient and generates good S-boxes.

4.5 Image Encryption and Decryption

Consider a communication channel between sender Alice and receiver Bob. Let $P_{n\times m}$ be an image that Bob made public and Alice wants to send a plain-image $I_{u\times v}$ to Bob. Compute the hash value of I and generate initial parameters of CML-systems. Produce two chaotic sequences from CML-systems and a string of points on an EC to construct permutations λ and σ. A set of PRNs is generated using $P_{n\times m}$ and a diffused image D is produced. Now shuffle rows and columns of D using permutations λ and σ respectively to get encrypted-image. Let S denote the symbol set of I. The steps of our cryptosystem are given below:

(1) Computation of SHA-256: Compute the hash value of I. Let $H(I)$ denote the SHA-256 hash value of I that we divide into 32 bytes from left-to-right. Let h_i, integer $i \in [1, 32]$ denote the i-th byte of 32 bytes in $H(I)$, i.e., $H(I) = \{h_1 h_2 \dots h_{32}\}$. Convert h_i, $i \in [1, 32]$ into decimal form if they are in hexadecimal.

(2) Selection of parameters for the CML-system to generate permutations: We generate two permutations $S(p, a, b, u)$ and $S(p, a, b, v)$ on $[1, u]$ and $[1, v]$, respectively. For this, select two subsets $D_u = \{h_1, h_2, \dots, h_{16}\}$ and $D_v = \{h_{17}, h_{18}, \dots, h_{32}\}$ of $\{h_1, h_2, \dots, h_{32}\}$ and calculate two sets of parameters $d_u = \{k_1^{(u)}, k_2^{(u)}, k_3^{(u)}, k_4^{(u)}\}$ and $d_v = \{k_5^{(v)}, k_6^{(v)}, k_7^{(v)}, k_8^{(v)}\}$ using Eqs. 4.3 and 4.4. Now, compute parameters k_i, $i = 1, 2, \dots, 8$ as $k_i = \sum_{i=r}^{s} h_i$, where $r = 4(i-1)+1$ and $s = 4i$. Suppose that q is the arithmetic mean of hash values in the set $\{h_1, h_2, \dots, h_{32}\}$ then elements of sets d_u and d_v are computed as

$$\begin{cases} k_1^{(u)} = 3.75 + \mathrm{mod}(\frac{k_1}{2^n} + q, 0.25), \text{and} \\ k_i^{(u)} = \mathrm{mod}(\frac{k_i}{2^n} + q, 1), \text{for } i = 2, 3, 4. \end{cases} \tag{4.3}$$

$$\begin{cases} k_5^{(v)} = 3.75 + \text{mod}(\frac{k_5}{2^n} + q, 0.25), \text{and} \\ k_i^{(v)} = \text{mod}(\frac{k_i}{2^n} + q, 1), \text{for } i = 6, 7, 8. \end{cases} \tag{4.4}$$

(3) Construction of permutations: For the permutation $S(p, a, b, u)$ set the initial conditions of the CML as follow: $\mu = k_1^{(u)}$, $\xi = k_2^{(u)}$, $x_o = k_3^{(u)}$ and $z = k_4^{(u)}$ and using the EC parameters (p, a, b, G) generate a string of points $S = \langle G \rangle$ on an EC. Now generate an S-box $S(p, a, b, u)$ using the algorithm given in Section 4.4. For permutation $S(p, a, b, v)$ set the initial conditions of a CML as follow: $\mu = k_5^{(v)}$, $\xi = k_6^{(v)}$, $x_o = k_7^{(v)}$ and $z = k_8^{(v)}$ and an EC parameters (p, a, b, G) then use the method in Section 4.4 to construct an S-box $S(p, a, b, v)$. For convenience, let σ and λ denote $S(p, a, b, u)$ and $S(p, a, b, v)$, respectively.

(4) Generation of parameters for PRNs: In this scheme, suppose the sender and the receiver agree on a precomputed $\mathrm{E}_{p',0,b'}$ for a prime $p' \equiv 2 \pmod 3$ and a parameter $b' \in [1, p'-1]$. Select $m_\mathrm{x} = h_i \times h_j$ and $m_\mathrm{y} = h_i \times h_j$ for some $i, j \in [1, 32]$, where $h_i, h_j \neq 0$. Finally compute the sequence $\gamma_{p',b',m_\mathrm{x},m_\mathrm{y},P}$ for the plain-image P.

(5) Local diffusion: For each integer $i \leq uv$, generate a diffused image D of plain-image I such that $d(i; D) = (d(i; I) + \gamma_{p,b,m_\mathrm{x},m_\mathrm{y},P}(i, d(i; P))) \pmod{|S|}$.

(6) Global diffusion: Let $R_i, i \in [1, u]$ denote i-th row of D from top to bottom. Then we get an image $\sigma(D)$ such that the i-th, $i \in [1, u]$ row of $\sigma(D)$ is $R_{\sigma(i)}$.

Let $C_i, i \in [1, v]$ denote i-th column of $\sigma(D)$ from left to right. Then we get a ciphertext $\lambda(\sigma(D))$ of I such that the i-th, $i \in [1, u]$ column of $\lambda(\sigma(D))$ is $C_{\lambda(i)}$.

Fig. 4.5 illustrates the flowchart of the encryption algorithm. In Fig. 4.6, an example of our encryption scheme is shown for an image of size 4 × 4.

Lemma 4.1. *Suppose that $\mathrm{E}_{p',0,b'}$ is an MEC with $p' \equiv 2 \pmod 3$ and $b' \in [1, p'-1]$. Consider $m_\mathrm{x}, m_\mathrm{y} \in [1, p']$ are two integers, a CML-system with the total number of lattices τ to be*

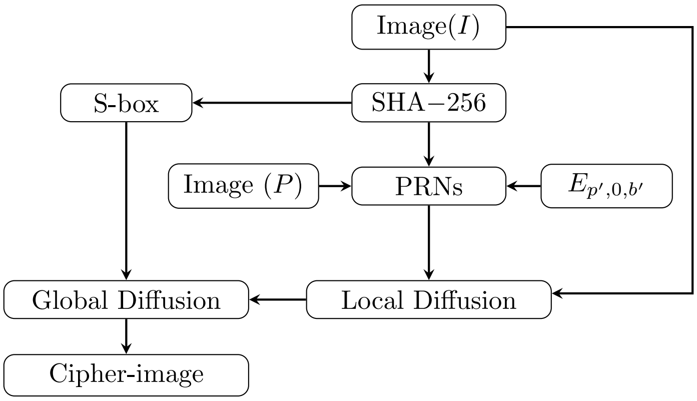
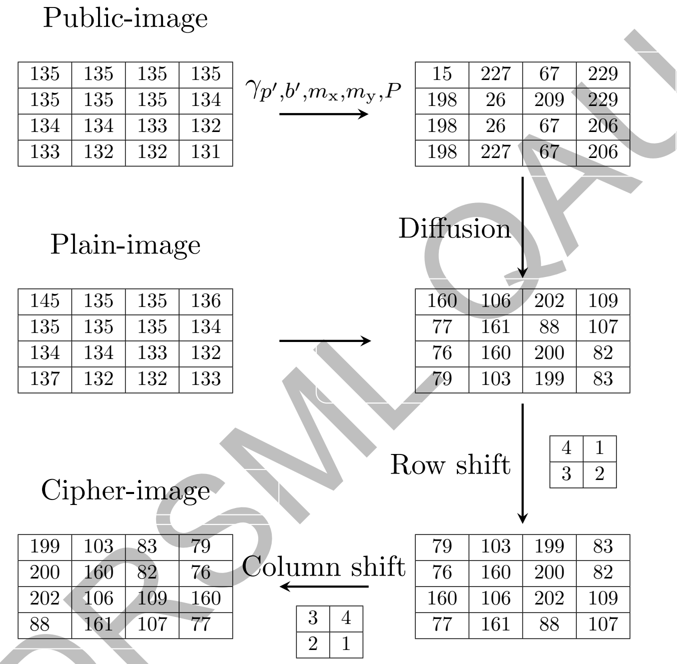


Figure 4.5: Flowchart of the encryption algorithm.



Public-image

135	135	135	135
135	135	135	134
134	134	133	132
133	132	132	131

15	227	67	229
198	26	209	229
198	26	67	206
198	227	67	206

Plain-image

145	135	135	136
135	135	135	134
134	134	133	132
137	132	132	133

160	106	202	109
77	161	88	107
76	160	200	82
79	103	199	83

4	1
3	2

79	103	199	83
76	160	200	82
160	106	202	109
77	161	88	107

3	4
2	1

Cipher-image

199	103	83	79
200	160	82	76
202	106	109	160
88	161	107	77

Figure 4.6: An example of the encryption scheme for an image of size 4 × 4 when $p' = 1048847$ and $(m_x, m_y) = (256, p')$.

generated and $\mathrm{E}_{p,a,b}$ *is an arbitrary EC such that* G *is its generator (or generator of a subgroup of the* $E_{p,a,b}$ *of large order if it is not cyclic EC). Let* $I_{u\times v}$ *be a plain-image and* $P_{m\times n}$ *be a public-image such that* $uv \leq mn$*. Then the proposed encryption algorithm can be implemented in time* $\mathcal{O}(uv + \tau(u+v))$*.*

Proof. We proved the time complexity of the algorithm in the following steps:

(I) By step (5) of Section 4.5, for $I_{u\times v}$, a set of random numbers of size uv is required to diffused I. To generate a random number $\gamma_{p',b',m_{\mathrm{x}},m_{\mathrm{y}},P}(i,d(i;P))$, where $d(i;P)$ is the i-th pixel of P, we need to compute the x-coordinate of $y = i + d(i;P) \pmod{m_{\mathrm{y}}}$ on the EC $\mathrm{E}_{p',0,b'}$ by Eq. 4.1 of Section 4.3. By precomputing the EC and sorting it w.r.t. y-coordinate, we can access y in constant time $\mathcal{O}(1)$. Therefore, the total computation time required to generate a set of PRNs of size uv from the sequence $\gamma_{p',b',m_{\mathrm{x}},m_{\mathrm{y}},P}$ is $\mathcal{O}(uv)$.

(II) At step (6) of Section 4.5, two S-boxes $S(p,a,b,u)$ and $S(p,a,b,v)$ are used to scramble the diffused image D. For each S-box $S(p,a,b,u)$ in step (6) of Section 4.4, a set of integers X based on an EC $\mathrm{E}_{p,a,b}$ is computed and a set of integers Y based on a CML-system is generated to shuffle the initial S-box C_o. By step (2) of Section 4.4, the maximum number of points of an EC $\mathrm{E}_{p,a,b}$ required to generate the set X is u. Therefore, the time complexity to construct the set X is $\mathcal{O}(u)$ when we ignore the time complexity of arithmetic operations to perform scalar multiplication. And at step (5) of Section 4.4, the number of floating point operations used to generate the sequence Y based on the CML-system are τu, where τ is the total number of lattices to be generated. By step (6) of Section 4.4, since one loop of length u is required to shuffle the initial S-box C_o using sets X and Y, so its time complexity is $\mathcal{O}(u)$. Hence the time complexity to generate the S-box $S(p,a,b,u)$ is $\mathcal{O}(\tau u + u + u)$ which is further simplified as $\mathcal{O}(\tau u)$. Now, using the same arguments, the time complexity to generate the S-box $S(p,a,b,v)$ is $\mathcal{O}(\tau v)$. Thus, the time complexity to generate S-boxes $S(p,a,b,u)$ and $S(p,a,b,v)$ is $\mathcal{O}(\tau u + \tau v) = \mathcal{O}(\tau(u+v))$.

(III) By step (6) Section 4.5, we need to shuffle the diffused image D using S-boxes $S(p,a,b,u)$ and $S(p,a,b,v)$, respectively. Therefore, the time complexity to shuffle the rows and columns of D is $\mathcal{O}(u)$ and $\mathcal{O}(v)$, respectively. So, the time complexity to shuffle D is $\mathcal{O}(u+v)$.

The time complexity of steps (I)–(III) is $\mathcal{O}(uv+\tau(u+v)+(u+v)) = \mathcal{O}(uv+\tau(u+v))$. Hence, the proposed encryption algorithm has the time complexity $\mathcal{O}(uv+\tau(u+v))$. □

4.5.1 Decryption

In this scheme, decryption is possible and it is the reverse process of the encryption method. For this, it is necessary to know the inverse S-boxes λ^{-1} and σ^{-1}. These S-boxes can be completely obtained from parameters of the CML and $E_{p,a,b}$. Alice can get the image $I_{\mathrm{u}\times\mathrm{v}}$ by the following procedure:

(1) Compute the hash value of I.

(2) Generate permutations λ and σ using the method given in Sec. 4.5. Compute inverses λ^{-1} and σ^{-1} of S-boxes λ and σ, respectively and recover $d(i,D)$ using Eq. 4.5,

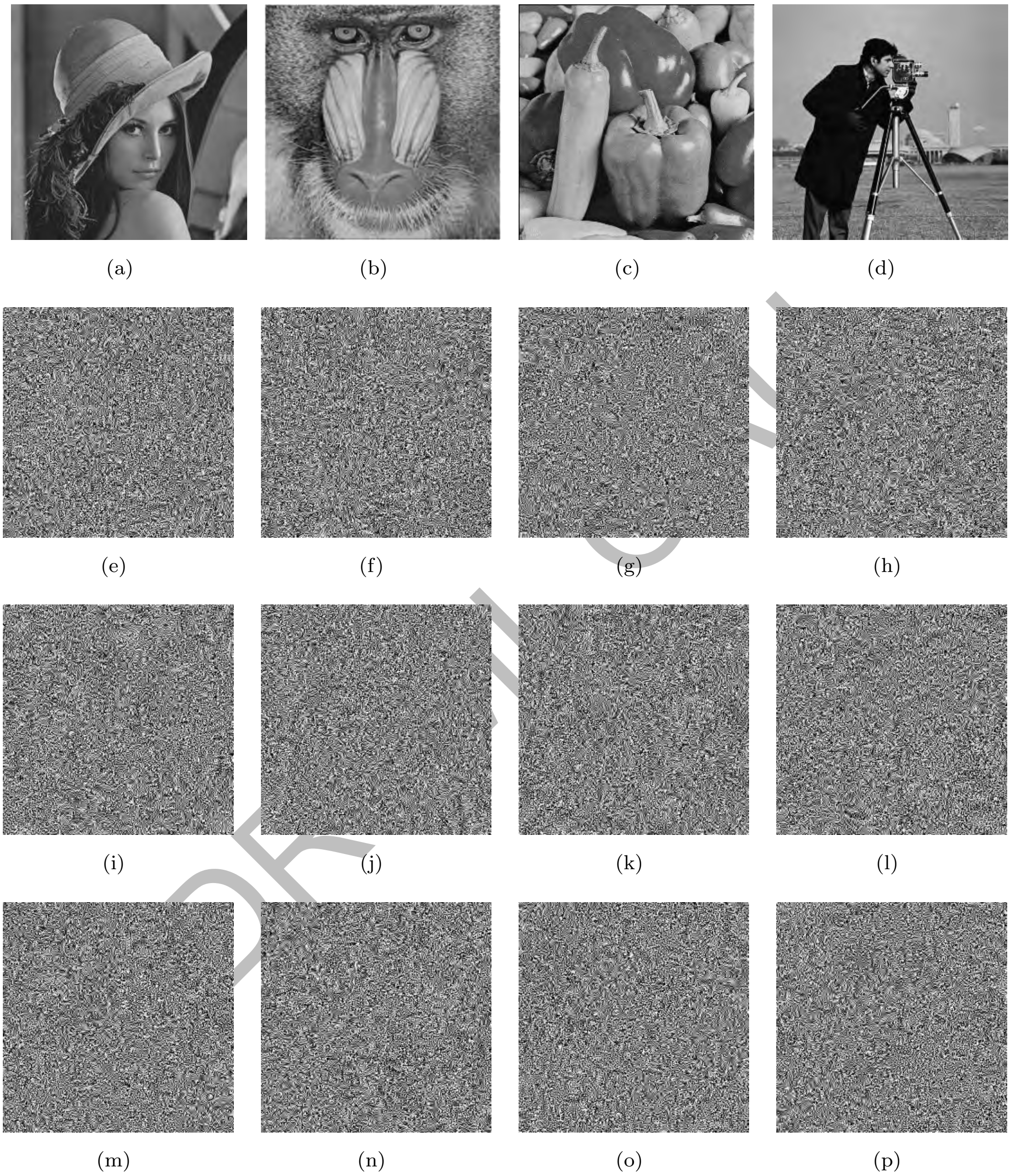
$$d(i,D) = \sigma^{-1}(\lambda^{-1}(C_I)). \tag{4.5}$$

(3) Compute the sequence $\gamma_{p,b,m_{\mathrm{x}},m_{\mathrm{y}},P}$ for the plain-image P.

(4) For each integer $i \leq uv$, find plain-image I such that $d(i;I) = (d(i;D) - \gamma_{p,b,m_{\mathrm{x}},m_{\mathrm{y}},P}(i, d(i;P)))$ (mod $|S|$).

4.6 Security Analysis

For encryption purposes all images are taken from the database USC-SIPI and the standard Lena image of size 256 × 256. In the database USC-SIPI, all images are of size $k \times k, k = 256, 512, 1024$. Security analysis for all the images is performed. We used MATLAB R2017a to run the encryption programs. All experiments are performed on a PC with Intel(R) Core(TM) i5, CPU 3.20GHZ, 8.00GB/RAM and the Microsoft Windows 10/64-bits. In this section all the analysis are done using the all-white image as a public-image P. Hash value of the Lena image of size 256 × 256 used for this section is `"66312459512417010018518672425113134182229771031039224311617216611313149152221972"`. The set of parameters (p, a, b, G) for an $E_{p,a,b}$ is the same as given in Table 4.2.


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

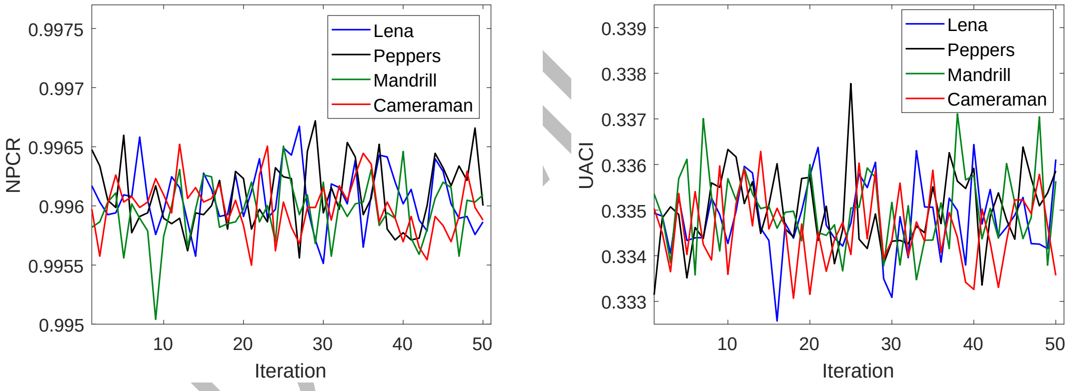
Figure 4.7: (a) Lena with $(m_x, m_y) = (19912, 40885)$; (b) Mandrill with $(m_x, m_y) = (39600, 54056)$; (c) Peppers with $(m_x, m_y) = (30600, 49494)$; (d) Cameraman with $(m_x, m_y) = (36084, 49952)$; (e-h) Ciphered images of images (a)-(d), when P is the all-white image; (i)-(l) Ciphered images of images (a)-(d), when P is the Lena image; (m)-(p) Ciphered images of images (a)-(d), when P is the Cameraman image.

