CAPITAL UNIVERSITY OF SCIENCE AND
TECHNOLOGY, ISLAMABAD

An Encryption Scheme Based on Chaotic Map

by

Farzana Abrar Kazmi

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

in the

Faculty of Computing
Department of Mathematics

2021

Dedicated to

My Parents And Brothers

without their effort and support, I would never have reached so far

CERTIFICATE OF APPROVAL

An Encryption Scheme Based on Chaotic Map

by

Farzana Abrar Kazmi

(MMT173029)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Atta Ullah	NUTECH, Islamabad
(b)	Internal Examiner	Dr. Dur e Shehwar Sagheer	CUST, Islamabad
(c)	Supervisor	Dr. Samina Rashid	CUST, Islamabad

Dr. Samina Rashid
Thesis Supervisor
December, 2021

Dr. Muhammad Sagheer
Head
Dept. of Mathematics
December, 2021

Dr. Muhammad Abdul Qadir
Dean
Faculty of Computing
December, 2021

Author's Declaration

I, **Farzana Abrar Kazmi** hereby state that my M-Phil thesis titled "**An Encryption Scheme Based on Chaotic Map**" 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. +

(Farzana Abrar Kazmi)
Registration No: MMT173029

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**An Encryption Scheme Based on Chaotic Map**" 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.

(Farzana Abrar Kazmi)
Registration No: MMT173029

Acknowledgement

ALLAH is the most beneficent and the most merciful whose blessings are abundant and favors are unlimited. This thesis would not have been possible without the inspiration and support of a wonderful individuals, my thanks and appreciation to all of them for being part of this journey and making this thesis possible. First of all, I would like to express my sincere gratitude to my supervisor Dr. Samina Rashid for her patience, motivation, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. My sincere thanks also goes to Dr. Rashid Ali, Dr. Muhammad Sagheer and Dr. Shafqat Hussain for their appreciation and support. Finally, I would like to thank my family for being always with me and bringing all the care and support for my career. Specially, I am grateful to my parents, who have given all the love and care and brought me up in this stage.

(Farzana Abrar Kazmi)

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 chaotic map. A chaotic map is an evaluation map that exhibits some sort of chaotic behavior (e.g.randomness). In this thesis encryption algorithm based on Feistal structure and S-box has been discussed. By using the three dimensional Logistic map equation a new S-box has been constructed which have good properties and analysis results.

Contents

List of Figures

List of Tables

Abbreviations

AES	Advanced Encryption Standard
BIC	Bit Independence Criteria
DES	Data Encryption Standard
DPA	Differential Power Analysis
GF	Galois Field
LCG	Linear Congruence Generator
SAC	Strict Avalanche Crierion
SET	S-box Evaluation Tool
SPN	Substitution Permutation Network
SNR	Signal to Noise Ratio

Symbols

$\mathbb{G}$	Group
$\mathbb{Z}$	Set of Integers
$\mathbb{R}$	Set of Real Numbers
$\mathbb{Q}$	Set of Rational Numbers
$\mathbb{F}_p$, $\mathbb{Z}_p$	Finite Field of order prime p

Chapter 1

Introduction

From ancient time to today, the secure transfer of private data over the public network is a big issue. There is a major need of secure channel for wireless networking and secret communication. Roman people knew some cryptographic methods and used the shift cipher or Caesar Cipher [1] while communicating with each other. Later, many ciphers were introduced for sending codes or secret messages. For example, monoalphabetical cipher, polyalphabetical cipher [2], Playfair cipher, four square cipher and Hill cipher of different orders. In this context, there are many contributions to cryptography see for example [1, 3]. Due to the advancement in network technology, security of the data is a big challenge. The rapid growth of technology extends to all areas of scientific research including digital image processing and transmission [4]. Popular use of multimedia technologies and enhanced network communication capacity slowly lead us to get clear and direct information through the images. In many fields, such as military, medical, industrial, digital, communication or even personal, millions of images are stored or transmitted everyday via the internet. The need to defend certain photos from unauthorized users has become a problem, depending on the application domain. Data security can be done by following various ways such as cryptography, watermarking and stenography etc.

1.1 Background

Cryptography is the science of secret communication which is used to alter the original transmission into unreadable form in the presence of a third-party over an

insecure channel. Converted message is called ciphertext and original message is called plaintext. To convert the plaintext into ciphertext an algorithm is needed known as encryption algorithm. The algorithm that converts the ciphertext back into plaintext is called the decryption algorithm. For encryption and decryption, cryptographic schemes need special information which is shared between sender and receiver, is called a key. A cryptographic scheme that consists of a message space, a ciphertext space, a key space, an encryption algorithm and decryption algorithm is called a cryptosystem.

Modern cryptography is an art of science which is now considered as branch of mathematics and computer science. It uses sophisticated mathematical equations (algorithms) to provides secrecy, integrity, authentication and anonymity to data [5]. Cryptography is broken down into two major branches on the basis of keys. Symmetric (private) key cryptography and asymmetric (public) key cryptography. In symmetric (private) key cryptography, only one key is used for both the data encryption and decryption. The trader and the acceptor are bound to share the key for data encryption and decryption with each other. For example Data Encryption Standard (DES) [5], Double Data Encryption Standard (2DES) [6], Triple Data Encryption Standard (TDES) [6], Advanced Encryption Standard (AES) [7], and Blowfish [8].

