CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Dynamic Key Dependent S-box for Symmetric Cryptosystem and its Application in Image Encryption

by

Kiran Tabassum

A thesis submitted in partial fulfillment for the degree of Master of Philosophy

in the

Faculty of Computing
Department of Mathematics

2022

Dedicated to my beloved parents and siblings

CERTIFICATE OF APPROVAL

Dynamic Key Dependent S-box for Symmetric Cryptosystem and its Application in Image Encryption

by

Kiran Tabassum

(MMT203009)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Ayesha Rafiq	IST, Islamabad
(b)	Internal Examiner	Dr. Muhammad Afzal	CUST, Islamabad
(c)	Supervisor	Dr. Rashid Ali	CUST, Islamabad

Supervisor Name
Dr. Rashid Ali
November, 2022

Dr. Muhammad Sagheer
Head
Dept. of Mathematics
November, 2022

Dr. M. Abdul Qadir
Dean
Faculty of Computing
November, 2022

Author's Declaration

I, **Kiran Tabassum** hereby state that my M.Phil thesis titled "**Dynamic Key Dependent S-box for Symmetric Cryptosystem and its Application in Image Encryption**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my M.Phil Degree.

(Kiran Tabassum)

Registration No: MMT203009

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**Dynamic Key Dependent S-box for Symmetric Cryptosystem and its Application in Image Encryption**" 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 Capital University of Science and Technology 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 after award of M.Phil Degree, the University reserves the right to withdraw/revoke my M.Phil 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 work.

(Kiran Tabassum)

Registration No: MMT203009

Acknowledgement

First of all I would like to thanks **Allah Almighty**, The Beneficent, The Merciful, The most Kind, who gave me the strength to complete this work. Secondly, I am Thankful to my parents and siblings for always being supportive in my whole educational career. Further, I would like to thank those who helped me in achieving this goal and completing this task. I am profoundly grateful to my generous supervisor **Dr. Rashid Ali** for his encouragement. He was always there whenever I found any problem. I am grateful to him for helping and guidance. His suggestions and guidance helped me in completing the thesis. I would like to thank all of my friends for motivating me during my degree program. Mostly, I would like to thank Phd scholar **Tahir Ali** also helped me a lot and guided me whenever I needed it.

(Kiran Tabassum)

Abstract

An S-box is a main component of many symmetric cryptographic algorithms. The most important characteristics of an S-box is to add non-linearity in the corresponding encryption scheme. The design of S-boxes is to increase the confusion ability of the cipher. Some researchers purposed different S-boxes based on different techniques. In this thesis, first an S-box based on simple mathematics operation is reviewed. The S-box is constructed using MATLAB, then the generated S-box is used for the image encryption scheme. In the scheme a compound chaotic map namely, the tent-logistic map is used. The tent logistic map is used for the generation of chaotic sequence. The secret key used in the construction of the S-box is further extended for using it as the initial conditions of the compound chaotic map. Results and security analysis demonstrate the good performance of the algorithm as a secure communication method for images.

Contents

Author's Declaration iv

Plagiarism Undertaking v

Acknowledgement vi

Abstract vii

List of Figures x

List of Tables xi

Abbreviations xiii

Symbols xiv

1 Introduction 1
- 1.1 S-boxes in Cryptography . . . 2
- 1.2 Image Encryption in Cryptography . . . 3
- 1.3 Literature Review . . . 4
- 1.4 Thesis Contribution . . . 6
- 1.5 Thesis Layout . . . 6

2 Preliminaries 8
- 2.1 Cryptography . . . 8
  - 2.1.1 Symmetric Key Cryptography . . . 9
  - 2.1.2 Public Key Cryptography . . . 9
- 2.2 Mathematical Background . . . 10
- 2.3 Boolean Function . . . 14
  - 2.3.1 Application of Boolean Function in S-boxes . . . 17
- 2.4 Substitution Boxes . . . 21
  - 2.4.1 Characteristics of S-box . . . 22
  - 2.4.2 Classification of S-boxes . . . 22
  - 2.4.3 General Properties of S-Box . . . 23
- 2.5 Key Dependent S-Box . . . 30

2.6 Software Tools in the Analysis of S-box 30

3 Construction of Dynamic Key Dependent S-box for Symmetric Cryptosystem 32

3.1 Operations Used for Generating S-box 32
3.2 Proposed Algorithm for Generating S-box 34
3.3 Finding and Result 38
3.3.1 Nonlinearity 38
3.3.2 Strict Avalanche Criteria 39
3.3.3 Differential Probability 40
3.3.4 Bit Independence Criterion 41
3.3.5 Linear Probability 42
3.3.6 Difference Percentage 42
3.3.7 Other Properties of S-box 43
3.4 Comparison Between Newly Generated S-boxes of Proposed Method 44

4 Application of S-box in Cryptography 47

4.1 Chaos Theory 47
4.1.1 Properties of Chaotic System 48
4.1.2 Lyapunov Exponent 49
4.1.3 Bifurcation Diagram 49
4.1.4 Logistic Map 50
4.1.5 Tent Map 51
4.1.6 The Tent Logistic system 52
4.2 Proposed Cryptosystem for Image Encryption 53
4.2.1 Key management 53
4.3 Results and Discussion 58
4.3.1 Performance Analysis 59
4.3.1.1 Histogram Test 59
4.3.1.2 Key Space Analysis 61
4.3.1.3 Correlation Coefficient 61
4.3.1.4 Key Sensitivity 61
4.3.1.5 Entropy Test 62

5 Conclusion 64

A Key Dependent S-boxes 65

Bibliography 76

List of Figures

3.1 Working flow of proposed substitution method 37

4.1 Bifurcation Diagram of Logistic map 50
4.2 Bifurcation Diagram of Tent map 51
4.3 Bifurcation Diagram of Tent-Logistic map 52
4.4 Flow diagram of proposed image encryption 55
4.5 Experimental result for Original images of Lena (a) and aeroplane (d), Encrypted image (b) and (e), Decrypted image (c) and (f) . . . 59
4.6 Histogram of Original image red component (a) Original image green component (b) Original image blue component (c) Encrypted image red component (d) Encrypted image green component (e) Encrypted image blue component (f) 60
4.7 Key sensitivity analysis for Original image (a) and (d), Encrypted image (b) and (e), Decrypted image with slightly wrong key (c) and (f) 62

List of Tables

2.1 Left Circular Shift . . . 13
2.2 Right Circular Shift . . . 13
2.3 Truth Table for XOR . . . 15
2.4 Truth Table for AND . . . 15
2.5 Truth Table of Boolean Function . . . 16
2.6 Truth Table of $GF(2^3)$ . . . 18
2.7 Hamming Distance of Boolean Function . . . 21
2.8 Truth Table of $GF(2^4)$ . . . 24
2.9 Truth Table . . . 27
2.10 S-Box of fixed point . . . 29
2.11 S-Box of opposite fixed point . . . 29

3.1 Key dependent dynamic S-Box . . . 36
3.2 Key dependent inverse S-Box . . . 36
3.3 NL of Boolean function of proposed S-Box . . . 38
3.4 Comparison of NL between proposed S-box with existing S-box . . . 38
3.5 Depedency matric for SAC . . . 39
3.6 Comparison of SAC between proposed S-box with existing S-box . . . 39
3.7 Differential approximation probability . . . 40
3.8 Comparison of DP between proposed S-box with existing S-box . . . 40
3.9 BIC-NL . . . 41
3.10 Comparison of BIC-NL between proposed S-box with existing S-box 41
3.11 Comparison of LP between proposed S-box with existing S-box . . . 42
3.12 New S-Box by changing key . . . 43
3.13 Different properties of newly generated S-boxes . . . 44

4.1 Correlation coefficient of two neighbouring pixels . . . 61
4.2 Entropy Test . . . 63

A.1 S_1 . . . 65
A.2 S_2 . . . 65
A.3 S_3 . . . 66
A.4 S_4 . . . 66
A.5 S_5 . . . 66
A.6 S_6 . . . 67
A.7 S_7 . . . 67

A.8 S_8 . . . 67
A.9 S_9 . . . 68
A.10 S_{10} . . . 68
A.11 S_{11} . . . 68
A.12 S_{12} . . . 69
A.13 S_{13} . . . 69
A.14 S_{14} . . . 69
A.15 S_{15} . . . 70
A.16 S_{16} . . . 70
A.17 S_{17} . . . 70
A.18 S_{18} . . . 71
A.19 S_{19} . . . 71
A.20 S_{20} . . . 71
A.21 S_{21} . . . 72
A.22 S_{22} . . . 72
A.23 S_{23} . . . 72
A.24 S_{24} . . . 73
A.25 S_{25} . . . 73
A.26 S_{26} . . . 73
A.27 S_{27} . . . 74
A.28 S_{28} . . . 74
A.29 S_{29} . . . 74
A.30 S_{30} . . . 75

Abbreviations

AES	Advanced Encryption Standard
ANF	Algebraic Normal Form
AE	Avalanche Effect
BIC	Bit Independence Criterion
BIC-NL	Bit Independence Criterion - Nonlinearity
DES	Data Encryption Standard
DP	Differential Probability
GF	Galois Field
IDEA	International Data Encryption Algorithm
LE	Lyapunov Exponent
LP	Linear Probability
NL	Non-Linearity
RSA	Rivest Shamir Adleman
RC4	Rivest cipher 4
RC6	Rivest cipher 6
SAC	Strict Avalanche Criteria
S-Box	Substitution Box
SPN	Substitution Permutation Network
TLS	Tent-Logistic System
WT	Walsh Transform
XOR	Exclusive OR

Symbols

G	Group
$\mathbb{Z}$	Set of integers
$\mathbb{R}$	Set of real numbers
$\mathbb{F}$	Field
K	Key
μ	Parameter
$\triangle$	Autocorrelation
$\oplus$	XOR
(RGB)	Red, Green, Blue component

Chapter 1

Introduction

From ancient times to the present day, cryptography ensures security during communication. The word cryptography comes from the Greek kryptos, which means hidden. The "crypt-" means "hidden", and the "-graphy" means "writing". Cryptography is the science of secure secret communication so that no third party can read or modify the information. Cryptography converts the secret messages into non readable form or secret code with the help of algorithm. The converted message is called ciphertext and the original message is called plaintext. A process which converts plaintext into ciphertext is called as Encryption and an algorithm which is used in encryption is known as Encryption algorithm. A process which converts ciphertext back to its plaintext is called as Decryption and an algorithm which is used in decryption is known as Decryption algorithm. For the encryption and decryption, cryptographic schemes need some important information which is shared between both sender and receiver, is called a Key [1]. A cryptographic scheme that consists of a message, a ciphertext, a key, an encryption algorithm and decryption algorithm is called a cryptosystem. On the basis of cryptosystem, the cryptography is divided in the following two types. Name as, the Symmetric Key Cryptography and the Public Key Cryptography.

In Symmetric Key Cryptography, a similar key is used for both encryption and decryption. To encrypt and decrypt all messages, both the sender and the receiver must know the secret key. Data Encryption Standard [1], Advanced Encryption

Standard [1], Triple Data Encryption Standard [2], RC4 [1], RC6, Blowfish [2] are examples of symmetric key cryptography. In Public Key Encryption, public and private keys are used separately for encryption and decryption. Since the public key is meant for widespread use, everyone on the network has access to it. One needs to be aware of the recipient's public key in order to encrypt the plaintext. Using their own private key, only the authorised person is able to decrypt the encrypted text. The public is not allowed to see the private key. RSA [2] and ElGamal [3] are examples of public key cryptography.

1.1 S-boxes in Cryptography

Substitution box is an essential component of symmetric key algorithms that performs substitution in cryptography. Substitution boxes are essentially Boolean vectorial functions given as look-up tables. An S-box takes a small block of bits and replaces them with another bit block. For efficient decryption, this substitution must be one to one. The substitution box usually takes m input bits and transforms into n output bit. An $m \times n$ S-box can be viewed as a look up table of 2^m words of n bits each. To strengthen the cryptosystem, an S-box must be designed in such a way that every output bit will depend on each input bit.

A symmetric key cryptosystem has stream cipher or block cipher. A block cipher converts whole block of plaintext into block of ciphertext using the secret key at a time whereas stream cipher encrypts one bit data at a time. So a block cipher has two basic requirement, size of block and size of key. The block ciphers are based on the Shannon's theory of confusion and diffusion that is also implemented in Substitution Permutation networks. Such networks essentially consist of a number of mathematical operations which are interconnected. It takes plaintext and key as input and apply many rounds of S-box to get desired ciphertext. The inverse S-box is used with the same key for decryption to obtain plaintext. Data Encryption Standard and Advanced Encryption Standard are examples of SPN cryptosystem.

The majority of currently used cipher blocks are build on the static natured S-boxes, which is a fundamental weakness in symmetric ciphers. As a result, the most serious flaw in symmetric cryptosystem is predetermined (fixed) substitution since it results in insecure ciphers [4] due to the fixed and predefined qualities of diffusion and confusion. The primary building block of security in an encryption scheme is substitution, despite the fact that permutation has its own effects.

Furthermore, predefined (static) S-boxes do not depend on the secret key, so these static S-boxes are responsible for easy doorways for attackers to launch algebraic attacks. Therefore, scaling the established S-box structure with dynamic strategies is the next challenge for today's symmetric cryptosystem in order to defend against linear and differential attacks as discussed in [5] and [6]. By generating S-boxes dynamically, the strength of ciphers can be increased as stated in [7, 8].

1.2 Image Encryption in Cryptography

An image is something you can see, but its not physically there. It can be a photograph, a painting, a picture on a television or computer screen or other two-dimensional representation. In a world where multimedia technology uses digital images extensively, maintaining user privacy is more crucial than ever. Image encryption is the process of encoding an image with the help of some encryption algorithm. Image encryption is crucial to ensuring the user's security and privacy and to guard against any unauthorised user access. Applications for image and video encryption can be found in a variety of industries, telemedicine, medical imaging, multimedia systems, the internet, and military correspondence [9]. So, the transmitting and receiving images from open platform such as the internet may not always be safe. In order to assure secure image transfer, a secure image sharing technique is needed. A secure image sharing approach that ensures secure image transmission is being achieved with the help of cryptography. Due to its increasing popularity and necessity, a variety of image encryption techniques have been created. Because of features like high correlation analysis, large data capacities, and other characteristics, images are fundamentally different from texts. Therefore, well-known techniques like the AES and the IDEA are always not effective.

1.3 Literature Review

Numerous S-boxes have been designed by researchers, and numerous new construction methods have been suggested. Several attempts have been made in earlier years to replace the static AES S-box structure with dynamic features.

Kazlauskas et al. [10] designed the S-boxes by carrying out different operations on round key.

Fahmy et al. [11] used GNU-C and ISO-C as the two parameters to create the symmetric S-box, replacing the inverse S-box with a new transformation called the shift row transformation.

Agarwal et al. [12] generated 256 key-dependent S-boxes using the affine values ranging from 0 to 255 from the list of 30 irreducible polynomials.

Sahoo et al. [13] employed the affine transformation to construct the static S-boxes.

Nadaf and Desai [14] used the construction strategy of the multi-operation S-box, which also depends on the static AES S-box. Anees and Chen [15] also used the similar construction strategy.

Manjula and Mohan [16] constructed the dynamic S-boxes in which the static S-box was left rotating based on the resulting 16-byte values of round key after performing the exclusive OR operation.

Balajee and Gnanasekar [17] used the pseudo random numbers to generate dynamic S-box values. Alabaichi and Salih [18] also used the similar strategy. A detailed analysis demonstrates that the some S-box approaches are better than other.

Ejaz et al. [19] proposed a secure key dependent dynamic substitution method for symmetric cryptosystems. The generated S-box has dynamic and unique values at each time during the execution and have strong cryptographic properties and randomness and achieved the basic goal of security of symmetric cryptosystems, which details are discused later in Chapter 3.

Recent studies have shown that chaotic-based S-box image encryption has greatly increased security. Similarly, researchers have designed several image encryption techniques in which some are based on different chaotic maps and some are based on permutation and substitution. With the passage of time, many new techniques have been proposed for building strong image encryption techniques.

Pareek et al. [20] have proposed an image encryption scheme using the logistic map.

Zhang et al. [21] suggested chaotic image encryption method based on a key stream buffer and circular substitution box. To generate random numbers, the system has used the logistic map and piecewise linear chaotic map.

Sheela et al. [22] proposed image encryption based on a modified Henon map using hybrid chaotic shift transform.

Supriyo et al. [23] proposed image encryption technique using permutation and substitution. The Arnold's Cat map was used in the construction of the image permutation technique and the logistic map used to create an S-box which was further used for key expansion.

Tyagi and Chaudhary [24] proposed method for image encryption by using two skew chaotic map and external key of 128-bit. The initial conditions for both the skew tent maps are derived using the external secret key.

Li et al. [25] suggested a technique that jointly permutes and diffuses (JPD) the pixels. Similarly, there are many other techniques for image encryption which are developing day by day for generation the strong cipher.

1.4 Thesis Contribution

The objective of this thesis is to study the scheme of Ejaz et al. [19] for the construction of strong key dependent dynamic S-box and this key dependent S-box is used in image encryption scheme. S-box is only nonlinear component of block cipher. Many S-boxes are generated from different scheme. In [19] S-box is generated with only simple mathematical operation like circular shift, XOR and nibble swap. This scheme is implemented in MATLAB. This scheme is used to generate various S-boxes and the cryptographic properties of these S-boxes are analysed with the help of tool [26]. The strength of S-boxes are measured on the basis of certain properties that are discussed in Chapter 2. The generated different key dependent S-boxes is also present in Chapter 3. Finally, we applied this S-box in image encryption scheme. We use compound chaotic map for image encryption. A chaotic map is an evaluation map that exhibits some sort of chaotic behavior (e.g.randomness). The method is effectively implemented to encrypt and decrypt the image in MATLAB and the security analysis of scheme, which includes key sensitivity and statistical analysis, is also determined.

1.5 Thesis Layout

The dessertation is composed as follow:

- **Chapter 2** describes the basic definitions, fundamental ideas, mathematical background and Boolean function that involve in the construction and analysis of S-box.

- **Chapter 3** describes the construction of dynamic key dependent secure S-boxes by using simple mathematical functions or operations and the comparison between newly generated S-boxes of the proposed method. All the calculation are obtained with the help of MATLAB.

- **Chapter 4** describes the image encryption technique, by using obtained key dependent S-box and compound chaotic map. With the help of MATLAB, all the calculation are obtained.

- **Chapter 5** includes the conclusion of the thesis.

Chapter 2

Preliminaries

This chapter describes the definitions, fundamental ideas, mathematical background and basic concepts of group theory and algebra that involve in the construction and analysis of S-box.

2.1 Cryptography

Cryptography is the science of secure secret communication so that no third party can read or modify the information. Cryptography converts the messages into non readable form with the help of algorithm. In cryptography, there are number of techniques which are useful for security purpose. For converting the messages or data into secret codes, we need a scheme or system. Such system is known as Cryptosystem [1]. A cryptosystem consist of five basic components:

- **Plaintext**: It is the original or readable form of message.
- **Ciphertext**: It is the converted or unreadable form of message.
- **Encryption Algorithm**: It converts plaintext into ciphertext.
- **Decryption Algorithm**: It converts ciphertext into plaintext.

- **Key**: It is the special information used in encryption and decryption algorithms which must be known to sender and receiver.

The cryptography is divided in the following two types.

- **Symmetric Key Cryptography**
- **Public Key Cryptography**

2.1.1 Symmetric Key Cryptography

In this type of encryption mechanism, the key used for encryption is same as the key used for decryption. Therefore, it is important to share the key before the transmission of information [1]. The sender and receiver must have the secret key so they can encrypt and decrypt all messages. There are several symmetric key algorithms like Data Encryption Standard (DES) [2], Advanced Encryption Standard (AES) [1, 2], Triple Data Encryption Standard (Triple DES) [2], RC4 [27], RC6, Blowfish [2]. Key sharing is the main flaws of symmetric key cryptography.

2.1.2 Public Key Cryptography

For public key cryptography, two different keys are used for encryption and decryption. These are private key and public key. The public key is for general use, and it is available to everyone on the network. Anyone who wants to encrypt the plaintext must know the public key of the receiver. Only the authorized person who has private key can decrypt the encrypted text. The private key is always kept secret from the outside world. RSA [2] and ElGamal [3] are examples of public key cryptography. The drawback of public key cryptography is that it is slower than symmetric key cryptography but it resolves key sharing issue of symmetric key cryptography. A symmetric key cryptosystem has either block cipher or stream cipher.

Definition 2.1.1. (Stream Cipher)

A stream cipher is symmetric key cipher whose each bit of data is sequentially encrypted using one bit of the key.

Definition 2.1.2. (Block Cipher)

A block cipher is an encryption/decryption scheme in which each block of plaintext is processed as a whole and used to produce another equal-length block of ciphertext.

2.2 Mathematical Background

In this section, some basic principles of group theory are introduced to understand the explanation for the formation and success of the S-boxes.

Definition 2.2.1.