4.6.1 Differential Cryptanalysis

We implemented NPCR/UACI tests for 50 times on different images by changing one pixel at different positions of plain-images. Average value of NPCR/UCAI is given in Table 4.6, where NPCR and UACI values for each iteration for given images is shown in Fig. 4.8. In Table 4.10, NPCR and UACI results for the standard Lena image are shown. Further we run the NPCR/UACI test for all plain-images given in database USC-SIPI and graphical results are plotted in Fig. 4.9(a) and Fig. 4.9(b).

Table 4.6: NPCR-UACI results of the proposed scheme.

Plain-image	Lena	Mandrill	Pepper	Cameraman
NPCR	99.61	99.60	99.61	99.60
UACI	33.49	33.51	33.50	33.44



(a)

(b)

Figure 4.8: NPCR-UACI values of 50 iterations of different images; (a) The NPCR values values of 50 iterations of different images; (b) The UACI values.

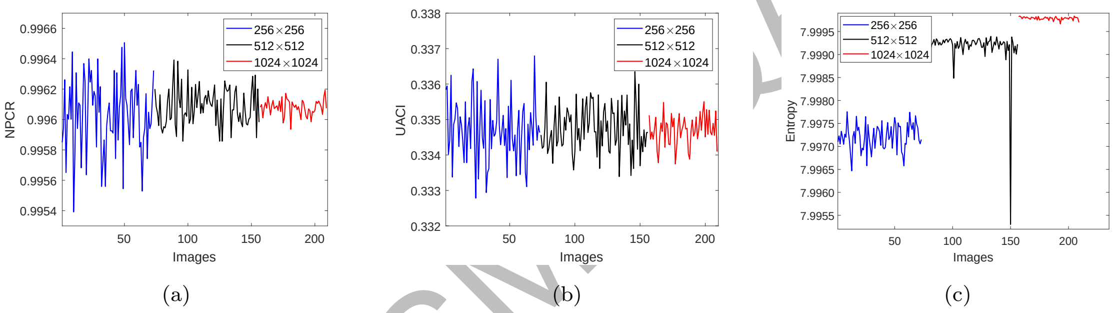
4.6.2 Statistical Cryptanalysis

An encryption scheme is acceptable for real-time encryption if it satisfies well-known tests such as entropy, histogram, and correlation. In the following, each test and related outcomes are discussed in detail.

Entropy Test: In image data, high entropy implies high randomness. The entropy of different images is given in Table 4.7. The entropy of encrypted-images for comparison is shown in Table 5.11 and the entropy of all encrypted-images is closer to the ideal value when compared with schemes in [106, 107, 140–148]. We also test the proposed algorithm for all plain-images in the database USC-SIPI. The results are displayed in Fig. 4.9(c) and the entropy is between $7.995 \leq H(I) \leq 7.9999$. Hence, the proposed algorithm produces high randomness.

Table 4.7: Entropy value.

Image	Lena	Mandrill	Pepper	Cameraman
Plain-image	7.4204	7.2641	7.5571	7.1048
Cipher-image	7.9975	7.9971	7.9976	7.9972

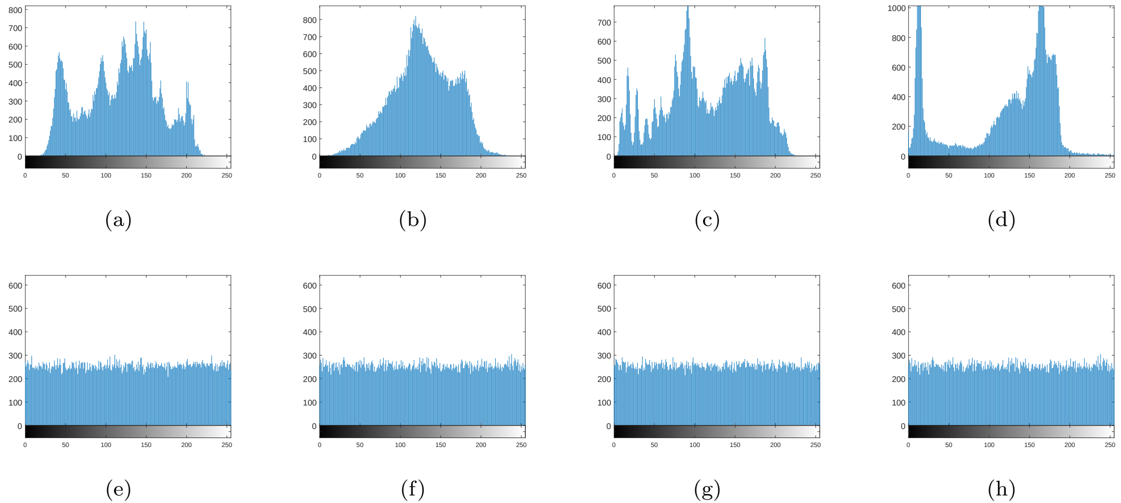


(a) (b) (c)

Figure 4.9: (a) The distribution of NPCR of different size images; (b) The distribution of UACI of different size images; (c) The distribution of entropy of different size images.

Histogram Test: A cipher scheme is considered secure if histograms of encrypted-images are uniform. The histograms of plain-images in Figs. 4.7(a)-(d) and their encrypted-images are given in Figs. 4.10(a)-(d) and Figs. 4.10(e)-(h) respectively. Uniform distribution of histograms of encrypted-images is an evidence of security of the proposed encryption scheme.

Correlation Test: The correlation of different images is given in Table 4.8. The results in Table 5.11 are also comparable to the schemes in [105–107, 140–151]. For $\text{Lena}_{256\times256}$ plain and encrypted images pixels correlation is plotted in Fig. 4.12 The correlation of all plain-images in the database USC-SIPI is displayed in Fig. 4.11. The results predict that the proposed algorithm satisfies the correlation test.



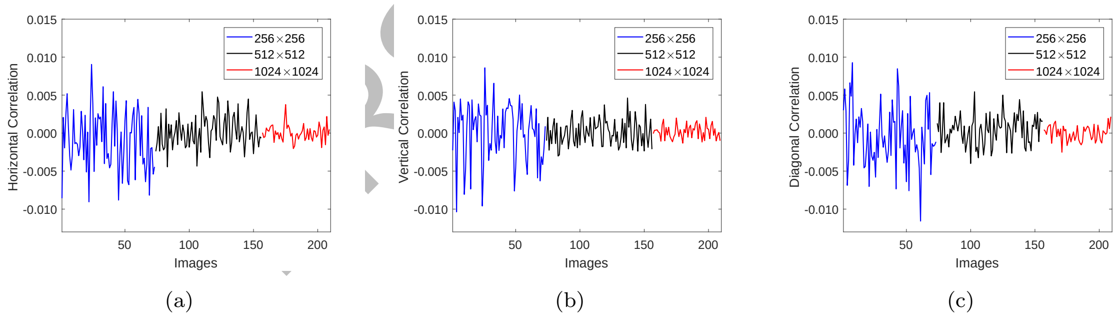
(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.10: (a)-(d) Plots of histogram of the plain-images in Fig. 4.7(a)-(d), respectively; (e)-(h) Plots of histogram of the ciphertexts in Fig. 4.7(e)-(h), respectively.

Table 4.8: Correlation results of the proposed scheme.

Plain-image	Correlations of plain-image			Correlations of cipher-image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena	0.9390	0.9136	0.9680	0.0007	0.0022	0.0085
Mandrill	0.9048	0.8899	0.8199	0.0032	0.0042	0.0020
Pepper	0.9429	0.9441	0.9025	-0.0042	0.0039	-0.0046
Cameraman	0.9331	0.9592	0.9075	0.0015	0.0023	0.0025

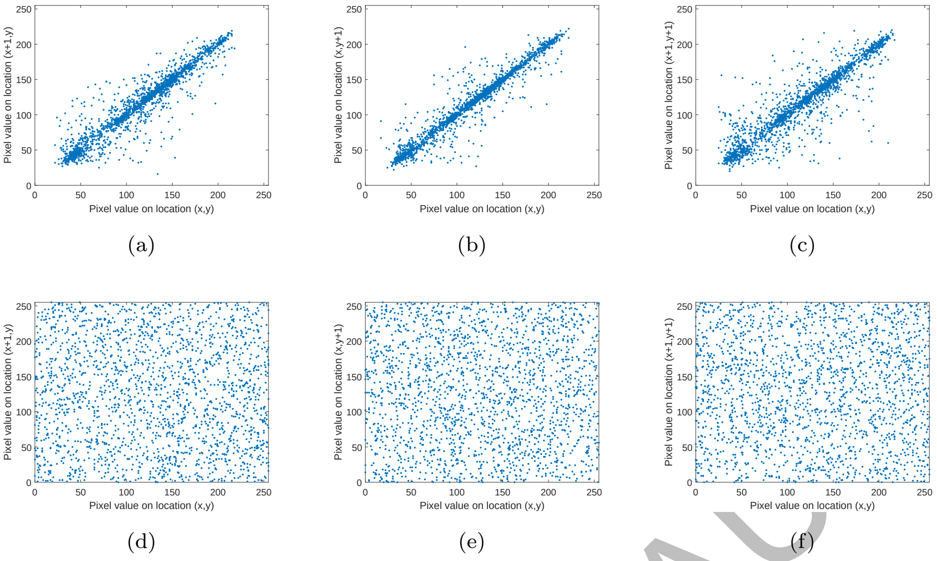


(a) (b) (c)

Figure 4.11: Two adjacent pixels distribution in different directions of images of size 256 × 256, 512 × 512 and 1024 × 1024 in the database USC-SIPI encrypted by the proposed encryption method; (a) Horizontal; (b) Vertical; (c) Diagonal.

4.6.3 Key Analysis

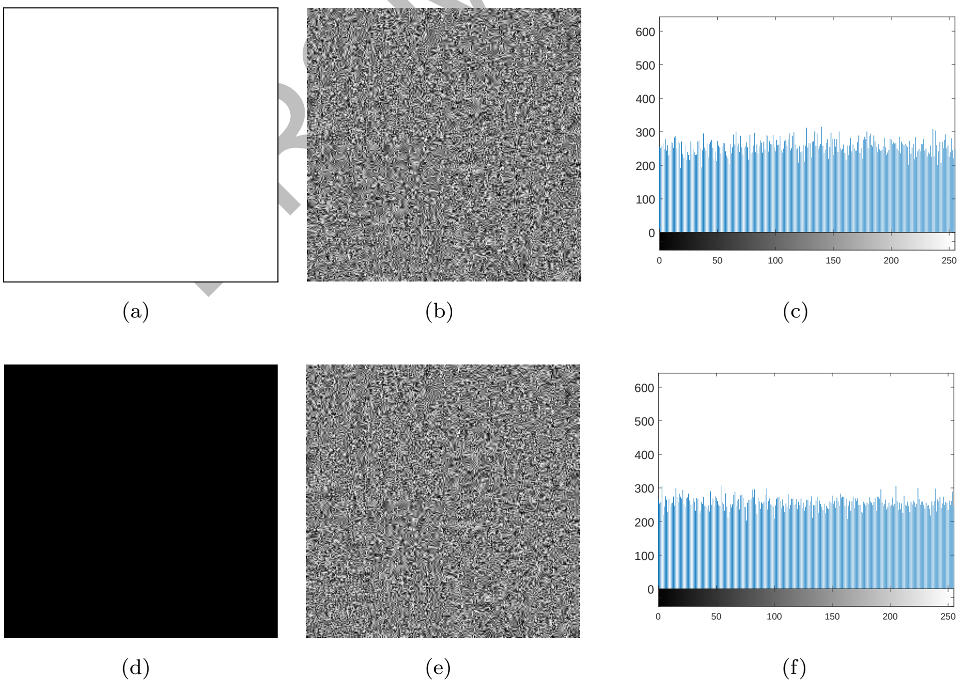
Keyspace: In the proposed cryptosystem, we used parameters p, a, b, G, p', b' and SHA-256 as keys, where each parameters length is 256-bits. So, the keyspace of the our cryptosystem is much larger than 2^{128}. Thus, the proposed encryption technique can resist brute-force attacks.



(a) (b) (c)

(d) (e) (f)

Figure 4.12: (a)-(c) Two adjacent pixels distribution in different directions of the plain-image Lena$_{256\times256}$; (d)-(f) Two adjacent pixels distribution in different directions of the cipher-image Lena$_{256\times256}$.



(a) (b) (c)

(d) (e) (f)

Figure 4.13: (a) All-white image $(m_x, m_y) = (24806, 33807)$; (b) Cipher-image; (c) Histogram; (d) All-black image $(m_x, m_y) = (20022, 46968)$; (f) Cipher-image; (e) Histogram.

Key Sensitivity: We use the hash value of I to generate parameters of the CML. The sensitivity of the CML-system to the initial conditions is shown in Fig. 4.4, which shows that the proposed system is highly sensitive to keys.

Table 4.9: Security analysis.

Image	NPCR	UACI	Correlation			Entropy
			Horizontal	Vertical	Diagonal	
All white	99.68	33.45	0.0028	0.0050	-0.0002	7.9946
All black	99.63	33.60	-0.0056	0.0076	0.0046	7.9941

Table 4.10: Comparison for Lena$_{256\times256}$ image.

Algorithm	NPCR	UACI	Correlation			Entropy
			Horizontal	Diagonal	Vertical	
Proposed	99.610	33.670	0.0030	0.0096	0.0026	7.9975
Ref. [105]	99.614	33.472	-0.0018	-0.0009	0.0011	7.9975
Ref. [106]	99.617	33.392	0.0029	-0.0003	0.0080	7.9965
Ref. [107]	99.625	33.423	0.0069	0.0075	0.0479	7.9972
Ref. [150]	99.621	33.413	0.0003	-0.0003	-0.00004	7.9977
Ref. [142]	99.614	33.546	-0.0016	-0.0026	0.0043	7.9970
Ref. [149]	99.617	33.452	-0.0008	-0.0011	-0.0014	7.9975
Ref. [152]	99.614	33.458	-0.0063	-0.0016	0.0065	7.9975
Ref. [143]	99.626	33.458	-0.0003	-0.0029	0.0016	7.9972
Ref. [144]	99.580	34.080	-0.0003	-0.0066	-0.0013	7.9967
Ref. [151]	99.609	33.451	-0.0018	0.0040	-0.0006	7.9976
Ref. [141]	99.606	33.463	0.0004	0.0051	0.0051	7.9974
Ref. [145]	99.594	33.505	0.0119	0.0011	0.0092	7.9970
Ref. [146]	99.599	33.456	-0.0029	0.0004	-0.0017	7.9971
Ref. [148]	99.605	33.419	0.0086	0.0009	0.0014	7.9970
Ref. [140]	99.695	33.384	0.0015	0.0057	0.0041	7.9962
Ref. [147]	99.612	33.457	-0.0052	-0.0003	0.0031	7.9973
Ref. [153]	99.610	32.790	0.0097	0.0178	0.0136	7.9971
Ref. [154]	99.618	33.415	0.0003	0.0004	0.0025	7.9972
Ref. [155]	99.620	33.510	-0.0230	-0.0034	0.0019	7.9974

4.6.4 Known-plaintext/chosen-plaintext attac

A set of RNs is used to generate diffusion in the plain-image I and further a pair of S-boxes $S(p, a, b, u)$ and $S(p, a, b, v)$ is used to create confusion in the diffused image. Since SHA-256 is purely plain-image dependent, hence the pair of S-boxes will be different for different plain-images. This implies that the proposed scheme will withstand plaintext attack.

To prove that an intruder cannot break the proposed cryptosystem using chosen-plaintext attack, let B_i for $i = 1, 2, \ldots, 2^n$ be the candidate S-boxes generated from the swap operation on initial S-box C_o by step (6) in Section 4.4. Since the set Y resulting from the CML-system, and the set X generated using an EC $\mathrm{E}_{p,a,b}$ are randomly generated over the set $\mathbb{Z}_{2^n}$, therefore each candidate S-box B_i for $i = 1, 2, \ldots, 2^n$ is randomly distributed over the symmetric group S_{2^n}. So, if the sets X, and Y are used to generate the final S-box $S(p, a, b, u)$ then the final S-box $S(p, a, b, u)$ is randomly distributed over the symmetric group S_{2^n}. Hence, for any known S-box $S(p, a, b, u)$ the candidate S-boxes are likely to take any value from S_{2^n}. Therefore, intruders finding the final S-box $S(p, a, b, u)$, using chosen-plaintext attack find no information about B_i. Consequently, the sets X and Y remain secret, and no information is obtained about the encryption keys. Similarly, an intruder will not find any information about keys if he tries to get the keys using S-box $S(p, a, b, v)$.

Furthermore, by step (4) in Section 4.3, the output of the sequence $\gamma_{p,b,m_{\mathrm{x}},m_{\mathrm{y}},A}$ is highly dependent on m_{x} and m_{y}. And both m_{x} and m_{y} are randomly distributed variables over the set $[1, p']$, therefore, if an adversary successfully finds the set of PRNs using chosen-plaintext attack then it is very hard to generate set $(a_1, a_2) \pmod{m_{\mathrm{y}}}$ for a large prime p'. Hence, if one can find the output of $\gamma_{p,b,m_{\mathrm{x}},m_{\mathrm{y}},A}$ then he will not be able to find secret keys m_{x} and m_{y}. From the above discussion, it is clear that if an intruder successfully deciphers a ciphertext using the chosen-plaintext attack, he will not be able to find any information about encryption keys. Hence, it proves that our algorithm has high immunity to a chosen-plaintext attack.

The attackers mainly use the all-black (resp. all-white) image to catch any orderliness in pixels of encrypted-image. Fig. 5.9 shows that there is no detectable pattern and histograms are uniform. Table 4.9 shows that the NPCR/UACI are very close to optimal values. Thus, the proposed scheme has high resistance against known-plaintext and chosen-plaintext attacks.

4.6.5 Computation Analysis

The encryption scheme has the time complexity $\mathcal{O}(uv + \tau(u + v))$ as proved in Lemma 4.1. For comparisons, we fix the total number of lattices $\tau = 8$ in the CML-system. So, the time complexity becomes $\mathcal{O}(uv+8(u+v))$. The time complexity comparison with schemes [132–137, 156] is presented in Table 4.11, which indicates the low time complexity of the proposed scheme than the methods in [132–137, 156]. The computation time in Table 4.13 is computed under the same environment. Run time comparison of our algorithm with algorithms in [141, 150, 156–159] over different operating systems is presented in Table 4.12. Our encryption scheme takes 0.641 seconds and 1.311 seconds to encrypt Lena$_{256\times256}$ and Lena$_{512\times512}$ images, respectively.

Table 4.11: The time complexity analysis for images of size $M \times N$ with related schemes.

Algorithm	System characteristics
Our	$\mathcal{O}(uv + 8(u + v))$
Ref. [132]	$\mathcal{O}(36uv + log(4uv))$
Ref. [133]	$\mathcal{O}(8uvlog(4uv) + 4uv)$
Ref. [134]	$\mathcal{O}(212uv + 4ulog(u) + 4vlog(v))$
Ref. [135]	$\mathcal{O}(54uv + 20u)$
Ref. [136]	$\mathcal{O}(61uv + 3(u + v))$
Ref. [137]	$\mathcal{O}(25uv))$
Ref. [156]	$\mathcal{O}(6uv)$

As shown in Table 4.13, the proposed encryption scheme is much faster than the methods in [14, 16, 107, 117, 132, 160, 161]. We used Lena images of size 256 × 256 and 512 × 512 for comparison in Table 4.13. Similarly, it is clear from results in Table 4.12 that the newly designed cryptosystem is comparatively faster than algorithms in [141, 150, 156–159]. The performance of

Table 4.12: Run time analysis for Lena image of size 256 × 256 with related schemes over different operating systems.