In 1976, Whitfield Diffie and Martin Hellman introduced a new scheme known as asymmetric key cryptography, also known as public key cryptography. Two keys are used in asymmetric cryptography one is for data encryption and the other is used for decryption. A person generates two keys one is kept secret, called secret key, and the other key is made public, called the public key. Anyone can encrypt data as the encryption key is public but only the person having the decryption key can decrypt the data because decryption key is private. Sender encrypts original text using public key and encryption algorithm to obtain ciphertext. The secret key and decryption algorithm are used to obtain original text. Examples are RSA [9], DSA, ELGamal [10] Diffie-Hellman key exchange [11] and Elliptic curve cryptosystem [12].

1.2 Substitution Boxes in Cryptography

A symmetric key cryptosystem is further categorized as either by stream cipher or block cipher. A block cipher will convert a whole block of plaintext into a block of ciphertext using the secret key at a time whereas stream cipher encrypts one bit or byte of data at a time. Thus a block cipher has two basic specification, block size and key size. The block ciphers are designed on the basis of Shannon's theory of confusion and diffusion which is also implemented in Substitution Permutation networks (SPN) [13]. Such networks basically consist of a number of mathematical operations which are linked together. It takes block of palintext, as input a key and apply many rounds of Substitution-box (S-box) or Permutation-box (P-box) to get desired ciphertext. For decryption process the inverse S-box or P-box is used in reverse order with the same key. The examples of SPN are the Data Encryption Standard (DES) [14] and Advanced Encryption Standard (AES) [15] cryptosystems. S-boxes are basically vectorial boolean functions expressed as look-up tables. An S-box takes in a small block of bits and substitute them by another block of bits. This substitution should be one to one to make decryption effective. Generally, the S-box takes and transforms m input bits into n output bits. So an S-box $(m \times n)$ can be regarded as a search table of 2^m words of n bits each. The output length can be the same as the input length in AES or can be different from that in a DES. An S-box should be designed in such a way that each output bit will depend on every input bit for making cryptosystem strong.

There are many defined methods for making good S-boxes. Some examples are twofish [16], DES [14], AES[15], and GOST [17] etc. Researchers and Cryptographers proposed many approaches and methods for the construction of a strong S-box. The security arguments of symmetric encryption algorithms are basically depending on the properties of S-boxes and so they are really crucial in cryptography. In this scenario a main question arises that, can some S-boxes are better than others. Obviously answer is yes, so the main focus was to investigate those measures which would differentiate between bad and good substitutions, and for those techniques which would construct good substitutions. Cryptanalysis attacks

depend on the weakness of cryptosystem so new attacks produce a demand for the new security parameters.

1.3 Objective of Thesis

The proposal of different methods has been made to design an S-boxes in conventional cryptography over the past decade. Many methods have been proposed by researchers [18], [19], [20] to generate a strong S-box. That is an S-box which provide more resistance against cryptanalysis. The strength of S-boxes are measured on the basis of certain properties that are discussed in chapter 2 among those the most important one is the non-linearity property. Some researchers proposed to generate an S-box by using chaotic maps. Some of them believe that there is a strong relationship between the science of chaos and cryptography. A chaotic map is an evaluation map that exhibits some sort of chaotic behavior (e.g.randomness). These maps may be parameterized by a continuous time or a discrete-time parameter. For a brief discussion on chaotic map see section 2.7. We focused our work to review the artice [21] and the construction of S-box based on one dimensional chaotic map and text encryption algorithm is discussed in thesis. We extend our work to three dimensional chaotic logistic map [22] to construct a new S-box and found that it is better in many ways.

1.4 Thesis Layout

The rest of the thesis is organized as follows:

- **Chapter 2** This chapter introduces Galois field, Boolean functions, chaos theory and their general properties has also been discussed. Different cryptographic properties are also explained according to the general design criteria of S-boxes.

- **Chapter 3** presents the construction of S-box based on one dimensional chaotic maps, sine map and linear congurence generator. Calculations are performed with the help of MATLAB.

- **Chapter 4** In this chapter a new S-box based on three dimensional logistic equation and text encryption algorithm based on Feistal structure is presented. All the calculations are performed with the help of MATLAB. Properties and comparison of S-boxes based on one dimensional chaotic map and three dimensional chaotic map has also been presented.

- **Chapter 5** Finally the conclusion of thesis is presented in this chapter.

Chapter 2

Preliminaries

This chapter describe the fundamental ideas, mathematical background and definitions related to the thesis which will be later used in proceeding chapters.

2.1 Cryptography

Cryptography is the art and science for transforming the secret messages into an unreadable format, called ciphertext. Only those who have a secret key can decipher the ciphertext into original message. Cryptography can also be used for user authentication. A system in which we convert data or message into secret codes using encryption algorithm and convert secret codes back into message using decryption algorithm is know as cryptosystem. There are five basic components in a cryptosystem.

1. **Plaintext**: It is the orignal form of data or message.
2. **Ciphertext**: It is coded form of data or message.
3. **Encryption algorithm**: It convert plaintext into ciphertext.
4. **Decryption algorithm**: It is important for the algorithm to run in reverse manner as well. It generates original plaintext with the help of secret key and ciphertext.

5. **Secret Key**: The information used in encryption and decryption which is known only to sender and receiver.

On the basis and design of cryptosystem the cryptography is further divided in the following two main types.