Let G be a non-empty set with binary operation $(*)$ on G. Then $(G, *)$ is called a **Group**, if the following properties are satisfied:

1. Closure: For all $c, d \in G$, $c * d \in G$.
2. Associative: For all $c, d, e \in G$, $(c * d) * e = c * (d * e)$.
3. Identity: For all $c \in G$, there exists an element $f \in G$ such that
$$c * f = f * c = c$$
4. Inverse: For each $g \in G$, then there exist an element $g^{-1} \in G$ such that
$$g * g^{-1} = g^{-1} * g = f$$

Moreover, G is said to be **Abelian or Commutative Group**, if the group holds

$$c * d = d * c \quad \text{for all } c, d \in G$$

Example 2.2.2.

Following are some examples of group:

1. Set of integers $\mathbb{Z}$ is a group with respect to usual addition operation of integer.
2. Set of all invertible matrices of order $n \times n$ with ordinary matrix multiplication forms a group.
3. Set of real number $\mathbb{R}$ is a group under addition.
4. The set $\mathbb{R}$ and set of integers $\mathbb{Z}$ are the examples of abelian groups with respect to addition.
5. The set of $\mathbb{R}\backslash\{0\}$ is an example of an abelian group with respect to multiplication.

Definition 2.2.3.

A non-empty set $\mathbb{F}$ with two binary operation addition (+) and multiplication (.) is called a **Field**, if it satisfies the following properties:

1. ($\mathbb{F}$,+) is commutative group.
2. ($\mathbb{F} \setminus \{0\}$,.) is commutative group.
3. Distributivity of multiplication over addition.

   For all $c, d, e \in \mathbb{F}$
   $$c \cdot (d + e) = (c \cdot d) + (c \cdot e)$$

Moreover, a field that contains finitely many elements is called **Finite Field**.

Example 2.2.4.

Following are some examples of field:

1. Set of real numbers $\mathbb{R}$ are field under usual addition and multiplication.

2. Set of complex numbers $\mathbb{C}$ are field under usual addition and multiplication.

3. Set of integer $\mathbb{Z}$ is not a field as there are no multiplicative inverses in $\mathbb{Z}$.

Definition 2.2.5.

A finite field in which the number of elements are of the form p^n is called **Galois Field** where p is prime and n is positive integer. It is denoted by $GF(p^n)$.

The elements of Galois field $GF(p^n)$ is defined as [28]

$$GF(p^n) = \{0, 1, 2, ..., p-1\} \cup \{p, p+1, p+2, ..., p+p-1\}$$

$$\cup\{p^2, p^2+1, p^2+2, ..., p^2+p-1\} \cup \dots$$

$$\cup\{p^{n-1}, p^{n-1}+1, p^{n-1}+2, ..., p^{n-1}+p-1\}$$

The order of the Galois field is given by p^n and p is the characteristic of the field whereas characteristics of the field is defined as the minimum number of times the multiplicative identity (one) must be used in a sum to obtain the additive identity (zero).

From the perspective of cryptography, one will focus on the following cases:

- $GF(p),\ n = 1$
- $GF(2^n),\ p = 2$

All polynomials of degree less than n with coefficients from $GF(p)$ are the elements of $GF(p^n)$. There are 256 elements in the finite field $GF(2^8)$.

Definition 2.2.6.

In mathematics, **Circular Shift** is the operation of rearranging the entries in a tuple, either by moving the last entry to the first place, while moving all the other entries to the next place, or by performing the operation inverse. Circular shifts are often used in cryptography to rearrange sequences of bits.

There are two types of circular shift, one is left circular shift and other is right circular shift which are discussed below:

1 Left Circular Shift

Left circular shift of n bits moves the first bits in the ending of the bit string while moving all other bits to the previous position.

Example 2.2.7.

Table 2.1 are examples of left circular shift:

Table 2.1: Left Circular Shift

8 Bit Sequence	**Shift by 3**
10010111	10111100
11100101	00101111
01111001	11001011
11001101	01101110
01001101	01101010

2 Right Circular Shift

Right circular shift of n bits moves the last bits in the beginning of the bit string while changing all other bits to the next position.

Example 2.2.8.

Table 2.2 are examples of right circular shift:

Table 2.2: Right Circular Shift

8 Bit Sequence	**Shift by 3**
11110010	01011110
01100101	10101100
00101111	11100101
10111001	00110111
00111001	00100111

Definition 2.2.9.

In the nibble swap, the nibble is four binary bits or "half a byte". The terms "byte" almost always refer to 8 bits. In the **Nibble Swap** operation, a byte is split from the middle, into two nibbles, and then the two nibbles change position with each other. Equivalently, swap the two hexadecimal numbers.

Example 2.2.10.

Consider a number $(93)_{10}$ in binary representation is $(01011101)_2$ and in hexadecimal form is $(5D)_{16}$.

It has two nibbles 0101 and 1101.

After swapping the nibbles, we get $(11010101)_2$ which is $(213)_{10}$ in decimal form and $(D5)_{16}$ in hexadecimal form.

2.3 Boolean Function

Boolean function is define as $f{:}GF(2^m) \rightarrow GF(2)$ where m is non-negative integer. In which m tuples $\{b_1, b_2, b_3 \ldots, b_m\} \in GF(2^m)$ as input and produces only one of the two output bits $\{0, 1\} \in GF(2)$ [19]. By using Boolean function, output values of Boolean can be determined with the help of some logical calculations of input values of Boolean. Boolean function has only the two possible outcome: one is true (one) and the other is false (zero). These functions are useful for designing electronic circuits, integrated circuits and digital computer chips. These functions also play a significant role for designing a substitution boxes (S-box) in cryptography.

A Boolean function can be expressed in two different ways.

- Truth Table (TT)
- Algebraic Normal Form (ANF)

Definition 2.3.1.

Truth Table represents the possible outcomes of a Boolean function in tabular form. The first two columns show the possible input and the last column shows the executed output of function. Boolean function f can be represented as a binary vector of size $(2^m \times 1)$, with entries $f(b)$ indexed by the vectors $b \in GF(2^m)$.

Example 2.3.2.

Consider the Boolean function $f = b_1 \oplus b_2$ of two variables b_1 and b_2. Truth Table of $m = 2$ is shown below in Table 2.3.

Table 2.3: Truth Table for XOR

b_1	b_2	f
0	0	0
0	1	1
1	0	1
1	1	0

Example 2.3.3.

Consider the Boolean function $f = b_1 \cdot b_2$ of two variables b_1 and b_2. Truth Table of $m = 2$ is shown below in Table 2.4.

Table 2.4: Truth Table for AND

b_1	b_2	f
0	0	0
0	1	0
1	0	0
1	1	1

Example 2.3.4.

Consider a mapping f: $GF(2^4)$ to $GF(2)$ given by

$$f(b_1, b_2, b_3, b_4) = b_1b_2b_3 \oplus b_2b_3b_4 \oplus b_1$$

For all possible values, the input bits b_1, b_2, b_3 and b_4, the output bit f are shown in Table 2.5.

TABLE 2.5: Truth Table of Boolean Function

b_1	b_2	b_3	b_4	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Definition 2.3.5.

Boolean function $f : GF(2^m) \rightarrow GF(2)$ of **Algebraic Normal Form**(ANF) described in the following form of polynomial.

$$f(b_1, b_2, \dots, b_m) = a_0 \oplus a_1 b_1 \oplus a_2 b_2 \oplus \dots \oplus a_m b_m \oplus$$

$$a_{1,2} b_1 b_2 \oplus \dots \oplus a_{m-1,m} b_{m-1} b_m \oplus \dots \oplus$$

$$a_{1,2\dots m} b_1 b_2, \dots, b_m$$

where $b_1, b_2, \dots, b_{1,2,\dots m} \in \{0,1\}^m$.

The ANF representation of Boolean function is most commonly used in cryptography. Boolean functions are widely used due to its unique properties. In study of S-boxes and Boolean functions, ANF plays an important role.

Consider the Boolean function f of two variables b_1 and b_2 . The ANF of 'XOR' Boolean function is represented as

$$f(b_1, b_2) = b_1 \oplus b_2 \oplus b_1 b_2$$

2.3.1 Application of Boolean Function in S-boxes

In cryptography, Boolean functions are important element for the construction of S-box. The function P defined as

$$P : GF(2^n) \to GF(2^m)$$

takes n bits as input and returns m output bits where $m > 1$ is a vectorial Boolean function. Basically, an S-box is a vectorial Boolean function.

Definition 2.3.6.

The sequence of form $\{(-1)^{f(\beta_0)}, (-1)^{f(\beta_1)}, \ldots, (-1)^{f(\beta_{2^n-1})}\}$ is known as **Sequence of Boolean function**. A sequence in which ones and minus ones or zeros has equal number known as balanced sequence whereas a sequence which has unequal number of ones and minus ones or zero known as unbalanced sequence.

Example 2.3.7.

Consider the following Boolean function which has three b_1, b_2 and b_3 as input bits.

$$f(b_1, b_2, b_3) = b_3 \oplus b_1 b_2$$

and it is shown in Table 2.6.

The sequence of Boolean function can be written as:

TABLE 2.6: Truth Table of $GF(2^3)$

b_1	b_2	b_3	b_1b_2	f
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	0

$\{(-1)^{f(\beta_0)}, (-1)^{f(\beta_1)}, (-1)^{f(\beta_2)}, (-1)^{f(\beta_3)}, (-1)^{f(\beta_4)}, (-1)^{f(\beta_5)}, (-1)^{f(\beta_6)}, (-1)^{f(\beta_7)}\}$

$\{(-1)^0, (-1)^1, (-1)^0, (-1)^1, (-1)^0, (-1)^1, (-1)^1, (-1)^0\}$

$\{1, -1, 1, -1, 1, -1, -1, 1\}$

It contains equal numbers of 1s and -1s. Hence, the sequence of Boolean function is balanced.

Definition 2.3.8.

A Boolean mapping $f : GF(2^m) \to GF(2)$ which can be written in the form of linear combination is known as **Linearity**, expressed as

$$f(x_1, x_2, ..., x_m) = d_1x_1 \oplus d_2x_2 \oplus \ldots \oplus d_mx_m$$

where $d_1, d_2 \ldots, d_m \in 2^m$ and the symbol used for XOR operation is $\oplus$ and linear combination of two Boolean functions $f(x), g(x)$ is define as

$$(f \oplus g)x = f(x) \oplus g(x)$$

Definition 2.3.9.

A Boolean mapping $f : GF(2^m) \rightarrow GF(2)$ which is combination of linearity and the constant is known as **Affine Function**, expressed as

$$f(x_1, x_2, ..., x_m) = d_1x_1 \oplus d_2x_2 \oplus \ldots \oplus d_mx_m \oplus d_0$$

For an Affine cipher, a Boolean function over modulo e is used. It is a basic substitution cipher. Due to less security, affine cipher can easily breakable.

This cipher performs the addition and multiplication using the function;

$$f(x) = (Bx \oplus D) \mod e$$

where B and D are key for the cipher. It is used for the encryption.

Definition 2.3.10.

The number of non-zero digits in a binary sequence is called **Hamming Weight**. It is represented by $H(w)$, where $w \in GF(2^m)$

Example 2.3.11.

For $m = 8$, take a sequence

$$w = (10100111)$$

in which the number of zeros is three and the number of ones is five.

Thus Hamming weight is

$$w = (10100111) = H(10100111) = 5$$

Definition 2.3.12.

Hamming Distance between two Boolean functions $k(u)$ and $l(u)$ with mapping $k(u), l(u) : GF(2^m) \rightarrow GF(2)$ is define as:

$$d(k, l) = H(k(u) \oplus l(u))$$

$$k(u) \oplus l(u) = k(u_0) \oplus l(u_0) \oplus k(u_1) \oplus l(u_1) \oplus \ldots \oplus k(u_{2^m-1}) \oplus l(u_{2^m-1})$$

where $u = (u_0, u_1, ..., u_{2^m-1}) \in GF(2^m)$.

Hamming distance is also defined as the number of bit positions in which the two bits are different. For calculation of Hamming distance, we perform the XOR operation and then count the number of $1s$ in the result.

Example 2.3.13.

Consider the two Boolean functions

$$k(u) = 10011111$$

$$l(u) = 10101011$$

And after taking XOR operation, we get

$$k(u) \oplus l(u) = 00110100$$

Thus, the Hamming distance is $d(k, l) = 3$

Example 2.3.14.

Consider the two Boolean function $k(u)$ and $l(u)$ with u_1, u_2 and u_3 as input bits where

$$k(u) = u_1u_2u_3, \quad l(u) = u_1 \oplus u_2u_3$$

The Hamming distance of Boolean functions are

$$d(k, l) = H(k(u) \oplus l(u))$$

also write as

$$d(k, l) = H((u_1u_2u_3) \oplus (u_1 \oplus u_2u_3))$$

the calculation of Hamming distance are shown in Table 2.7

Table 2.7: Hamming Distance of Boolean Function

u_1	u_2	u_3	$k = u_1u_2u_3$	$l = u_1 \oplus u_2u_3$	$k(u) \oplus l(u)$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	0	1

Thus, the Hamming distance is $d(k,l) = 5$

Definition 2.3.15.

The correlation measurement between Boolean function g and all the linear combinations is called the **Walsh Transform**. The Boolean function of the Walsh transform is defined as:

$$WT_g(\beta) = \sum_{\beta \in GF(2^m)} (-1)^{g(u) \oplus \beta . u} \tag{2.1}$$

for all $u \in GF(2^m)$.

2.4 Substitution Boxes

Substitution box (S-box) is an necessary component of symmetric key algorithms that performs substitution. An S-box takes a small bits block and replaces them with another bit block. For efficient decryption, this substitution must be one to one. The substitution box usually takes m input bits and transforms into n output bit. To strengthen the cryptosystem, an S-box must be designed in such a way that every output bit will depend on each input bit. The non-linearity is most important feature of the substitution box. Because S-box are non-linear so that it provides security in particular against linear cryptanalysis.

2.4.1 Characteristics of S-box

In cryptosystem, S-box is only the highly nonlinear Boolean function. Actually, there are two main reasons for studying the S-box design:

- **Designing new Ciphers**

  S-box design is the most important area for designing a new cipher, due to the fact it is the solely nonlinear part of the system. The strength of cipher depends on this part. As with development of cryptography, hackers are also creating new methods of attacks, so S-box design have to be secured in advance to assurance cipher security.

- **Private use of S-Box Design**

  Adversaries used the back-doors to generate key for certain ciphers such as AES, therefore, each agency and particularly governments prefer to have a secure device only relevant to their agency with a more safety layer which is feasible solely if they design their S-boxes for their particular system.

2.4.2 Classification of S-boxes

There are three types of S-box.

- **Straight S-box**

  A Straight S-box takes an input and produces output of the similar size. This type of S-box had been recommended by Rijndeel cipher. It is the easiest and common category of S-box. Advanced Encryption Standard is an example of such S-box.

- **Expanded S-box**

  It collects lesser bits as input and create an output of more bits. Such S-box can be build by duplicating some input or output bits.

- **Compressed S-box**

  This type of S-box takes more input bits and produce lesser output bits. Example of this type of S-Box is Data Encryption Standard in which 6 bits of input are taken as one block of input and 4 bits in one block are returned as output block.

2.4.3 General Properties of S-Box

Substitution boxes (S-Boxes) are an important part of many cryptosystem. Its perform substitution in cryptography, so it should satisfy the following properties for developing a strong S-box. Some properties of the strong S-Box are given below.

Definition 2.4.1.

A sequence of Boolean function g is called **Balanced** if the occurrence of both zeros and ones are equal.

Example 2.4.2.

Consider the Boolean function with mapping

$$GF(2^4) \to GF(2)$$

such that

$$g(b_1, b_2, b_3, b_4) = b_1 \oplus b_2 \oplus b_3 b_4$$

where b_1, b_2, b_3 and b_4 are taken as inputs bits which are given in Table 2.8.

In Table 2.8, the last column has eight zeros and the eight ones. Thus, the sequence g is balanced.

TABLE 2.8: Truth Table of $GF(2^4)$

b_1	b_2	b_3	b_4	b_3b_4	g
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	1	1

Definition 2.4.3.

Bijection is a mapping, which gives a unique output with the each input bits. Let n be the possible input bits such as $(0,1)^n$, there exists a unique output bit. All the output vector must appear once. For calculating the bijective property a method was introduced for the $n \times n$ [29] S-boxes. An $n \times n$ S-boxes are said to satisfy the bijective property if for $(1 \leq j \leq n)$ the Boolean functions g_j of S are such that:

$$H(\sum_{j=1}^{n} a_j g_j) = 2^{n-1} \tag{2.2}$$

where $a_j \in \{0,1\}$ for $(a_1, a_2, \ldots, a_n) \neq (0,0,\ldots,0)$ and H is the Hamming weight.

The Condition 2.2 guarantees that every Boolean function g_j and all their combination are balanced.

Definition 2.4.4.

The **Correlation Immunity** [30] of a Boolean function denotes the independence size between the linear combination of input and output bits. By using the relationship between Walsh transform and Hamming weight of its input, functional order can be determine.

When $WT_g(\beta) = 0$, and $1 \leq H(w) \leq$ p, a Boolean function is said to have correlation immunity.

Definition 2.4.5.

Algebric Immunity of two Boolean functions $g(u)$ and $h(u)$ is defined as the lowest degree of non-zero function h such that either

$$(g + 1)h = 0 \text{ or } g \cdot h = 0$$

where the function h for which $g \cdot h = 0$ is called annihilator of g [31].

Definition 2.4.6.

The **Absolute Indicator** of Boolean function $g(u)$ is defined as the maximum absolute value of autocorrelation, which are:

$$\triangle_g = \max | \triangle_g (d)| \quad \text{where} \quad d \in GF(2^n)$$

And the **Autocorrelation** of Boolean function $g(u)$ is defined as:

$$\triangle_g(d) = \sum (-1)^{g(u)+g(u+d)} \quad \text{where} \quad u \in GF(2^n)$$

The **Sum of Square Indicator** of Boolean function $g(u)$ also derived from the autocorrelation function $\triangle_g(d)$ which are

$$\sigma_g = \sum_{d \in GF(2^n)} (\triangle_g(d))^2$$

Definition 2.4.7.

An **Algebraic degree** is associated with the nonlinearity measures. An algebric degree of Boolean function $g(u)$ is defined as the highest degree of a function g, which are

$$deg(g) = n - 1$$

Higher algebraic degree are more better than the lower algebraic degree.

Definition 2.4.8.

The **Non-linearity** of a Boolean function with mapping $g(u) : GF(2^n) \rightarrow GF(2)$ is defined as the minimum hamming distance of g from the set of all n-variable affine functions.

$$NL(g) = \min d(k, l)$$

Nonlinearity of Boolean function g can be shown by using Walsh transform from the following formula

$$NL(g) = 2^{n-1}(1 - 2^{-n}) \max_{\beta \in GF(2^n)} |WHT_g(\beta)|$$

Thus a Boolean function g which is a maximally nonlinear is known as bent function.

Example 2.4.9.

Let u_1 and u_2 are input bits and $g(u)$ is a Boolean function such that

$$g(u_1, u_2) = u_1 \oplus u_2$$

Truth Table is given below:

TABLE 2.9: Truth Table

u_1	u_2	$g(u)$	0	$u_1 \oplus u_2$	$g(u) \oplus 0$	$g(u) \oplus u_1$	$g(u) \oplus u_2$	$g(u) \oplus (u_1 \oplus u_2)$
0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	1	0	0
1	0	1	0	1	1	0	1	0
1	1	1	0	0	1	0	0	1

where $0, u_1, u_2, u_1 \oplus u_2$ are the possible linear function of u_1 and u_2

and

$$d_1(g(u), 0) = 3, \quad d_2(g(u), u_1) = 1, \quad d_3(g(u), u_2) = 1, \quad d_4(g(u), u_1 \oplus u_2) = 1$$

So,

$$NL_g = \min(d_1, d_2, d_3, d_4) = 1$$

Definition 2.4.10.

If by changing single binary bit in an input results in significant change in an output, then the binary sequence is called **Avalanche Effect** [19].

When any algorithm shows higher AE, it means that algorithm is strong cryptographic and also secure against the attacks.

Definition 2.4.11.

If single input bit is change as a result each output bit changes with the probability of 0.5 or 50% is called **Strict Avalanche Criteria** [19].

The values of Strict Avalanche Effect of S-box depend on the dependency matrix.

Definition 2.4.12.

If single input bit changes as a result output bits will change independently is known as **Bit Independence Criterion**.

To measure the the relationship between pairs of avalanche variables, correlation must be calculated.

Definition 2.4.13.

Linear Probability is used to compute the resistance of linear cryptanalysis, which is estimated as:

$$LP = \max_{\alpha_z, \beta_z \neq 0} \left| \frac{R\{z \in M \mid z.\alpha_z = S(z) \cdot \beta_z\}}{2^n} - \frac{1}{2} \right| \tag{2.3}$$

where M represent the all possible inputs, α_z is the input bit and β_z is output bit and 2^n with $n = 8$ which is $2^8 = 256$ number of elements.