Algorithm	Time(s)	System characteristics
Our	0.6460	MATLAB R2017a, CPU i5-3.2 GHz, 8 GB RAM
Ref. [150]	1.1247	MATLAB R2016a, CPU 3.0 GHz, 8 GB RAM
Ref. [141]	2.4600	MATLAB R2017a, CPU 2.8 GHz, 8 GB RAM
Ref. [157]	1.4816	MATLAB 7.14, CPU i7-3.4 GHz, 16 GB RAM
Ref. [158]	2.2234	MATLAB R2016a, CPU i7-2.7 GHz, 8 GB RAM
Ref. [159]	1.8936	MATLAB R2016b, CPU 2.40 GHz, 12 GB RAM
Ref. [156]	4.6880	MATLAB R2016b, CPU 2.8 GHz, 8 GB RAM

Table 4.13: Run time analysis in seconds(s) for Lena images applied on same operating system.

Image size	References							
	Proposed	[107]	[14]	[16]	[160]	[132]	[117]	[161]
256 × 256	0.641	1.286	8.744	3.035	0.834	2.913	1.654	15.45
512 × 512	1.311	4.333	2731.450	3.082	2.849	11.26	6.353	76.777

our suggested algorithm is better than existing approaches and that it is applicable to real-time image encryption.

4.7 Conclusion

In this chapter, a novel image encryption scheme to encrypt digital images for real-time encryption is proposed. The proposed scheme is based on an EC over large primes and a CML chaotic system. Public-key dependent PRNs are generated to generate diffusion in the plain-image. An efficient S-box generator for generation of plain-image dependent dynamic S-box obtain from an EC and a CML is designed. The proposed S-boxes are used to create confusion in the plain-image. Advantages of this work are:

- Table 4.4 reflects that out method is efficient than generator in [14, 33, 45, 55, 101].
- In Table 4.10, NPCR and UACI results for the standard Lena image are better than the schemes in [141, 144–146, 148, 151].
- The proposed cryptosystem has low time complexity when compared with schemes in [132–137, 156] theoretically and less running time than the algorithms in [14, 16, 107, 117, 141, 150, 157–161].
- The proposed algorithm works for large parameters of an EC.

Moreover, the suggested technique has more security measures to resist chosen-plaintext and known-plaintext attacks due to the use of key-dependent dynamic S-boxes and the key-stream of PRNs. Further, the large keyspace guarantees security against the brute-force attacks.

Chapter 5

Elliptic Curves-Based Robust Image Cryptosystem with High Security and Sensitivity

5.1 Introduction

We have designed an image cryptosystem to tackle the issues related to small-key sizes, noise attacks, plaintext attacks, and data loss during transmission. The proposed cryptosystem is divided into three parts. In the first part, we have developed a PRNG to diffuse the pixels of the plain-image. Our PRNG is highly efficient and generates random-numbers with high randomness. In the second step, an S-box generator is constructed to generate permutation S-boxes with high NL. The generator has various advantages over the existing S-box generation algorithms in terms of the NL and robustness. In the final step, an image encryption technique is presented to encrypt grayscale images. The image cryptosystem is sensitive to the plain-image and robust against data loss and noise attacks. The rest of the chapter is set up as. The related work to the new image cryptosystem is given in Sec. 5.2. In Sec. 5.3, a PRNG based on an EC is proposed and analyzed. An S-box generator is introduced in Sec. 5.4. A description of our cryptosystem is presented in Sec. 5.5. Sec. 5.6 presents a security evaluation of the suggested image encryption method. Finally, Sec. 5.7 offers conclusions.

5.2 Related Work to the Cryptosystem

More and more study is being done to figure out how to successfully prevent unauthorized access to image data including private information during network transmission. With the use of image encryption techniques, the private information contained in plain-images can be efficiently protected. For example, a plain-image can be transformed into a cipher image that has the appearance of noise or texture [162]. However, it is simple for the attacker to get a hint from the noise-like cipher image, which prompts the attacker to launch an initial attack. The common security flaws of the current image encryption techniques are enumerated below based on cryptanalysis: (i) A single round of the permutation plus diffusion structure, (ii) a small keyspace, (iii) a key stream generated independently of the plain-image, and (iv) computation time. There are a number of image encryption techniques including PRNs-based encryption, chaos-based encryption, and EC-based encryption schemes designed to overcome common security flaws.

The location of pixels in the image is changed during the confusion phase to make the secret appear meaningless. On the other hand, during diffusion or substitution, pixels of an image are changed to other values [163]. The histogram remains intact while the pixel positions are modified in a permutation-based encryption method [164], which is weak and easily breakable [165]. The confusion phase and diffusion phase are utilized to strengthen the security of the encryption methods [166].

An encryption method for wireless sensor networks that makes use of chaotic maps and ECC over a prime field is designed in [167]. In [168], a new two-dimensional chaotic system is designed to generate the random sequence to create confusion and diffusion in an image pixels. Image encryption scheme in [169] describes the creation of an asymmetric cryptography algorithm based on a quantum chaotic system. Six-dimensional chaotic maps are used in the novel encryption method described in [170]. The use of a PRNG is critical for determining the unpredictability of several cryptographic techniques. In [171], a new S-box generator based on a deterministic approach over ECs is constructed to construct highly dynamic and secure S-boxes with low computing cost. In [172], a unique EC-based image encryption approach has been suggested to attain security near theoretically ideal levels while requiring low computing time. A PRNG over isomorphic ECs is also defined to generate PRNs to encrypt an image data. To achieve excellent security against statistical results, an optimization problem is constructed with an objective function in terms of entropy and correlation. In [173], a novel image encryption technique is introduced. The chaos theory and ECs-based PRNG is the foundation of the encryption algorithm.

5.3 Designing of a New PRNG

In this section, we proposed a dynamic algorithm to develop a PRNG over ECs. The newly designed method is based on ECs and an arbitrary set of rational numbers. The technique has low computation time over large primes and constructs random-numbers with high entropy values which are very advantageous for cryptographic uses. The construction procedure of the proposed PRNG is as follows:

(i) Choose an EC $E_{p,a,b}$ with a generator G of a subgroup of largest order in $E_{p,a,b}$. Generate a set

$$A = \{(x_i, y_i) \in E_{p,a,b}, i = 1, 2, \dots, \lambda\} \text{ s.t. } |A| = \lambda. \tag{5.1}$$

(ii) Define a mapping e such that

$$e : A \times (0, 1) \to A_\epsilon \times A_\epsilon \tag{5.2}$$

$$((x, y), \epsilon) \mapsto (\epsilon x, \epsilon y)$$

where $A_\epsilon \times A_\epsilon = \{(\epsilon x, \epsilon y) \ \forall \ (x, y) \in A\}$ and $\epsilon \in (0, 1)$ is an arbitrary point.

(iii) Choose a gain factor 10^{δ} to normalize points of the set A_{ϵ}. *e.g.*,

$$\tilde{x} = \epsilon x \times 10^{\delta}, \tag{5.3}$$

$$\tilde{y} = \epsilon y \times 10^{\delta}. \tag{5.4}$$

(iv) Convert $\tilde{x}$ and $\tilde{y}$ into binary strings $\mathcal{S}_x$ and $\mathcal{S}_y$, respectively. Concatenate($\|$) $\mathcal{S}_x$ and $\mathcal{S}_y$ to generate a string $\mathcal{S}_z$ of binary bits. Again for each point of the EC concatenate the binary strings $\mathcal{S}_z$ to a string $\mathcal{S}$ of required length $\ell \geq n2^{2n}$, where n is an integer.

(v) Construct a set $\mathcal{B} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{\flat}\}$ by dividing $\mathcal{S}$ from left to right into subsequences ($\mathcal{S}_i$) each of length n and discarting extra right most bits, where $\flat$ is equal to 2^{2n}.

(vi) Define a mapping

$$\mathcal{G} : \mathcal{B} \rightarrow [0, 2^n - 1], \tag{5.5}$$

$$\mathcal{S}_i \mapsto n_i \in [0, 2^n - 1]. \tag{5.6}$$

Where $\mathcal{G}$ converts binary strings to their decimal forms, *e.g.*, $n_i = \text{bin2dec}(\mathcal{S}_i)$.

(vii) Output of function $\mathcal{G}$ is the proposed set of random-numbers $\gamma(p, a, b, G, \epsilon, n)$.

The randomness of binary sequences is tested using NIST-800-22 statistical tests [8]. We have generated 100 binary sequences each of length 10^6 for different values of parameter (ϵ) and set of parameters of an $E_{p,a,b}$ in Table 5.8. In Table 5.1, the results of NIST statistical tests are given. The proposed PRNG has the ability to generate binary sequences with good cryptographic properties. We have generated random-numbers sequences of different lengths. Analysis of the period and entropy bounds of these sequences are given in Table 5.2. In Fig. 5.1, a bar chart is given for random-number sets of size 1024 and 65530. We have plotted the outcome

Algorithm 2 The PRNs generation algorithm.

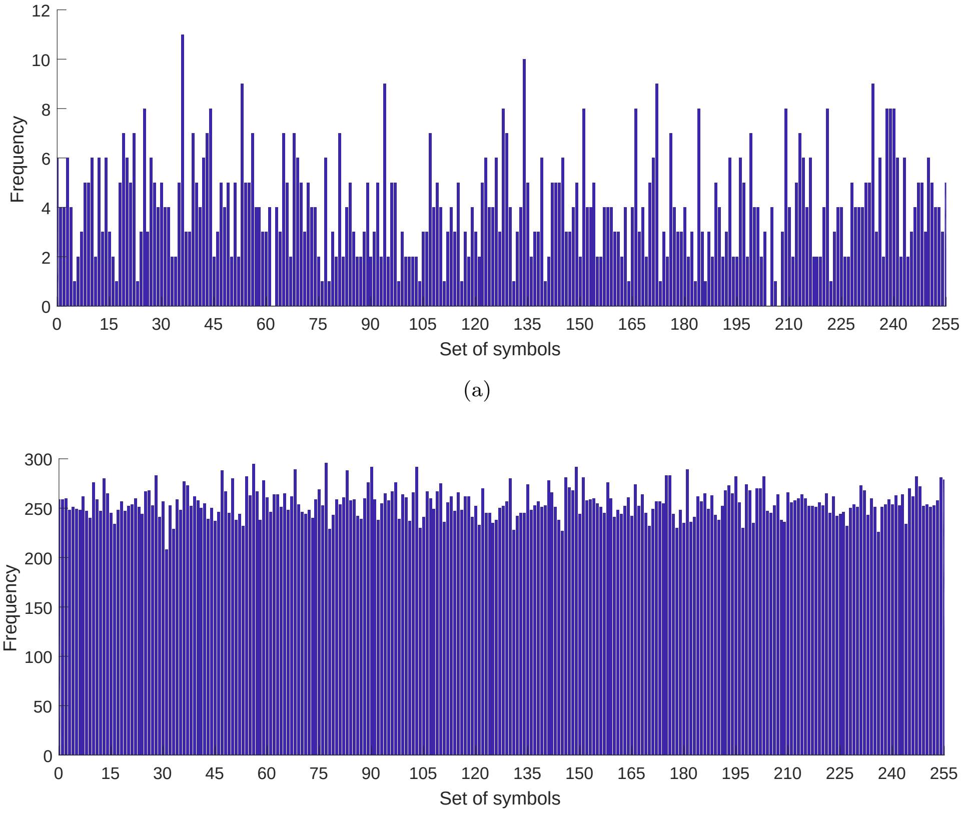
```
Require: Initial parameters $\epsilon$, $p$, $a$, $b$ and a base point $G$ of an $E_{p,a,b}$.
Ensure: Set of PRNs $\gamma(p, a, b, G, \epsilon, n)$.
$\mathcal{S} = \{\}$;  /* $\mathcal{S}$ is a set contains binary string generated after the implementation of map $e$ on points of the EC $E_{\mathrm{p,a,b}}$.
while $|\mathcal{S}| \leq \ell$ do
    /* Where $\ell = n2^{2n}$ for a set $\gamma$ of PRNs and $n$ is bit-length of each random-number in $\gamma$. e.g., $r_i \in [0, 2^n - 1]$ $\forall\, r_i \in \gamma$
    Generate $G_i = iG$ and compute $(\epsilon x_i, \epsilon y_i)$ using map $e$.
    Construct the binary string $\mathcal{S}_z = [\mathcal{S}_x; \mathcal{S}_y]$ using Eqs. 5.3 and 5.4;
    $\mathcal{S} = [\mathcal{S}; \mathcal{S}_z]$;
end while
$\mathcal{B} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\flat\}$;  /* Each $\mathcal{S}_i$ is a $n$-bits subsequence of $\mathcal{S}$ where $\mathcal{S}$ is from left to right and discarting extra right most bits, here $\flat = 2^{2n}$.
$\mathcal{G} : \mathcal{B} \to [0, 2^n - 1]$, $\mathcal{S}_i \mapsto n_i$;  /* Where $\mathcal{G}$ converts binary strings to their decimal forms.
Output of $\mathcal{G}$ is the proposed $\gamma(p, a, b, G, \epsilon, n)$
```

Table 5.1: NIST test results.

Statistical Test	The Proposed PRNG		
	p-Value	Pass Rate	Status
Frequency (monobit)	0.759756	99/100	Success
Block-frequency	0.474986	100/100	Success
Cumulative sums (Forward)	0.574903	99/100	Success
Cumulative sums (Reverse)	0.759756	99/100	Success
Runs	0.213309	100/100	Success
Longest run of Ones	0.262249	99/100	Success
Rank	0.304126	100/100	Success
FFT	0.383827	100/100	Success
Non-overlapping templates	0.500963	99/100	Success
Overlapping templates	0.085587	98/100	Success
Universal	0.935716	100/100	Success
Approximate entropy	0.071177	100/100	Success
Random-excursions	0.437274	61/62	Success
Random-excursions Variant	0.964295	62/62	Success
Serial 1	0.759756	99/100	Success
Serial 2	0.514124	98/100	Success
Linear complexity	0.224821	100/100	Success

Table 5.2: Analysis of period and entropy for different sets of PRNs, when p, a, b, G are the same as given in Table 5.8 and $\epsilon = 0.0102310371$.

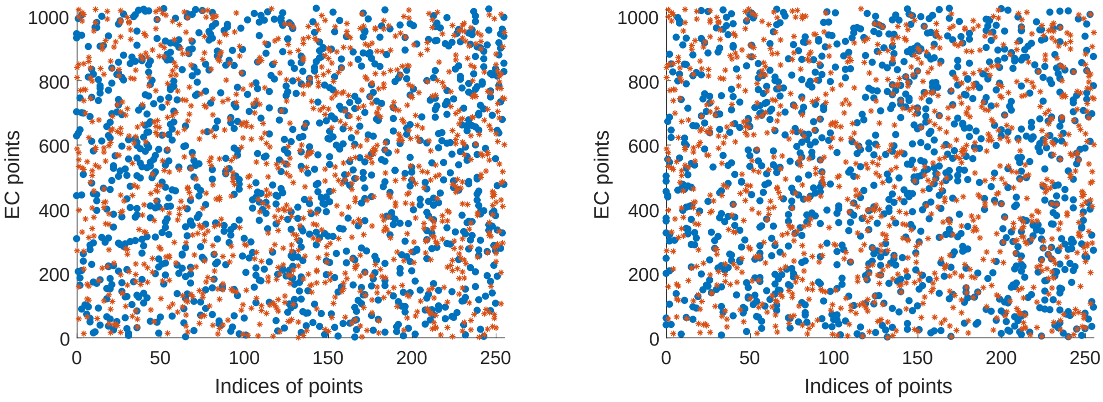
PRNs	Period		Entropy	
	Obtained	Optimal	Obtained	Optimal
$\gamma(p, a, b, G, \epsilon, 4)$	512	512	3.9656	4
$\gamma(p, a, b, G, \epsilon, 5)$	1024	1024	4.9760	5
$\gamma(p, a, b, G, \epsilon, 6)$	4096	4096	5.9894	6
$\gamma(p, a, b, G, \epsilon, 7)$	16384	16384	6.9935	7
$\gamma(p, a, b, G, \epsilon, 8)$	65536	65536	7.9976	8
$\gamma(p, a, b, G, \epsilon, 9)$	262144	262144	8.9986	9



(a)

(b)

Figure 5.1: For $\epsilon = 0.0102310371$ and the $E_{p,a,b}$ parameters in Table 5.8 histograms of PRNs; (a) The histogram of $\gamma(p, a, b, G, \epsilon, 10)$, where frequency of an integer in the γ is the height the corresponding bar. (b) The histogram of $\gamma(p, a, b, G, \epsilon, 16)$, where frequency of an integer in γ is the height the corresponding bar.



(a)

(b)

Figure 5.2: Sensitivity of the output of $\gamma(p, a, b, G, \epsilon, n)$ to parameters ϵ; (a) Plot of the output of $\gamma(p, a, b, G, \epsilon, n)$, where blue points are output $\gamma(p, a, b, G, \epsilon, n)$ when $\epsilon = 0.0102310371000$ and orange plot is when $\epsilon = 0.523569200000$; (b) Blue points are output $\gamma(p, a, b, G, \epsilon, n)$ when $\epsilon = 0.5235692000000$ and orange plot is when $\epsilon = 0.523569200005$.

of the proposed PRNG in Fig. 5.2 to check the sensitivity of the PRNG to the parameter (ϵ). Random-number analysis shows that the PRNG is able to generate cryptographically secure PRNs. Also, the method generates sequences of RNs with long periods and high entropy which is considered good for encryption purposes.

5.4 An S-box Generator

The proposed generator is built up using the PRNs generated by Algorithm 2 and a swap function to create NL in an initial S-box. Our generator has a low computational cost and in return constructs highly dynamic S-boxes. The main steps of the generator are given below.

- For an $m \times m$ S-box select parameters p, a, b for an EC $E_{p,a,b}$ with a generator G of a subgroup of largest order in $E_{p,a,b}$.
- Construct a set $A = \{(x_i, y_i) \in E_{p,a,b}, i = 1, 2, \ldots, \lambda\}$ s.t. $|A| = \lambda$. Define a mapping $\tilde{e}$ such that $\tilde{e} : A \times (0, 1) \to A_\epsilon$ s.t. $((x, y), \epsilon) \mapsto (\epsilon x, \epsilon y) \in A_\epsilon$, where $A_\epsilon = \{(\epsilon x, \epsilon y) \ \forall \ (x, y) \in A\}$ and $\epsilon \in (0, 1)$ is an arbitrary point.
- Choose a gain factor 10^δ to normalize points of the set A_ϵ. $\tilde{x} = \epsilon x \times 10^\delta$ and $\tilde{y} = \epsilon y \times 10^\delta$. Convert $\tilde{x}$ and $\tilde{y}$ into binary strings s_x and s_y respectively. Concatenate($\|$) s_x and s_y to generate a string s_z. Generate a binary bit string of length $m2^m$.
- Divide it from left to right into subsequences (s) each of length n. Now, convert these subsequences into decimal numbers $n \in [0, \ell]$, where $\ell = 2^m - 1$. Let $\mathcal{N} = (n_0, n_1, \ldots, n_\ell)$ be the sequence of integers obtained from the decimal form of the subsequences.
- Now for an initial S-box $\sigma(p, a, b, G, \epsilon, m)(i) = i, \ i = 0, 1, \ldots, \ell,$ generate S-boxes as follows:

$$\text{swap}(\sigma(i), \sigma(\mathcal{N}(i))), \ \text{for } i = 1, 2, \ldots, 2^m. \tag{5.7}$$

Algorithm 3 The S-box generation algorithm.

```
Require: Initial parameters $\epsilon$, $p$, $a$, $b$ and a base point $G$ of an $E_{p,a,b}$
Ensure: S-box $\sigma(p, a, b, G, \epsilon, m)$
Generate a string $\mathcal{S}$ s.t. $|\mathcal{S}| = m2^m$ using Algorithm 2.
Assemble a set $\mathcal{N} = [0, 2^m - 1]$ from $\mathcal{S}$ s.t. $|\mathcal{N}| = 2^m$
for $i = 1, 2, \ldots, 2^m$ do
    swap($\sigma(i), \sigma(\mathcal{N}(i))$)
end for
Then output is S-box $\sigma(p, a, b, G, \epsilon, m)$
```

Then the output of Eq. 5.7 is our final S-box $\sigma(p, a, b, G, \epsilon, m)$.