1. **Symmetric Key Cryptography**
2. **Public Key Cryptography**

2.1.1 Symmetric Key Cryptography

A system in which same key is used for both encryption and decryption is called symmetric key cryptography [23]. For example, Data Encryption Standard (DES) [24], Double Data Encryption Standard [6], Triple Data Encryption (3DES) [6] and Advance Encryption Standard (AES) [25]. A model of symmetric key cryptography is shown in the Figure 2.1

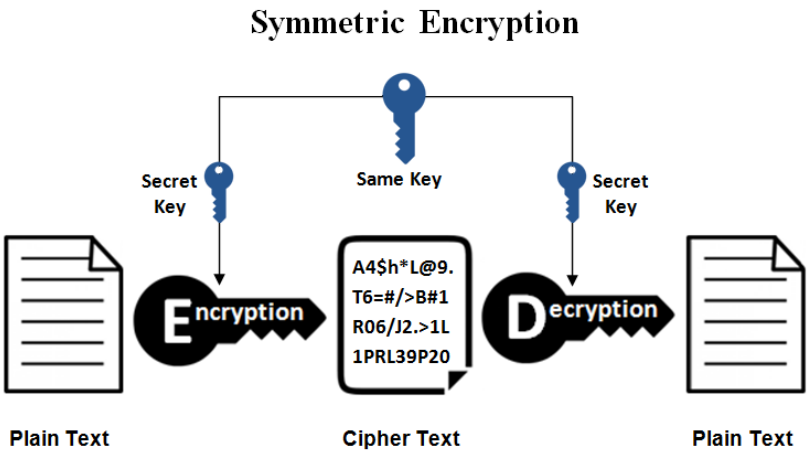

FIGURE 2.1: Symmetric Key

key sharing is the main drawback of symmetric key cryptography, which means that the secret key must be passed on to each party involved in communication. Electronic communication used for this purpose may not be a safe way of exchanging keys, as communication networks may be accessed by anyone. The only safe way to share keys is to secretly exchange them, but it may be a very difficult job.

2.1.2 Public Key Cryptography

Public key cryptosystem is proposed by Diffie-Hellman in 1976. Two keys are used for encryption and decryption in public key cryptography [6], one is known to all is public key, and the other is called the private key of the owner which is kept confidential. The public key cryptography is shown in the Figure 4.5 . Here, the sender uses public key and encryption algorithm to encrypt the original text to receive the ciphertext. The secret key and decryption algorithm are used to obtain orignal text by the receiver end.

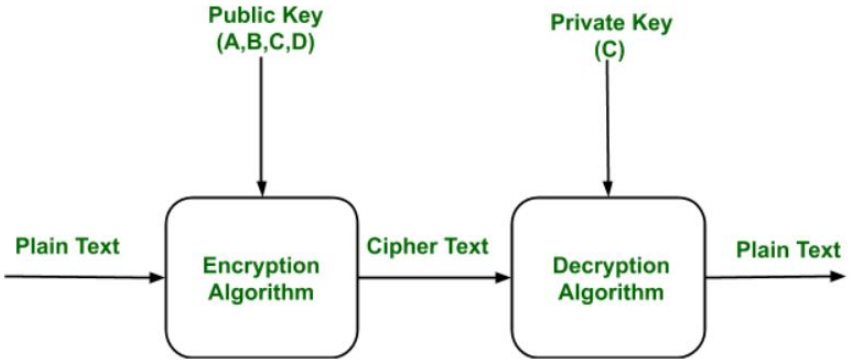

FIGURE 2.2: Asymmetric Key

RSA cryptosystem [26] and ElGamal cryptosystem [10] are examples of asymmetric key cryptography. Block and stream ciphers are the two main types of ciphers used in cryptography.

Definition 2.1.1. (Block Cipher)

"A block cipher is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length". [27]

Definition 2.1.2. (Stream Cipher)

"A stream cipher is one that encrypts a digital data stream one bit or one byte at a time". [27]

2.2 Feistel Structure

Cryptographic scheme based on the Feistel structure uses the same algorithm for encryption and decryption. As shown in Figure 2.3, the Feistel structure consists of couple of rounds of processing of the plaintext, with each round such as a substitution step accompanied by means of a permutation step.

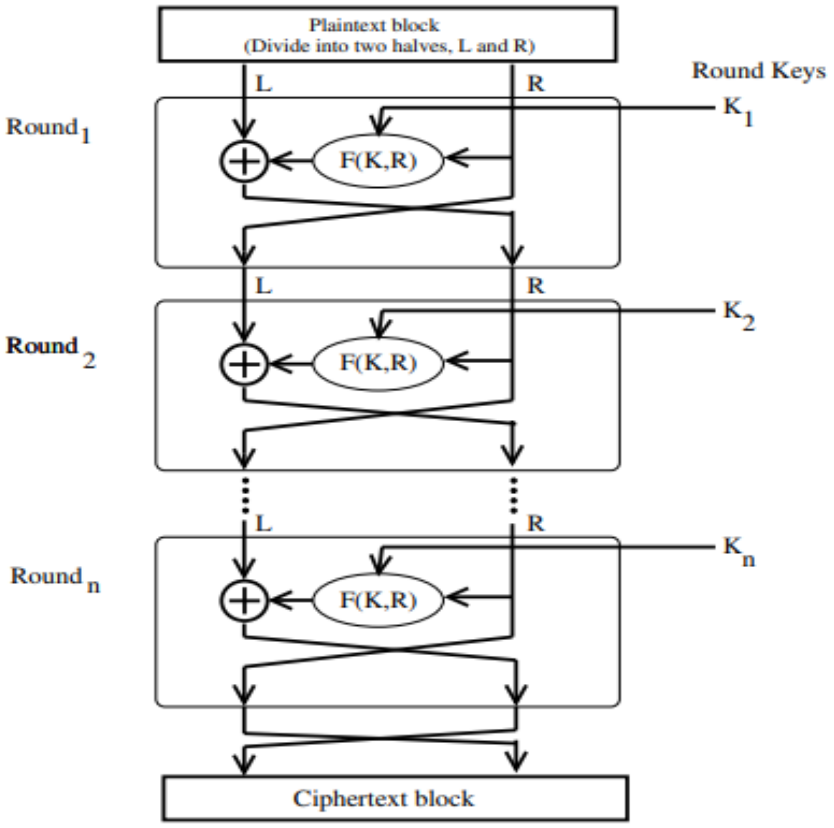

FIGURE 2.3: Feistel Structure

The structure suggested by Feistel is represented on the left-hand side of Figure 2.3. The inputs to the encryption algorithm are a plaintext block of length of $2w$ bits and a key. There are two halves of the plaintext block, L_0 and R_0. The two halves of the data move through n processing rounds and then combine to create the block of ciphertext. Each i round has L_{i-1} and R_{i-1} as inputs derived from the previous round, as well as a K_i subkey derived from the masterkey K . Generally speaking, the K_i subkeys vary from K and from each other. n rounds are used in Figure 2.3, but it is possible to introduce any number of rounds. All rounds have