Definition 2.4.14.

Differential Probability is use to measured the performance against differential cryptanalysis, which is estimated as:

$$DP = \max_{\Delta_z \neq 0, \Delta_x} \left(\frac{R\{z \in M \mid S(z) \oplus S(z \oplus \Delta z) = \Delta x\}}{2^n} \right) \tag{2.4}$$

where M represent the all possible inputs, Δz is the input differentials, Δx is output differentials and 2^n with $n = 8$ which is $2^8 = 256$ number of elements. An S-box with has lower Differential Probability can withstand differential cryptanalysis better.

Definition 2.4.15.

Take an S-box $P : GF(2^n) \to GF(2^m)$ and for $v \in GF(2^n)$. A point is called the **fixed point** of S-box if

$$p(v) = v$$

and the point is called the **opposite fixed point** of S-box if

$$p(v) = v^{'}$$

where $v^{'}$ is the reverse of v.

Any S-box is supposed to be good against differential cryptanalysis attacks which does not have fixed points and opposite fixed points as compared to those that has a fixed point and opposite fixed points.

Example 2.4.16.

Take an S-box (2×2) with two Boolean functions as shown in Table 2.10

TABLE 2.10: S-Box of fixed point

$GF(2)$	Binary format of $GF(2)$	S-Box	Binary format of S-Box
0	00	1	01
1	01	3	11
2	10	2	10
3	11	0	00

In Table 2.10, the '2' element shows the fixed point of S-Box.

Example 2.4.17.

Take an S-box (2×2) with two Boolean functions as shown in Table 2.11

TABLE 2.11: S-Box of opposite fixed point

$GF(2)$	Binary format of $GF(2)$	S-Box	Binary format of S-Box
0	00	1	01
1	01	2	10
2	10	3	11
3	11	0	00

In Table 2.11, the '1' element shows the opposite fixed point of S-Box.

2.5 Key Dependent S-Box

With the changing in technology day by day, people are searching for new features of data communication through the network with optimal data security. It is highly possible that during the transmission of data, information can be accessed by unauthorized individuals, putting any network systems security at risk [32]. However, the data security is more important and no risk is acceptable.

Substitution is the main source of nonlinearity in cipher block of symmetric cryptosystem and S-Box is only nonlinear part of cryptosystem. The primary weakness of symmetric ciphers is the existence of S-boxes, which are based on fixed (static) nature. Specified (fixed) substitution is the most obvious flaw in symmetric cryptosystems because it causes insecure ciphers due to the fixed and predefined qualities of diffusion and confusion [4]. Although permutation has its own effects, the essential building block of security in an encryption system is substitution.

Furthermore, static natured S-boxes do not depend on the secret key, so these static S-boxes are responsible for easy doorways for attackers to launch algebraic attacks. So there are need for design a S-boxes which are depend on key in order to resist differential and linear attack.

2.6 Software Tools in the Analysis of S-box

Some tools are available for the studying of S-box properties. A brief description of such available tools are given below:

1. **Boolfun Package in R**

   Free open source mathematical program is R, used for statistical computing. It runs on various windows UNIX and Mac OS platforum, despite the fact that the standard variant R does not support boolean function of evaluation, but a package name **Boolfun** can be loaded which gives feature related to cryptographic analysis of Boolean functions [33].

2. **S-box Evaluation Tool (SET)**

   This technique for evaluating the cryptographic properties of the Boolean function and S-boxes was once proposed by Stjepan Picek and his team. SET stands for S-box Evaluation Tool. It is also a free tool for open source mathematics that is convenient and handy to use. It works in VS(visual studio) [34].

3. **Sage Math**

   The Sage Math library is a free, open source mathematics tool that consists of a Boolean function module and an S-box. Through this method, we can test the algebraic properties and measure a range of cryptographic properties for S-boxes and Boolean functions associated to the linear approximation matrix and distinction distribution table.

4. **VBF**

   VBF is the short form of Boolean Function Library Vector. It was introduced by Alverez-Cubero and Zufiria [35]. This tool is used for the test and analysis of cryptographic properties of S-boxes [35].

5. **SAMT**

   SAMT is another tool for the analysis and evaluation of cryptographic properties of Boolean function and S-box. It works on MATLAB.

Chapter 3

Construction of Dynamic Key Dependent S-box for Symmetric Cryptosystem

S-box plays a significant role in cryptosystem. Recently a method is proposed by Ejaz et al. [19] for construction of dynamic key dependent secure S-boxes. In this chapter, a construction of S-box by using simple mathematical functions or operations are discussed. The operations used in S-box is given in Section 3.1. The proposed algorithm and flowchart of S-box is presented in Section 3.2. Findings and results of proposed S-box is given in Section 3.3 and the comparison between newly generated S-boxes of the proposed method are given in Section 3.4.

3.1 Operations Used for Generating S-box

There are number of methods of S-boxes proposed and designed by different researcher. Some researcher uses the static S-box for their research and some uses the dynamic S-box. But the analysis shows that the some S-boxes which are designed, either fixed (static) in nature, optimally not secure or deficient in dynamicity and randomness due to which these are quiet vulnerable or exposed to the modern attack.

The proposed technique approach differs from previously developed AES based S-box and its other variants since it does not employ affine polynomials to generate values for the S-box. The suggested substitution method makes use of a few straightforward yet essential mathematical operations or functions.

In the proposed method, S-boxes are constructed dynamically from secret key of 128-bits (16 bytes in Hexadecimal) by performing some basic and simple operations including Circular Shift, XOR and Nibble Swap.

Circular Shift is the operation of rearranging the entries in a tuple, either by moving the last entry to the first place, while moving all the other entries to the next place, or by performing the operation inverse. The some details are already discussed in Section 2.2. Formally, a permutation σ is defined as a circular shift of entries n in each tuple such that:

$$\sigma(j) \equiv (j+1) \mod m, \quad \text{for all entries} \quad j = 1, \ldots, m$$

Exclusive OR (XOR) is a logical operation that only produces a true value if certain conditions are met like one input is true and other input is false. Both inputs must be different. Its symbol is as follows:

$$X \,\mathrm{XOR}\, Y \quad \text{is written as} \quad X \oplus Y$$

In **Nibble Swap**, the term nibble originally means “half an octet” or “half a byte”. The terms ‘byte’ mean eight binary bits and ‘nibble’ mean four binary bits. In nibble swap operation, a byte is divided from middle into two nibbles and then each nibble shift with another. Some details are already discussed in Section 2.2.

The S-boxes produced by the proposed technique are entirely distinct from the fixed (static) S-boxes of AES since they are built using mathematical functions or operations to defend against algebraic attacks.

3.2 Proposed Algorithm for Generating S-box

Many researchers [16, 18, 36, 37] have proposed the design approaches of key dependent dynamic S-boxes generation with various cryptographic strengths. However, most of these approaches, lack in randomness and dynamicity. Ejaz et al. [19] proposed a secure key dependent dynamic substitution method for symmetric cryptosystem. This approach uses the simple mathematics operations and functions for creating S-box. The algorithm for designing S-box is explained below:

Algorithm 3.2.1.

Input: Hexadecimal sequence input by the user, where Key $(K) = 16$ characters

Output: S-box and Inv S-box

Step 1: Convert the 16 characters of K (in Hexadecimal) into binary sequence of 128 bits of key then $K = 128$ bits.

Step 2: Count the $n = number$ of $1s$ from K and perform the nth time left circular shift.

Step 3: Divide the $K = 128$ bits into two halves, then

$$\text{Left Key } (LK) = 64 \text{ bits and Right Key } (RK) = 64 \text{ bits}$$

Step 4: Perform XOR operation on both halves. The resultant value are stored as R' and the previous Right Key (RK) are stored as L'.

Step 5: Convert the $R' = 64$ bits into 8-bytes. After that perform nibble swap on each bytes.

Step 6: Store values (8-bytes) in array of size '8' followed by the loop. After that a conditional statement is used to avoid duplication and ensure uniqueness of S-box.

Step 7: After storing the values of R' in S-box, R' is reconverted into binary sequence of 64 bits.

Step 8: After that, both L' and R' rejoin here to make binary sequence of 128 bits, and then control moves back to the Step 1 as:

$$K = L' + R'$$

Step 9: Then, all actions are carried out in the previous order, and each step is managed by a conditional statement included within a conditional loop. This process is repeated until the S-box generates 256 distinct values in hexadecimal format.

Step 10: For inverse S-box, a new loop is generate in which the indexes and values of the generated S-box are swapped with each other to create the inverse S-box.

The Algorithm 3.2.1 implementation is performed on PC with MATLAB R2019b having operating system window 10 pro 64 bit, Core i7-4600U with 2.10 GHz CPU and 8GB Ram.

By taking as an Example of key in hexadecimal form, a S-box is constructed, which is shown in Table 3.1 and inverse S-Box is constructed, which is shown in Table 3.2.

Example:

Key value (in hex): $7468617473206D79206B756E67206675$

TABLE 3.1: Key dependent dynamic S-Box

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	a8	08	26	38	24	10	16	d1	9b	60	58	a2	61	aa	c6	82
1	0b	36	37	5b	b6	39	9c	9f	c9	97	7e	7c	9a	19	4f	b8
2	f6	8c	64	dc	40	88	76	90	14	d6	da	db	0d	0e	7b	3b
3	21	b2	73	7d	2b	ff	3c	f3	33	f9	22	17	e8	89	af	e4
4	5c	86	91	68	a3	e7	0f	cb	9d	b9	6e	2c	d4	e9	ae	84
5	1a	b5	fe	23	01	30	e6	1d	95	29	55	ce	6a	71	c4	96
6	6c	f5	5a	70	ef	2f	94	48	b0	4b	7f	77	2a	a1	69	ab
7	ca	a4	ad	f1	b7	dd	57	a7	51	12	87	6d	de	ee	d7	42
8	1f	a6	0c	92	11	15	fc	80	b1	45	28	4e	31	47	6f	ec
9	e2	34	c1	44	03	35	50	04	cc	0a	ea	8b	e0	1b	d9	ba
a	b4	3a	7a	c0	65	54	83	07	43	81	bc	1e	f8	32	b3	ed
b	99	3e	5d	2e	f4	e3	93	3d	a9	e1	18	a5	52	3f	66	05
c	c3	6b	20	d2	d8	d5	bd	25	63	56	fb	8a	4a	e5	8e	27
d	df	d3	4c	8f	bb	02	98	85	59	f2	be	ac	74	f7	bf	79
e	49	72	5f	9e	8d	4d	53	5e	cf	13	c2	eb	46	c7	fa	41
f	67	d0	fd	09	1c	a0	c8	2d	06	78	cd	f0	75	62	c5	00

TABLE 3.2: Key dependent inverse S-Box

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	ff	54	d5	94	97	bf	f8	a7	01	f3	99	10	82	2c	2d	46
1	05	84	79	e9	28	85	06	3b	ba	1d	50	9d	f4	57	ab	80
2	c2	30	3a	53	04	c7	02	cf	8a	59	6c	34	4b	f7	b3	65
3	55	8c	ad	38	91	95	11	12	03	15	a1	2f	36	b7	b1	bd
4	24	ef	7f	a8	93	89	ec	8d	67	e0	cc	69	d2	e5	8b	1e
5	96	78	bc	e6	a5	5a	c9	76	0a	d8	62	13	40	b2	e7	e2
6	09	0c	fd	c8	22	a4	be	f0	43	6e	5c	c1	60	7b	4a	8e
7	63	5d	e1	32	dc	fc	26	6b	f9	df	a2	2e	1b	33	1a	6a
8	87	a9	0f	a6	4f	d7	41	7a	25	3d	cb	9b	21	e4	ce	d3
9	27	42	83	b6	66	58	5f	19	d6	b0	1c	08	16	48	e3	17
a	f5	6d	0b	44	71	bb	81	77	00	b8	0d	6f	db	72	4e	3e
b	68	88	31	ae	a0	51	14	74	1f	49	9f	d4	aa	c6	da	de
c	a3	92	ea	c0	5e	fe	0e	ed	f6	18	70	47	98	fa	5b	e8
d	f1	07	c3	d1	4c	c5	29	7e	c4	9e	2a	2b	23	75	7c	d0
e	9c	b9	90	b5	3f	cd	56	45	3c	4d	9a	eb	8f	af	7d	64
f	fb	73	d9	37	b4	61	20	dd	ac	39	ee	ca	86	f2	52	35

The generated S-box and inverse S-box values are in hexadecimal format. While the proposed approach is very useful and capable of producing endless S-boxes and their inverse S-boxes, only one example of an S-box and its inverse S-box are provided in Tables 3.1 and 3.2, respectively. The proposed method is key dependent and a single bit change in a key can produce various S-boxes with distinct values. Here is the flow diagram of the proposed method.

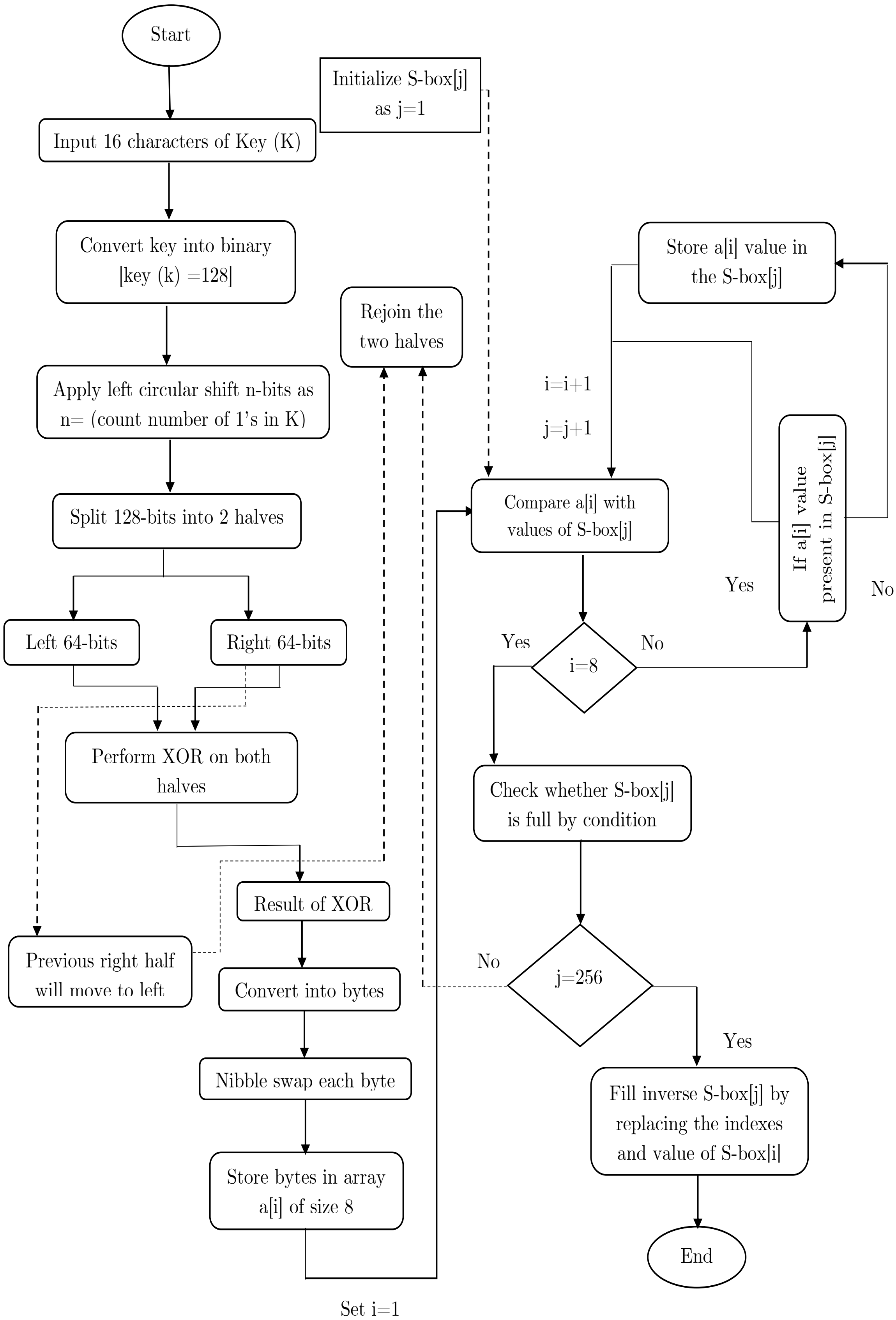

FIGURE 3.1: Working flow of proposed substitution method

3.3 Finding and Result

The evaluation of constructed S-box in Table 3.1 is performed by using tool which is presented in [26]. Some cryptographic properties are describe in this section and some properties are also described in Chapter 2 is presented in this section.

3.3.1 Nonlinearity

Any good S-box should not map an input to an output linearly because it compromises the security of the cipher. The cipher is more resistant to linear cryptanalysis when the nonlinearity value is higher. The calculation formula for nonlinearity of S-boxes are already defined in Section 2.4.3. The average value of NL is 104.25 with a minimum of 102 and maximum of 108. The NL values of all eight component of the Boolean functions in proposed S-box is shown in Table 3.3.

TABLE 3.3: NL of Boolean function of proposed S-Box

0	1	2	3	4	5	6	7	Average
108	102	106	108	102	102	104	102	104.25

and the comparison of nonlinearity of S-box with previous S-boxes are shown in Table 3.4.

TABLE 3.4: Comparison of NL between proposed S-box with existing S-box

S-boxes	NL		
	Min	Max	Avg
Hussain et al. [38]	98	108	104
Vaicekauskas et al. [39]	98	108	102.5
Alkhaldi et al. [40]	98	108	104
Hussain et al. [41]	100	108	104.75
Khan et al. [42]	102	108	105.25
Proposed [19]	102	108	104.25

Thus the nonlinearity of S-box shows that it is a good indicator to resist the linear attack.

3.3.2 Strict Avalanche Criteria

Strict avalanche criteria is a necessary component for cryptographic S-box. This criterion indicates, if only single input changes as a result each output bit changes with the probability of 0.5. The SAC values of S-box depend on dependency matrix. The average value of SAC is 0.504395 with a minimum of 0.406250 and maximum of 0.593750. The dependency matrix for SAC of proposed method is shown in Table 3.5.

TABLE 3.5: Depedency matric for SAC

0.5625	0.5781	0.5000	0.5156	0.5156	0.5156	0.5468	0.5468
0.4531	0.4844	0.5312	0.4844	0.4531	0.4687	0.5156	0.5000
0.5468	0.5156	0.5156	0.5468	0.5468	0.4531	0.5000	0.4844
0.5156	0.4687	0.5312	0.5000	0.4218	0.5468	0.4687	0.4687
0.5468	0.5000	0.4687	0.4687	0.5000	0.5312	0.4687	0.5000
0.5781	0.5156	0.4687	0.4844	0.4844	0.4218	0.5625	0.4687
0.5937	0.4844	0.4844	0.5937	0.5781	0.4531	0.5468	0.5312
0.4062	0.4844	0.4218	0.5156	0.5468	0.4844	0.5000	0.4844

A comparison of minimum value, maximum value and average SAC value of the proposed S-box with the SAC values of existing S-boxes are shown in Table 3.6.

TABLE 3.6: Comparison of SAC between proposed S-box with existing S-box

S-boxes	SAC		
	Min	Max	Avg
Jakimoski. [43]	0.3761	0.5975	0.5058
Khan et al. [42]	0.3906	0.6250	0.5039
Wang et al. [44]	0.4850	0.5150	0.5072
Çavuşoğlu et al. [45]	0.4218	0.5937	0.5039
Özkaynak et al. [46]	0.3906	0.5703	0.4931
Proposed [19]	0.4062	0.5937	0.5044

Thus, the proposed method also meets the criteria of SAC and its values is close to 0.5 as compared with the others.

3.3.3 Differential Probability

Differential cryptanalysis for S-boxes was demonstrated by Biham and Shimar in [47]. Differential probability is use to measure the performance against the differential cryptanalysis and its formula is already defined in Section 2.4.3. Table 3.7 provides a summary of all the differential values of the proposed S-box.

Table 3.7: Differential approximation probability

0	8	8	6	6	6	8	8	8	6	10	6	8	8	6	8
6	6	8	8	6	6	6	10	6	6	8	6	8	6	8	6
6	8	8	6	8	8	6	8	6	10	6	6	6	8	6	8
8	8	6	6	10	6	6	8	6	8	6	8	8	6	6	6
8	8	6	6	8	6	6	6	6	8	6	8	6	6	8	6
6	6	8	6	6	6	6	6	6	8	6	6	10	6	10	8
6	8	6	8	6	6	6	6	6	8	6	6	6	8	8	6
6	6	6	6	6	8	8	8	8	6	6	6	6	6	6	8
6	6	4	6	8	8	8	6	8	8	10	6	8	6	6	6
6	6	10	6	8	8	6	6	8	6	6	6	6	4	8	8
6	6	6	8	6	6	8	8	6	6	6	8	6	8	6	6
6	8	8	6	6	6	8	8	6	6	6	6	8	8	8	6
6	8	8	6	6	10	6	8	6	6	6	6	8	6	6	6
6	8	6	6	6	6	10	8	8	6	6	8	8	10	6	6
6	6	10	8	8	6	8	6	8	6	6	6	4	6	6	6
6	8	8	10	8	6	6	8	6	6	6	6	6	6	8	8

The maximum value in Table 3.7 is 10, which only appears thirteen times in the Table. When 10 is divided by 256, the value of DP is equal to 0.03906. Table 3.8 displays the comparison of S-box values with other DP values.