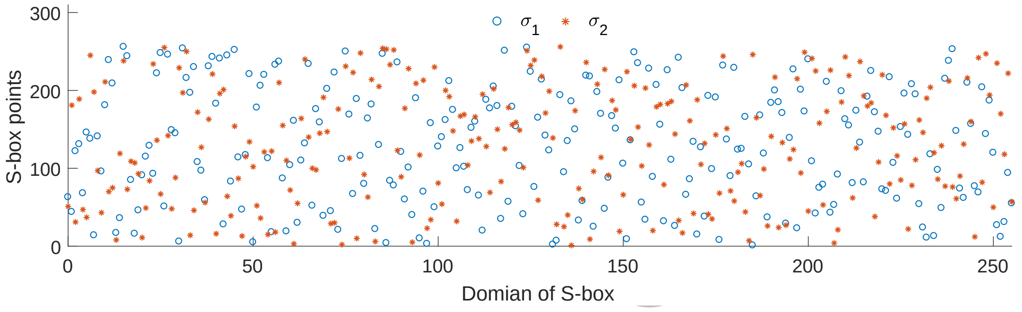


Figure 5.3: Sensitivity of Algorithm 3 to the parameter ϵ; (a) S-box $\sigma(p, a, b, G, \epsilon, m)$ when $\epsilon = 0.0001310463700$; (b) S-box $\sigma(p, a, b, G, \epsilon, m)$ when $\epsilon = 0.0001310463705$.

Table 5.3: The proposed S-box $\sigma(p, a, b, G, \epsilon_i, i = 1, 2, m)$ when $\epsilon_i = 0.01023103710, i = 1, 2$.

166	5	17	241	88	54	129	53	253	199	15	160	107	244	79	254
204	191	6	2	188	153	29	38	173	10	113	181	18	103	126	239
42	151	132	62	118	108	130	168	164	205	165	91	123	77	202	227
207	221	137	76	176	228	252	234	171	89	177	83	220	230	119	218
236	182	117	162	217	233	31	240	223	232	20	247	114	121	24	7
59	43	28	112	213	212	201	98	26	237	32	35	105	75	161	0
99	222	231	23	111	216	142	14	187	56	116	150	225	92	193	127
66	167	179	180	101	140	246	63	203	149	49	194	33	110	243	11
58	46	136	70	152	224	170	139	106	156	74	229	206	235	51	138
226	73	85	1	146	61	250	209	248	86	102	104	145	95	40	172
131	30	122	47	109	97	155	159	144	185	143	189	120	64	55	148
16	169	163	100	68	128	215	50	4	208	13	94	157	34	211	200
80	124	12	45	242	52	115	134	3	158	174	19	65	183	195	71
90	125	141	27	147	57	21	190	44	238	82	251	22	219	96	186
178	133	48	41	196	8	175	154	198	72	9	60	37	25	36	69
184	93	197	255	135	78	81	84	210	67	39	245	87	249	192	214

To evaluate how secure the suggested encryption method is, we first assess the developed S-boxes robustness. We specifically create dynamic S-boxes using random initialization parameters (ϵ) and p, a, b, G given in Table 5.8, then gauge each generated S-box strength by performing a series

Table 5.4: The proposed S-box $\sigma(p, a, b, G, \epsilon_i, i = 1, 2, m)$ when $\epsilon_i = 0.131046370, i = 1, 2$.

133	5	35	102	50	69	204	44	219	240	210	90	180	122	61	134
24	9	147	26	54	181	36	148	208	232	28	227	107	112	188	211
196	255	223	163	4	175	143	138	114	77	87	52	132	92	101	248
161	250	73	245	231	88	71	242	197	224	167	174	185	236	66	25
109	126	159	201	140	95	177	169	78	27	154	96	195	56	183	125
89	99	192	203	15	22	191	234	249	68	149	113	229	193	103	170
1	213	206	184	171	10	67	21	202	62	16	59	251	14	207	12
187	198	142	93	75	164	189	235	55	72	205	144	156	85	194	215
162	129	91	70	237	225	136	64	186	230	226	81	221	155	209	79
123	57	145	166	46	116	182	110	7	2	40	246	111	30	84	37
200	105	80	151	243	228	212	74	0	131	152	172	199	65	127	3
233	244	41	214	106	128	217	160	141	29	124	94	153	165	146	48
20	49	6	33	220	13	82	39	254	157	19	104	32	216	97	179
86	247	53	241	252	51	76	139	173	8	38	150	168	239	115	17
34	63	120	118	117	18	137	23	98	218	42	11	60	45	121	130
119	238	178	108	135	31	190	222	83	158	43	253	58	47	100	176

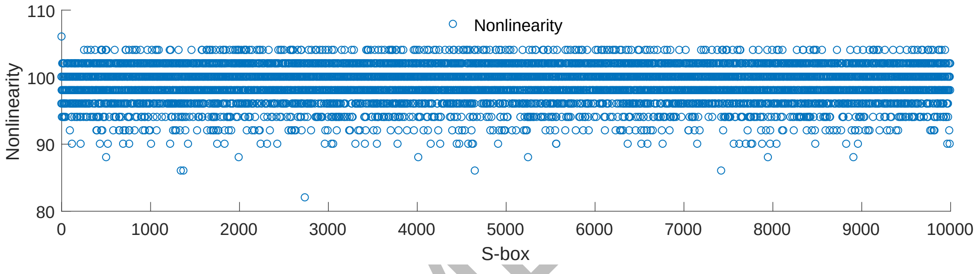


Figure 5.4: Plot of NL for 10000 S-boxes.

Table 5.5: Analysis of the S-boxes.

S-boxes	NL	LAP	DAP	AC	SAC		BIC	
					min.	max.	min.	max.	NL
S-box in Table 5.3	106	0.133	0.016	255	0.375	0.594	0.471	0.547	100
S-box in Table 5.4	106	0.141	0.016	255	0.391	0.594	0.475	0.547	100

of common S-box tests, such as DAP, NL, LAP, BIC and SAC [37, 39, 40, 174]. Table 5.3 and Table 5.4 display examples of S-boxes in the 16 × 16 format. Table 5.6 presents the outcomes of our S-boxes. The ideal values in Table 5.7 are extremely similar to all findings in Table 5.5 for the proposed S-boxes. Comparison of the NL of the S-boxes produced by the suggested scheme with S-boxes generated by other techniques in [14, 33, 45, 46, 55, 74, 101] are shown in Fig. 5.5. In Fig. 5.4, a plot of the NL of each of 10000 S-boxes with random keys (ϵ) is shown. The results show that the suggested generator constructs S-boxes with good NL than other approaches. Fig. 5.3 shows the sensitivity of Algorithm 3 to the initial random key (ϵ).

Table 5.6: Security comparison of the proposed S-boxes.

S-boxes	NL	LAP	DAP	AC	SAC		BIC		
					min.	max.	min.	max.	NL
Proposed	106	0.148	0.047	254	0.406	0.641	0.459	0.523	100
Ref. [43]	96	0.023	0.050	254	0.391	0.625	0.477	0.531	92
Ref. [44]	103	0.132	0.039	255	0.398	0.570	0.472	0.535	96
Ref. [47]	100	0.129	0.039	255	0.422	0.594	0.477	0.525	98
Ref. [55]	104	0.141	0.054	253	0.406	0.594	0.461	0.522	98
Ref. [48]	100	0.152	0.039	255	0.391	0.586	0.468	0.537	100
Ref. [49]	96	0.125	0.047	255	0.422	0.609	0.471	0.547	96
Ref. [58]	98	0.133	0.054	254	0.422	0.609	0.477	0.535	94
Ref. [59]	96	0.125	0.039	255	0.359	0.609	0.477	0.541	98
Ref. [61]	104	0.133	0.039	255	0.359	0.609	0.457	0.535	96
Ref. [62]	104	0.133	0.039	254	0.438	0.641	0.475	0.547	98
Ref. [23]	104	0.145	0.039	255	0.391	0.625	0.471	0.531	98
Ref. [56]	104	0.148	0.047	254	0.438	0.578	0.482	0.543	96
Ref. [50]	104	0.094	0.023	255	0.391	0.578	0.476	0.529	103

Table 5.7: Standard tests of S-box with optimal values.

Test	NL	LAP	DAP	SAC	BIC	AC
Optimal values	120	0	0	0.5	0.5	255

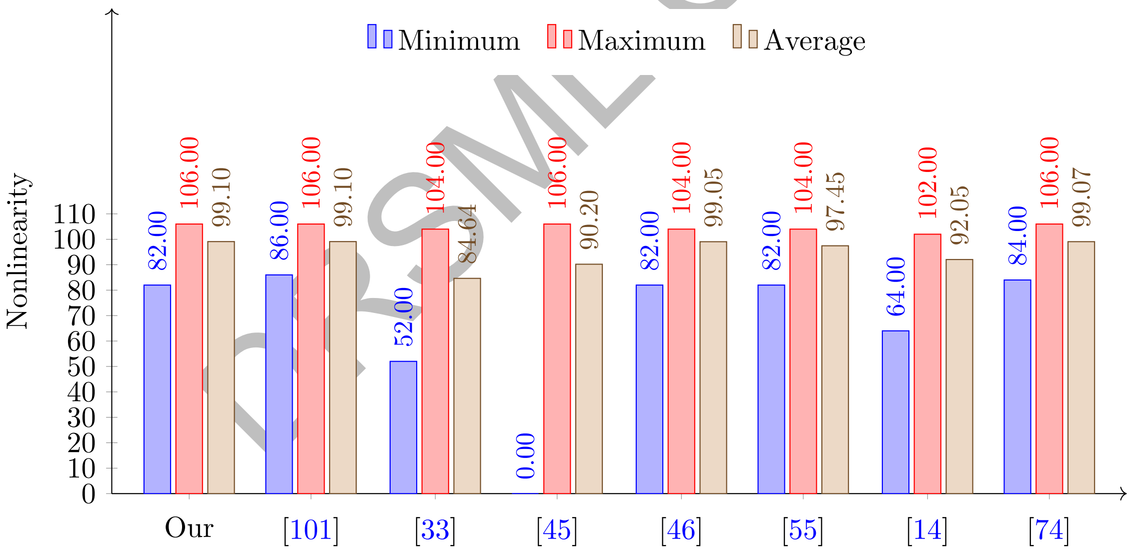


Figure 5.5: Comparison of NL of our 10000 S-boxes with other schemes.

Our method is highly key-dependent and very sensitive to the initial key as shown in Fig. 5.3.

5.5 Image Encryption

We develop an encryption scheme to encrypt greyscale images. The hash function is used to generate the hash value of 256 bits of plaintext. Let $H(I)$ be the hash value of the plain image I. To make the encryption algorithm plain image dependent, generate the parameters ϵ, ϵ_o and ϵ_1 from $H(I)$. A set of PRNs of size uv is generated using Algorithm 2 to mask I. Two S-boxes are constructed using Algorithm 3 to vanish the correlation between neighboring pixels of the masked image. The encryption method is described in the below steps.

(1) Define the initial values for S-boxes: Let H be the hash value of image I computed using the hash-256. Now split the hash bits into 8-bits sequences (d_i) to get a set $\{h_1, h_2, \ldots, h_{32}\}$, where each h_i is decimal form of corresponding d_i, $i \in [1, 32]$. Then construct the sets $m_1 = \{h_1, h_3, \ldots, h_{31}\}$; $m_2 = \{h_2, h_4, \ldots, h_{32}\}$. Define the parameters: $mean_1 = mean(H)$; $max_o = max\{h_1, h_3, \ldots, h_{31}\}$; $max_1 = max\{h_2, h_4, \ldots, h_{32}\}$; $sum_1 = sum(H)$; $\epsilon_o = mod((sum_1 * max_o + mean_1) \div 257 + 0.0315, 1)$; $\epsilon_1 = mod((sum_1 * max_1 + mean_1) \div 257 + 0.0415, 1)$.

(2) Compute $\epsilon = (\sum\{h_1, h_2, \ldots, h_{16}\}) \div 257, 1)$ and generate a set $A = \{(x_i, y_i) \in E_{p,a,b}, i = 1, 2, \ldots \lambda\}$ s.t. $|A| = \lambda$ using the generator of G of largest subgroup of the $E_{p,a,b}$, where λ is number of points needed to encrypt an image. Generate a set R_n of PRNs size uv using Algorithm 2.

(3) Diffusion in the plain image: Let $I_{u \times v}$ be the plain image. Convert I into a row matrix I_r, *e.g.*, $I_r = I(:)$. Diffused the plain image I using the set R_n to generate diffused image $\mathcal{D}$ such that $D(i) = mod(I_r(i) + R_n(i), 256)$ for $i = 1, 2, \ldots, uv$.

(4) Generate two permutations: Now for parameter ϵ_o and ϵ_1 generate S-boxes $\sigma(p, a, b, \epsilon_o, u)$ and $\sigma(p, a, b, \epsilon_1, v)$, respectively. Now, use these S-boxes as permutations ρ_u and ρ_v. Use ρ_u for the permutation of rows and ρ_v for the permutation of columns of the image $\mathcal{D}$.

(5) Confusion of neighboring pixels: Create confusion in neighboring pixels of $\mathscr{D}$ by shuffling rows and columns. Let $\mathscr{R}(i), i \in [1, u]$ denote i-th row of $\mathscr{D}$ from top to bottom. Then we get an image $\sigma(\mathscr{D})$ such that the i-th, $i \in [1, u]$ row of $\sigma(\mathscr{D})$ is $\mathscr{R}(\rho_u(i))$.

Let $C_i, i \in [1, v]$ denote i-th column of $\rho_u(\mathscr{D})$ from left to right. Then we get a ciphertext $\rho_v(\rho_u(\mathscr{D}))$ of I such that the i-th, $i \in [1, u]$ column of $\rho_v(\rho_u(\mathscr{D}))$ is $\mathscr{C}(\rho_v(i))$.

Table 5.8: Parameters of the EC used for security analysis.

p =69940347812394391875468529501296093702908827571801221847695030011446965607235415402011299983600983482216264978818695023069093987522038004143591906776633681509747788907966745303055743322600442179925371713036509994574118177396481789936479642771655689696225428834428166275862150589022692246011064194057224704113
a =254540141305309354872002345220102798143324237941067887654702627766676097499761435395162900011283366282692219135480739600802535751929683549114930961373081240812294949631461161662315573754289206382304287095511194957252880074854217317551416078372031379499850324280820711783333789587454461131935458323562353997 93
b =153769139591715540903454106075260152094514970066571984106747057250597665383489390864992500070015889667286295934270735287544400099620690663763769114023651133201241845145739210211853465452360240364085885203166044880610135548385899253094396922586207974472547723415666497028644023471081511014126861630762442227 39
G_x = 107529020820106531116239476583779674588952999486812383344848108522314181360378884682777854063270040290198636409114588514862895689739470662039161734326443953068378715097828346818112347413530129350343793355667756686249288553276628165885479798787081164402026185547070354731645388723606895922073115659639864370992
G_y = 673950489787130748633033981246613609963935964270916301083579078280706965642235901404500699177599394219230156484957857967235013503185713474116483862155776306852604761904528166430803777838503013989507785118357229649886816370900372642527046143191316237964031258753448968086808366390661525414749906393186572111 56

5.5.1 Decryption of process

The decryption of our encryption is possible and the original plain image can be obtained using the original initial keys. The decryption method is divided into the following steps.

- Set the original keys p, a, b, G and ϵ_i. $i = 1, 2, 3$.
- Compute the set of PRNs R_n and two permutations ρ_u and ρ_v. As the inverses of two permutations ρ_u and ρ_v exists so computes the inverses ρ_u^{-1} and ρ_v^{-1} of two permutations ρ_u and ρ_v respectively.

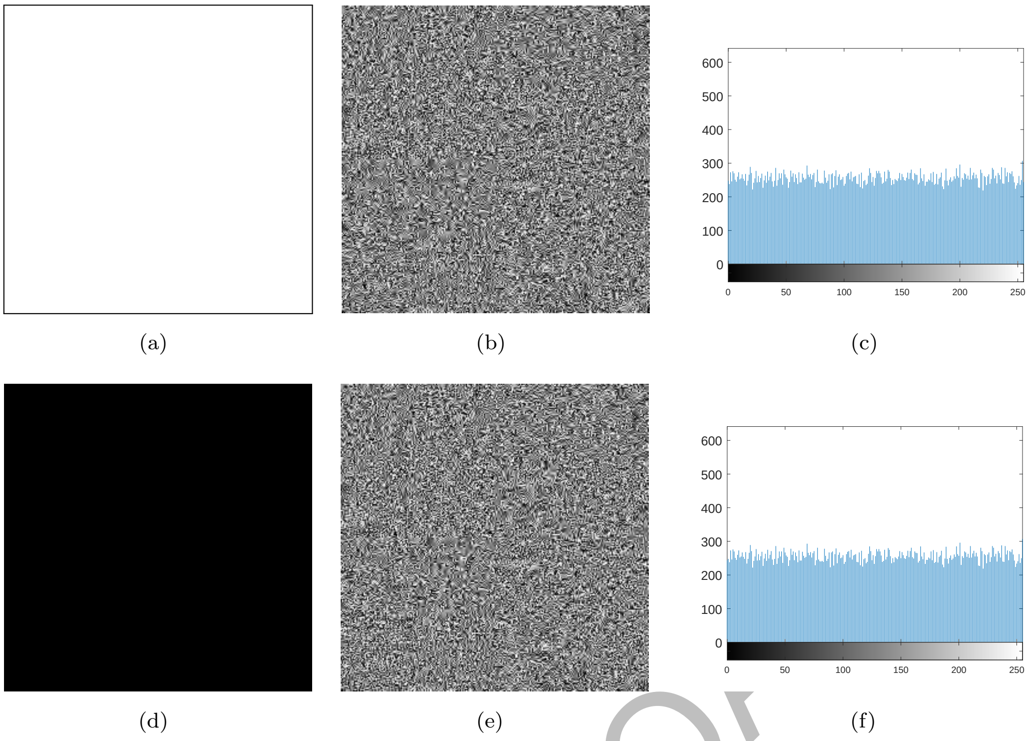


Figure 5.6: Encryption. (a) All-white; (b) Encrypted; (c) Histogram; (d) All-black; (e) Encrypted; (c) Histogram.

- Apply ρ_v^{-1} and ρ_u^{-1} to image $\mathcal{C}(\rho_v(i))$, respectively *e.g.*, $\rho_u^{-1}(\rho_v^{-1}(\mathcal{C}(\rho_v(i))))$ to get diffused iimage $\mathcal{D}$. Then get the plain image I as $I(i) = mod(\mathcal{D}(i) - R_n(i), 256)$.

5.6 Analysis of the Proposed Image Cryptosystem

We applied different cryptographic tests to analyze the security performance of the encryption algorithm. For convenience, we use an EC $E_{p,a,b}$ parameters given in Table 5.8 throughout this chapter. The security analysis of our encryption technique is given in the below subsections.