the same structure. On the left half of the results, a substitution is performed. This is achieved by applying to the right half of the data a round function F and then taking the exclusive XOR of that function's output and the left half of the data. For each round, the round function has the same general structure but is parameterized by the round subkey. Another way to express this is to say that F is a right-halfbit block function and a subkey of y bits that generates an output value of w bits in length: $F(RE_i, K_{i+1})$. A permutation consisting of the interchange of the two halves of the data is performed following this substitution. This structure is a basic form of Shannon's proposed substitution-permutation network (SPN). The exact realization of a Feistel network depends on the selection of the following parameters and design functionality:

1. Block size :

   Larger block sizes mean more security (all other things being equal) but reduced encryption/decryption speed for a given algorithm. The greater security is achieved by greater diffusion. Traditionally, a block size of 64 bits has been considered a reasonable tradeoff and was nearly universal in block cipher design.

2. Key size :

   Larger key size means greater security but may decrease encryption/decryption speed. The greater security is achieved by greater resistance to brute-force attacks. Key sizes of 64 bits or less are now widely considered to be inadequate, and 128 bits has become a common size.

3. Number of rounds :

   The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.

4. Subkey generation algorithm :

   Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.

5. Round function F :

   More complex function generally means greater resistance to cryptanalysis.

Definition 2.2.1. Confusion

"In Shannon's original definitions, confusion refers to making the relationship between the ciphertext and the symmetric key as complex and involved as possible.". [27]

Definition 2.2.2. Diffusion

"Diffusion means that if we change a character of the plaintext, then several characters of the ciphertext should change, and similarly, if we change a character of the ciphertext, then several characters of the plaintext should change". [27]

2.3 Mathematical Background

This section is about some basics tools in mathematics that are used in the thesis.

Definition 2.3.1. (Group)

"Let $\mathbb{G}$ be a non empty set together with binary operations $*$ on $\mathbb{G}$. Then $(\mathbb{G}, *)$ is called a group if it satisfies the following properties:

i. **Closure**: For all $a, b \in \mathbb{G}$, $a * b \in \mathbb{G}$,

ii. **Associative**: For all $a, b, c \in \mathbb{G}$ $(a * b) * c = a * (b * c)$,

iii. **Identity**: There is element $e \in \mathbb{G}$ such that $a * e = e * a = a$,

iv. **Inverse**: If $a \in \mathbb{G}$, then there is an element $b \in \mathbb{G}$ such that $a * b = b * a = e$".[28]

Definition 2.3.2. (**Abelain Group**)

A group $\mathbb{G}$ is called abelian group, if binary operation "*" is commutative that is [28]

$$a * b = b * a \ \forall \ a, b \in \mathbb{G}."$$

The following are examples of group.

Example 2.3.1.

i. Set of integers $\mathbb{Z}$ is a group with respect to addition of integers.

ii. Set of all invertible matrices with ordinary matrix multiplication form a group.

iii. Set of real numbers (only non zero elements) $\mathbb{R}$ form a group under multiplication.

Definition 2.3.3. (Monic Polynomial):

"A monic polynomial, is a mathematical expression that consists of coefficients and a single variable, with the leading coefficient equal to one. The leading coefficient is found in the term that contains the variable with the highest degree or exponent." [29]

Definition 2.3.4. (Unique Factorization):

"Every monic polynomial $f(x)$ is either irreducible or can be factorized into a product of monic polynomial factors. Further if a factor is not irreducible, it can be factored further. Since factor degrees are decreasing but bounded below by 1, we must eventually arrive at a product of monic irreducible(prime) polynomials". [29]

Definition 2.3.5. (Floor Function)

" The floor function is the function that takes as input a real number x, and gives

as output the greatest integer less than or equal to x, denoted floor(x) or $\lfloor x \rfloor$. The integral part or integer part of x, often denoted $[x]$". [30]

Example 2.3.2.

floor(2.4) = $\lfloor 2.4 \rfloor = 2$.

Definition 2.3.6. (Celling Function)

" The ceiling function ceiling(x) is defined as the function that outputs the smallest integer greater than or equal to x". [30]

Example 2.3.3.

ceil(2.4) = $\lceil 2.4 \rceil = 3$, while $\lfloor 2 \rfloor = \lceil 2 \rceil = 2$

Definition 2.3.7. (Field)

"A nonempty set $\mathbb{F}$ with two binary operation addition $(+)$ and $(\cdot)$ is called a field and finite field is a field that contain finite numbers of element, if it satisfies the following properties:

i. ($\mathbb{F}$,+) is an abelian group.

ii. ($\mathbb{F}$,$\cdot$) is an abelian group.

iii. Distributivity of addition over multiplication". [31]

Examples of field are:

i. Set of real and complex numbers are fields under usual addition and multiplication.