Table 3.8: Comparison of DP between proposed S-box with existing S-box

S-boxes	Khan et al. [42]	Wang et al. [44]	Özkaynak et al.[46]	Proposed [19]
Max DP	0.03906	0.0468	0.0468	0.03906

Thus, the proposed method is strong enough to withstand the differential attack.

3.3.4 Bit Independence Criterion

A function (g) justifies the bit-independence criterion for the input (u) and output (v, w) in such a way that if the input bit (u) is changed then the output bits (v, w) should change independently. Correlation must be calculated to measure relationship between the avalanche variable sets. The average value of BIC-NL is 103.857 with a minimum of 96 and maximum of 108. The BIC-NL of the proposed S-box method is shown in Table 3.9.

TABLE 3.9: BIC-NL

–	100	106	102	102	104	102	106
100	–	102	108	104	98	104	106
106	102	–	96	102	106	106	106
102	108	96	–	104	106	108	104
102	104	102	104	–	98	104	106
104	98	106	106	98	–	106	106
102	104	106	108	104	106	–	106
106	106	106	104	106	106	106	–

The comparison of BIC-NL values of proposed S-box with the existing S-box methods are shown in Table 3.10.

TABLE 3.10: Comparison of BIC-NL between proposed S-box with existing S-box

S-boxes	Çavuşoğlu et al.[45]	Khan et al.[42]	Alkhaldi et al.[40]	Proposed[19]
BIC-NL	103.3	100.3	102.9	103.857

Thus, it shows that the proposed S-box method significantly justifies nonlinearity based bit independence criterion.

3.3.5 Linear Probability

The linear probability was introduced in 1993 to break the 8-rounds of Data Encryption Standard [48]. The use of LP is to compute the resistence of the linear cryptanalysis and its formula is already defined in Section 2.4.3. Maximum LP of proposed S-box is just 0.1328, indicating that it is resistant to linear cryptanalysis.

Table 3.11 shows the maximum LP value and its comparison with maximum LP value of earlier S-Box methods.

TABLE 3.11: Comparison of LP between proposed S-box with existing S-box

S-boxes	Hussain et al. [41]	Khan et al. [42]	Hussain et al. [38]	Proposed [19]
Max LP	0.125	0.140	0.1328	0.1328

Thus, the comparison shows that proposed method is effective and resistant to linear attacks.

3.3.6 Difference Percentage

This is a very important factor in analyzing the strength of S-boxes. This test analyzes that if only single bit of key is changed, then how many values are relocated to distinct position than previously generated S-box. Another important feature of this test is that it enhances the avalanche effect and enhances safety. To conduct this test, many S-boxes were created using various keys, and then further S-boxes were created by altering one bit of each key from various keys. Then all the newly S-boxes which were generated compared with previous S-boxes. But the proposed method can generate a new S-box with a unique value by only changing in one bit of the input key.

Table 3.12 shows the newly generated S-box with different key, where the key is 74 73 66 79 20 75 20 74 75 6B 20 67 61 6E 6D 68

TABLE 3.12: New S-Box by changing key

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	08	32	80	3c	8c	96	3a	19	0a	53	7e	76	d9	14	1e	ae
1	63	98	44	82	d0	a1	87	40	f2	a0	a3	93	13	0e	1f	60
2	6a	b8	b2	7f	e3	9d	09	8d	1a	34	a8	cf	35	06	20	25
3	c5	aa	6d	c9	e9	cb	00	90	28	46	77	22	65	21	95	7b
4	62	49	f1	9c	ba	10	01	c0	4d	5c	e8	d2	ee	2b	6e	72
5	ed	4a	2a	c4	b3	5d	2d	6f	ff	7a	a4	fa	e5	37	a6	92
6	6c	f3	83	ef	a9	e4	f7	74	7c	0b	b9	9a	f5	e2	f6	59
7	f9	75	17	47	2e	61	bb	5f	c6	85	0d	02	55	5e	50	de
8	fb	4f	0c	97	73	78	3f	cc	52	ec	04	64	bc	88	84	3d
9	86	c3	45	e7	48	23	5b	e6	d5	16	8e	3b	3e	b5	4c	d7
a	33	d4	9b	b6	8b	c2	a7	39	dd	89	07	56	b4	be	d1	15
b	ab	ac	d6	e1	ca	c1	7d	a5	fe	4b	05	5a	d3	58	6b	24
c	c7	df	c8	9f	cd	29	2c	36	1d	f4	41	fd	68	03	18	da
d	af	dc	f0	79	11	30	ea	94	51	26	d8	2f	f8	57	8f	12
e	27	1c	81	0f	fc	bf	54	42	1b	b0	67	eb	db	66	43	4e
f	69	b7	9e	a2	b1	ad	e0	70	71	ce	38	8a	99	31	91	bd

The comparison of Tables 3.1 and 3.12 demonstrates that, even after changing the key, the generation of new S-boxes using the suggested method fully satisfies the criteria of difference percentage.

3.3.7 Other Properties of S-box

Here are the result of some other properties of proposed S-box. These properties are described in Chapter 2.

- S-box is Bijective.
- The number of fixed point is 1 and opposite fixed point of S-box is 3.
- S-box is Balanced.

- Algebraic degree is 7.
- Bent nonlinearity value of S-box is 116.6863.
- Hamming weight of all Boolean functions of S-box are given below:

S_j	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
HW	128	128	128	128	128	128	128	128

3.4 Comparison Between Newly Generated S-boxes of Proposed Method

With the help of proposed method, different S-boxes were generated with different keys and their values of different properties was calculated. The few generated different S-boxes are given in Appendix A and there calculated values are given below:

TABLE 3.13: Different properties of newly generated S-boxes

S-boxes	NL			BIC-NL	DP	SAC			LP
	Min	Max	Avg			Min	Max	Avg	
S_1	102	106	105	103.286	0.0468	0.4063	0.6094	0.5032	0.125
S_2	96	108	102.5	103.929	0.0468	0.4063	0.6094	0.5054	0.125
S_3	96	106	103.75	103.714	0.0468	0.3906	0.5938	0.4951	0.125
S_4	94	108	102.5	103.714	0.0468	0.3594	0.5938	0.4954	0.1328
S_5	100	108	105.25	102.357	0.03906	0.4063	0.5781	0.5005	0.1172
S_6	100	108	104.25	103.500	0.0468	0.3906	0.5781	0.5048	0.125
S_7	96	106	101.25	103.643	0.03906	0.3906	0.5938	0.4985	0.1328
S_8	98	108	103.75	103.214	0.0468	0.3906	0.6250	0.4983	0.1484
S_9	102	108	104.25	104.000	0.0468	0.3750	0.5938	0.5034	0.1484

Continued...

Continued from previous page

S-boxes	NL			BIC-NL	DP	SAC			LP
	Min	Max	Avg			Min	Max	Avg	
S_{10}	100	108	104.75	103.643	0.0468	0.3750	0.6250	0.4951	0.1406
S_{11}	102	108	104.5	103.786	0.03906	0.4063	0.6563	0.5068	0.1328
S_{12}	102	108	104.75	104.500	0.0468	0.4063	0.5781	0.5007	0.1406
S_{13}	98	104	102	103.429	0.0468	0.4219	0.5781	0.4954	0.125
S_{14}	100	104	102.75	102.429	0.03906	0.3906	0.6406	0.5034	0.1328
S_{15}	100	106	104	103.571	0.03906	0.3438	0.5938	0.4995	0.1172
S_{16}	102	108	105.75	102.643	0.0468	0.4063	0.6094	0.5049	0.1328
S_{17}	102	108	106	104.286	0.0468	0.3906	0.5781	0.4998	0.1328
S_{18}	98	108	101.5	102.357	0.0546	0.4063	0.6250	0.5081	0.1328
S_{19}	100	106	103	103.500	0.0468	0.4219	0.6250	0.4985	0.1328
S_{20}	102	106	104.25	102.786	0.03906	0.3438	0.6094	0.4946	0.1406
S_{21}	100	108	103.75	103.714	0.03906	0.3906	0.5938	0.5007	0.1406
S_{22}	98	106	103.25	102.071	0.03906	0.3906	0.6094	0.5081	0.125
S_{23}	98	108	102.75	102.643	0.03906	0.4219	0.5938	0.5083	0.1406
S_{24}	102	106	103.25	104.214	0.03906	0.3750	0.6563	0.5046	0.1328
S_{25}	102	106	104.25	104.000	0.03906	0.3750	0.5938	0.5046	0.1328
S_{26}	98	104	101.75	103.857	0.0468	0.3750	0.6250	0.4902	0.125
S_{27}	100	108	103.75	104.071	0.03906	0.4219	0.5938	0.5071	0.125
S_{28}	100	108	104.75	103.714	0.03906	0.3750	0.5938	0.4954	0.125
S_{29}	102	108	104.25	103.714	0.03906	0.4063	0.6094	0.5088	0.1172
S_{30}	104	106	104.75	103.929	0.03906	0.4063	0.6094	0.5054	0.1328

Table 3.13 shows the values of different S-boxes which are generated through proposed method in which the highest AvgNL value is 106 and lowest AvgNL value is 101.25, the highest DP value is 14/256 which is 0.0546 and lowest DP value is

10/256 which is 0.03906, the highest LP value is 0.1484 and lowest LP value is 0.1172, the highest BIC-NL value is 104.500 and lowest BIC-NL value is 102.071 and the SAC value are close to 0.5.

Thus, this shows that the proposed technique can produce several S-boxes with good properties.

Chapter 4

Application of S-box in Cryptography

An S-box (substitution-box) is a fundamental building block of symmetric key algorithms in cryptography that performs substitution. They usually serve to ensure Shannon's property of confusion in block ciphers by masking the connection between the key and the ciphertext. The performance and security level of an encryption method are directly impacted by the substitution box, which is the nonlinearity component of a symmetric key encryption technique. In this chapter, an efficient and secure method is proposed for dynamic S-box and chaos key-generator based image encryption scheme.

4.1 Chaos Theory

Chaos is the study of unexpected and nonlinear surprises. That is a simple technique to foresee the unexpected. Chaos theory is the branch of mathematics concerned with the behaviour of dynamic systems. Weather, stock market volatility, our mental states, and other nonlinear processes that are difficult to stabilize or regulate are all covered by the chaos theory. Almost all chaos-based cryptographic algorithms employ dynamic systems based on a set of real numbers, which are challenging for practical realization and circuit implementation. Chaos cryptography refers to the use of chaos in secret writing. The study of a quick secure system design is known as chaos cryptography. The system dynamics can function in a guaranteed state of chaos, as determined by the traffic model. A dynamic system is one whose operation depends on a time-dependent point in a geometrical space, such as the motion of a pendulum or the flow of water through a pipe. Any map that exhibits chaotic activity is said to as chaotic. The time parameter could be discrete or continuous. Discrete maps are appropriate forms of iterated functions.

4.1.1 Properties of Chaotic System

In practically all nonlinear deterministic systems, the chaos phenomenon can be encountered. When long-term mathematical function progresses in a continuous and haphazard manner, chaos appears to exist. Chaotic systems include the following characteristics:

- **Apparently arbitrary but totally deterministic behaviour**

  Although the behaviour of a chaotic system seems random but it is completely predictable. Therefore, the same output value set is produced by an iterative chaotic system with the identical initial conditions.

- **Long-term Prediction**

  Small variations in the initial conditions, such as those brought on by measurement mistakes or rounding errors in numerical computation, can result in dramatically different results for these dynamical systems making long-term prediction is generally challenging.

- **Sensitivite to Initial Conditions**

  Sensitivity to initial conditions implies that a systems behaviour might diverge quickly due to slightly altered conditions, making it unpredictable.

4.1.2 Lyapunov Exponent

In the study of dynamical systems, the term "Lyapunov Exponent" has been frequently used [49]. LE describes the degree of divergence between two closely spaced dynamical system trajectories. A positive LE means that the two trajectories diverge more and more with each iteration until they are completely different, regardless of how close they are to one another. The LE of a chaotic dynamical system is hence positive. The number of LE in a multidimensional dynamical system may be many. Its close paths exponentially diverge in various dimensions if it has more than one positive LE. Hyperchaotic behaviour is the term given to this occurrence. The performance of a dynamical system with hyperchaotic behaviour is very high in terms of chaos, and the results are unpredictable. LE can be defined as:

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln \mid g^{'}(y_i) \mid \tag{4.1}$$

where $g(y_i)$ is the chaotic system's function. There are three dynamical scenarios for the LE.

1. All LE are less than zero if the orbit attracts toward a stable point.
2. The system is neutrally stable when the LE is zero such system are conservative and in a steady state mode. They display Lyapunove stability.
3. All LE are greater than zero if the system is chaotic.

4.1.3 Bifurcation Diagram

A bifurcation, also known as a period doubling or transition from an M-point attractor to a $2M$-point attractor, takes place when the control parameter is changed. A bifurcation diagram is a visual representation of the series of period-doubling that take place as the control parameter r rises. Figure 4.1 shows the bifurcation diagram of logistic chaotic map, with r as the horizontal axis. The system is allowed to settle for each value of r before plotting successive values of x over a few hundred iterations.

4.1.4 Logistic Map

One of the most well-known 1D chaotic maps is the logistic map, which has a straightforward mathematical foundation but complex chaotic behaviour. The logistic map's mathematical representation is

$$x_{n+1} = rx_n(1 - x_n) \tag{4.2}$$

where r is the control parameter and x_n is the initial condition of the equation (4.2).

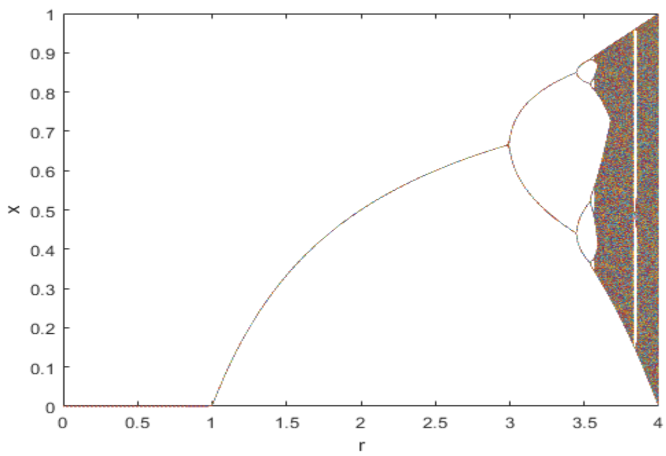

FIGURE 4.1: Bifurcation Diagram of Logistic map

Figure 4.1 makes it obvious that every point is shown at zero when the value is $r \leq 1$. As a result, there is only one point attractor for $r \leq 1$. There are still one point attractors for $r \in (1, 3)$, but the value of x that is attracted increases as r rises. Bifurcation occurs at $r=$ 3, 3.45, 3.54, 3.564, 3.569, etc. up until immediately after 3.57 is the point at which chaos takes over. However, the system is not chaotic for all values of $r \in [3.57, 4]$, and there are some points where three point attractors are visible.

4.1.5 Tent Map

Tent map is a 1D discrete chaotic iterative map that displays tent-like shape. It is also referred to as a triangular map. The following is mathematical model of tent map:

$$x_{n+1} = \begin{cases} \mu x_n & if \quad 0 < x_n < 0.5 \\ \mu(1 - x_n) & if \quad 0.5 \leq x_n < 1 \end{cases} \tag{4.3}$$

where $\mu \; \varepsilon \; [0, 2]$ is the control parameter which takes a positive real number and $x_0 \; \varepsilon \; [0, 1]$ is the initial condition of equation (4.3).

Figure 4.2 shows bifurcation diagram which reveals the following details. For $\mu \; \varepsilon \; [0, 1)$ the equation converges to $x = 0$ in the tent map, which has one fixed point at $x = 0$. For $\mu = 1$ all values of $x \leq 0.5$ are fixed points of system. For μ between 1 and 2, the system produces an unstable, chaotic sequence. Tent map exhibits fixed point behaviour when the $\mu < 1$ and chaotic behaviour when the $\mu > 1$.

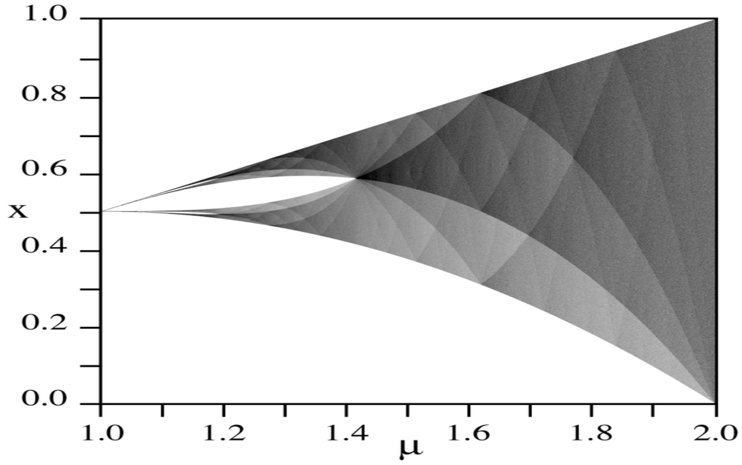

FIGURE 4.2: Bifurcation Diagram of Tent map

The logistic map and tent map both experienced issues with the output state values being distributed unevenly and having a tiny chaotic range.

4.1.6 The Tent Logistic system

Lu et al. [50] developed the tent-logistic system, a new compound system that combines the tent and logistic map, to address the problems with the logistic and tent maps (TLS). This is its mathematical model:

$$x_{n+1} = \begin{cases} \frac{4(9-\mu)}{9} x_n(1-x_n) + \frac{2\mu}{9}(x_n) & x_n < 0.5 \\ \frac{4(9-\mu)}{9} x_n(1-x_n) + \frac{2\mu}{9}(1-x_n) & x_n \geq 0.5 \end{cases} \tag{4.4}$$

where $\mu \in [0,9]$ is the control parameter. When $\mu = 0$, the above equation behaves as a logistic map; however, when $\mu = 9$, it degenerates into a tent-shaped chaotic map. Figure 4.3 displays the TLS bifurcation diagram. It is clear from Figure 4.3 that the chaotic range was significantly greater than the logistic or tent map ranges, being the entire range $[0,9]$. Its output sequences are uniformly distributed. Therefore, compared to the logistic and tent maps, the TLS performed better under chaotic conditions.

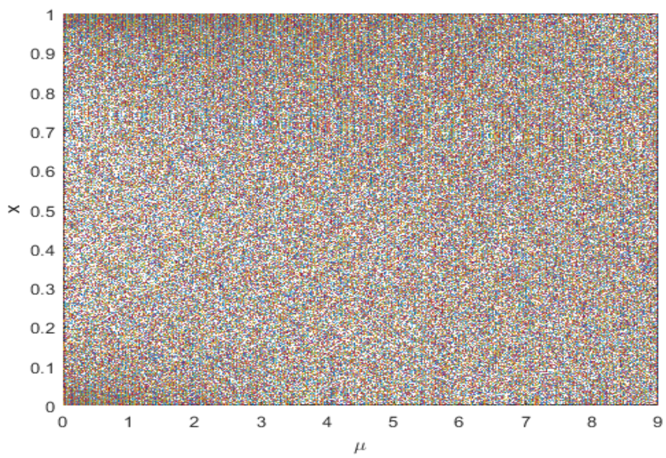

Figure 4.3: Bifurcation Diagram of Tent-Logistic map

In comparison to logistic and tent maps, the tent-logistic approach has two advantages. First, the tent-logistic system's chaotic range was significantly larger than that of the logistic and tent maps. If the system parameter was used as the cryptosystem secret key, the key space of the cryptosystem based on the new system would be significantly bigger. Second, over the entire 0 to 1 value range, the tent-logistic system had equally spaced output sequences. These advantages made the suggested tent-logistic technique more appropriate for cryptography applications.

4.2 Proposed Cryptosystem for Image Encryption

In this section, we go over the detailed process for both the proposed image encryption and decryption utilizing chaotic tent-logistic system and S-box. S-box is generated through the Algorithm 3.2, used as a lookup table for the replacement of pixels. Then the three chaotic sequences are generated and the bitwise XOR operation is implemented with substituted pixel values of each image component. Figure 4.4 show the flow diagram of proposed algorithm.