5.6.1 Differential Attacks

The parameters $\epsilon_i,\ i = 1, 2$ in Algorithms 2 and 3 are hash function dependent which makes the scheme secure against differential attacks. We applied this test to different images by changing pixels at different positions. For images of different sizes at database USC-SIPI, the numerical results of NPCR and UACI are given in Table 5.9 and Table5.10, respectively. The differential

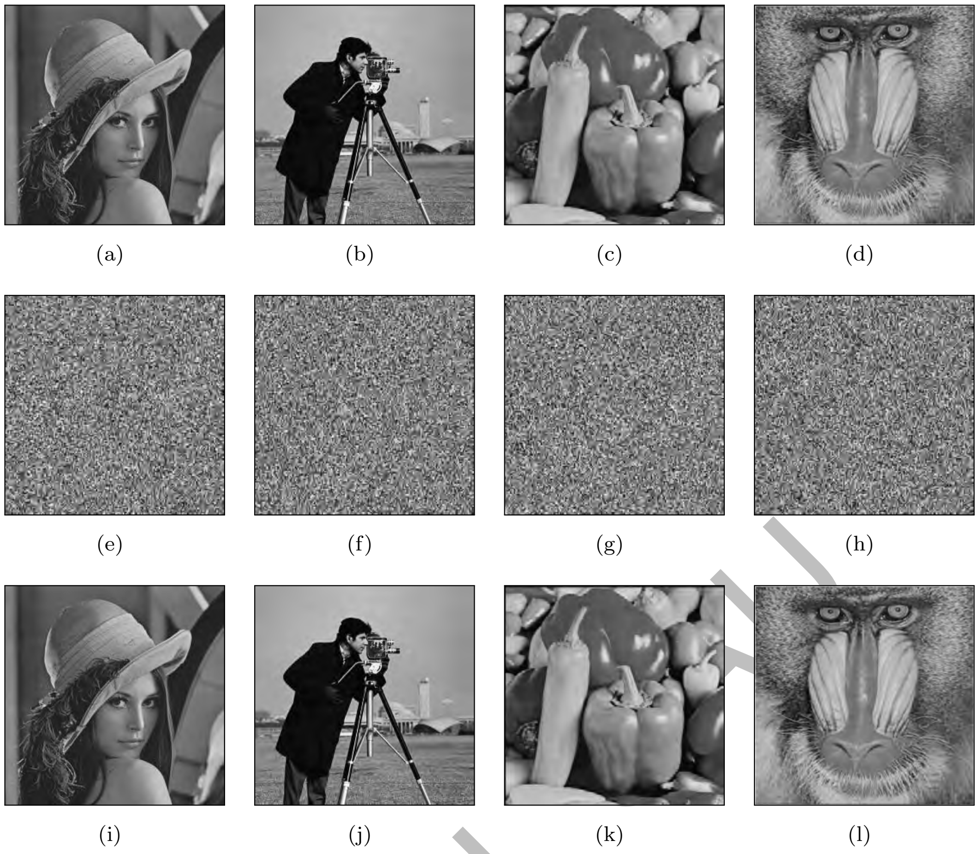


Figure 5.7: (a)-(d) Plain images Lena, Cameraman, Peppers and Mandrill of size 256 × 256; (e)-(h) Encrypted images; (i)-(l) Decrypted images.

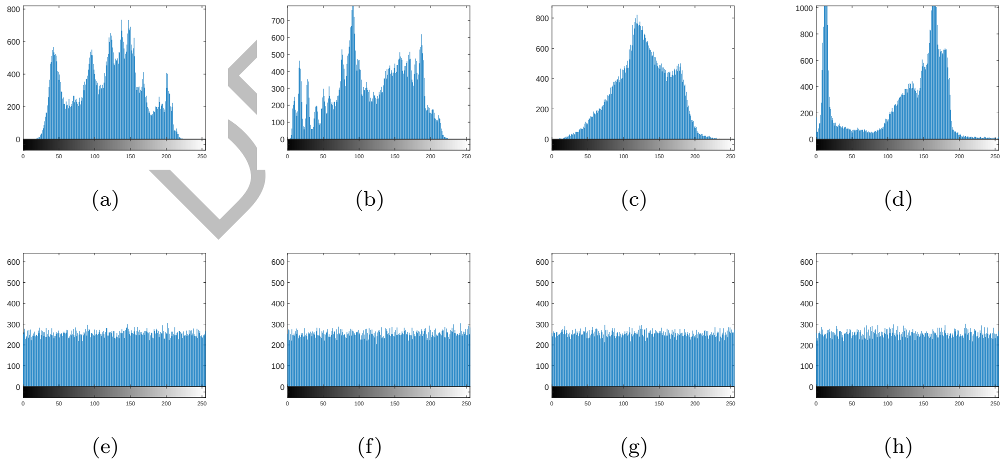


Figure 5.8: (a)-(d) Histograms of plain images in 5.7(a)-(d); (e)-(h) Histogram encrypted images in 5.7(e)-(h).

analysis indicates that the new encryption scheme is highly secure against differential attacks.

Table 5.9: NPCR comparison with existing schemes for different standard images.

Images	NPCR				
	[132]	[175]	[176]	[177]	proposed
256 × 256	$N_\epsilon^* \geq 99.5693$				
5.1.09	99.5575	99.6246	99.6064	99.6124	99.6136
5.1.10	99.5544	0.0092	99.6154	99.5972	99.6046
5.1.11	99.8123	99.6445	99.6244	99.5956	99.6098
5.1.12	99.6109	99.5972	99.5703	99.6017	99.6120
5.1.13	99.7421	99.6582	99.6109	99.6552	99.6077
5.1.14	99.6933	99.5987	99.6364	99.6002	99.6182
512 × 512	$N_\epsilon^* \geq 99.5893$				
5.2.08	99.6101	99.6216	99.5870	99.6220	99.6115
5.2.09	99.7025	99.6048	99.6260	99.6208	99.6087
5.2.10	99.6120	99.5861	99.6124	99.5968	99.6100
7.1.01	99.5190	99.6162	99.5992	99.6181	99.6107
7.1.02	99.7200	99.6025	99.6075	99.6140	99.6135
7.1.03	99.4072	99.5998	99.6079	99.6166	99.6069
7.1.04	99.6037	99.6033	99.5988	99.6227	99.6024
7.1.05	99.4572	99.6307	99.6170	99.5960	99.6048
7.1.06	99.5213	99.6105	99.6272	99.6212	99.6068
7.1.07	99.5007	99.6029	99.5931	99.6113	99.6073
7.1.08	99.6902	99.6120	99.6094	99.5914	99.6063
7.1.09	99.7181	99.6048	99.6162	99.6067	99.6058
7.1.10	99.5163	99.6212	99.6045	99.6056	99.6048
boat.512	99.7128	99.6067	99.6154	99.6021	99.6107
gray21.512	99.6120	99.6094	99.6022	99.6239	99.6065
ruler.512	99.3118	99.6113	99.6120	99.5930	99.6128
1024 × 1024	$N_\epsilon^* \geq 99.5994$				
5.3.01	99.6040	99.6116	99.5931	99.6100	99.6118
5.3.02	99.4789	99.6223	99.6128	99.6129	99.6091
7.2.01	99.7578	99.6042	99.6156	99.5964	99.6102
Pass	15/25	23/25	23/25	24/25	25/25
Mean	99.6010	95.6286	99.6088	99.6098	99.6091

5.6.2 Histogram Analysis

The graphs in Figs. 5.8(a)-(d) show histograms before the encryption method is applied, while the graphs in Figs. 5.8(e)-(h) show histograms after the encryption method is applied to the plain-image. The images show that the histograms are uniform. Furthermore, histograms of all-white (resp. all-black) image in Fig. 5.6(c) (resp. in Fig. 5.6(f)) are also uniform, which is highly recommended for a secure encryption scheme.

Table 5.10: UACI comparison with existing schemes for different standard images.

Images	UACI				
	[132]	[175]	[176]	[177]	proposed
256×256	$(U_\epsilon^{*-}, U_\epsilon^{*+}) = (33.3730, 33.5541)$				
5.1.09	33.5119	33.5980	33.5527	33.4425	33.4922
5.1.10	33.4263	33.5366	33.4381	0.0011	33.4801
5.1.11	33.4192	33.4398	33.4390	33.484	33.4413
5.1.12	33.2672	33.4228	33.4373	33.3609	33.5442
5.1.13	33.4252	33.4205	33.3488	33.3039	33.4538
5.1.14	33.2919	33.4696	33.5133	33.5008	33.4569
512×512	$(U_\epsilon^{*-}, U_\epsilon^{*+}) = (33.2824, 33.6447)$				
5.2.08	33.4509	33.4720	33.4377	33.5233	33.4454
5.2.09	33.4543	33.4921	33.4939	33.4834	33.4490
5.2.10	33.4365	33.4914	33.3888	33.4532	33.4621
7.1.01	33.4811	33.5212	33.5553	33.3369	33.4945
7.1.02	33.4762	33.4846	33.4342	33.4121	33.4636
7.1.03	33.5346	33.4647	33.4585	33.4970	33.4527
7.1.04	33.3450	33.5202	33.4830	33.4412	33.4718
7.1.05	33.5380	33.5400	33.4393	33.4753	33.4774
7.1.06	33.4766	33.5254	33.5634	33.4571	33.4712
7.1.07	33.4695	33.5205	33.5241	33.3844	33.4520
7.1.08	33.4258	33.5678	33.4251	33.3863	33.4499
7.1.09	33.4954	33.5223	33.4606	33.3879	33.4406
7.1.10	33.4389	33.4325	33.4119	33.4615	33.4535
boat.512	33.4693	33.5097	33.4993	33.4589	33.4437
gray21.512	33.4667	33.3930	33.4634	33.3857	33.4676
ruler.512	33.5154	33.5129	33.5090	33.5253	33.4740
1024×1024	$(U_\epsilon^{*-}, U_\epsilon^{*+}) = (33.4183, 33.5088)$				
5.3.01	33.4973	33.4532	33.4698	33.5380	33.4747
5.3.02	33.5109	33.4853	33.4820	33.4525	33.4659
7.2.01	33.4826	33.4965	33.4878	33.4348	33.4654
Pass	22/25	24/25	23/25	22/25	25/25
Mean	33.4523	33.4917	33.4687	32.1035	33.4657

5.6.3 Information Entropy

In Table 5.13, the entropy of all-white and all-black images is given. In Table 5.11, a comparison of the entropy for Lena$_{256\times256}$ image is given. A comparison of different images of size 512×512 for the entropy is given in Table 5.12. These comparisons predict that the proposed scheme is significantly better than other methods.

5.6.4 Correlation Analysis

Table 5.13 contains the correlation values for all-white and all-black images. In Table 5.11 correlation values of Lena$_{256\times256}$ image are compared. Furthermore, a comparison of different

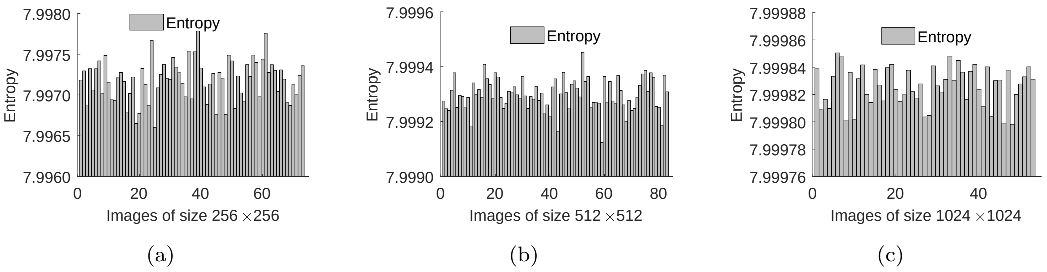


Figure 5.9: Entropy of different images;(a) 256 × 256; (b) 512 × 512; (c) 1024 × 1024.

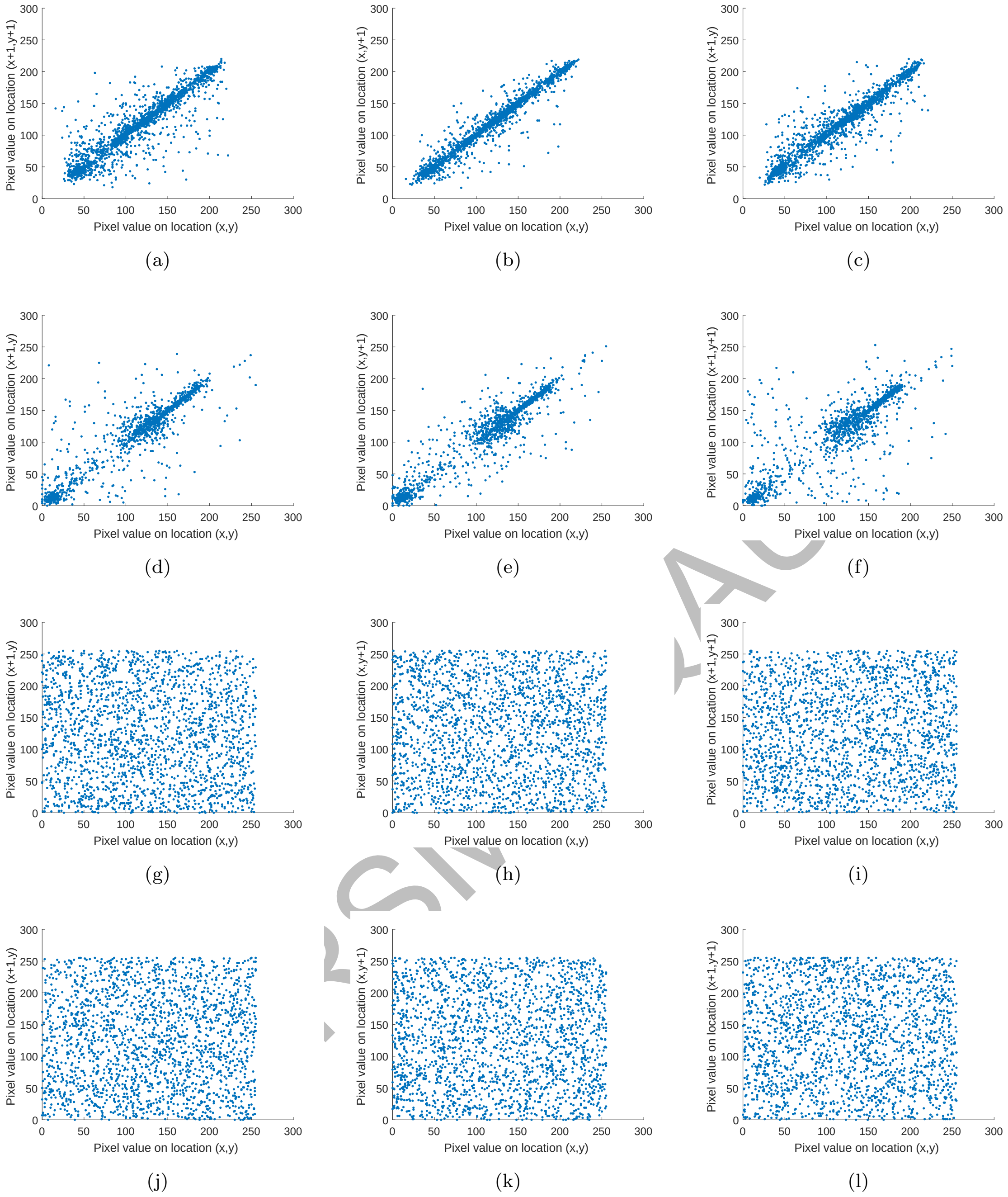
images of size 512 × 512 is given in Table 5.12, which is significantly better than the compared schemes. In Fig. 5.10, correlation between original images and encrypted images is given.

Table 5.11: Comparison for Lena$_{256\times256}$ grayscale image.

Algorithm	NPCR	UACI	Correlation			Entropy
			Horizontal	Diagonal	Vertical	
Proposed	99.658	33.368	-0.0048	0.0035	0.0022	7.9979
Ref. [105]	99.614	33.472	-0.0018	-0.0009	0.0011	7.9975
Ref. [106]	99.617	33.392	0.0029	-0.0003	0.0080	7.9965
Ref. [107]	99.625	33.423	0.0069	0.0075	0.0479	7.9972
Ref. [151]	99.609	33.451	-0.0018	0.0040	-0.0006	7.9976
Ref. [150]	99.621	33.413	0.0003	-0.0003	-0.00004	7.9977
Ref. [142]	99.614	33.546	-0.0016	-0.0026	0.0043	7.9970
Ref. [149]	99.617	33.452	-0.0008	-0.0011	-0.0014	7.9975
Ref. [152]	99.614	33.458	-0.0063	-0.0016	0.0065	7.9975
Ref. [143]	99.626	33.458	-0.0003	-0.0029	0.0016	7.9972
Ref. [144]	99.580	34.080	-0.0003	-0.0066	-0.0013	7.9967
Ref. [141]	99.606	33.463	0.0004	0.0051	0.0051	7.9974
Ref. [145]	99.594	33.505	0.0119	0.0011	0.0092	7.9970
Ref. [146]	99.599	33.456	-0.0029	0.0004	-0.0017	7.9971
Ref. [148]	99.605	33.419	0.0086	0.0009	0.0014	7.9970
Ref. [140]	99.695	33.384	0.0015	0.0057	0.0041	7.9962
Ref. [147]	99.612	33.457	-0.0052	-0.0003	0.0031	7.9973
Ref. [153]	99.610	32.790	0.0097	0.0178	0.0136	7.9971
Ref. [154]	99.618	33.415	0.0003	0.0004	0.0025	7.9972
Ref. [155]	99.620	33.510	-0.0230	-0.0034	0.0019	7.9974

5.6.5 Key Analysis

Keyspace: We use plain image hash value to generate initial keys in the suggested encryption method to encrypt the grayscale images. Furthermore, parameters of EC $E_{p,a,b}$ are also used as initial keys. We use hash 256 bits to generate the random parameters ϵ_i, $i = 1, 2, 3$ and a



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 5.10: Two adjacent pixels distribution in different directions. (a)-(c) Pixels of the Lena image; (d)-(f) Pixels of the Cameraman image; (g)-(i) Pixels of the encrypted image Lena; (k)-(l) Pixels of the encrypted image Cameraman.

prime p of 1024 bits as a key. So, the initial key length is much more than 2^{128} which indicates that our scheme withstands brute-force attacks.

Key Sensitivity: We assess the suggested technique's key sensitivity during the encryption process and the method of decryption. First, we encrypt a plain image using three random

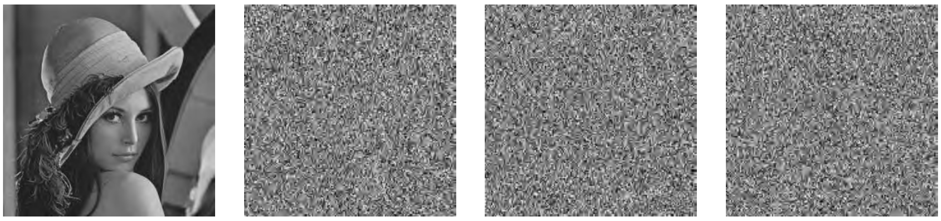
Table 5.12: Comparison for different images.