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

Definition 2.3.8. Galois Field

"A finite field whose order is the form of p^n, where n is positive integer and p is prime number is called Galois Field denoted by $GF(p^n)$. In Galois field, elements are defined as
$GF(p^n) = (0, 1, 2, ..., p-1) \cup (p, p+1, p+2, ..., p+p-1) \cup (p^2, p^2+1, p^2+2, ..., p^2+p-1) \cup .... \cup (p^{n-1}, p^{n-1}+1, p^{n-1}+2, ..., p^{n-1}+p-1)$.
The order of Galois field is given by p^n while p is characteristics of field and the degree of the polynomials in $GF(p^n)$ is less than n, while coefficients is at most $p-1$ [32]."

Example 2.3.4.

$GF(3^2)$=$(0, 1, 2, x, x+1, x+2, 2x, 2x+1, 2x+2)$ consist of $3^2 = 9$ elements where each of the polynomials have degree less than 2 and coefficients are less than 3.

Example 2.3.5.

Finite field $\mathbb{F}_2$ that is, {0,1} with addition and multiplication is defined in TABLE 2.1 and TABLE 2.2 respectively.

TABLE 2.1: Addition

+	0	1
0	0	1
1	0	0

TABLE 2.2: Multiplication

.	0	1
0	0	0
1	0	1

2.3.1 Computation in Galois Field

In this section, the algebraic operation will explained in Galois field. In Galois field algebraic expression needs some additional steps. In the Euclidean space, algebraic operations $(+,-,\times)$ in Galois Field need some additional steps.

2.3.2 Addition and Subtraction in Galois Field

In Galois field, the operation of addition is quite simple. If $v_1(x)$, $g_1(x)$ are any two polynomials in $GF(p^n)$ and $h_1(x) = v_1(x) + g_1(x)$ with the coefficients of $v_1(x)$, $g_1(x)$ and $h_1(x)$ are $A = a_{n-1}, a_{n-2}, ..., a_1 a_0, B = b_{n-1}, b_{n-2}, ...b_1, b_0$, and $C = c_{n-1}, c_{n-2}, ...c_1, c_0$ respectively. Let a_k, b_k and c_k are the coefficients of $v_1(x)$, $g_1(x)$ and $h_1(x)$ respectively then

$$c_k = a_k + b_k \bmod p \text{ for } k = 0, 1, 2, 3, ...n - 1$$

Likewise if $h_1(x) = v_1(x) - g_1(x)$ then $c_k = a_k - b_k \bmod p$ where $k \in \{0, 1, 2, 3.....n - 1\}$.

Note that in Galois field $GF(2^n)$ addition can be performed using XOR operation. The element of Galois field can be represent by a unique n-bit pattern. We can transform polynomials of Galois field in binary number system from which we can convert it into any number system.

Example 2.3.6.

Conversion of polynomials into different number systems

Let $x^7 + x^5 + x^4 + x + 1$ is the polynomial in $GF(2^8)$.

The binary representation

$$x^7 + x^5 + x^4 + x + 1 = (10110011)_2.$$

In hexadecimal representation

$$x^7 + x^5 + x^4 + x + 1 = (B3)_{16}.$$

In decimal representation

$$x^7 + x^5 + x^4 + x + 1 = (10110011)_2 = 179.$$

Example 2.3.7.

Suppose we are working in $GF(2^4)$, then compute $v(x) + g(x)$ if $v(x) = x^3 + x^2 + x + 1$, $g(x) = x^2 + 1$ under the mod $m(x)$ where $m(x) = x^4 + x^3 + x + 1$ then

$$v(x) + g(x) = (x^3 + x^2 + x + 1) + (x^2 + 1)$$

$$v(x) + g(x) = (x^3 + x) \mod (x^4 + x^3 + x + 1)$$

Alternatively, from binary number system

$$v(x) = x^3 + x^2 + x + 1 = (1111)_2 \text{ and } g(x) = x^2 + 1 = (0101)_2$$

$$v(x) + g(x) = 1111 \oplus 0101$$

$$= 1010$$

$$= x^3 + x \mod (x^4 + x^3 + x + 1)$$

2.3.3 Multiplication and Multiplicative Inverse

In Galois Field, multiplication involves more hard work. Suppose $v_1(x)$, $g_1(x)$ be any two polynomials in $GF(p^n)$ and suppose $m_1(x)$ be irreducible polynomial. The degree of product of $v_1(x)$ and $g_1(x)$ should be less than n in $GF(p^n)$. If $h_1(x)$ represent the product of $v_1(x)$, $g_1(x)$ then

$$h_1(x) = v_1(x) \cdot g_1(x) \mod p.$$

Suppose $a_1(x)$ represent the multiplicative inverse of $v_1(x)$ then

$$v_1(x) \cdot a_1(x) = 1 \mod p.$$

Note that in evaluating the product of any two polynomials and their inverses need both reducing polynomial $m_1(x)$ and coefficients in modulo p. The most feasible method to calculate the multiplicative inverse of polynomials is Extended Euclidean Algorithm. The multiplication is elaborated by an example given below.

Example 2.3.8.
Let $v_1(x) = x^2 + 1$ and $g_1(x) = x^2 + x + 1$ with irreducible polynomial $m_1(x) = x^3 + x^2 + 1$ in $GF(2^3)$ then

$$\begin{aligned} v_1(x).g_1(x) &= (x^2+1)(x^2+x+1) \\ &= (x^2+1)(x^2+x+1) \\ &= x^4 + x^3 + x^2 + x^2 + x + 1 \\ &= x^4 + x^3 + x + 1 \\ &= 1 \mod (x^3 + x + 1) \end{aligned}$$

2.3.4 Modular Multiplicative Inverse

In this section we will explain how to find multiplicative inverses modulo some integer n.