4.2.1 Key management

An external secret key is used for generating initial condition of chaotic map. The 128-bit form of the external secret key is represented by

$$W(i) = w_{127} w_{126} \cdots w_0 \tag{4.5}$$

and

$$W_i = k_1 k_2 \cdots k_{16} \tag{4.6}$$

Each k_i is an 8 bit block of the secret key. From the above blocks, the following unique initial condition x and three parameters μ_1, μ_2, μ_3 are derived:

$$x = \frac{k_1 \oplus k_2 \oplus k_3 \oplus \cdots \oplus k_{16}}{2^8} \tag{4.7}$$

and

$$\mu_1 = (w_1 \times 2^3 + w_2 \times 2^2 + w_3 \times 2^1 + w_4 \times 2^0 + w_5 \times 2^{-1} + \cdots + w_{11} \times 2^{-7}) \mod 9$$

$$\mu_2 = (w_{12} \times 2^3 + w_{13} \times 2^2 + w_{14} \times 2^1 + w_{15} \times 2^0 + w_{16} \times 2^{-1} + \cdots + w_{22} \times 2^{-7}) \mod 9$$

$$\mu_3 = (w_{23} \times 2^3 + w_{24} \times 2^2 + w_{25} \times 2^1 + w_{26} \times 2^0 + w_{27} \times 2^{-1} + \cdots + w_{33} \times 2^{-7}) \mod 9$$

Algorithm 4.2.1. (Image encryption algorithm)

Input: Image I, Algorithm 3.2, Secret key k, Tent Logistic map

Output: Encrypted image C

Step 1: Read the provided image I.

Step 2: Separate the colour image I into its Red, Green, and Blue (RGB) primary colour components.

Step 3: Input 128 bits secret key (16 hexadecimal) in Algorithm 3.2 to generate an S-box.

Step 4: Use S-box(1D) as lookup table and apply on every component of image I to get the substitution.

Step 5: Convert the substituted colour components into a one-dimensional array.

Step 6: Iterate the tent-logistic map for L times with the initial state x and the control parameters μ_1, μ_2, μ_3 to generate three chaotic sequences.

Step 7: To remove the negative effects of transient process, discard the first n_0 values from L, i.e., $L1 = L - n_0$.

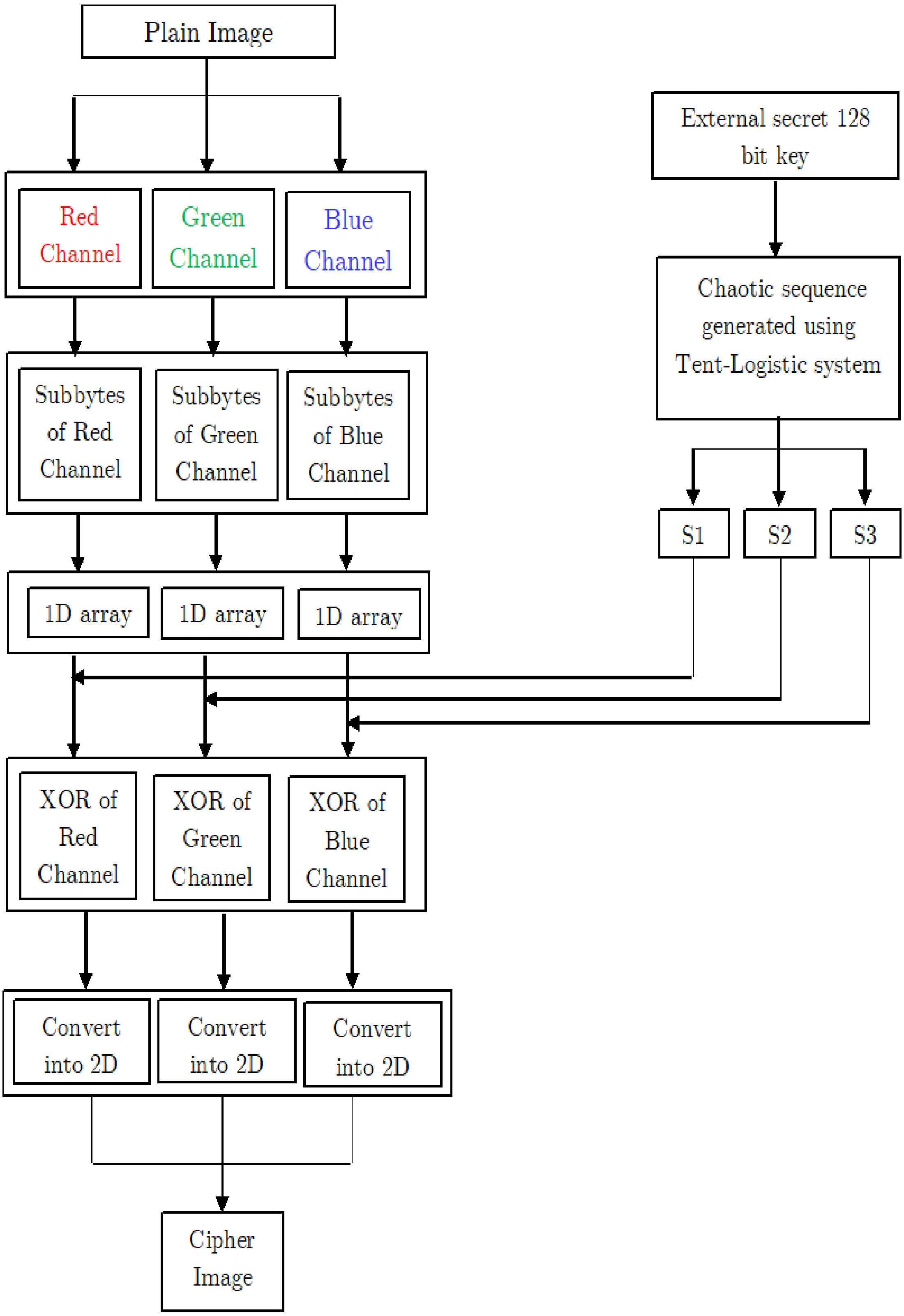

FIGURE 4.4: Flow diagram of proposed image encryption

Step 8: Apply the below given relations to transform the obtained sequence into 8-bit integer values

$$x_i = \mod(floor(x_i \times 10^{14}), 256), i = 1, 2, ..., L1,$$

$$y_i = \mod(floor(y_i \times 10^{14}), 256), i = 1, 2, ..., L1,$$

$$z_i = \mod(floor(z_i \times 10^{14}), 256), i = 1, 2, ..., L1,$$

where $floor(x)$ returns the greatest integer less than or equal to x and mod returns the residual after dividing by 256. As a result, the output sequences fall within the $[0, 255]$ range.

Step 9: Using the previously created chaotic sequence, diffuse each colour component's separability as follows:

For red component:

$$R^{'}(1) = R(1) \oplus x(1) \mod 256$$

$$R^{'}(i) = ((R(i) \oplus x(i)) \oplus R^{'}(i-1)) \mod 256 \quad ; \ 2 \leq i \leq L1$$

For green component:

$$G^{'}(1) = G(1) \oplus y(1) \mod 256$$

$$G^{'}(i) = ((G(i) \oplus y(i)) \oplus G^{'}(i-1)) \mod 256 \quad ; \ 2 \leq i \leq L1$$

For blue component:

$$B^{'}(1) = B(1) \oplus z(1) \mod 256$$

$$B^{'}(i) = ((B(i) \oplus z(i)) \oplus B^{'}(i-1)) \mod 256 \quad ; \ 2 \leq i \leq L1$$

Step 10: Convert the obtained $R^{'}, G^{'}$ *and* $B^{'}$ components into two dimensional array and combine these color components to get the ciphered image C.

The following decryption algorithm can be used to restore the original image of the cipher image C.

Algorithm 4.2.2. (Image decryption algorithm)

Input: Cipher image C, Algorithm 3.2, Secret key k, Tent Logistic map

Output: Original image I

Step 1: Read the cipher image C.

Step 2: Separate the cipher image C into its Red $(R^{'})$, Green $(G^{'})$, and Blue $(B^{'})$ primary colour components.

Step 3: Convert the colour components into a one-dimensional array.

Step 4: Iterate the tent-logistic map for L times with the initial state x and the control parameters μ_1, μ_2, μ_3 to generate three random sequences.

Step 5: To remove the negative effects of transient process, discard the first n_0 values from L, i.e., $L1 = L - n_0$.

Step 6: Apply the below relation to transform the obtained sequence into 8-bit integer values

$$x_i = \mod (floor(x_i \times 10^{14}), 256), i = 1, 2, ..., L1,$$

$$y_i = \mod (floor(y_i \times 10^{14}), 256), i = 1, 2, ..., L1,$$

$$z_i = \mod (floor(z_i \times 10^{14}), 256), i = 1, 2, ..., L1,$$

where $floor(x)$ returns the greatest integer less than or equal to x and mod returns the residual after dividing by 256. As a result, the output sequences fall within the $[0, 255]$ range.

Step 7: Using the previously created chaotic sequence, decrypt each colour component's separability as follows:

For red component:

$$R(i) = ((R^{'}(i) \oplus x(i)) \oplus R^{'}(i-1)) \mod 256 \quad ; \ 2 \leq i \leq L1$$

$$R(1) = R^{'}(1) \oplus x(1) \mod 256$$

For green component:

$$G(i) = ((G^{'}(i) \oplus y(i)) \oplus G^{'}(i-1)) \mod 256 \quad ; \ 2 \leq i \leq L1$$

$$G(1) = G^{'}(1) \oplus y(1) \mod 256$$

For blue component:

$$B(i) = ((B^{'}(i) \oplus z(i)) \oplus B^{'}(i-1)) \mod 256 \quad ; \ 2 \leq i \leq L1$$

$$B(1) = B^{'}(1) \oplus z(1) \mod 256$$

Step 9: Convert the obtained $R, G \ and \ B$ components into two dimensional array.

Step 10: Input 128 bits secret key (16 hexadecimal) in Algorithm 3.2 to generate an S-box and then its inverse S-box(1D).

Step 11: Use inverse S-box as lookup table and apply on the every component of cipher image C and combine these color components to get the original image I.

4.3 Results and Discussion

The experimental findings are presented in this section. For the verification of our scheme, colour images named as Lena and aeroplane are taken. Results of the proposed scheme are shown in the Figure 4.5.

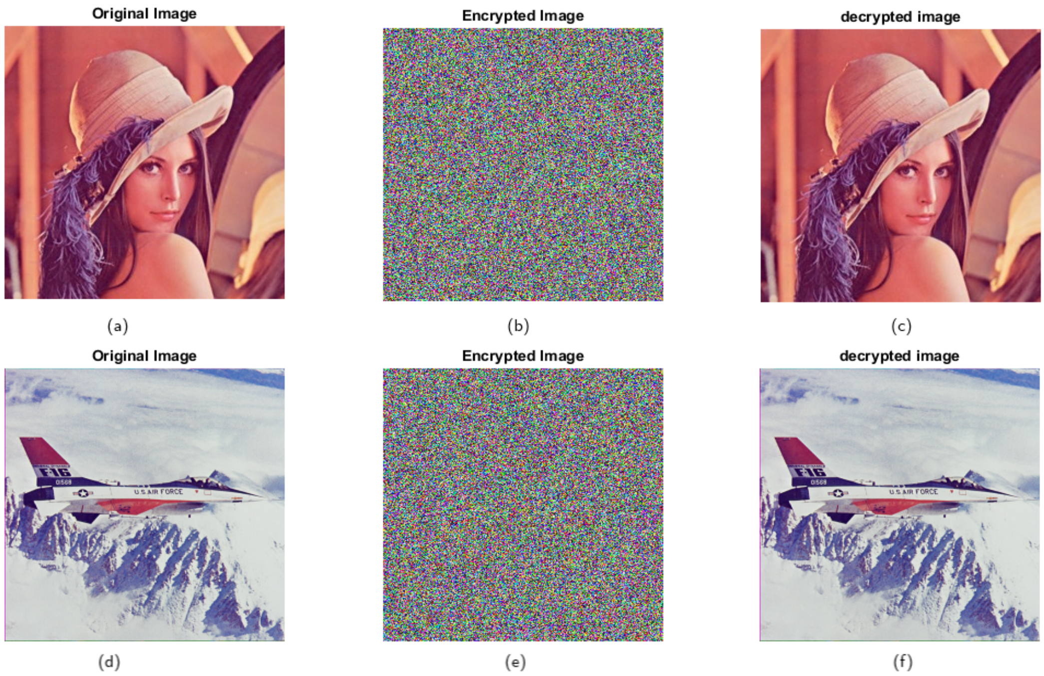

FIGURE 4.5: Experimental result for Original images of Lena (a) and aeroplane (d), Encrypted image (b) and (e), Decrypted image (c) and (f)

4.3.1 Performance Analysis

We applied the some popular security test to check the resistance of the several attacks to the proposed scheme. For this purpose, image of Lena of same size (256 $\times$ 256) are used.

- **Statistical Analysis**

  A secure cryptosystem must resist different types of attacks efficiently. For examining the resistance of proposed cryptosystem, we use the histogram test, key space analysis, key sensitivity, correlation coefficient and entropy test.

4.3.1.1 Histogram Test

The pictorial representation of each pixel intensity value and their frequencies is known as histogram [51]. A good encryption scheme should always give a cipher image with a uniform histogram distribution for any plain images. Figure 4.6a, 4.6b and 4.6c show the red, green and blue components of histogram of original image and Figure 4.6d, 4.6e and 4.6f give the histogram of encrypted image. It is clear that the histogram of encrypted image is different from the histogram of the original image and the histogram of cipher image is almost uniform then we conclude that the attacker cannot break the security of the proposed method.

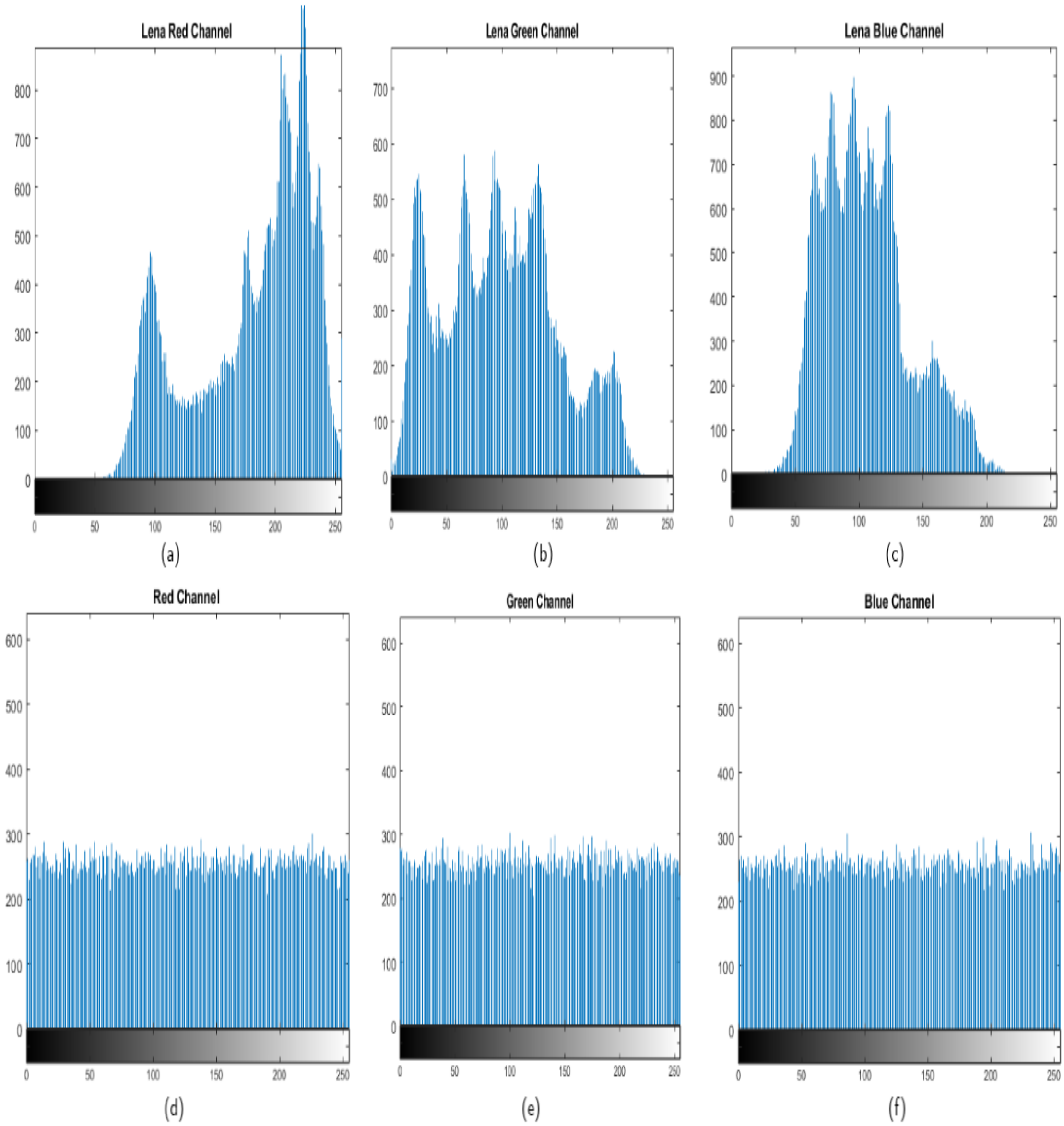

FIGURE 4.6: Histogram of Original image red component (a) Original image green component (b) Original image blue component (c) Encrypted image red component (d) Encrypted image green component (e) Encrypted image blue component (f)

4.3.1.2 Key Space Analysis

To prevent a brute force attack, a good encryption method should have a large key space. The suggested technique uses a 128-bit key. It includes 2^{128} distinct keys combinations. Therefore, the proposed scheme ensures a sufficiently large key space which are greater than 2^{104} to prevent the brute force attack.

4.3.1.3 Correlation Coefficient

Correlation means the relation of the neighbouring pixels in horizontal, vertical and diagonal directions. In digital images, pixels are highly correlated with each other. A cryptosystem is considered good if its break this strong correlation by applying the encryption algorithm [23]. Table 4.1 shows the correlation of the original image and encrypted images.

The value of correlation coefficient falls between the -1 and 1. In contrast to the -1 value, which indicates a decreasing linear relationship, the 1 value indicates an increasing linear relationship. The value "0" indicates that the two images are independent. Table 4.1 demonstrates that the correlation coefficient of the suggested technique is close to the zero, indicating that the plain image and cipher image are not linearly related with one another.

TABLE 4.1: Correlation coefficient of two neighbouring pixels

Scheme	Vertical	Horizontal	Diagonal
Original Image	0.9804	0.9585	0.9425
Encrypted Image	-0.0042	0.0010	-0.0020
Supriyo et al. [23]	0.0022	0.0026	-0.0008
Hafsa et al. [52]	-0.006	-0.0003	0.00014
Abduljabbar et al.[53]	0.0070	0.0033	0.0027

4.3.1.4 Key Sensitivity

In general, the key sensitivity means that a minor change in the keys would generate unique different cipher images. A good image encryption algorithm should be

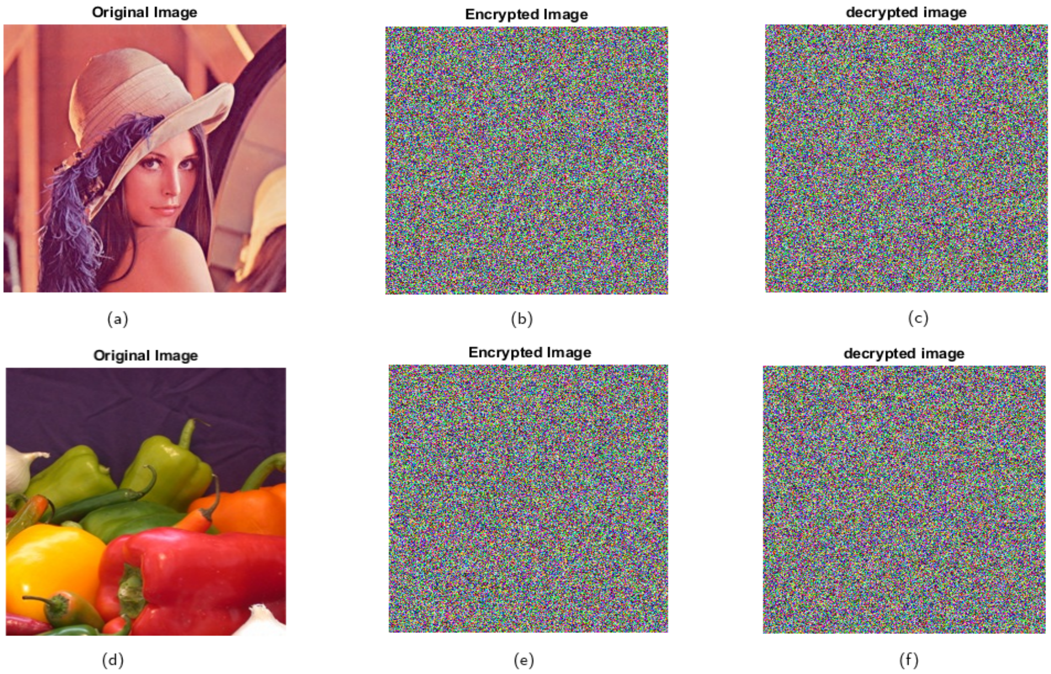

FIGURE 4.7: Key sensitivity analysis for Original image (a) and (d), Encrypted image (b) and (e), Decrypted image with slightly wrong key (c) and (f)

very sensitive to key that is utilized [54]. Suppose a key is obtained by changing the single bit of the original key . Then by the sensitivity of keys, we assume that the original image will not reveal because we change the original key. Suppose we consider the new key '7468617473206D79206B756E67206676'. Figure 4.7a and 4.7d indicate the original images, 4.7b and 4.7e indicate the encrypted images by using original key and 4.7c and 4.7f show the decrypted images after changing the key. Thus 4.7c and 4.7f indicate that the original images are not revealed. Hence, the proposed scheme is sensitive to key.