Algorithm	Algorithm	NPCR	UACI	Correlation			Entropy
				Horizontal	Diagonal	Vertical	
Lena(512)	Proposed	99.611	33.441	0.0013	-0.0007	0.0011	7.9993
	Ref. [178]	99.57	33.35	0.0033	-0.0002	-0.0040	7.9990
	Ref. [179]	99.64	34.14	0.0242	0.0245	0.0261	7.9965
	Ref. [180]	99.58	33.08	-0.0006	-0.0243	0.0048	7.9968
	Ref. [181]	99.63	33.48	0.0299	0.0030	0.0081	7.9992
	Ref. [182]	99.62	33.48	-0.0015	0.0107	-0.0063	7.9982
	Ref. [160]	58.98	24.60	-0.0004	-0.0378	0.0037	7.9993
	Ref. [183]	99.60	33.48	0.0005	0.0028	-0.0025	7.9993
Cameraman(512)	Proposed	99.606	33.458	-0.0001	0.0026	-0.0015	7.9993
	Ref. [178]	99.56	33.40	0.0026	-0.0015	0.0002	7.9991
	Ref. [180]	99.60	33.15	-	-	-	7.9904
	Ref. [181]	99.62	33.44	0.0102	-0.0202	0.0066	7.9991
	Ref. [182]	99.60	33.46	-0.0077	0.0083	0.0061	7.9983
	Ref. [160]	76.26	28.30	-0.0061	0.0166	0.0058	7.9992
	Ref. [183]	99.60	33.44	0.0025	-0.0016	-0.0034	7.9994
Peppers(512)	Proposed	99.608	33.498	-0.0019	0.0001	-0.0018	7.9993
	Ref. [178]	99.60	33.35	0.0068	0.0005	-0.0022	7.9992
	Ref.[179]	99.61	33.17	-	-	-	7.9950
	Ref. [180]	99.71	32.19	-	-	-	7.9961
	Ref. [181]	99.62	33.35	0.0131	0.0030	0.0022	7.9992
	Ref. [182]	99.62	33.46	0.0052	0.0215	0.0039	7.9978
	Ref. [160]	51.64	20.31	0.0049	0.0068	0.0099	7.9993
	Ref. [183]	99.61	33.50	0.0018	0.0018	0.0010	7.9993
Mandrill(512)	Proposed	99.618	33.418	0.0004	0.0001	-0.0002	7.9992
	Ref. [178]	99.63	33.17	0.0023	-0.0008	-0.0002	7.9986
	Ref. [179]	99.62	33.56	-	-	-	7.9962
	Ref. [180]	99.59	31.56	-	-	-	7.9971
	Ref. [181]	99.61	33.46	-0.0068	0.0036	-0.0082	7.9993
	Ref. [182]	99.62	33.51	-0.0009	0.0186	-0.0130	7.9992
	Ref. [160]	60.28	21.40	0.0124	-0.0215	-0.0018	7.9993
	Ref. [183]	99.62	33.42	0.0006	0.0046	0.0021	7.9993

Table 5.13: Security analysis.

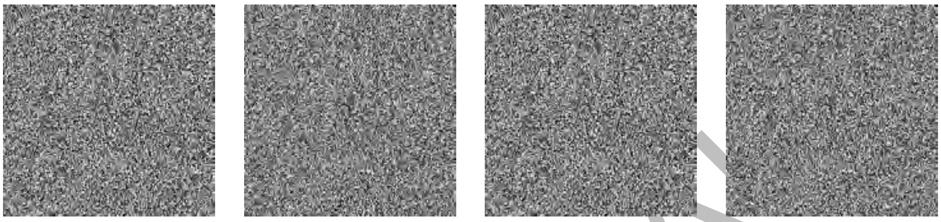
Image	NPCR	UACI	Correlation			Entropy
			Horizontal	Vertical	Diagonal	
All-white	99.605	33.44	-0.0038	-0.0023	0.0011	7.9976
All-black	99.601	33.49	-0.0090	-0.0096	0.0065	7.9973

parameters $\epsilon_i,\ i = 1, 2, 3$ with slight change (10^{-14}). Fig. 5.11 shows different encrypted images with different $\epsilon_i,\ i = 1, 2, 3$. The updated cipher-images are entirely different from the original cipher-images. Then, to retrieve a cipher-image, we employ three decryption keys with a negligible difference (10^{-14}) in each $\epsilon_i,\ i = 1, 2, 3$. Fig. 5.12 displays the recovered decrypted images along with the original decrypted image. All of the obtained cipher-images are false.



(a) (b) (c) (d)

Figure 5.11: Sensitivity to input keys in encryption; (a) Lena image; (b) Cipher-image with original keys; (c) Cipher-image with slight change (10^{-14}) in ϵ_1; (d) Cipher-image with slight change (10^{-14}) in ϵ_2; (e) Cipher-image with slight change (10^{-14}) in ϵ_3.



(a) (b) (c) (d)

Figure 5.12: Sensitivity to input keys in decryption of cipher-image; (a) Lena image; (b) Decryption with different keys; (c) Decryption with slight change (10^{-14}) in ϵ_1; (d) Decryption with slight change (10^{-14}) in ϵ_2; (e) Decryption with slight change (10^{-14}) in ϵ_3.

The encryption and decryption operations for grayscale images are thus sensitive to the initial keys.

5.6.6 Plaintext Attacks

Intruders frequently use particular types of images such as all-black or all-white to decrypt encryption methods. The attacker can identify detectable patterns in these images and utilize them to derive key-related data from the generated cipher images. The images are encrypted to demonstrate how the suggested encryption can remove any traces of chosen patterns from a plain image. In Figure 5.6(b) and Figure 5.6(e) all-white and all-black images are displayed after encryption, respectively. It is evident from Figure 5.6 that there is no identifiable patterns. The security analysis of both images is given in Table 5.13, and results predict that the images are completely randomized with no patterns. The attacker is unable to extract any meaningful information from the encrypted known-plain image or chosen-plain image for the proposed technique. These findings demonstrate that the suggested encryption technique is immune to differential attacks.

5.6.7 Noise Attacks

During the transmission, some data of an image may be lost due to the salt noise and Gaussian noise. An encryption scheme should have anti-noise ability. We add 0.01, 0.05 and 0.1 salt and pepper noises, respectively, to the decrypted image to address the impact of noise. The performance results of the proposed scheme against the noise attack are presented in Table 5.14. In Fig. 5.14 results are shown. The figure shows that we can still recognize the image when the noise intensity goes to 0.1, which predicts that our scheme is highly efficient against noise attacks.

Table 5.14: Occlusion attack results.

Occlusion ratio	NPCR	UACI	Correlation		
			Horizontal	Vertical	Diagonal
0	0	0	0.9247	0.9503	0.8809
1/64	0.0156	0.0046	0.8889	0.9154	0.8644
1/16	0.0622	0.0183	0.7599	0.7874	0.7328
1/4	0.2489	0.0721	0.4302	0.4384	0.3787

5.6.8 Data Loss Attacks

For an encryption scheme, it is mandatory to restore data lost in cipher-image to the greatest extent. We examine the impact on the decrypted image of replacing 1/64, 1/16, and 1/4 of the encrypted image with black pixels. The performance results against data loss attacks are shown in Table 5.15. The results for the data loss attack are given in Fig. 5.14.

Table 5.15: Noise attack results.

Noise ratio	NPCR	UACI	Correlation		
			Horizontal	Vertical	Diagonal
0	0	0	0.9247	0.9503	0.8809
0.01	0.0101	0.0030	0.8955	0.9331	0.8717
0.05	0.0496	0.0141	0.7962	0.8195	0.7741
0.10	0.1002	0.0294	0.6567	0.6754	0.6407

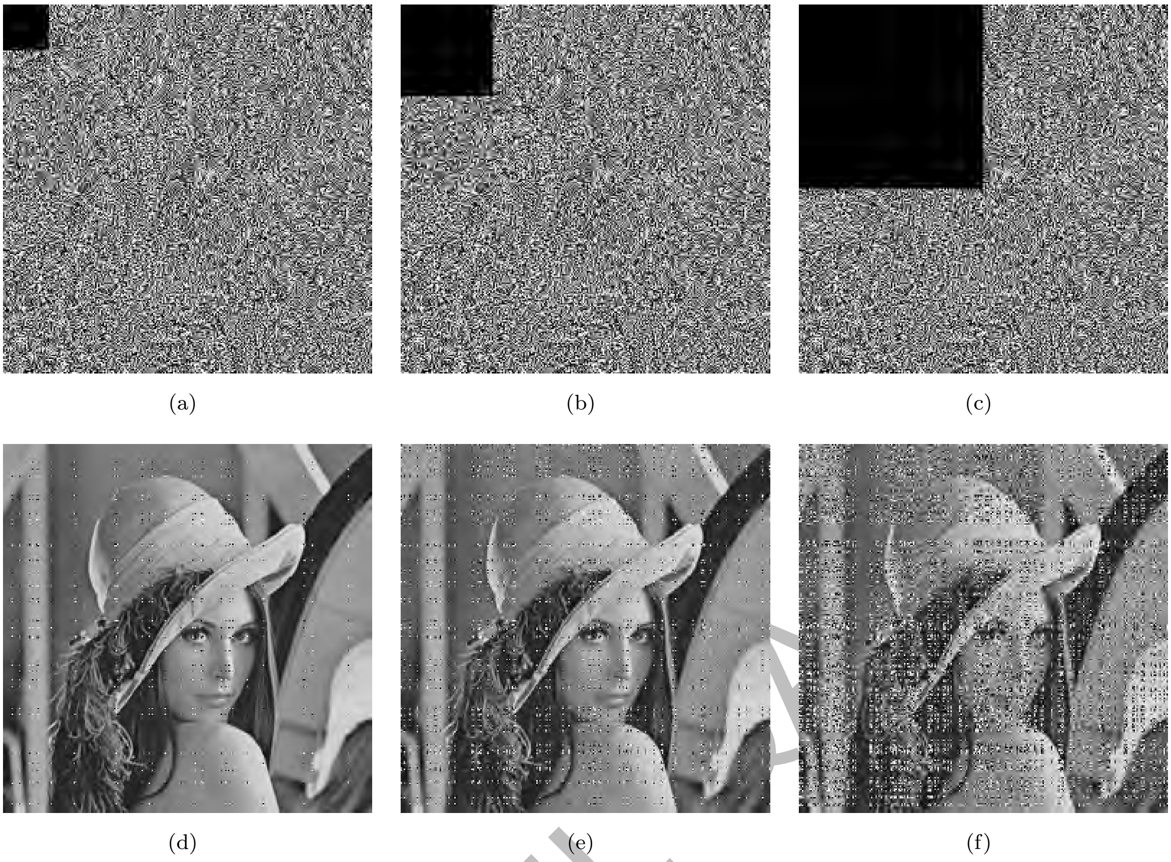


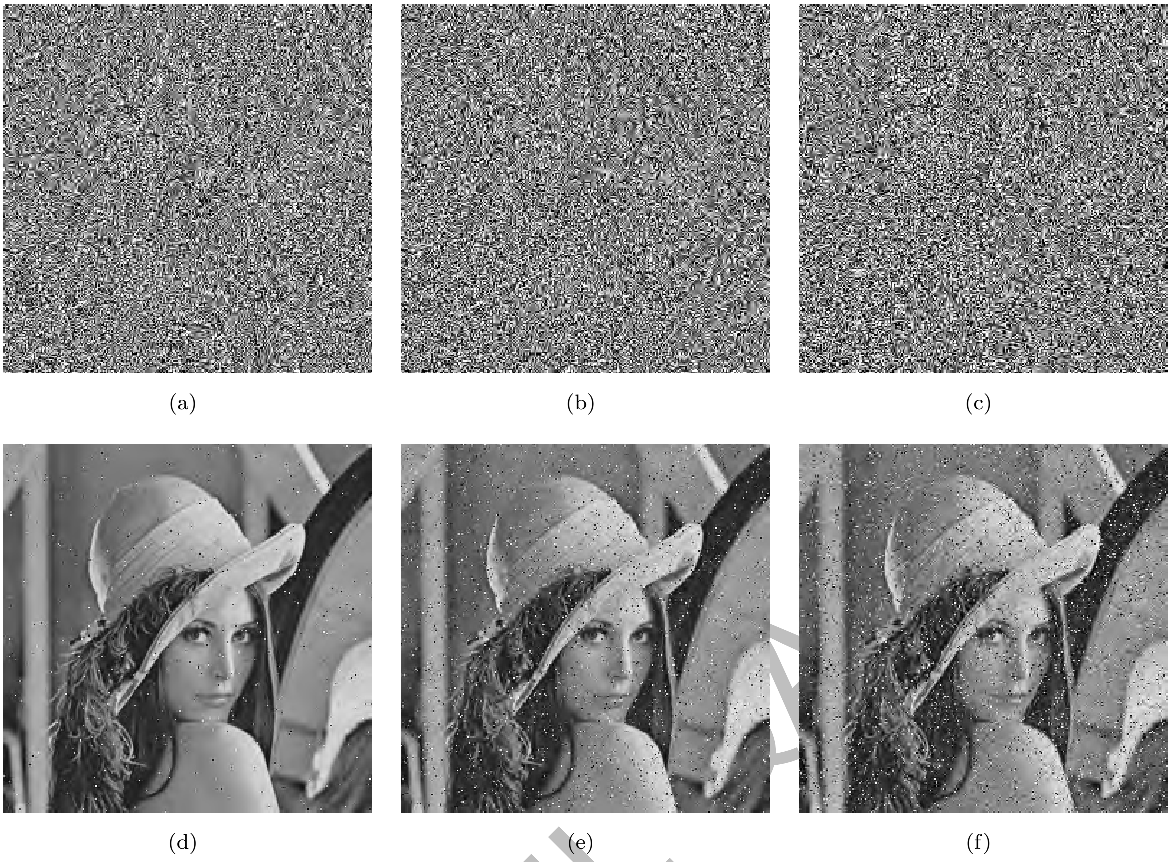
Figure 5.13: Lena encrypted images with data loss; (a) 1/64; (b) 1/16; (c) 1/4 . (d)-(f) Decrypted images of (a)-(c) data lost images, respectively.

Table 5.16: Run time analysis in seconds(s) for Lena images applied on same operating system.

Image size	References								
	Proposed	[107]	[14]	[16]	[160]	[132]	[117]	[161]	[184]
256 × 256	0.556	1.286	8.744	3.035	0.834	2.913	1.654	15.45	0.641
512 × 512	2.721	4.333	2731.450	3.082	2.849	11.26	6.353	76.777	1.311

5.6.9 Time Complexity Analysis

For the run time analysis, we used MATLAB R2017a, a PC with CPU i5-3.2 GHz, and 8 GB RAM running on Windows 8. We have conducted a speed test using grayscale images of different sizes 256 × 256 and 512 × 512. Our results show that encryption speed is fast when compared with schemes in [14, 16, 107, 117, 132, 160, 161] is given in Table 5.16.


(a) (b) (c)

(d) (e) (f)

Figure 5.14: Lena encrypted images with; (a) 0.01 noise; (b) 0.05 noise; (c) 0.1 noise. (d)-(f) Decrypted images of (a)-(c) noisy images, respectively.

5.7 Conclusion

In this chapter, ECs are utilized for the image encryption. For the encryption of plain images, a very complex structured PRNG is used to create diffusion, and a highly nonlinear S-box generator is used for the image. The encryption time is shown in Table 5.16. The encryption scheme is effective against cryptographic attacks as it is clear from Tables 5.11 and 5.12. The encryption scheme has the following advantages:

- ECs over large primes are used to generate PRNs sequences and their pattern are very complex to predict.
- An S-box generator is proposed to design dynamical S-boxes and the generated S-boxes are cryptographically strong when compared with S-boxes in [23, 43, 44, 47–50, 55, 55,

56, 58, 59, 61].

- The encryption method can withstand chosen-plaintext and ciphertext attacks since it is very sensitive to plaintexts.
- A robust EC-based encryption modal is proposed. The results show that the given scheme has better entropy analysis than schemes in [106, 107, 140, 142–144, 178–182]. The algorithm has low computation time as compared to methods in [14, 16, 107, 117, 132, 160, 161].

In this study, we have used ECs to make the algorithm more complex and efficient. It is difficult to find a fast way to encrypt data quickly. It seems that the proposal is a good one.

Chapter6

Summary and Conclusion

This chapter provides a summary of the research reported in this thesis and offers some suggestions for further research. The outcomes of this work are as follows:

(1) An efficient S-box generator based on finite ECs is developed that can generate S-boxes with good cryptographic strength and can be used for lightweight cryptography.

(2) A new parametrization of the resonant triads to develop a new algorithm for the enumeration of all resonant triads in a specific box is proposed. A new total order is defined using the proposed parametrization. The newly developed total order and the parametrization are employed to design an S-box generator.

(3) A novel image cryptosystem based on an EC over large primes and a couple map lattices is constructed to encrypt digital images for real-time encryption.

(4) ECs over large primes are used to design PRNG and S-box generators. The newly designed PRNG and S-box generators are used to propose image encryption methods with the desired cryptographic security.

Several ideas that have not been implemented yet came to mind while working on the current plans. Some of them are as follows:

(1) Developing new methods based on ECs for the generation PRNs and S-boxes with low time and space complexity.

(2) Increasing the cryptographic strengths of the proposed algorithms and optimizing them for color images.

(3) Designing a new software to assess the security of $n \times n$ S-boxes with $n \geq 9$.

(4) Introducing new encryption algorithms for the Internet of Things utilizing ECs.

(5) Investigating CML-systems and ECs-based S-box generators and integrating them with heuristic optimization techniques to increase the NL of S-boxes.

Bibliography

[1] L. C. Washington, *Elliptic curves: number theory and cryptography.* CRC press, 2008.

[2] V. S. Miller, “Use of elliptic curves in cryptography,” in *Conference on the theory and application of cryptographic techniques*, pp. 417–426, Springer, 1985.

[3] N. Koblitz, “Elliptic curve cryptosystems,” *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[4] M. Srinivas and S. Porika, “Encryption and decryption using elliptic curves for public key cryptosystems,” in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1300–1303, IEEE, 2017.

[5] H. Marzouqi, M. Al-Qutayri, and K. Salah, “An FPGA implementation of NIST 256 prime field ECC processor,” in *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*, pp. 493–496, IEEE, 2013.

[6] M. S. Hossain, E. Saeedi, and Y. Kong, “High-speed, area-efficient, FPGA-based elliptic curve cryptographic processor over nist binary fields,” in *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pp. 175–181, IEEE, 2015.

[7] M. Amara and A. Siad, “Elliptic curve cryptography and its applications,” in *International workshop on systems, signal processing and their applications, WOSSPA*, pp. 247–250, IEEE, 2011.

[8] Fips186-2., “Digital signature standard (DSS),” *National Institute of Standards and Technology (NIST)*, vol. 20, no. 13, p. 5, 2000.

[9] X. Ansi, “Public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ECDSA),” *ANSI X9 Catalog*, 1999.

[10] I. ISO, "IEC 15946-2: Information technology–security techniques–cryptographic techniques based on elliptic curves–part 1: Digital signatures," *International Organization for Standardization*, 2002.

[11] Microprocessor and M. Committee, "IEEE standard specifications for public-key cryptography," *IEEE Computer Society*, pp. 1–226, 2000.

[12] D. D. Wheeler, "Problems with chaotic cryptosystems," *Cryptologia*, vol. 13, no. 3, pp. 243–250, 1989.

[13] Y. Zheng and J. Jin, "A novel image encryption scheme based on hénon map and compound spatiotemporal chaos," *Multimedia Tools and Applications*, vol. 74, no. 18, pp. 7803–7820, 2015.

[14] U. Hayat and N. A. Azam, "A novel image encryption scheme based on an elliptic curve," *Signal Processing*, vol. 155, pp. 391–402, 2019.

[15] U. Hayat, I. Ullah, N. A. Azam, and S. Azhar, "A novel image encryption scheme based on elliptic curves over finite rings," *Entropy*, vol. 24, no. 5, p. 571, 2022.

[16] N. A. Azam, I. Ullah, and U. Hayat, "A fast and secure public-key image encryption scheme based on mordell elliptic curves," *Optics and Lasers in Engineering*, vol. 137, p. 106371, 2021.

[17] W. Li, X. Chang, A. Yan, and H. Zhang, "Asymmetric multiple image elliptic curve cryptography," *Optics and Lasers in Engineering*, vol. 136, p. 106319, 2021.

[18] 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*, pp. 705–714, Springer, 2016.

[19] S. Toughi, M. H. Fathi, and Y. A. Sekhavat, "An image encryption scheme based on elliptic curve pseudo random and advanced encryption system," *Signal processing*, vol. 141, pp. 217–227, 2017.

[20] T. Shah and D. Shah, “Construction of highly nonlinear S-boxes for degree 8 primitive irreducible polynomials over $\mathbb{Z}_2$,” *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 1219–1234, 2019.