Given any two integer r and s, the problem is to find an integer t such $r.t \equiv 1 \mod s$ and $r^{-1} \equiv t \mod s$, where $1 \leq t \leq s-1$.
The multiplicative inverse of r mod s are relatively prime that is, $gcd(r, m) = 1$.

Algorithm (Multiplicative inverse in finite field)

To find the multiplicative inverse in $\mathbb{Z}_p$, we can implement Euclidean Algorithm

[33] in the computer algebra system [34] ApCoCoA.

Following is the method of finding the inverse of r mod s.

Input: An integer r and an irreducible integer s.

Output: r^{-1} mod s

i. Initialize six integers V_i and W_i for i=1,2,3 as
$(V_1, V_2, V_3) = (1, 0, m)$
$(W_1, W_2, W_3) = (0, 1, r)$

ii. If W_3=0, return V_3=gcd(r, s); no inverse of r exist in mod s

iii. If W_3=1 then return W_3 = gcd (r, s) and $W_2 = r^{-1}$ mod s

iv. Now divide V_3 by W_3 and find the quotient Q when V_3 is divided by W_3

v. Set $(P_1, P_2, P_3) = ((V_1 - QW_1), (V_2 - QW_2), (V_3 - QW_3))$

vi. Set $(V_1, V_2, V_3) = (W_1, W_2, W_3)$

vii. Set $(W_1, W_2, W_3) = (P_1, P_2, P_3)$

viii. Go to step (ii).

2.4 Boolean Function

Boolean function is a function which is define as $f: GF(2^n) \to GF(2)$ where n is non-negative integer. Every value of n where $(n = 1, 2, ..., 8)$ can be written as $x_1, x_2, x_3, x_4, ..., x_n$. A Boolean function explain how Boolean output values determine with the help of some logical calculations of Boolean input values. These functions are also helpful to design the circuits and chips of digital computers [35]. In cryptography, Boolean functions plays an important role for designing a substitution boxes.

Example 2.4.1.

For $n = 3$, we have a mapping from $GF(2^n)$ to $GF(2)$.

$$f(x_1, x_2, x_3) = x_1 \oplus x_2 x_3$$

with input bits x_1, x_2 and x_3.

TABLE 2.3: Truth table of Boolean function

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Definition 2.4.1. (Sequence of the function)

The sequence of the form $\{(-1)^{f(\alpha_0)}, (-1)^{f(\alpha_1)}, ..., (-1)^{f(\alpha_{2^n-1})}\}$ is known as sequence of Boolean function f. Such a sequence is said to be balanced if it contain equal number of ones and minus ones. A function is called a balanced if its sequence is balanced. We have to show an example given below,

Example 2.4.2. Consider the following Boolean function with input bits v_1, v_2, v_3 and v_4

$$f(v_1, v_2, v_3, v_4) = v_1 v_2 v_3 + v_2 v_3 v_4 + v_1$$

So it is defined as below:

TABLE 2.4: Truth table of $GF(2^4)$

i	$\alpha = v_1v_2v_3v_4$	$f\alpha(i)$
0	0000	0
1	0001	0
2	0010	0
3	0011	0
4	0100	0
5	0101	0
6	0110	1
7	0111	1
9	1001	1
10	1010	1
11	1011	1
12	1100	1
13	1101	1
14	1110	0
15	1111	1

So the sequence of function f can be written as

$$\{(-1)^{f(\alpha_0)}, (-1)^{f(\alpha_1)}, (-1)^{f(\alpha_2)}, (-1)^{f(\alpha_3)}, (-1)^{f(\alpha_4)}, (-1)^{f(\alpha_5)}, (-1)^{f(\alpha_6)}, (-1)^{f(\alpha_7)}$$
$$, (-1)^{f(\alpha_8)}, (-1)^{f(\alpha_9)}, (-1)^{f(\alpha_{10})}, (-1)^{f(\alpha_{11})}, (-1)^{f(\alpha_{12})}, (-1)^{f(\alpha_{13})},$$
$$(-1)^{f(\alpha_{14})}, (-1)^{f(\alpha_{15})}\}$$

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

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

Definition 2.4.2.

A Boolean function $f : GF(2^n) \rightarrow GF(2)$, composed of linear function and a constant is called **Affine function** which can be expressed as,

$$f(x) = a \cdot x \oplus \varepsilon$$

where a, $x \in GF(2^n)$ and $\varepsilon \in GF(2)$. Set of all n variable affine Boolean function is denoted by A_n. [**?**]

Definition 2.4.3.

"The number of non-zero digits in a binary sequence is called **Hamming weight**. It is denoted by **wt(x)**, where $x \in GF(2^n)$". [36]

Example 2.4.3. For n=8. Let

$$x = 00110101$$

then

$$\mathbf{wt}(00110101) = 4$$

Definition 2.4.4.

" The **Hamming distance** between two Boolean functions f, g: $GF(2^n) \rightarrow GF(2)$ is defined as: [36]

$$d(f, g) = \mathbf{wt}(f(v) \oplus g(v))$$

Here,

$$f(v) \oplus g(v) = f(v_0) \oplus g(v_0) \oplus f(v_1) \oplus g(v_1) \oplus \cdots \oplus f(v_2^{n-1}) \oplus g(v_2^{n-1})''.$$

where $v = (v_0, v_1, ..., v_2^{n-1}) \in GF(2^n)$ It is considered as the number of inputs where the functions differ or how many bits need to be changed in truth table of f to get g.

Example 2.4.4.

Consider the two Boolean functions

$$f(x) = 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1$$

$$g(x) = 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0$$

$$d(f, g) = 4$$

Example 2.4.5.