4.3.1.5 Entropy Test

It is a measuring tool to decide the level of irregularity of a data sequence. An ideal random data of 8 bit sequence should achieve the entropy value 8 [23]. It is a fundamental and efficient test for actually looking at whether the pixels of the encrypted images are arbitrary or not. Table 4.2 shows the values of plain image and cipher image.

It is clear that all the entropy values are close to 8, thus the proposed cryptosystem is appropriate for making high randomness in cipher images.

TABLE 4.2: Entropy Test

Encrypted Scheme	Entropy value
Encrypted Image	7.9990
Supriyo et al. [23]	7.9990
Hafsa et al. [52]	7.9998
Abduljabbar et al.[53]	7.99913

Chapter 5

Conclusion

In this thesis, reviewed of the scheme [19] based on key dependent S-box is discussed. This scheme used the simple functions like circular shift, XOR and nibble swap. The key used for generating S-box has 128 bits long. A straightforward and efficient method is used to build the S-box. The analysis shows that the constructed S-box has a good cryptographic properties.

Then we use this key dependent S-box in our image encryption scheme. In encryption phase, S-box is used for the confusion purpose and the compound chaotic map (tent-logistic map) is used for generating chaotic sequence. After that, a mixing technique was used to combine the values of the substituted image pixels with the generated sequence to get the cipher image. The decryption procedure is similarly the inverse of the encryption procedure. We obtained the plain image by reverse the order.

The suggested algorithm has offered resistance against various cryptographic attacks. The efficiency of the suggested method is demonstrated by the security analysis.

As a future work, the proposed scheme by Ejaz et al. [19] can be extended to 192 or 256 bits of different sizes of key and then the generated S-boxes can be used in image encryption scheme.

Appendix A

Key Dependent S-boxes

TABLE A.1: S_1

Key: 5B4BDA8BF0ED6914F8511338E9BE32DF

36	95	63	a2	b6	97	47	df	6c	94	71	3e	67	bb	b0	60
2b	b8	3c	8e	65	0f	25	d9	10	3b	b1	0b	a7	03	01	74
eb	e7	ef	08	e4	24	80	09	98	11	d5	cc	7a	f3	0c	c3
f7	0e	ce	8a	fe	05	bd	c0	ca	a6	f2	32	39	8c	de	ea
1f	0d	92	62	2d	1b	53	d4	fd	2e	9a	9d	e0	43	58	8b
20	81	84	69	ec	b4	ac	b9	bf	af	17	a4	4a	31	a0	48
90	b7	a9	6f	fc	1e	9b	68	2c	dc	ee	2f	86	34	50	1d
35	a8	27	61	38	07	91	51	44	7f	9c	fb	59	7b	ff	1a
c9	c6	9e	3f	aa	5f	40	6e	d6	21	54	ab	8f	fa	d0	4f
6d	f9	b2	f1	4c	d2	b3	1c	c7	a3	2a	22	16	cd	77	52
f8	e6	bc	13	3d	46	5a	8d	37	6a	7d	4d	a5	c2	42	28
c8	f0	e5	3a	0a	96	d7	6b	14	29	be	79	30	82	72	b5
76	d1	19	cb	db	15	99	e3	04	41	c4	f4	4e	26	23	89
06	83	d3	64	cf	5c	18	93	87	12	ba	85	55	00	5e	c1
5d	ed	45	56	e8	4b	7c	5b	f5	78	ad	c5	f6	e1	66	02
d8	75	33	88	da	49	9f	57	73	7e	e2	e9	ae	dd	70	a1

TABLE A.2: S_2

Key: D3E0F96232596B3BDE5D6457253C9CC9

bd	9d	3d	15	67	f5	f7	02	29	ba	63	24	36	44	fb	38
c4	85	08	30	14	ee	d8	a0	b6	e6	5a	93	ce	6e	62	f8
96	ea	8b	d2	1f	35	12	a9	87	91	2a	1c	19	cd	da	1e
a1	7c	86	73	7e	ef	69	5e	b5	39	6f	18	eb	82	76	b0
d4	57	89	7a	00	ff	5b	6b	3c	b7	b4	0d	d9	0e	27	9a
68	4a	77	b9	80	79	3a	e1	16	dc	7b	b8	e7	c1	55	34
65	7f	d6	32	37	f6	d3	8d	f4	88	e3	a2	5c	01	ae	cb
f0	81	f1	be	1a	49	ca	10	17	13	bf	c0	ab	a5	46	3f
51	fc	e2	b3	c2	94	ec	47	50	59	8c	c3	7d	95	31	fe
78	bb	66	bc	1d	f9	33	40	4c	e0	de	8e	a3	22	3e	98
af	a8	a6	97	71	cf	3b	74	a7	25	72	83	09	41	9e	e8
07	9f	2e	92	56	8f	dd	ad	11	e5	4e	0b	5d	48	99	04
0f	21	6d	1b	53	2b	d5	03	d7	0a	f2	20	06	9b	54	23
61	2f	c8	c5	b2	ac	45	64	2d	6c	84	60	c6	fd	a4	75
58	d1	4b	d0	df	43	fa	4d	db	0c	8a	aa	cc	ed	6a	b1
26	2c	c7	70	5f	f3	28	e9	52	05	90	42	4f	9c	c9	e4

TABLE A.3: S_3

Key: AE6021961EFA5DBDDA193797D1B00CAA

8e	2f	c2	30	e9	49	2a	e2	9e	77	0c	e8	a6	e5	8c	cc
c5	5b	af	0e	f7	54	6c	45	d5	fd	15	2e	92	ba	ee	ce
7c	58	fc	6d	d8	05	56	12	74	59	f6	63	25	7d	dc	91
fa	0a	3b	4f	96	c9	14	44	46	57	b4	95	c4	f3	3e	43
df	5f	3a	83	c3	21	64	bf	ab	88	11	9a	d6	ac	c1	98
b6	1e	69	d7	75	52	1b	48	37	1d	0d	27	a8	19	e1	06
20	9c	17	0f	a4	16	8a	b8	97	cd	ff	50	ec	e3	61	4c
8d	79	34	29	18	c7	71	6f	ae	b0	5e	ef	f4	39	8f	41
1f	36	53	07	2b	d9	13	4b	f8	9f	73	ea	3d	bd	fe	33
1a	5a	f1	9b	b5	04	68	dd	24	55	7b	d0	6e	c6	51	b1
7f	94	93	8b	42	de	38	1c	81	e4	b3	6a	85	d1	5d	72
67	40	00	5c	c8	c0	4e	bb	86	9d	7e	66	08	02	bc	a9
62	01	23	a5	a0	a1	2c	89	b9	a2	3c	a3	f5	65	82	70
76	aa	d3	87	f9	26	d4	a7	80	7a	da	f2	ed	78	09	60
b2	22	be	2d	cb	e6	ca	0b	f0	32	db	fb	4d	31	10	4a
e0	3f	90	e7	eb	84	35	b7	47	28	ad	03	cf	6b	d2	99

TABLE A.4: S_4

Key: B421B8CEA9FEB5332B451604756D7EAA

d7	29	bb	b2	27	f4	e2	66	be	6d	22	02	30	d8	aa	18
3b	af	db	59	97	dc	52	89	2c	c8	8a	a0	d9	43	f3	c7
a6	b3	ce	ef	b4	19	ec	bc	d1	b1	3c	57	c4	fb	14	7d
f5	9e	9f	71	8e	01	15	4d	47	fd	c5	e6	0f	85	68	96
54	6c	f8	77	33	82	f6	0a	98	0e	80	ee	ca	8d	87	a8
9a	34	5f	d6	49	b7	63	d0	64	ad	e1	7c	20	10	31	ab
e0	40	6a	cc	1b	bf	6f	70	a3	7b	eb	06	25	4b	c0	fa
ba	3e	c9	62	8f	1a	0b	ea	05	3f	e4	4a	c1	48	5a	fe
1f	df	9c	a4	f1	c3	23	a2	53	5c	7e	55	b0	16	24	8c
41	93	cf	88	26	90	46	cb	04	74	56	50	58	de	28	92
5d	b6	a9	b9	c6	91	36	83	11	2a	e3	ac	60	42	69	99
84	76	95	12	b5	bd	2e	a5	4c	d3	1c	07	1d	81	cd	d2
f0	dd	7f	a1	a7	0d	03	da	f9	4f	0c	38	e7	3d	ed	ff
e5	73	2f	2b	65	9b	35	6e	94	08	67	f2	f7	45	b8	17
79	5b	ae	e9	00	e8	37	8b	9d	13	09	fc	d5	44	61	2d
86	51	3a	d4	5e	4e	72	75	78	1e	32	39	c2	6b	7a	21

TABLE A.5: S_5

Key: 9B98C245F17B647A0A74373786AD2D45

19	ce	5f	27	77	6d	94	f3	b2	e4	1b	34	b0	87	44	1c
14	af	41	7e	03	43	5d	4e	99	d7	d6	56	e6	9d	d3	a7
fb	5b	2f	28	1f	b9	bf	46	c8	05	e3	e9	a4	c0	8c	68
13	eb	39	11	bb	ef	f5	61	54	f9	d4	22	cf	69	9f	71
bd	aa	ba	c5	a1	e8	75	31	66	29	18	00	c4	97	8f	21
5c	92	60	57	64	e7	79	ff	b1	ab	9a	7a	91	a6	6f	c9
24	0f	1d	ea	42	6b	02	48	49	23	7c	7d	4b	b8	2d	b7
95	93	1e	15	a3	0d	58	78	df	01	da	b6	62	35	c1	0e
10	f0	2e	37	f1	ed	83	86	db	5e	16	e0	63	06	de	3e
d1	b5	73	e5	a2	17	cb	7b	6a	c3	72	ee	f4	ad	c7	ca
84	40	a0	ec	38	8a	45	f8	9c	2c	90	f6	ae	4c	53	6c
d9	3d	09	96	f2	2b	3c	a5	d2	4a	3a	c2	2a	82	26	dd
0c	70	a8	08	59	3f	cd	51	76	80	1a	65	88	ac	07	e1
d5	85	55	12	5a	32	d0	be	0a	fc	36	bc	33	8b	b4	cc
89	8d	fe	fa	fd	3b	4f	a9	47	8e	30	9b	74	81	98	f7
b3	dc	4d	50	9e	52	67	6e	04	25	d8	e2	c6	0b	7f	20

Table A.6: S_6

Key: 997F13E286E63951AF630FC5AF921442

63	c1	72	92	47	d2	31	f1	75	ad	25	20	49	d3	c6	2c
a5	f4	95	07	59	41	ce	f0	5c	b0	ba	3c	27	9e	bc	14
97	29	24	f5	2b	e0	d7	6a	4b	cb	1e	a0	b2	9f	1c	04
df	b3	1f	39	cc	08	3f	02	90	cf	4f	7e	d9	33	7a	51
fe	32	c0	c8	e4	a9	0c	96	db	6c	e3	9d	89	de	0e	06
16	9b	26	c2	e5	28	68	05	4d	98	bd	a6	2e	8c	91	a2
a1	54	b5	b8	44	35	94	0a	50	f8	fa	4e	74	40	69	88
43	5b	aa	85	34	a4	1a	9c	cd	6e	c9	e7	f9	0b	09	2d
d6	53	1d	6b	12	d4	f6	e8	62	00	84	b4	8d	64	c4	21
5e	7f	a3	3d	70	b1	11	86	5f	e2	87	7d	be	dd	b6	66
58	0f	ac	dc	bb	52	38	3a	ae	80	22	d0	4c	b9	17	45
d1	83	2a	13	f3	d8	57	93	18	ab	5a	55	ea	73	15	76
ec	81	ff	ef	99	8f	9a	1b	6f	46	8e	30	36	fc	19	5d
23	3e	8b	da	f2	71	fd	e1	82	e6	4a	03	79	37	c7	d5
61	01	0d	56	7c	6d	e9	48	67	8a	77	ed	7b	f7	42	78
60	a7	b7	3b	fb	bf	ca	10	eb	65	ee	2f	c3	c5	a8	af

Table A.7: S_7

Key: 80C8847BD00DE9AC0317CD556D35F85B

0f	b7	9e	52	7d	7a	20	e3	f3	74	e6	38	db	ec	be	49
03	bb	66	44	98	73	b3	21	f0	0a	bd	ce	eb	63	77	5f
6f	7f	2a	e0	d6	93	ea	36	c7	4c	cc	b9	d5	4e	4f	32
19	0c	65	e4	4b	2e	df	30	57	05	33	22	2f	28	7c	ba
d8	1f	97	61	a5	e9	d3	76	c9	68	46	94	e2	1e	a1	79
11	2d	25	12	39	ca	9a	2b	75	e5	59	70	1c	af	c2	6c
de	72	53	b4	ef	1a	85	54	04	c3	91	84	0e	ab	a4	09
fb	35	8a	e8	aa	c4	7b	3a	8b	8d	3c	ad	b2	18	55	3b
56	d2	dd	86	ee	87	a2	f9	fc	34	d7	c0	1b	14	8c	89
cf	9f	d0	ac	7e	06	80	64	07	e7	01	6d	16	5c	d4	5d
9c	6b	ae	f6	92	69	3e	f7	1d	82	40	95	d9	31	5a	bf
da	62	8f	a8	90	17	e1	24	3d	ed	96	a6	41	fa	d1	ff
23	c8	51	8e	9d	4d	bc	cb	a9	60	9b	b5	5e	0b	c6	02
0d	f4	00	fd	71	81	26	a3	2c	c1	fe	b8	b0	50	b1	f8
a7	42	15	f5	37	47	48	43	45	6a	88	99	f2	3f	cd	a0
f1	6e	27	78	67	08	b6	c5	58	4a	83	29	10	13	dc	5b

Table A.8: S_8

Key: 199450DF1BC2BAB32E53C21FDF8DD6F7

3e	9c	06	26	72	6b	22	b1	a7	a5	43	5d	b9	d3	39	65
e2	d6	28	fe	ef	84	67	c4	01	38	2d	7b	f6	c0	6c	c6
8b	11	03	3c	9d	a9	aa	d8	98	27	13	66	ab	b2	53	fc
4c	81	be	1c	b8	b7	73	45	91	8e	46	a4	8a	61	52	d9
fb	25	19	cf	57	54	85	78	10	47	0b	df	b0	36	3f	f3
c5	99	1d	e6	4e	69	77	2e	9f	c9	9e	00	90	62	ac	d7
97	40	ee	35	70	15	7c	41	f5	e5	09	50	29	0e	f1	bd
4a	a6	75	96	74	a1	c2	24	f0	eb	07	55	f9	48	79	33
b4	82	ea	bb	0d	7d	0c	ed	ae	cd	ff	8c	14	5e	cb	49
a0	ce	71	6f	59	da	5b	56	4f	6d	c8	d1	16	e8	02	e3
a3	18	e1	0f	7a	21	23	a2	ba	e0	c3	bf	3b	2a	2b	e7
d2	34	fa	cc	44	17	05	76	d5	dc	7f	9a	04	b6	fd	3d
af	ec	89	8f	4b	88	f8	80	3a	b3	1f	de	4d	db	ca	2c
7e	ad	31	5f	83	5c	9b	c7	0a	37	58	20	b5	8d	f4	60
a8	64	5a	2f	86	94	92	12	c1	6a	dd	bc	63	d4	1b	95
d0	f2	1a	93	42	32	6e	30	1e	e9	e4	51	f7	87	08	68

TABLE A.9: S_9

Key: 32520F5B7025D503A7B20D9809A57597

4a	fa	00	61	b1	cc	50	40	dd	e1	d6	21	30	94	ef	7b
32	99	cf	f6	62	bf	ca	a2	b9	25	ed	33	90	9d	d4	4f
24	9a	2d	67	e6	26	3e	29	a4	a9	c4	70	c2	9e	bc	de
4d	9c	2a	6c	f3	48	55	47	58	ba	6e	fc	d7	3b	a8	85
9f	fb	f4	13	6a	22	76	b5	38	03	44	c8	53	06	27	14
b2	5b	39	5c	6d	35	cb	e8	45	16	18	0f	c3	0c	04	a3
96	0a	91	1a	d3	4c	1d	73	e7	ab	93	ea	54	68	46	e2
8d	e0	42	f5	ce	43	e9	08	c9	34	7f	80	51	49	05	3d
69	81	6b	c6	8c	5a	0b	a5	75	89	63	ac	82	1c	af	78
1e	bb	ec	20	92	3c	7e	c5	31	d0	d2	f7	f1	a6	66	87
2c	f2	b7	57	17	aa	e5	d5	1f	d9	98	5e	f8	df	d8	37
60	bd	4b	8f	b3	71	eb	15	28	11	77	23	88	65	8b	e4
86	0e	83	8e	0d	db	ad	56	be	7c	41	dc	59	fe	b0	2e
f0	b8	8a	a7	a1	a0	c1	79	ee	da	5f	e3	12	74	c0	02
95	10	01	09	19	4e	6f	9b	07	b4	f9	2f	64	c7	1b	52
d1	3a	7a	ae	fd	2b	b6	72	cd	ff	7d	3f	5d	36	97	84

TABLE A.10: S_{10}

Key: E4BAF860EA5405AC3F069F81AED0CD6D

bd	cb	76	1e	44	48	8c	1c	50	79	81	7e	ae	b9	c8	b8
eb	82	e2	3f	83	d0	a5	35	a0	10	b4	f6	59	e3	fd	97
60	c4	67	99	68	6b	6e	ac	2c	9d	0d	ad	2e	20	ec	a8
d9	bb	e7	93	24	11	18	c2	f0	2a	b7	dc	b5	21	ba	38
ef	6c	29	66	88	90	ce	dd	87	ee	47	d3	f4	5d	32	69
e8	4f	f3	80	b1	09	64	c1	8f	53	ca	8d	37	86	02	6a
3b	36	0c	56	be	30	1d	23	a6	cd	9f	55	c9	df	58	7a
ea	33	0b	04	91	d6	77	3d	39	de	e9	71	5c	19	a2	40
62	fe	98	00	2f	fb	3e	25	7d	75	d1	3a	a7	8e	6f	5e
96	c7	1a	0e	b2	7c	5f	e6	63	27	94	78	f8	aa	06	46
f7	c3	16	03	d4	54	01	e5	cc	43	8b	9b	70	12	41	f5
d8	5a	17	db	07	73	05	51	28	85	ab	9a	a9	b0	74	e1
a3	34	d7	52	c5	e0	7f	c6	2b	4e	9c	22	a1	31	bc	95
84	45	a4	14	fc	4b	bf	d2	af	0f	42	b6	92	0a	6d	57
ed	9e	1b	b3	fa	49	4c	65	2d	ff	89	13	08	3c	5b	8a
f1	7b	f2	1f	4a	cf	d5	61	da	f9	72	c0	e4	4d	26	15

TABLE A.11: S_{11}

Key: F665E74C071CCEBF77B937E5F1A8CA93

37	24	7a	ad	0d	01	2b	70	14	27	59	26	cf	77	da	d5
e8	c9	f8	41	18	4b	d9	e6	fb	c3	68	44	f5	29	ab	ba
8b	c1	4c	57	25	d4	a5	93	e2	2c	20	3f	a4	76	7c	6d
72	02	79	80	eb	f7	3e	51	48	56	af	85	b1	54	97	5a
95	6a	8c	05	96	75	c4	5f	10	7b	78	aa	3d	83	2f	0b
0e	f4	ea	a3	bb	53	bd	ed	7f	5d	d0	92	67	9a	39	a2
d3	90	0f	28	35	a7	0a	f3	31	65	c5	58	12	7e	9e	4f
be	5b	cd	86	ae	42	a8	94	69	21	1e	04	71	f9	36	ac
1f	b8	e4	c8	9d	13	16	e3	06	00	1a	03	a1	84	4a	34
e0	47	64	a0	62	b3	11	9f	d7	23	1b	46	2e	c6	3c	07
30	dc	5e	b5	2a	8f	d6	40	33	ee	9b	98	de	6f	3b	f1
e9	c2	32	09	49	b9	1d	d8	55	e1	50	ff	45	ce	89	ca
8d	4e	08	a6	f2	e7	4d	fd	99	fe	22	fc	63	bc	cc	a9
fa	8e	db	0c	66	c7	6b	43	df	f0	b4	ec	15	d2	3a	17
61	cb	74	60	91	c0	8a	dd	6c	19	bf	7d	2d	38	82	9c
5c	73	b6	6e	81	1c	ef	87	b0	f6	88	d1	b7	b2	e5	52