[21] I. Hussain, T. Shah, M. A. Gondal, W. A. Khan, and H. Mahmood, “A group theoretic approach to construct cryptographically strong substitution boxes,” *Neural Computing and Applications*, vol. 23, no. 1, pp. 97–104, 2013.

[22] N. A. Azam, U. Hayat, and M. Ayub, “A substitution box generator, its analysis, and applications in image encryption,” *Signal Processing*, p. 108144, 2021.

[23] U. Hayat, N. A. Azam, and M. Asif, “A method of generating 8× 8 substitution boxes based on elliptic curves,” *Wireless Personal Communications*, vol. 101, no. 1, pp. 439–451, 2018.

[24] U. Hayat, N. A. Azam, H. R. Gallegos-Ruiz, S. Naz, and L. Batool, “A truly dynamic substitution box generator for block ciphers based on elliptic curves over finite rings,” *Arabian Journal for Science and Engineering*, pp. 1–13, 2021.

[25] N. A. Azam, U. Hayat, and I. Ullah, “An injective S-box design scheme over an ordered isomorphic elliptic curve and its characterization,” *Security and communication networks*, vol. 2018, 2018.

[26] J. H. Silverman, *The arithmetic of elliptic curves*, vol. 106. Springer, 2009.

[27] M. D. Bustamante and U. Hayat, “Complete classification of discrete resonant Rossby/drift wave triads on periodic domains,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 9, pp. 2402–2419, 2013.

[28] G. S. Kopp, “The arithmetic geometry of resonant Rossby wave triads,” *SIAM Journal on Applied Algebra and Geometry*, vol. 1, no. 1, pp. 352–373, 2017.

[29] N. Kishimoto and T. Yoneda, “A number theoretical observation of a resonant interaction of Rossby waves,” *Kodai Mathematical Journal*, vol. 40, no. 1, pp. 16–20, 2017.

[30] C. E. Shannon, “Communication theory of secrecy systems,” *The Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.

[31] H. Liu, A. Kadir, X. Sun, and Y. Li, "Chaos based adaptive double-image encryption scheme using hash function and S-boxes," *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 1391–1407, 2018.

[32] Ü. Çavuşoğlu, A. Zengin, I. Pehlivan, and S. Kaçar, "A novel approach for strong S-box generation algorithm design based on chaotic scaled Zhongtang system," *Nonlinear dynamics*, vol. 87, no. 2, pp. 1081–1094, 2017.

[33] M. F. Khan, A. Ahmed, and K. Saleem, "A novel cryptographic substitution box design using gaussian distribution," *IEEE Access*, vol. 7, pp. 15999–16007, 2019.

[34] A. H. Zahid, M. J. Arshad, and M. Ahmad, "A novel construction of efficient substitution-boxes using cubic fractional transformation," *Entropy*, vol. 21, no. 3, p. 245, 2019.

[35] A. M. Abbas, A. A. Alharbi, and S. Ibrahim, "A novel parallelizable chaotic image encryption scheme based on elliptic curves," *IEEE Access*, vol. 9, pp. 54978–54991, 2021.

[36] C. Carlet, Y. Crama, and P. L. Hammer, "Boolean functions for cryptography and error-correcting codes.," 2010.

[37] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 386–397, Springer, 1993.

[38] M. T. Sakallı, B. Aslan, E. Buluş, A. Ş. Mesut, F. Büyüksaraçoğlu, and O. Karaahmetoğlu, "On the algebraic expression of the AES S-box like S-boxes," in *International Conference on Networked Digital Technologies*, pp. 213–227, Springer, 2010.

[39] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[40] A. Webster and S. E. Tavares, "On the design of S-boxes," in *Conference on the theory and application of cryptographic techniques*, pp. 523–534, Springer, 1985.

[41] N. A. Azam, U. Hayat, and I. Ullah, "Efficient construction of a substitution box based on a mordell elliptic curve over a finite field," *Frontiers of Information Technology & Electronic Engineering*, vol. 20, no. 10, pp. 1378–1389, 2019.

[42] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT)*, vol. 1, no. 2, pp. 31–38, 2011.

[43] M. Khan, T. Shah, H. Mahmood, M. A. Gondal, and I. Hussain, "A novel technique for the construction of strong S-boxes based on chaotic lorenz systems," *Nonlinear Dynamics*, vol. 70, no. 3, pp. 2303–2311, 2012.

[44] G. Tang, X. Liao, and Y. Chen, "A novel method for designing S-boxes based on chaotic maps," *Chaos, Solitons & Fractals*, vol. 23, no. 2, pp. 413–419, 2005.

[45] D. Lambić, "A novel method of S-box design based on discrete chaotic map," *Nonlinear Dynamics*, vol. 87, no. 4, pp. 2407–2413, 2017.

[46] D. Lambić, "A novel method of S-box design based on chaotic map and composition method," *Chaos, Solitons & Fractals*, vol. 58, pp. 16–21, 2014.

[47] G. Jakimoski and L. Kocarev, "Chaos and cryptography: block encryption ciphers based on chaotic maps," *IEEE transactions on circuits and systems i: fundamental theory and applications*, vol. 48, no. 2, pp. 163–169, 2001.

[48] F. Özkaynak and A. B. Özer, "A method for designing strong S-boxes based on chaotic lorenz system," *Physics Letters A*, vol. 374, no. 36, pp. 3733–3738, 2010.

[49] B. B. Cassal-Quiroga and E. Campos-Cantón, "Generation of dynamical S-boxes for block ciphers via extended logistic map," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[50] G. Ivanov, N. Nikolov, and S. Nikova, "Cryptographically strong S-boxes generated by modified immune algorithm," in *International Conference on Cryptography and Information Security in the Balkans*, pp. 31–42, Springer, 2015.

[51] I. Hussain, T. Shah, H. Mahmood, and M. A. Gondal, "A projective general linear group based algorithm for the construction of substitution box for block ciphers," *Neural Computing and Applications*, vol. 22, no. 6, pp. 1085–1093, 2013.

[52] N. A. Azam, "A novel fuzzy encryption technique based on multiple right translated AES gray S-boxes and phase embedding," *Security and Communication Networks*, vol. 2017, 2017.

[53] A. Razaq, H. Alolaiyan, M. Ahmad, M. A. Yousaf, U. Shuaib, W. Aslam, and M. Alawida, "A novel method for generation of strong substitution-boxes based on coset graphs and symmetric groups," *IEEE Access*, vol. 8, pp. 75473–75490, 2020.

[54] M. A. Yousaf, H. Alolaiyan, M. Ahmad, M. Dilbar, and A. Razaq, "Comparison of pre and post-action of a finite abelian group over certain nonlinear schemes," *IEEE Access*, vol. 8, pp. 39781–39792, 2020.

[55] A. H. Zahid and M. J. Arshad, "An innovative design of substitution-boxes using cubic polynomial mapping," *Symmetry*, vol. 11, no. 3, p. 437, 2019.

[56] A. Razaq, A. Yousaf, U. Shuaib, N. Siddiqui, A. Ullah, and A. Waheed, "A novel construction of substitution box involving coset diagram and a bijective map," *Security and Communication Networks*, vol. 2017, 2017.

[57] M. Khan and N. A. Azam, "Right translated AES gray S-boxes," *Security and Communication Networks*, vol. 8, no. 9, pp. 1627–1635, 2015.

[58] A. A. Abd el Latif, B. Abd-el Atty, M. Amin, and A. M. Iliyasu, "Quantum-inspired cascaded discrete-time quantum walks with induced chaotic dynamics and cryptographic applications," *Scientific reports*, vol. 10, no. 1, pp. 1–16, 2020.

[59] A. A. Abd EL-Latif, B. Abd-El-Atty, and S. E. Venegas-Andraca, "A novel image steganography technique based on quantum substitution boxes," *Optics & Laser Technology*, vol. 116, pp. 92–102, 2019.

[60] G. Chen, "A novel heuristic method for obtaining S-boxes," *Chaos, Solitons & Fractals*, vol. 36, no. 4, pp. 1028–1036, 2008.

[61] M. B. Farah, A. Farah, and T. Farah, "An image encryption scheme based on a new hybrid chaotic map and optimized substitution box," *Nonlinear Dynamics*, pp. 1–24, 2019.

[62] T. Farah, R. Rhouma, and S. Belghith, "A novel method for designing S-box based on chaotic map and teaching–learning-based optimization," *Nonlinear Dynamics*, vol. 88, no. 2, pp. 1059–1074, 2017.

[63] D. Bhattacharya, N. Bansal, A. Banerjee, and D. RoyChowdhury, "A near optimal S-box design," in *International Conference on Information Systems Security*, pp. 77–90, Springer, 2007.

[64] M. Ahmad, I. A. Khaja, A. Baz, H. Alhakami, and W. Alhakami, "Particle swarm optimization based highly nonlinear substitution-boxes generation for security applications," *IEEE Access*, vol. 8, pp. 116132–116147, 2020.

[65] M. Ahmad, M. N. Doja, and M. S. Beg, "ABC optimization based construction of strong substitution-boxes," *Wireless Personal Communications*, vol. 101, no. 3, pp. 1715–1729, 2018.

[66] S. Farwa, T. Shah, and L. Idrees, "A highly nonlinear S-box based on a fractional linear transformation," *SpringerPlus*, vol. 5, no. 1, pp. 1–12, 2016.

[67] L. C. N. Chew and E. S. Ismail, "S-box construction based on linear fractional transformation and permutation function," *Symmetry*, vol. 12, no. 5, p. 826, 2020.

[68] M. Ahmad, E. Al-Solami, A. M. Alghamdi, and M. A. Yousaf, "Bijective S-boxes method using improved chaotic map-based heuristic search and algebraic group structures," *IEEE Access*, vol. 8, pp. 110397–110411, 2020.

[69] C. Wang, D. Wang, G. Xu, and D. He, "Efficient privacy-preserving user authentication scheme with forward secrecy for industry 4.0," *Science China Information Sciences*, vol. 65, no. 1, pp. 1–15, 2022.

[70] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE transactions on industrial informatics*, vol. 15, no. 9, pp. 4957–4968, 2019.

[71] S. Qiu, D. Wang, G. Xu, and S. Kumari, “Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices,” *IEEE Transactions on Dependable and Secure Computing*, 2020.

[72] I. Ullah, N. A. Azam, and U. Hayat, “Efficient and secure substitution box and random number generators over mordell elliptic curves,” *Journal of Information Security and Applications*, vol. 56, p. 102619, 2021.

[73] B. Premananda, K. Nikhil, and N. Jain, “MEC S-box based PRESENT lightweight cipher for enhanced security and throughput,” in *2020 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, pp. 212–217, IEEE, 2020.

[74] S. Ibrahim and A. M. Abbas, “Efficient key-dependent dynamic S-boxes based on permutated elliptic curves,” *Information Sciences*, vol. 558, pp. 246–264, 2021.

[75] K. McKay, L. Bassham, M. S. Turan, and N. Mouha, “Report on lightweight cryptography (NISTIR8114),” *National Institute of Standards and Technology (NIST)*, 2017.

[76] V. A. Thakor, M. A. Razzaque, and M. R. Khandaker, “Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities,” *IEEE Access*, 2021.

[77] Federal Information Processing Standards Publication 180-2, “Announcing the secure hash standard,” *US DoC/NIST, 2002.*

[78] J. Kim and R. C.-W. Phan, “Advanced differential-style cryptanalysis of the NSA’s skipjack block cipher,” *Cryptologia*, vol. 33, no. 3, pp. 246–270, 2009.

[79] W. Horton and A. Hasegawa, “Quasi-two-dimensional dynamics of plasmas and fluids,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 4, no. 2, pp. 227–251, 1994.

[80] K. L. Harper, B. E. Quinn, S. V. Nazarenko, and M. D. Bustamante, *Zonostrophy and Other Quadratic Invariants in Drift and Quasi-Geostrophic Wave Turbulence*. Cambridge University Press, 2019.

[81] B. Galperin and P. L. Read, *Zonal jets: Phenomenology, genesis, and physics*. Cambridge University Press, 2019.

[82] V. Petoukhov, S. Rahmstorf, S. Petri, and H. J. Schellnhuber, "Quasiresonant amplification of planetary waves and recent Northern Hemisphere weather extremes," *Proceedings of the National Academy of Sciences*, vol. 110, no. 14, pp. 5336–5341, 2013.

[83] D. Coumou, J. Lehmann, and J. Beckmann, "The weakening summer circulation in the Northern Hemisphere mid-latitudes," *Science*, vol. 348, no. 6232, pp. 324–327, 2015.

[84] E. Kartashova and A. Kartashov, "Laminated wave turbulence: generic algorithms I," *International Journal of Modern Physics C*, vol. 17, no. 11, pp. 1579–1596, 2006.

[85] E. Kartashova and A. Kartashov, "Laminated wave turbulence: generic algorithms II," *Comm. Comp. Phys*, vol. 2, no. 4, pp. 783–794, 2007.

[86] J. M. Lewis, "Carl-Gustaf Rossby: A study in mentorship," *Bulletin of the American Meteorological Society*, vol. 73, no. 9, pp. 1425–1439, 1992.

[87] N. A. Phillips, "Carl-Gustaf Rossby: His times, personality, and actions," *Bulletin of the American Meteorological Society*, vol. 79, no. 6, pp. 1097–1112, 1998.

[88] C.-G. Rossby, "Relation between variations in the intensity of the zonal circulation of the atmosphere and the displacements of the semi-permanent centers of action," *Journal of Marine Research*, vol. 2, pp. 38–55, 1939.

[89] J. G. Charney, "The dynamics of long waves in a baroclinic westerly current," *Journal of Atmospheric Sciences*, vol. 4, no. 5, pp. 136–162, 1947.

[90] J. G. Charney, "On the scale of atmospheric motions," *Geofysiske Publikasjoner*, vol. 17, pp. 3–17, 1948.

[91] A. Hasegawa and K. Mima, "Stationary spectrum of strong turbulence in magnetized nonuniform plasma," *Physical Review Letters*, vol. 39, no. 4, p. 205, 1977.

[92] A. Hasegawa and K. Mima, "Pseudo-three-dimensional turbulence in magnetized nonuniform plasma," *The Physics of Fluids*, vol. 21, no. 1, pp. 87–92, 1978.

[93] P. Lynch, "Resonant Rossby wave triads and the swinging spring," *Bulletin of the American Meteorological Society*, vol. 84, no. 5, pp. 605–616, 2003.

[94] J. Pedlosky, *Geophysical fluid dynamics*, vol. 710. Springer, 1987.

[95] D. Cox, "Primes of the fom $z^2 + ny^2$: Fermat, class field theory, and complex multiplication, John Wiley &," *Sons: New York*, 1989.

[96] U. Hayat, S. Amanullah, S. Walsh, M. Abdullah, and M. D. Bustamante, "Discrete resonant Rossby/drift wave triads: Explicit parameterisations and a fast direct numerical search algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 79, p. 104896, 2019.

[97] T. Johansson, "Analysis and design of modern stream ciphers," in *IMA International Conference on Cryptography and Coding*, pp. 66–66, Springer, 2003.

[98] P. C. Van Oorschot, A. J. Menezes, and S. A. Vanstone, *Handbook of applied cryptography.* CRC press, 1996.

[99] A. H. Zahid, M. Ahmad, A. Alkhayyat, M. T. Hassan, A. Manzoor, and A. K. Farhan, "Efficient dynamic S-box generation using linear trigonometric transformation for security applications," *IEEE Access*, vol. 9, pp. 98460–98475, 2021.

[100] I. Hussain, A. Anees, T. A. Al-Maadeed, and M. T. Mustafa, "Construction of S-box based on chaotic map and algebraic structures," *Symmetry*, vol. 11, no. 3, p. 351, 2019.

[101] G. Murtaza, N. A. Azam, and U. Hayat, "Designing an efficient and highly dynamic substitution-box generator for block ciphers based on finite elliptic curves," *Security and Communication Networks*, vol. 2021, 2021.

[102] I. Ullah, U. Hayat, and M. D. Bustamante, "Image encryption using elliptic curves and Rossby/drift wave triads," *Entropy*, vol. 22, no. 4, p. 454, 2020.

[103] Z. Li, "Application research of digital image technology in graphic design," *Journal of Visual Communication and Image Representation*, vol. 65, p. 102689, 2019.

[104] K. Kobayashi, "(7) artificial intelligence technology and medical image processing," *No Shinkei geka. Neurological Surgery*, vol. 48, no. 7, pp. 654–664, 2020.

[105] M. Li, M. Wang, H. Fan, K. An, and G. Liu, "A novel plaintext-related chaotic image encryption scheme with no additional plaintext information," *Chaos, Solitons & Fractals*, vol. 158, p. 111989, 2022.

[106] C. Zou, X. Wang, and H. Li, "Image encryption algorithm with matrix semi-tensor product," *Nonlinear Dynamics*, vol. 105, no. 1, pp. 859–876, 2021.

[107] K. M. Hosny, S. T. Kamal, M. M. Darwish, and G. A. Papakostas, "New image encryption algorithm using hyperchaotic system and fibonacci q-matrix," *Electronics*, vol. 10, no. 9, p. 1066, 2021.

[108] S. Weng, Y. Shi, W. Hong, and Y. Yao, "Dynamic improved pixel value ordering reversible data hiding," *Information Sciences*, vol. 489, pp. 136–154, 2019.

[109] C. Wang, X. Wang, Z. Xia, and C. Zhang, "Ternary radial harmonic fourier moments based robust stereo image zero-watermarking algorithm," *Information Sciences*, vol. 470, pp. 109–120, 2019.

[110] K. D. Patel and S. Belani, "Image encryption using different techniques: A review," *International Journal of Emerging Technology and Advanced Engineering*, vol. 1, no. 1, pp. 30–34, 2011.

[111] G. Ye, H. Zhao, and H. Chai, "Chaotic image encryption algorithm using wave-line permutation and block diffusion," *Nonlinear Dynamics*, vol. 83, no. 4, pp. 2067–2077, 2016.

[112] M. Lahdir, H. Hamiche, S. Kassim, M. Tahanout, K. Kemih, and S.-A. Addouche, "A novel robust compression-encryption of images based on spiht coding and fractional-order discrete-time chaotic system," *Optics & Laser Technology*, vol. 109, pp. 534–546, 2019.

[113] L. Gong, H. Luo, R. Wu, and N. Zhou, "New 4D chaotic system with hidden attractors and self-excited attractors and its application in image encryption based on RNG," *Physica A: Statistical Mechanics and its Applications*, vol. 591, p. 126793, 2022.

[114] L. Gong, R. Wu, and N. Zhou, “A new 4D chaotic system with coexisting hidden chaotic attractors,” *International Journal of Bifurcation and Chaos*, vol. 30, no. 10, p. 2050142, 2020.

[115] Z. Huang and N. Zhou, “Image encryption scheme based on discrete cosine stockwell transform and DNA-level modulus diffusion,” *Optics & Laser Technology*, vol. 149, p. 107879, 2022.

[116] S. Li, X. Mou, Y. Cai, Z. Ji, and J. Zhang, “On the security of a chaotic encryption scheme: problems with computerized chaos in finite computing precision,” *Computer physics communications*, vol. 153, no. 1, pp. 52–58, 2003.

[117] Y. Zhou, L. Bao, and C. P. Chen, “A new 1d chaotic system for image encryption,” *Signal processing*, vol. 97, pp. 172–182, 2014.

[118] G. Ye, “Image scrambling encryption algorithm of pixel bit based on chaos map,” *Pattern Recognition Letters*, vol. 31, no. 5, pp. 347–354, 2010.

[119] K. Kaneko, “Pattern dynamics in spatiotemporal chaos: Pattern selection, diffusion of defect and pattern competition intermettency,” *Physica D: Nonlinear Phenomena*, vol. 34, no. 1-2, pp. 1–41, 1989.

[120] S. Wang, W. Liu, H. Lu, J. Kuang, and G. Hu, “Periodicity of chaotic trajectories in realizations of finite computer precisions and its implication in chaos communications,” *International journal of modern physics B*, vol. 18, no. 17n19, pp. 2617–2622, 2004.