Consider the two Boolean functions $f(v) = v_1v_2v_3$ and $g(v) = v_1 \oplus v_2v_3$ with input bits v_1, v_2, v_3. Hamming distance of these boolean functions is:

TABLE 2.5: Truth table of $GF(2^3)$

i	$v_i = v_1v_2v_3$	$(f \oplus g)(v_i)$
0	000	0
1	001	0
2	010	0
3	011	0
4	100	1
5	101	1
6	110	1
7	111	1

$$d(f,g) = \mathbf{wt}(f(v) \oplus g(v)) = \mathbf{wt}(v_1v_2v_3 \oplus v_1 \oplus v_2v_3)$$

Hence, the hamming distance of f and g is 4.

Definition 2.4.5.

"An **Algebric Immunity** (AI) of two Boolean functions $f(v)$ and $g(v)$ is defined as the lowest degree of non-zero function g such that either

$$(f+1)g = 0$$

or

$$f.g = 0$$

where a Boolean function f is said to admit an annihilating function g if $f.g = 0$." [37]

Example 2.4.6. Consider the two boolean functions

$$f(v) = v_1 + v_2$$

and

$$g(v) = v_2$$

to compute the algebric immunity;

TABLE 2.6: Truth table of AI

$v = v_1v_2$	$f(v)$	fg	$(f+1)$	$(f+1)g$
00	0	0	1	0
01	1	0	0	0
10	1	0	0	0
11	0	0	1	0

Last column of the above table shows that $(f+1)g = 0$.

2.5 S-box

An S-box (substitution-box) is a fundamental component of symmetric key algorithms which perform substitution. They are usually used in block ciphers to obscure the relationship between the key and the ciphertext. It increases the security of cryptosystem against known attacks. The S-box usually takes a certain number of input bits m and converts them to a certain number of output bits n, where n is not necessarily equal to m. S-boxes are classified on the basis of some important properties. Among them the most important characteristics of substitution box is non-linearity. In-fact it is the only operation in symmetric encryption schemes that are nonlinear and hence provide security particularly against linear cryptanalysis. In input and output vector bijection needs an onto and one to one mapping. It is necessary that the function must be bijective to be used for the construction of S-box. Many researchers have examined previously that how S-box is designed and provide security against known cryptanalysis attacks. There are many methods for making good S-boxes such as the construction used in blowfish.

2.5.1 Significance of S-boxes

The only nonlinear part of a SPN as a cryptosystem is the S-box because S-boxes are composed of highly nonlinear Boolean functions. Without them, adversaries would compromise the system with ease.

Actually, there are three main reasons for studying the S-box design.

1. **Critical to Block Ciphers**

   If you are not studying S-box design criteria then you are bound to adopt

an important part of block ciphers just like a black box with no real understanding of what is their design and how it is affecting the whole system.

2. **Designing New Ciphers**

   For designing a new cipher, S-box design is the most significant area because it is the only nonlinear part of the system. So basically a cipher strength depends on this part. As with advancement of cryptography, hackers are also developing new methods of attacks, so S-box design should be secured in advance to guarantee cipher security.

3. **Need of Developing Private S-boxes**

   Interest and awareness in this topic was increased especially when back-doors are used by the adversaries to generate keys for certain ciphers such as AES [15], therefore, every organization and especially governments want to have a secure system only applicable to their organization with an extra security layer which is possible only if they design their individual S-boxes for their specific system.

2.5.2 Classification of S-boxes

S-boxes are categorized into three types,

1. **Straight S-box**

   A straight S-box takes input and gives output of the same size *i.e.* $m = n$ and this S-box had been proposed by Rijndeel cipher. It is the simplest and easiest type of S-box. AES [15] is an example of such S-box.

2. **Expanded S-box**

   It receives fewer bits as input and generates an output of more data bits *i.e.* $n < m$. By duplicating some input or output bits such S-box can be constructed.

3. **Compressed S-box**

   A design of S-box which takes in more bits and output fewer bits is called compressed S-Box. An excellent example of compressed S-box is DES in which 6 input bits are taken as one input block and 4 bits in one block are returned as output block *i.e.* $n > m$. Now we will move toward the properties of S-box.

Definition 2.5.1.

" A binary sequence of Boolean function f is called **Balanced** if there are equal number of zeros and ones. [36]"

Example 2.5.1.

To show binary sequence is balanced. Consider the example with boolean function,

$$f(v_1, v_2, v_3, v_4) = v_1 \oplus v_2 v_3 \oplus v_4$$

is given in Table 2.7. The last column contains 8 zeros and 8 ones, so the sequence of f is balanced.

Definition 2.5.2.

Non-linearity is very important part of S-boxes. The non-linearity, $NL(f)$, of a boolean function $f(v) : GF(2^n) \to GF(2)$ is defined as the minimum hamming distance of f from the set of all n-variable affine functions. Using Walsh transform, non-linearity can be shown as

$$NL(f) = 2^{n-1}(1 - 2^{-n}). \text{ [36]}$$

If n is even $f(v)$ attains maximum non-linearity, that is, $2^{n-1}(1 - 2^{-n})$, such functions are called bent functions. Non-linearity can be measured in terms of

TABLE 2.7: Truth table of $GF(2^4)$

i	$\alpha = v_1v_2v_3v_4$	$f(\alpha_i)$
0	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	0
3	0 0 1 1	1
4	0 1 0 0	0
5	0 1 0 1	1
6	0 1 1 0	1
7	0 1 1 1	0
8	1 0 0 0	1
9	1 0 0 1	0
10	1 0 1 0	1
11	1 0 1 1	0
12	1 1 0 0	1
13	1 1 0 1	0
14	1 1 1 0	0
15	1 1 1 1	1