Table A.12: S_{12}

Key: DC67081378F34553B727CDD5A03CFADD

86	81	8b	bd	91	7f	1f	dc	41	1c	31	5a	e1	f2	b6	45
24	6c	ca	62	9e	f1	51	06	ac	02	ef	9c	73	78	fb	22
48	c1	d7	eb	d6	2a	a0	a6	08	23	bb	fe	c4	fc	cd	9d
09	01	77	bc	ee	c2	a8	be	33	df	c7	1b	5e	4e	03	8e
c5	af	6e	0b	68	ce	70	79	ba	fa	99	c6	b1	d1	19	8d
69	0a	d5	e2	76	e9	d0	de	4f	46	ff	1a	05	97	f3	92
6b	16	b7	82	59	ea	27	5b	83	0d	cc	e5	ec	74	e7	20
55	53	0f	4d	56	da	c8	60	f0	f5	c0	30	15	35	95	36
89	a7	ad	7a	1d	49	ab	63	11	b3	b9	38	ed	5f	84	3d
13	c3	c9	12	8c	a5	f4	f7	4c	1e	0c	2c	67	44	94	42
6d	00	aa	d4	50	a2	6f	7c	26	e8	3f	88	64	6a	e4	b5
28	e6	07	57	75	a3	9b	2e	9a	f6	04	7e	b8	25	96	b2
db	8f	39	d3	3b	98	8a	d9	93	7b	e3	58	52	2d	14	2f
cb	71	3c	29	b4	4b	b0	d2	54	4a	87	61	3e	9f	72	3a
43	21	a4	66	7d	dd	fd	2b	d8	32	37	85	bf	0e	65	e0
a9	17	47	10	ae	40	cf	f9	90	34	18	a1	5d	5c	f8	80

Table A.13: S_{13}

Key: 73A44DA7EFC9B8BA95AF669F1C3F57E1

95	7c	f9	7b	a7	fd	03	2c	c1	93	3b	11	d4	57	fb	e4
89	55	f4	cd	5d	1d	a3	7e	f3	2b	f5	bb	b0	cc	bd	df
06	8b	4d	b9	47	15	6b	b1	ba	97	ff	e9	db	cb	e0	3e
62	a5	58	d7	d2	c7	d1	51	bc	5b	14	34	75	70	77	94
fe	90	2e	12	79	9a	3a	02	45	ec	a8	fc	5c	53	a1	6a
e8	c6	d8	83	61	d3	71	e3	68	c8	81	0a	27	e5	f8	48
54	18	1f	1b	23	74	21	dc	2a	5e	2d	91	f0	6e	7a	13
42	c5	a4	4b	f7	a2	80	84	e1	43	39	08	0e	63	64	49
0c	4f	de	8e	3d	26	2f	ce	e6	8a	46	73	76	05	10	1a
d5	6d	ae	1e	29	41	8d	a6	67	56	ee	85	09	01	b3	f1
96	e2	8c	7d	f6	8f	30	dd	07	37	4c	c0	66	17	fa	eb
9e	24	c9	38	b6	ab	a9	72	0f	78	d9	52	1c	b5	88	28
af	da	7f	59	82	20	00	c4	b8	d0	4e	50	b7	0d	6f	a0
f2	9f	65	44	60	92	22	ca	31	35	98	cf	ed	d6	5f	b2
c3	9c	e7	25	9d	40	99	ad	3c	aa	6c	19	87	32	3f	04
16	ea	86	36	be	ac	c2	5a	4a	33	9b	0b	b4	bf	69	ef

Table A.14: S_{14}

Key: 695E13E9CB1055497874E325DEC7802A

26	52	e5	91	28	ab	af	ca	d6	f6	42	df	93	43	fc	9a
d9	84	74	fd	18	f4	8d	56	3a	7a	b6	83	5f	a2	dd	34
5b	41	96	f7	76	11	65	62	01	0b	21	ac	7b	b2	cb	85
3b	bb	51	cd	ae	33	7e	53	6c	89	15	f1	81	c5	38	b7
e9	20	9b	90	9f	6f	77	a6	de	e7	ad	25	70	55	2c	12
f2	04	40	d0	50	08	a4	54	5d	fb	d3	0e	b9	63	eb	6a
cc	78	a3	80	48	da	1a	d4	06	c4	09	3d	ee	13	d8	cf
73	75	44	0d	9d	e4	bf	f9	49	ff	27	61	c1	8b	c9	24
1e	87	6b	a8	4d	2f	ec	2d	5a	c3	94	72	36	4a	8f	3c
fa	14	8a	17	6e	1d	8e	a0	f3	22	1b	71	31	d7	66	05
f8	02	57	07	b5	10	0c	1f	fe	c6	4c	59	a1	d2	dc	64
a9	47	e2	e8	e1	98	b4	67	e6	a5	d5	5c	f5	b3	ce	46
e0	4f	ea	ba	2a	b0	ed	37	8c	c0	82	60	79	c7	30	3f
95	00	a7	db	e3	03	b8	be	86	b1	9c	99	35	1c	19	7d
ef	0a	f0	69	2b	aa	29	16	2e	7f	d1	23	92	c2	3e	c8
88	5e	68	32	4b	bd	bc	39	45	97	7c	58	6d	9e	0f	4e

Table A.15: S_{15}

Key: 1072FFEFBB51788C9FAE1DF3EABAC8C8

9b	4c	83	3a	7d	06	98	f1	66	8b	99	fb	fe	30	c9	df
47	eb	c1	20	6b	9e	53	55	17	81	09	5a	74	ef	5c	f3
a9	7c	bd	60	33	85	03	57	78	f9	21	fc	04	1e	71	e3
39	35	d5	90	64	c5	0a	9d	29	52	d6	b7	12	c3	ad	d9
c6	15	7b	aa	2e	05	62	77	4b	b4	38	72	d1	2a	8e	36
f0	82	6c	e1	cb	94	4e	db	a3	a6	a5	1d	24	ea	49	a7
5e	07	18	3c	93	be	fa	41	b8	95	46	b0	ff	91	88	97
f7	5d	3f	a0	11	0c	ee	76	ab	27	9c	2c	00	d0	08	c2
ba	c4	bc	7f	13	e7	70	e6	34	b1	e9	63	1b	8c	9a	d7
bb	cc	b3	e5	ec	6e	56	51	01	f2	31	68	87	8a	de	b2
23	ac	ae	32	6d	75	54	4a	2d	cf	96	6a	89	10	3e	dc
a2	26	f4	f8	da	73	92	b9	a8	40	8d	a4	67	48	3b	b5
8f	d4	65	d8	c7	ed	84	f5	9f	e2	0d	22	dd	a1	5b	3d
59	02	37	16	c0	0b	c8	80	25	43	d3	86	5f	28	e8	bf
1c	7a	d2	b6	7e	2b	6f	61	0f	4f	44	79	1f	19	e0	14
ce	f6	1a	e4	af	45	58	0e	cd	ca	2f	fd	69	50	42	4d

Table A.16: S_{16}

Key: 17A15998219B9995BE127EC06E4A53D4

53	63	46	be	90	af	93	84	6b	c8	bb	c6	4a	7e	50	49
df	1f	9a	5f	6e	eb	c5	79	c3	bc	f1	d8	9d	4e	8f	f5
05	25	ab	f7	43	52	89	96	76	fa	9f	b2	8e	72	32	94
3a	9b	d5	44	0c	27	88	12	de	71	56	10	13	2c	38	20
4b	59	36	5b	37	ac	2d	08	6d	a1	ea	ff	1a	5c	e5	7c
24	fe	ad	3d	ba	cd	62	0f	39	ca	35	22	0a	01	d7	29
0e	41	bd	a5	82	f4	c2	f2	fb	5d	7d	ec	86	45	33	b9
8c	b3	17	d4	f3	fc	23	c0	6f	66	4c	d6	da	7a	4d	2e
40	65	95	6c	70	d0	c4	54	8d	3e	85	30	b5	1d	a2	e2
28	d1	09	ef	c7	9e	b1	ee	74	fd	e9	02	64	b7	5e	58
73	21	ae	14	98	a7	5a	e4	b6	f9	8a	0b	07	91	a6	26
57	7b	a8	ce	69	ed	97	3c	f0	a4	d9	2b	0d	a3	a9	04
78	48	db	e3	31	51	cf	99	d3	1e	2a	d2	dd	c9	dc	83
b0	c1	e7	3b	bf	aa	60	75	03	80	19	cb	55	34	1b	11
4f	00	f6	6a	68	8b	87	e0	b4	7f	18	b8	e1	81	61	67
92	42	cc	9c	15	47	1c	a0	e6	e8	77	f8	3f	16	2f	06

Table A.17: S_{17}

Key: B7DB2E3DBFF51ABB7A0AB4249861C624

dd	91	1a	29	97	d4	9c	cf	f7	e3	01	6a	fe	41	40	13
75	6d	fb	11	57	5e	cc	f8	10	cd	7d	c2	55	e5	36	4c
a0	3b	fd	04	49	34	62	21	af	c6	de	cb	bd	bb	ca	17
84	6f	1b	82	51	e7	72	4d	15	32	2f	09	3c	50	fa	94
b0	45	1f	71	a3	7a	d7	83	f5	86	1c	ea	a2	5f	59	d0
6b	aa	20	74	78	9b	a7	f6	b7	37	68	46	87	c8	ae	48
f1	92	9d	df	3d	4f	0a	b3	c0	73	53	64	7b	5d	52	26
88	b9	ab	ba	ac	a1	00	22	a5	08	12	24	4e	47	7c	3f
2d	8f	54	5a	05	d9	c1	96	ed	e6	e0	2a	eb	0b	2b	e1
dc	db	5c	7e	8c	65	0c	03	ce	c7	c5	b1	d1	80	35	4a
98	31	18	a8	58	39	76	63	27	60	95	e8	e4	a6	ec	28
fc	02	06	d6	f9	79	0d	e9	44	90	bf	1d	b5	6c	1e	d5
c3	8e	e2	8b	f2	6e	14	8d	25	bc	c9	85	38	43	70	a4
93	0e	ff	19	f3	4b	5b	d8	89	b8	42	b6	f4	69	a9	ad
23	81	c4	7f	56	67	b2	ef	33	d2	77	16	b4	66	30	9e
0f	f0	9f	07	2c	3a	8a	9a	be	2e	da	ee	61	3e	99	d3

Table A.18: S_{18}

Key: 26017B7846C8C1497A656385C062A14B

00	79	91	60	f3	16	aa	89	71	a2	65	20	72	5c	0a	3e
40	01	44	e1	80	c3	3c	4e	64	db	c7	7a	74	43	36	b1
a4	1f	dd	ce	87	ba	35	02	8f	7e	ac	7f	2a	f8	70	52
08	ae	6e	9e	66	af	0d	05	37	04	28	5d	5f	83	be	c9
a5	c2	11	7d	ec	d3	9b	0e	54	09	ff	78	fb	77	c6	57
d5	94	d1	cf	fe	c4	b3	9f	bc	6a	ab	10	95	4a	18	14
3d	23	25	9c	88	17	a0	85	22	f7	d2	e0	ef	03	29	8e
9d	b5	a7	56	ed	12	bd	fc	93	98	97	d8	a6	f0	d9	0c
6c	73	f2	cc	f1	e9	92	ee	5e	a3	e4	96	86	13	a9	a1
b4	0f	a8	d7	07	3f	62	e7	2d	1d	8a	63	26	e3	9a	d0
2b	19	b9	30	38	8d	de	ad	58	cb	32	21	48	bf	31	c5
27	6b	b0	f6	15	50	69	0b	b6	da	99	b7	42	41	24	f5
b2	d4	5b	47	2c	2e	e5	b8	67	f4	3a	5a	75	cd	c1	34
4f	4b	4c	e8	39	2f	49	68	ea	dc	46	1e	06	1b	33	1c
e2	c8	6d	45	7c	c0	df	84	82	fa	4d	7b	90	f9	6f	55
59	e6	bb	8c	ca	76	fd	53	3b	61	51	81	1a	8b	d6	eb

Table A.19: S_{19}

Key: 9116BE22DC0F52DE859BB1D53D67C178

25	43	f3	fd	58	2a	e4	89	16	91	b0	67	82	71	fa	3f
bc	84	ce	2f	c4	1b	54	97	f9	11	75	ef	45	13	20	e8
76	35	0c	e2	dd	60	12	2b	68	17	88	90	1c	04	0a	3a
fc	d0	a6	22	a4	e6	98	93	74	38	09	0f	10	ee	40	2c
5e	c1	a1	a2	1d	7a	03	a8	07	50	19	56	f0	14	e9	9b
4c	f8	cb	a7	9e	c6	4d	4e	9f	61	c5	f6	63	62	cd	83
d8	9c	c7	b2	33	96	5a	c3	4f	34	57	ea	d7	8a	70	87
6d	44	f2	8e	bf	24	7e	d1	42	18	3c	be	a5	5b	ac	f7
b6	51	73	ab	aa	b3	bd	00	d4	8c	81	ba	2d	b4	92	f5
41	01	15	c8	08	05	4b	37	47	94	80	99	b7	31	5d	72
53	af	64	3e	95	6a	4a	36	2e	cc	d6	86	7c	26	b8	d9
cf	eb	f4	e5	f1	b1	9a	ff	8f	d2	39	3b	23	a3	bb	6b
fe	0d	c0	d3	69	6e	1f	28	ae	0b	a9	1a	da	6c	66	7f
06	46	85	1e	ec	c9	5f	0e	59	7b	77	e3	27	b9	e1	a0
8b	49	6f	db	8d	dc	55	02	48	b5	5c	de	ad	ca	52	32
7d	e7	3d	21	df	65	9d	79	e0	ed	d5	30	78	29	fb	c2

Table A.20: S_{20}

Key: F8115F19A74E6E824489C314C278FA76

79	39	31	18	ca	6a	2d	e9	b3	a1	bf	ef	38	0e	93	6b
13	b4	b9	a7	83	86	67	61	da	19	eb	57	47	87	5d	58
34	48	99	4b	3d	dc	e1	1a	af	ab	ad	f2	00	8c	25	14
0c	c8	7c	b1	ce	45	46	7e	40	e6	d8	64	3e	24	bc	cf
9b	7b	41	e4	9c	f3	a4	d2	a9	d1	17	95	b5	96	e2	ba
2a	69	0f	23	32	29	3c	33	4e	5c	d6	d9	27	01	90	22
c9	78	c5	49	9e	d0	ae	1d	fd	e8	80	20	66	a0	59	0b
8f	f9	28	4a	c7	bd	aa	12	10	76	8e	c3	2c	c0	cc	ec
ea	08	75	02	ff	42	03	9d	be	cb	06	56	26	fa	f5	1b
d5	6d	04	5b	0a	dd	3b	52	43	d7	71	7a	cd	bb	b7	50
ed	51	62	88	8a	e5	a3	b0	55	82	68	fe	0d	ee	3a	53
9a	c4	98	de	73	e3	11	6f	a6	94	65	f7	60	63	44	35
b2	36	8d	a8	b6	21	89	6e	6c	92	c2	2b	1e	16	a5	f0
4f	1f	1c	4d	f6	15	37	f4	72	09	07	f8	05	ac	81	fc
f1	db	7d	a2	c6	e7	2f	c1	8b	97	70	2e	30	3f	5a	d3
5e	4c	df	e0	5f	91	77	9f	7f	84	fb	54	b8	85	d4	74

Table A.21: S_{21}

Key: 3F796D94C919DFD1586891BEBEC76C62

4c	f7	a0	d9	7f	ca	ce	9d	3b	71	6e	4f	be	e2	a4	9b
04	dd	4e	b1	ee	5e	5d	6c	3d	76	82	7b	20	25	35	da
3c	7e	f8	d4	1e	62	3f	98	17	55	a1	f1	02	fb	8e	07
44	8c	8f	d1	6a	c9	84	f0	9e	38	56	ae	d6	2e	23	48
93	10	0d	46	ab	b9	5f	7a	f5	06	9f	39	15	7d	e9	34
f4	ed	a3	ff	09	61	d0	c3	7c	e8	83	32	bc	2c	a5	fa
ba	5a	c8	18	16	bf	27	13	81	2b	a9	c2	88	fe	58	40
d8	2a	3e	d7	cb	0f	cd	ad	49	e0	4a	50	70	94	78	c4
e7	fd	41	b7	ac	9c	21	e1	f6	dc	ec	45	24	e6	2f	26
66	8d	e5	a6	01	91	d5	3a	22	af	00	28	77	31	43	33
cc	0c	1b	14	df	1f	60	47	12	b8	4b	f3	bd	73	63	9a
30	b6	97	75	72	89	f2	c6	0a	e4	a2	90	67	59	05	b0
c1	2d	cf	e3	57	5c	ea	87	95	8b	b2	c7	11	bb	80	52
de	8a	37	aa	4d	eb	96	85	ef	74	6b	68	0e	69	54	a8
f9	0b	99	64	51	c0	6f	65	1a	a7	03	86	c5	92	1d	29
79	d3	08	db	b3	36	fc	d2	19	b4	6d	5b	b5	1c	42	53

Table A.22: S_{22}

Key: FD500064F73607A3409ACC28FB630166

b8	b7	59	89	81	aa	d0	7f	45	fd	49	60	cf	d6	93	dc
b2	4c	2a	ff	41	98	b5	63	47	2c	bc	30	d2	09	36	e4
87	c5	3b	f1	5b	c0	dd	df	ac	ae	96	ee	d8	bb	70	a1
08	20	fa	61	73	64	86	01	ed	0e	8d	15	62	11	22	a4
a7	33	54	0b	fe	05	3d	10	88	c6	83	a5	c8	ef	f6	17
42	71	7d	de	48	1c	e8	18	6d	92	bd	07	77	d1	94	f8
c3	2f	00	14	e5	44	3c	28	e6	29	1b	9f	5a	e9	99	c4
21	f5	69	12	5c	7a	b4	af	5d	bf	d4	56	4f	8f	f2	55
78	26	d5	eb	a2	27	82	d3	4a	76	da	95	8a	2b	ec	0f
4e	0a	3e	65	a6	53	66	37	fb	ad	06	1a	b6	c9	7b	25
8c	13	2d	43	03	35	9e	3a	f9	19	4d	b3	e3	5e	57	9b
39	fc	db	c1	51	80	75	79	a9	6a	6f	32	2e	e7	38	34
58	c7	67	9a	cd	f0	46	ca	50	f3	91	97	85	0c	b0	ce
8e	d7	0d	a0	31	04	6b	f4	72	3f	7c	e1	7e	84	4b	8b
40	24	be	6e	5f	e0	b1	68	9d	9c	1f	1e	cc	74	ba	b9
52	c2	6c	ea	a8	f7	1d	cb	02	16	a3	d9	23	ab	90	e2

Table A.23: S_{23}

Key: 03B31CBF08914063F43C7D12D3A005FA

76	ed	d3	68	7b	c6	5c	61	f4	17	f8	d8	34	2f	c0	0b
c2	00	24	b7	58	82	bf	a8	22	de	0d	93	f2	db	a7	fa
d1	8d	6e	3a	5e	ff	8b	50	19	69	6f	08	97	f0	9b	25
e5	3c	20	f1	da	67	9a	fe	0a	3f	b0	31	2d	cf	4b	5f
79	78	fd	51	ef	42	21	29	dc	a9	71	33	eb	7d	b3	8c
90	3d	7f	9f	c9	a2	52	91	40	1d	1a	c5	b4	39	e9	56
8f	f3	4e	48	95	18	b6	46	d7	a4	88	e1	47	1e	49	a1
81	a6	bc	04	f7	01	15	e8	94	bb	df	d5	ea	dd	f9	57
27	73	ee	b1	6b	6c	2b	e4	7c	55	e7	be	ca	36	72	a3
ab	8e	0c	37	aa	70	c1	5d	ba	0e	60	53	c8	c7	3e	98
38	1b	9e	9c	b9	bd	c4	d4	16	96	ce	59	02	1c	6a	e3
13	8a	3b	2e	af	cd	30	26	09	23	80	d9	d2	10	03	92
89	07	32	62	86	4c	d0	f6	11	77	63	e0	1f	14	ac	44
74	05	e2	66	54	f5	a0	6d	cb	2c	b8	5a	41	06	35	4a
12	cc	85	45	fc	64	87	ec	c3	0f	2a	fb	84	a5	ad	83
99	b2	5b	b5	4f	65	43	d6	7a	e6	28	ae	7e	9d	4d	75