[121] X. Wang, J. Yang, and N. Guan, “High-sensitivity image encryption algorithm with random cross diffusion based on dynamically random coupled map lattice model,” *Chaos, Solitons & Fractals*, vol. 143, p. 110582, 2021.

[122] Y.-Q. Zhang, Y. He, P. Li, and X.-Y. Wang, “A new color image encryption scheme based on 2DNLCML system and genetic operations,” *Optics and Lasers in Engineering*, vol. 128, p. 106040, 2020.

[123] Y.-Q. Zhang and X.-Y. Wang, “A new image encryption algorithm based on non-adjacent coupled map lattices,” *Applied Soft Computing*, vol. 26, pp. 10–20, 2015.

[124] G. Ye, M. Liu, and M. Wu, "Double image encryption algorithm based on compressive sensing and elliptic curve," *Alexandria Engineering Journal*, vol. 61, no. 9, pp. 6785–6795, 2022.

[125] A. H. Brahim, A. A. Pacha, and N. H. Said, "Image encryption based on compressive sensing and chaos systems," *Optics & Laser Technology*, vol. 132, p. 106489, 2020.

[126] G. Ye, C. Pan, Y. Dong, Y. Shi, and X. Huang, "Image encryption and hiding algorithm based on compressive sensing and random numbers insertion," *Signal processing*, vol. 172, p. 107563, 2020.

[127] H. Liu and Y. Liu, "Cryptanalyzing an image encryption scheme based on hybrid chaotic system and cyclic elliptic curve," *Optics & Laser Technology*, vol. 56, pp. 15–19, 2014.

[128] Z. E. Dawahdeh, S. N. Yaakob, and R. R. bin Othman, "A new image encryption technique combining elliptic curve cryptosystem with hill cipher," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 3, pp. 349–355, 2018.

[129] M. Benssalah, Y. Rhaskali, and K. Drouiche, "An efficient image encryption scheme for tmis based on elliptic curve integrated encryption and linear cryptography," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 2081–2107, 2021.

[130] L. Liu, Y. Zhang, and X. Wang, "A novel method for constructing the S-box based on spatiotemporal chaotic dynamics," *Applied Sciences*, vol. 8, no. 12, p. 2650, 2018.

[131] G. Liu, "Designing S-box based on 4D-4wing hyperchaotic system," *3D Research*, vol. 8, no. 1, pp. 1–9, 2017.

[132] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," *Optics and Lasers in Engineering*, vol. 78, pp. 17–25, 2016.

[133] L. Teng and X. Wang, "A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive," *Optics Communications*, vol. 285, no. 20, pp. 4048–4054, 2012.

[134] Z. Hua, F. Jin, B. Xu, and H. Huang, "2D logistic-sine-coupling map for image encryption," *Signal Processing*, vol. 149, pp. 148–161, 2018.

[135] Z. Hua, B. Xu, F. Jin, and H. Huang, "Image encryption using Josephus problem and filtering diffusion," *IEEE Access*, vol. 7, pp. 8660–8674, 2019.

[136] A. Belazi, S. Kharbech, M. N. Aslam, M. Talha, W. Xiang, A. M. Iliyasu, and A. A. Abd El-Latif, "Improved sine-tangent chaotic map with application in medical images encryption," *Journal of Information Security and Applications*, vol. 66, p. 103131, 2022.

[137] X. Wang, Y. Hou, S. Wang, and R. Li, "A new image encryption algorithm based on CML and DNA sequence," *IEEE access*, vol. 6, pp. 62272–62285, 2018.

[138] A. A. Abd El-Latif and X. Niu, "A hybrid chaotic system and cyclic elliptic curve for image encryption," *AEU-International Journal of Electronics and Communications*, vol. 67, no. 2, pp. 136–143, 2013.

[139] X. Liu, X. Tong, Z. Wang, and M. Zhang, "Uniform non-degeneracy discrete chaotic system and its application in image encryption," *Nonlinear Dynamics*, vol. 108, no. 1, pp. 653–682, 2022.

[140] A. Pourjabbar Kari, A. Habibizad Navin, A. M. Bidgoli, and M. Mirnia, "A novel multi-image cryptosystem based on weighted plain images and using combined chaotic maps," *Multimedia Systems*, vol. 27, no. 5, pp. 907–925, 2021.

[141] X. Wang and M. Zhang, "An image encryption algorithm based on new chaos and diffusion values of a truth table," *Information Sciences*, vol. 579, pp. 128–149, 2021.

[142] X. Yan, X. Wang, and Y. Xian, "Chaotic image encryption algorithm based on arithmetic sequence scrambling model and DNA encoding operation," *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 10949–10983, 2021.

[143] Y. Niu and X. Zhang, "A novel plaintext-related image encryption scheme based on chaotic system and pixel permutation," *IEEE Access*, vol. 8, pp. 22082–22093, 2020.

[144] B. Abd-El-Atty, A. El-Latif, A. Ahmed, and S. E. Venegas-Andraca, "An encryption protocol for NEQR images based on one-particle quantum walks on a circle," *Quantum Information Processing*, vol. 18, no. 9, pp. 1–26, 2019.

[145] S. Amina and F. K. Mohamed, "An efficient and secure chaotic cipher algorithm for image content preservation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 60, pp. 12–32, 2018.

[146] X. Wang, X. Zhu, and Y. Zhang, "An image encryption algorithm based on Josephus traversing and mixed chaotic map," *IEEE Access*, vol. 6, pp. 23733–23746, 2018.

[147] H.-M. Yuan, Y. Liu, T. Lin, T. Hu, and L.-H. Gong, "A new parallel image cryptosystem based on 5D hyper-chaotic system," *Signal Processing: Image Communication*, vol. 52, pp. 87–96, 2017.

[148] X. Wang, S. Wang, Y. Zhang, and K. Guo, "A novel image encryption algorithm based on chaotic shuffling method," *Information Security Journal: A Global Perspective*, vol. 26, no. 1, pp. 7–16, 2017.

[149] Y. Chen, S. Xie, and J. Zhang, "A hybrid domain image encryption algorithm based on improved henon map," *Entropy*, vol. 24, no. 2, p. 287, 2022.

[150] X. Wang and Y. Li, "Chaotic image encryption algorithm based on hybrid multi-objective particle swarm optimization and DNA sequence," *Optics and Lasers in Engineering*, vol. 137, p. 106393, 2021.

[151] Y. Luo, S. Tang, J. Liu, L. Cao, and S. Qiu, "Image encryption scheme by combining the hyper-chaotic system with quantum coding," *Optics and Lasers in Engineering*, vol. 124, p. 105836, 2020.

[152] N. Iqbal and M. Hanif, "An efficient grayscale image encryption scheme based on variable length row-column swapping operations," *Multimedia Tools and Applications*, vol. 80, no. 30, pp. 36305–36339, 2021.

[153] A. U. Rehman, J. S. Khan, J. Ahmad, and S. O. Hwang, "A new image encryption scheme based on dynamic S-boxes and chaotic maps," *3D Research*, vol. 7, no. 1, pp. 1–8, 2016.

[154] Y. Liu, J. Wang, J. Fan, and L. Gong, "Image encryption algorithm based on chaotic system and dynamic S-boxes composed of DNA sequences," *Multimedia Tools and Applications*, vol. 75, no. 8, pp. 4363–4382, 2016.

[155] Y. Luo, L. Cao, S. Qiu, H. Lin, J. Harkin, and J. Liu, “A chaotic map-control-based and the plain image-related cryptosystem,” *Nonlinear Dynamics*, vol. 83, no. 4, pp. 2293–2310, 2016.

[156] X. Wang and H. Sun, “A chaotic image encryption algorithm based on improved Joseph traversal and cyclic shift function,” *Optics & Laser Technology*, vol. 122, p. 105854, 2020.

[157] P. Singh, A. Yadav, and K. Singh, “Phase image encryption in the fractional hartley domain using arnold transform and singular value decomposition,” *Optics and Lasers in Engineering*, vol. 91, pp. 187–195, 2017.

[158] K. Xuejing and G. Zihui, “A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system,” *Signal Processing: Image Communication*, vol. 80, p. 115670, 2020.

[159] A. A. Karawia and Y. A. Elmasry, “New encryption algorithm using bit-level permutation and non-invertible chaotic map,” *IEEE Access*, vol. 9, pp. 101357–101368, 2021.

[160] E. Z. Zefreh, “An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions,” *Multimedia Tools and Applications*, vol. 79, no. 33, pp. 24993–25022, 2020.

[161] T. J. Satish, M. N. S. Theja, G. G. Kumar, and V. Thanikaiselvan, “Image encryption using integer wavelet transform, logistic map and xor encryption,” in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 704–709, IEEE, 2018.

[162] A. Girdhar, H. Kapur, and V. Kumar, “A novel grayscale image encryption approach based on chaotic maps and image blocks,” *Applied Physics B*, vol. 127, no. 3, pp. 1–12, 2021.

[163] B. Mondal and T. Mandal, “A light weight secure image encryption scheme based on chaos & DNA computing,” *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 4, pp. 499–504, 2017.

[164] B. Mondal, N. Biswas, and T. Mandal, "A comparative study on cryptographic image scrambling.," in *RICE*, pp. 261–268, 2017.

[165] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE transactions on information forensics and security*, vol. 11, no. 2, pp. 235–246, 2015.

[166] J.-x. Chen, Z.-l. Zhu, and H. Yu, "A fast chaos-based symmetric image cryptosystem with an improved diffusion scheme," *Optik*, vol. 125, no. 11, pp. 2472–2478, 2014.

[167] K. Biswas, V. Muthukkumarasamy, and K. Singh, "An encryption scheme using chaotic map and genetic operations for wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2801–2809, 2014.

[168] H. Huang, S. Yang, and R. Ye, "Efficient symmetric image encryption by using a novel 2D chaotic system," *IET Image Processing*, vol. 14, no. 6, pp. 1157–1163, 2020.

[169] G. Ye, K. Jiao, and X. Huang, "Quantum logistic image encryption algorithm based on SHA-3 and RSA," *Nonlinear Dynamics*, vol. 104, no. 3, pp. 2807–2827, 2021.

[170] S. Zhu and C. Zhu, "Image encryption algorithm with an avalanche effect based on a six-dimensional discrete chaotic system," *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 29119–29142, 2018.

[171] M. A. M. Khan, N. A. Azam, U. Hayat, and H. Kamarulhaili, "A novel deterministic substitution box generator over elliptic curves for real-time applications," *Journal of King Saud University-Computer and Information Sciences*, 2022.

[172] T. Haider, N. A. Azam, and U. Hayat, "A novel image encryption scheme based on ABC algorithm and elliptic curves," *Arabian Journal for Science and Engineering*, pp. 1–21, 2022.

[173] S. Adhikari and S. Karforma, "A novel image encryption method for e-governance application using elliptic curve pseudo random number and chaotic random number sequence," *Multimedia Tools and Applications*, vol. 81, no. 1, pp. 759–784, 2022.

[174] W. Meier and O. Staffelbach, “Nonlinearity criteria for cryptographic functions,” in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 549–562, Springer, 1989.

[175] W. Liu, K. Sun, and C. Zhu, “A fast image encryption algorithm based on chaotic map,” *Optics and Lasers in Engineering*, vol. 84, pp. 26–36, 2016.

[176] Z. Hua, Y. Zhou, C.-M. Pun, and C. P. Chen, “2D sine logistic modulation map for image encryption,” *Information Sciences*, vol. 297, pp. 80–94, 2015.

[177] X. Wang, Q. Wang, and Y. Zhang, “A fast image algorithm based on rows and columns switch,” *Nonlinear Dynamics*, vol. 79, no. 2, pp. 1141–1149, 2015.

[178] M. Ghazvini, M. Mirzadi, and N. Parvar, “A modified method for image encryption based on chaotic map and genetic algorithm,” *Multimedia Tools and Applications*, vol. 79, no. 37, pp. 26927–26950, 2020.

[179] S. Noshadian, A. Ebrahimzade, and S. J. Kazemitabar, “Optimizing chaos based image encryption,” *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25569–25590, 2018.

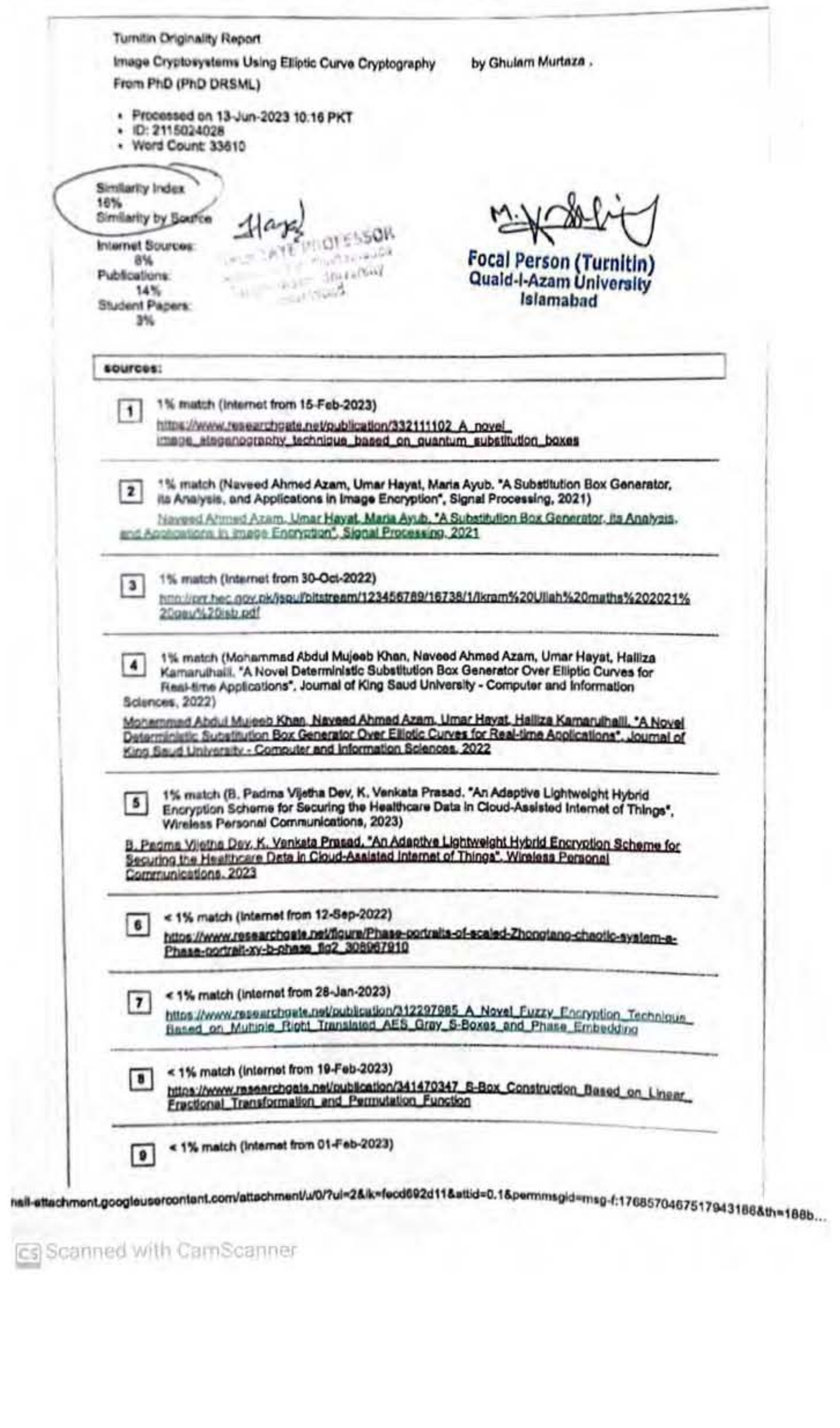
[180] S. Mozaffari, “Parallel image encryption with bitplane decomposition and genetic algorithm,” *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25799–25819, 2018.

[181] X. Chai, Z. Gan, K. Yuan, Y. Chen, and X. Liu, “A novel image encryption scheme based on DNA sequence operations and chaotic systems,” *Neural Computing and Applications*, vol. 31, no. 1, pp. 219–237, 2019.

[182] K. Zhan, D. Wei, J. Shi, and J. Yu, “Cross-utilizing hyperchaotic and DNA sequences for image encryption,” *Journal of Electronic Imaging*, vol. 26, no. 1, p. 013021, 2017.

[183] W. Zhou, X. Wang, M. Wang, and D. Li, “A new combination chaotic system and its application in a new bit-level image encryption scheme,” *Optics and Lasers in Engineering*, vol. 149, p. 106782, 2022.

[184] N. A. Azam, G. Murtaza, and U. Hayat, “A novel image encryption scheme based on elliptic curves and coupled map lattices,” *Optik*, p. 170517, 2023.



Turnitin Originality Report

Image Cryptosystems Using Elliptic Curve Cryptography by Ghulam Murtaza .

From PhD (PhD DRSML)

- Processed on 13-Jun-2023 10:16 PKT
- ID: 2115024028
- Word Count: 33610

Similarity Index
16%
Similarity by Source

Internet Sources:
8%
Publications:
14%
Student Papers:
3%

[illegible] PROFESSOR

Focal Person (Turnitin)
Quaid-I-Azam University
Islamabad

sources:

1 1% match (Internet from 15-Feb-2023)
https://www.researchgate.net/publication/332111102_A_novel_image_steganography_technique_based_on_quantum_substitution_boxes

2 1% match (Naveed Ahmed Azam, Umar Hayat, Maria Ayub. "A Substitution Box Generator, its Analysis, and Applications in Image Encryption", Signal Processing, 2021)
Naveed Ahmed Azam, Umar Hayat, Maria Ayub. "A Substitution Box Generator, its Analysis, and Applications in Image Encryption", Signal Processing, 2021

3 1% match (Internet from 30-Oct-2022)
http://prr.hec.gov.pk/jspui/bitstream/123456789/16738/1/Ikram%20Ullah%20maths%202021%20qau%20isb.pdf

4 1% match (Mohammad Abdul Mujeeb Khan, Naveed Ahmed Azam, Umar Hayat, Haliza Kamarulhaili. "A Novel Deterministic Substitution Box Generator Over Elliptic Curves for Real-time Applications", Journal of King Saud University - Computer and Information Sciences, 2022)
Mohammad Abdul Mujeeb Khan, Naveed Ahmed Azam, Umar Hayat, Haliza Kamarulhaili. "A Novel Deterministic Substitution Box Generator Over Elliptic Curves for Real-time Applications", Journal of King Saud University - Computer and Information Sciences, 2022

5 1% match (B. Padma Vijetha Dev, K. Venkata Prasad. "An Adaptive Lightweight Hybrid Encryption Scheme for Securing the Healthcare Data in Cloud-Assisted Internet of Things", Wireless Personal Communications, 2023)
B. Padma Vijetha Dev, K. Venkata Prasad. "An Adaptive Lightweight Hybrid Encryption Scheme for Securing the Healthcare Data in Cloud-Assisted Internet of Things", Wireless Personal Communications, 2023

6 < 1% match (Internet from 12-Sep-2022)
https://www.researchgate.net/figure/Phase-portraits-of-scaled-Zhongtang-chaotic-system-a-Phase-portrait-xy-b-phase_fig2_308967910

7 < 1% match (Internet from 28-Jan-2023)
https://www.researchgate.net/publication/312297985_A_Novel_Fuzzy_Encryption_Technique_Based_on_Multiple_Right_Translated_AES_Gray_S-Boxes_and_Phase_Embedding

8 < 1% match (Internet from 19-Feb-2023)
https://www.researchgate.net/publication/341470347_S-Box_Construction_Based_on_Linear_Fractional_Transformation_and_Permutation_Function

9 < 1% match (Internet from 01-Feb-2023)