Table A.24: S_{24}

Key: DFC046499907D7F3C3ED1BE6D3A4F43E

61	ea	7d	5a	15	19	6e	e8	18	9c	95	69	2f	79	90	f3
71	6f	43	db	85	09	e0	77	ab	58	de	52	cf	b3	e6	53
d2	25	35	e5	12	f8	f0	94	ac	63	65	17	26	f9	a7	74
3a	4d	66	0d	e2	bc	05	75	bd	72	2e	50	e3	34	27	d7
33	fa	d0	24	59	bf	42	e7	04	b2	d8	ad	41	7c	c6	6b
b6	e4	4e	10	a3	1b	ba	93	d6	78	21	57	06	c2	7b	96
32	5d	08	a4	f5	51	b8	46	31	cd	88	d1	f7	03	54	07
8d	62	28	b7	dd	1a	d4	7f	3f	1f	30	16	ce	f1	dc	48
d3	45	f6	a8	0a	9e	2d	2c	fb	cb	a5	8a	0f	c3	56	1e
3c	8f	99	23	84	ed	11	ae	c5	f2	0e	3e	1d	38	37	1c
3d	ec	81	a1	6c	d5	4a	ff	7a	af	76	70	c7	7e	e1	98
a9	b4	ee	8b	97	60	64	55	a6	ca	44	00	d9	67	c4	83
c1	cc	29	a2	91	a0	5c	14	9d	82	73	89	39	4c	13	40
8c	6a	c0	87	5e	01	49	da	fe	e9	b5	86	b0	be	4f	92
bb	c9	2a	8e	fc	2b	9f	aa	9b	df	6d	eb	22	3b	80	fd
47	4b	9a	5f	c8	20	f4	b1	36	5b	0b	0c	ef	68	b9	02

Table A.25: S_{25}

Key: 02EA0B0CB48686891C3F12F4160FA55E

7d	e1	5d	91	8f	2a	98	32	0d	f8	e8	ed	a6	e0	87	a7
8b	09	c1	86	1b	a5	a4	7b	1e	8c	b9	30	cc	1f	50	53
81	19	6e	f7	60	e2	5c	56	05	0f	07	c8	fc	45	ec	73
84	f9	1c	0b	f3	54	16	7a	28	3a	99	4e	40	e4	f1	c9
82	f5	ac	58	89	b8	d0	6b	3e	ba	61	2b	2e	06	2c	33
55	42	ff	03	c0	46	a0	1d	bb	2f	74	27	44	ee	4f	34
10	c4	6c	08	21	e5	ce	be	88	51	20	00	9f	c3	38	31
d2	76	62	d5	80	57	bf	9b	db	29	da	eb	e6	69	e7	41
9c	af	8e	95	c6	52	ab	7f	f0	23	48	c7	26	5a	12	f6
c5	66	b0	3d	01	f2	68	d8	cb	6f	a1	97	e9	72	a9	02
77	11	8a	fa	7c	04	ca	b3	78	3f	18	36	9a	96	15	ef
5e	5b	e3	3c	df	d9	cd	13	aa	83	6a	39	14	b7	35	17
4a	71	43	85	90	4c	cf	b1	67	4b	93	0a	dd	bc	a8	d4
a3	ae	ad	59	92	b2	1a	d6	fd	fb	37	70	0e	6d	75	3b
5f	0c	c2	d7	dc	94	65	22	63	8d	9d	b6	2d	25	9e	7e
d1	47	b4	fe	49	de	bd	ea	64	24	4d	b5	a2	f4	79	d3

Table A.26: S_{26}

Key: 6270C49F76860C72BD95FAA02A2CE7E1

fe	2f	f9	f1	ea	55	57	9c	b8	dc	fd	82	2d	7a	72	e6
8f	8c	90	c8	92	39	f6	31	26	08	88	f2	0b	ef	69	02
a2	71	4e	e7	6c	95	4c	91	6b	5c	6f	5e	48	80	89	ac
a1	9f	3b	af	62	53	b1	e8	06	37	77	d0	32	04	33	b9
ee	7c	3d	0c	e0	cc	46	cd	12	24	e5	97	d1	aa	01	6e
4f	21	03	70	0f	19	6d	be	38	29	50	a0	9a	1b	68	96
ae	ba	7b	ab	b6	99	25	a4	8a	c6	1e	5d	fb	42	c5	b0
eb	11	86	98	73	e1	f4	14	5a	74	1f	1d	7f	35	76	c0
dd	28	f8	a9	d7	56	0a	22	43	a3	db	ed	61	7d	8b	78
f7	2b	f0	d6	30	07	e3	60	09	3a	5f	1c	85	51	45	13
3f	87	63	10	49	cb	3c	47	fc	ec	6a	3e	d8	4d	c4	d4
c1	23	c9	18	8e	df	bb	c3	e2	15	27	0e	9d	e4	1a	0d
58	34	93	cf	bf	81	17	ad	84	05	94	2e	79	ce	36	9e
40	a5	e9	52	f3	67	75	54	fa	59	44	f5	d9	4a	2a	b2
bc	ca	bd	da	16	83	5b	2c	b5	d5	65	7e	ff	9b	b7	c7
20	d3	a6	de	d2	c2	a7	a8	00	4b	41	8d	b3	64	66	b4

Table A.27: S_{27}

Key: 14293B8741FA07CFB5B361806223854E

60	68	96	86	c1	f8	66	a0	fb	85	83	23	cd	ff	e8	d2
ed	9e	3c	4b	c7	00	2d	b0	6c	9d	8b	2f	12	31	f0	b8
c3	70	03	cb	ac	49	f4	f9	21	d6	b9	73	d3	56	17	b2
5d	aa	28	bd	e1	e7	1e	97	a1	b6	30	7d	33	65	d7	71
a8	88	26	e3	a5	08	35	bc	46	4a	90	34	0f	10	4f	dc
c9	bf	92	3d	76	ca	1f	e2	39	a9	2a	50	4e	7b	e6	7a
06	7f	62	f3	3f	72	4c	38	ce	c0	74	11	02	6d	19	d0
bb	87	5f	5a	c6	6f	6e	32	a2	1c	de	8e	94	77	61	a3
ee	2c	99	93	75	79	2b	3b	8d	59	54	80	b7	09	1a	51
41	e4	29	22	15	3e	d9	ea	89	64	b5	cf	fd	69	5b	52
fa	0a	c8	c4	da	7c	8f	8c	48	6b	af	d8	e0	f6	0b	43
9a	ba	9c	84	6a	df	b1	d5	a4	dd	58	0c	c2	9b	55	36
44	82	cc	ec	98	a7	07	78	45	67	04	53	fc	db	20	be
2e	0e	eb	1d	95	ef	81	7e	d1	f1	16	ad	47	ab	0d	27
01	14	8a	13	9f	63	d4	91	f7	b4	3a	b3	18	f5	42	05
ae	25	c5	40	1b	5c	37	e5	5e	24	e9	57	f2	4d	a6	fe

Table A.28: S_{28}

Key: B5FA3BB3367152AAFBA62075095853EF

bc	38	87	7c	5e	02	82	9a	b1	ff	88	4d	18	64	eb	78
f6	d6	62	8c	f2	9e	70	84	c0	42	c6	6d	49	06	e2	d5
db	a3	fb	66	41	29	68	d4	91	f9	9f	be	33	2d	e8	1d
cb	3b	1e	90	44	8d	65	24	8e	08	e4	cf	6a	03	fd	ba
50	9d	74	01	ca	d0	3a	7b	b5	98	c5	2c	ad	00	16	52
09	b3	92	32	56	46	8a	ab	a6	d2	d3	48	ea	f1	15	3e
71	0e	0c	6b	54	5d	23	4f	e5	1f	fa	25	f3	bb	53	d9
e3	1a	0b	96	c1	99	a1	3c	b4	de	da	dc	ce	c3	10	0a
61	5b	f5	0d	2f	76	d1	5c	a2	3f	39	fc	b2	07	a5	d7
ed	57	7a	a7	b0	c8	b6	11	b9	2e	73	28	f4	22	9b	bd
af	75	43	45	4e	fe	93	31	7f	dd	27	e1	2b	f0	19	cd
4c	8b	13	3d	e9	f8	36	a8	63	c9	77	e0	6e	60	17	80
ae	8f	97	c7	a0	94	df	35	55	7d	89	a4	7e	5a	6c	26
c2	21	69	05	1b	5f	67	ee	86	30	cc	47	34	b8	b7	9c
14	d8	6f	72	59	04	c4	ac	aa	81	f7	40	37	bf	58	12
4a	51	2a	85	83	e7	4b	e6	0f	79	ec	1c	a9	ef	95	20

Table A.29: S_{29}

Key: 456232F006BF6EB926109CEB90CB0C91

1a	d8	ac	8b	e6	95	1d	88	52	cf	2d	44	20	86	ef	30
7d	da	c3	be	19	93	e0	18	83	6e	39	80	87	70	f5	a1
b5	4b	60	76	9c	05	bf	3b	f3	a7	ab	78	69	a6	9e	c9
41	bb	5a	e7	d2	65	db	66	99	15	a2	91	56	28	af	de
94	9d	d0	eb	bd	75	5f	97	ff	23	4f	a3	17	64	e9	9b
10	74	e3	f2	c1	11	61	6d	4a	0d	09	2b	43	b8	14	36
57	f1	67	1c	40	f7	c8	aa	55	02	6a	3f	c5	a8	d5	68
fd	53	cd	ae	7f	0b	50	98	8f	e8	0a	c2	df	48	82	c7
6b	a0	4e	04	59	d4	0f	e5	22	c6	2f	96	2e	51	07	fa
47	ee	13	ea	35	b3	1f	16	b1	08	79	dc	fb	90	8a	45
29	63	7b	d9	72	ad	42	21	ec	c4	06	00	9a	b6	ca	01
d7	4d	12	89	38	6f	32	d3	a4	1b	f9	49	2c	77	e4	5c
a9	85	54	b9	62	bc	4c	26	81	3a	0e	f0	46	ed	b0	b4
cc	fe	b2	3e	9f	7e	73	c0	37	58	8e	ce	7a	d1	31	f8
0c	27	e1	6c	cb	71	34	25	5e	3d	e2	ba	7c	03	8d	f4
d6	8c	5b	a5	fc	b7	f6	dd	33	24	2a	3c	1e	84	92	5d

Table A.30: S_{30}

Key: 13778F8D864B606D4E00372CC1C5436E

a1	be	db	5c	a0	c3	17	81	86	29	3b	09	66	87	6a	14
9a	3f	72	92	ab	48	2f	ff	4f	d8	b6	3e	a9	cb	85	e3
1e	bf	0d	e7	d4	76	93	94	e5	97	79	36	91	96	23	42
41	4a	35	73	f6	c8	e1	7f	47	1a	7d	55	63	04	cc	21
31	7a	68	f8	dd	d1	06	0e	53	e6	62	82	ae	b2	08	b1
7c	bb	fd	25	88	a4	2b	75	f3	d6	6e	10	99	ea	74	6d
9c	fe	61	4c	d5	98	ac	f9	0c	11	49	bd	b4	b0	57	c9
58	9f	39	a2	f2	cd	2c	c6	02	ef	b9	22	50	6c	c2	f4
52	f1	7e	e8	8f	eb	69	fc	59	f7	56	9e	38	3d	8a	d7
77	d3	a5	6b	f5	e2	3c	5a	60	1b	cf	6f	5d	1f	d0	e4
89	40	33	15	65	20	3a	5f	1c	fb	c5	01	bc	dc	ee	45
24	0b	95	8e	12	0f	c0	70	37	c7	00	2e	a7	71	18	54
07	8b	b7	2a	de	aa	84	f0	4e	0a	e9	1d	8c	16	fa	ba
a3	c1	d9	df	34	7b	2d	44	b3	ad	a6	13	5e	32	ec	83
27	5b	67	64	e0	80	90	a8	ce	b8	9b	43	46	30	19	26
28	05	af	03	d2	b5	51	4b	4d	9d	c4	8d	ed	78	ca	da

Bibliography

[1] W. Stallings, "Cryptography and network security principles and practice," 5th Edition, Prentice Hall 2006.

[2] P. Patil, P. Narayankar, D. Narayan, and S. M. Meena, "A comprehensive evaluation of cryptographic algorithms: Des, 3des, aes, rsa and blowfish," *Procedia Computer Science*, vol. 78, pp. 617–624, 2016.

[3] R. Singh and S. Kumar, "Elgamal's algorithm in cryptography," *International Journal of Scientific & Engineering Research*, vol. 3, no. 12, pp. 1–4, 2012.

[4] T. Xu, F. Liu, and C. Wu, "A white-box aes-like implementation based on key-dependent substitution-linear transformations," *Multimedia Tools and Applications*, vol. 77, no. 14, pp. 18117–18137, 2018.

[5] F. Luma, H. Hilal, and A. Ekhlas, "New dynamical key dependent s-box based on chaotic maps," *IOSR Journal of Computer Engineering*, vol. 17, no. 4, pp. 2278–2661, 2015.

[6] T. Ara, P. G. Shah, and M. Prabhakar, "Dynamic key dependent s-box for symmetric encryption for iot devices," in *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, pp. 1–5, 2018.

[7] R. Hosseinkhani and H. H. S. Javadi, "Using cipher key to generate dynamic s-box in aes cipher system," *International Journal of Computer Science and Security (IJCSS)*, vol. 6, no. 1, pp. 19–28, 2012.

[8] I. A. Shoukat, U. Iqbal, A. Rauf, and M. R. Faheem, “Randomized substitution method for effectively secure block ciphers in iot environment,” *Arabian Journal for Science and Engineering*, vol. 45, no. 12, pp. 11019–11036, 2020.

[9] P. K. Das, P. Kumar, and M. Sreenivasulu, “Image cryptography: a survey towards its growth,” *Adv. Electron. Electr. Eng. Res. India Publ*, vol. 4, no. 2, pp. 179–184, 2014.

[10] K. Kazlauskas, G. Vaicekauskas, and R. Smaliukas, “An algorithm for key-dependent s-box generation in block cipher system,” *Informatica*, vol. 26, no. 1, pp. 51–65, 2015.

[11] A. Fahmy, M. Shaarawy, K. El-Hadad, G. Salama, and K. Hassanain, “A proposal for a key-dependent aes,” in *3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT)*, 2005.

[12] P. Agarwal, A. Singh, and A. Kilicman, “Development of key-dependent dynamic s-boxes with dynamic irreducible polynomial and affine constant,” *Advances in Mechanical Engineering*, vol. 10, no. 7, 2018.

[13] O. B. Sahoo, D. K. Kole, and H. Rahaman, “An optimized s-box for advanced encryption standard (aes) design,” in *2012 International Conference on Advances in Computing and Communications*, pp. 154–157, 2012.

[14] R. Nadaf and V. Desai, “Hardware implementation of modified aes with key dependent dynamic s-box,” *IEEE ICARET*, pp. 576–580, 2012.

[15] A. Anees and Y.-P. P. Chen, “Designing secure substitution boxes based on permutation of symmetric group,” *Neural Computing and Applications*, vol. 32, no. 11, pp. 7045–7056, 2020.

[16] G. Manjula and H. Mohan, “Constructing key dependent dynamic s-box for aes block cipher system,” in *2016 2nd international conference on applied and theoretical computing and communication technology (iCATccT)*, pp. 613–617, 2016.

[17] M. K. Balajee and J. Gnanasekar, "Evaluation of key dependent s-box based data security algorithm using hamming distance and balanced output," *Tem Journal*, vol. 5, no. 1, p. 67, 2016.

[18] A. Alabaichi and A. I. Salih, "Enhance security of advance encryption standard algorithm based on key-dependent s-box," in *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, pp. 44–53, 2015.

[19] A. Ejaz, I. A. Shoukat, U. Iqbal, A. Rauf, and A. Kanwal, "A secure key dependent dynamic substitution method for symmetric cryptosystems," *PeerJ Computer Science*, vol. 7, p. e587, 2021.

[20] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image and vision computing*, vol. 24, no. 9, pp. 926–934, 2006.

[21] X. Zhang, Z. Zhao, and J. Wang, "Chaotic image encryption based on circular substitution box and key stream buffer," *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 902–913, 2014.

[22] S. Sheela, K. Suresh, and D. Tandur, "Image encryption based on modified henon map using hybrid chaotic shift transform," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25223–25251, 2018.

[23] D. Supriyo, N. Bhanja, S. K. Dhara, S. Paul, and S. Das, "Color image encryption scheme based on key dependent s-box and arnold's cat map," *International Journal of Engineering Research and Technology (IJERT)*, 2021.

[24] S. Tyagi and D. Chaudhary, "Adapted encryption algorithm with multiple skew tent map," 2013.

[25] T. Li, J. Shi, and D. Zhang, "Color image encryption based on joint permutation and diffusion," *Journal of Electronic Imaging*, vol. 30, no. 1, 2021.

[26] Y. Wang, Q. Xie, Y. Wu, and B. Du, "A software for s-box performance analysis and test," in *2009 International Conference on Electronic Commerce and Business Intelligence*, pp. 125–128, 2009.

[27] A. Mousa and A. Hamad, "Evaluation of the rc4 algorithm for data encryption.," *Int. J. Comput. Sci. Appl.*, vol. 3, no. 2, pp. 44–56, 2006.

[28] C. J. Benvenuto, "Galois field in cryptography," *University of Washington*, vol. 1, no. 1, pp. 1–11, 2012.

[29] 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.

[30] R. A. Rueppel, "Correlation immunity and the summation generator," in *Conference on the Theory and Application of Cryptographic Techniques*, pp. 260–272, 1985.

[31] B. Preneel and A. BRAEKEN, "Cryptographic properties of boolean functions and s-boxes," *Departement elektrotechniek (ESAT)*, 2006.

[32] G. Thamilarasu and A. Odesile, "Securing wireless body area networks: challenges, review and recommendations," in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1–7, 2016.

[33] F. Lafitte, "The boolfun package: Cryptographic properties of boolean functions," vol. 26, pp. 365–369, 2012.

[34] S. Picek, L. Batina, D. Jakobović, B. Ege, and M. Golub, "S-box, set, match: a toolbox for s-box analysis," in *IFIP International Workshop on Information Security Theory and Practice*, pp. 140–149, 2014.

[35] J. A. Alvarez-Cubero and P. J. Zufiria, "A c++ class for analysing vector boolean functions from a cryptographic perspective," in *2010 International Conference on Security and Cryptography (SECRYPT)*, pp. 1–9, 2010.

[36] H. Singh and P. Singh, "Enhancing aes using novel block key generation algorithm and key dependent s-boxes," *Cyber-Security and Digital Forensics*, vol. 5, pp. 30–45, 2016.

[37] B. Maram and J. Gnanasekar, "A block cipher algorithm to enhance the avalanche effect using dynamic key-dependent s-box and genetic operations," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 10, pp. 399–418, 2018.

[38] I. Hussain, T. Shah, M. A. Gondal, M. Khan, and W. A. Khan, "Construction of new s-box using a linear fractional transformation," *World Appl. Sci. J*, vol. 14, no. 12, pp. 1779–1785, 2011.

[39] G. Vaicekauskas, K. Kazlauskas, and R. Smaliukas, "A novel method to design s-boxes based on key-depdendent permutation schemes and its quality analysis," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 4, pp. 93–99, 2016.

[40] A. H. Alkhaldi, I. Hussain, and M. A. Gondal, "A novel design for the construction of safe s-boxes based on tderc sequence," *Alexandria Engineering Journal*, vol. 54, no. 1, pp. 65–69, 2015.

[41] 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.

[42] 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.

[43] 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.

[44] Y. Wang, K.-W. Wong, X. Liao, and T. Xiang, "A block cipher with dynamic s-boxes based on tent map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 7, pp. 3089–3099, 2009.

[45] 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.

[46] F. Özkaynak and S. Yavuz, "Designing chaotic s-boxes based on time-delay chaotic system," *Nonlinear Dynamics*, vol. 74, no. 3, pp. 551–557, 2013.

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

[48] M. Matsui, "Linear cryptanalysis method for des cipher," *Lecture Notes in Computer Science*, vol. 765, pp. 386–397, 1994.

[49] R. Stoop and P. Meier, "Evaluation of lyapunov exponents and scaling functions from time series," *JOSA B*, vol. 5, no. 5, pp. 1037–1045, 1988.

[50] Q. Lu, C. Zhu, and G. Wang, "A novel s-box design algorithm based on a new compound chaotic system," *Entropy*, vol. 21, no. 10, p. 1004, 2019.

[51] M. A. S. Hassan and I. S. I. Abuhaiba, "Image encryption using differential evolution approach in frequency domain," *arXiv preprint arXiv:1103.5783*, 2011.

[52] A. Hafsa, M. Gafsi, J. Malek, and M. Machhout, "Fpga implementation of improved security approach for medical image encryption and decryption," *Scientific Programming*, vol. 2021, 2021.

[53] Z. A. Abduljabbar, I. Q. Abduljaleel, J. Ma, M. A. A. Sibahee, V. O. Nyangaresi, D. G. Honi, A. I. Abdulsada, and X. Jiao, "Provably secure and fast color image encryption algorithm based on s-boxes and hyperchaotic map," *IEEE Access*, vol. 10, pp. 26257–26270, 2022.

[54] K. H. Ranjan, S. S. Fathimath, G. Aithal, and S. Shetty, "A survey on key (s) and keyless image encryption techniques," *Cybernetics and Information Technologies*, vol. 17, no. 4, pp. 134–164, 2017.