Hamming weight and Hamming distance as well, that is

$$N_{(f)} = \min_{a \in A_n} d_H(f, a) \tag{2.1}$$

where A_n is set of affine fuction,

Example 2.5.2. Let v_1 and v_2 are input bits and $f(v)$ is a boolean function:

$$f(v_1, v_2) = v_1 \oplus v_2$$

TABLE 2.8: Truth table

v_1	v_2	$f(v)$	0	$v_1 \oplus v_2$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	0	0

Where $0, v_1, v_2, v_1 \oplus v_2$ are the possible linear function of v_1 and v_2 and $d_1(f(v), 0) = 3$, $d_2(f(v), v_1) = 1$, $d_3(f(v), v_2) = 1$, $d_4(f(v), v_1 \oplus v_2) = 1$.
So,

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

.

Definition 2.5.3.

The measurement of correlation between the Boolean function g and all of the linear combinations is known as **Walsh transform**. The Walsh transform of a Boolean function g is defined by

$$WHT_{(f)}(\beta) = \sum_{\beta \in GF(2^n)} (-1)^{f(v) \oplus \beta \cdot v} \tag{2.2}$$

where $\beta \in GF(2^n)$ for all $v \in GF(2^n)$.

The non-linearity of a Boolean function $f(v)$ can be given by Walsh transform by the following formula

$$N_f = 2^{n-1}(1 - 2^{-n}) \max_{\beta \in GF(2^n)} |WHT_{(f)}(\beta)|). \text{ [36]}$$

Definition 2.5.4.

Bijection is a mapping in which each input bit produce a unique output. Let n be the possible input bits such as $\{0, 1\}^n$ there exist a unique output bit. Every output vector should appear one time. A method for calculating the bijective property was introduced for the $n \times n$ [38] S-boxes. An $n \times n$ S-boxes are said to satisfy the bijective property if for $g_i (1 \leq i \leq n)$the boolean functions g_i of S are such that:

$$\mathbf{wt}(\sum_{i=1}^{n} c_i g_i) = 2^{n-1} \tag{2.3}$$

where $c_i \in \{0,1\}$, $(c_1, c_2, \cdots c_n) \neq (0, 0, \cdots 0)$ hamming weight is $\mathbf{wt()}$. The condition 2.3 guarantees that every boolean function g_i and all their combination are balanced of 0/1, this demonstrates equal numbers of zeros and ones. The S-box generated in this case has all the different output values from the interval of $[0, 255]$ so it meets the condition of the bijective property. We illustrate the above definition by the following example.

Example 2.5.3. Let us consider 4×4 S-box and show it is bijective.
inputs: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
S-box: [9 13 10 15 11 14 7 3 12 8 6 2 4 1 0 5]^t
where each elements of S-box can be represented as:

$$S = \begin{bmatrix} f_1 : 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ f_2 : 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ f_3 : 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ f_4 : 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

That is, S(0000) = 1 1 1 1, $S(0001) = 1\,0\,1\,1, \ldots\ldots$, $S(1111) = 1\,0\,0\,0$. Since S-box is used both encryption and decryption, it should be a bijective mapping. This is to make sure that every S-box also has an inverse S-box.

Definition 2.5.5.

If half of the output bits changed as the result of changing single input bit then this is called **Avalanche Effect**. To understand Avalanche effect, choose a pair of n-bit plaintext vectors X and X_j which is dissimilar only in j_{th} bit, and their corresponding output bits are $f(X)$ and $f(X_j)$ which are different at least in bit

i. After this, taking XOR of output bits and we get:

$$V_j = f(X) \oplus f(X_j)$$

Each V_j contain n-bits, are called Avalanche variables. If the above procedure is repeated for $1 \leq j \leq n$, for each j one half of the variables are equal to 1, then f having a good avalanche effect. Now, we will define the **Strict Avalanche** property. To explain it we can use the above process but, we will apply an alternate method. First of all, n-bit random plaintext vector X is generated and find its corresponding ciphertext vector Y. Then, the n-vectors plaintext is:

$$(X_1,\ X_2,\ \cdots, X_n)$$

formed which in such a way that X and X_i are dissimilar only in j_{th} bit. The corresponding ciphertexts vectors are:

$$(Y_1, Y_2, ..., Y_n)$$

Then, we have:

$$Y_i = f(X_i)$$

Thus, we obtained avalanche vectors:

$$(V_1, V_2, ..., V_n)$$

such that

$$V_i = Y \oplus Y_i$$

Now value of V_i added in dependence matrix A. We repeat this process for large numbers of time. The degree of repeated procedure depends on the number of randomly generated plaintext vectors which is said to be r, and every element of matrix A divided by r. In matrix the value of 0 show that the ciphertext bits is totally independent of plaintext bits and 1 show that any change in plaintext will change the ciphertext.

Definition 2.5.6.

"An **Algebraic degree** of S-box is related with the nonlinearity measures. An algebric degree of boolean function $h(v)$ is defined as the highest degree of a function h, which can be expressed as

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

. Higher algebraic degree is considered more better than the lower algebraic degree." [37]

Definition 2.5.7.

The **Transparency order** of S-box is small provides a high resistance against differential power analysis (DPA) attacks. The smaller value of transparency order, provides a high resistance against differential power analysis (DPA) attacks where, DPA (differential power analysis) is a strong cryptanalytic technique which is used to remove secret data from cryptographic device. If the transparency order of a S-box is high then S-box cannot achieve its resistance against differential power analysis (DPA) attacks depends on the quality of the measurements an attacker can achieve.

Definition 2.5.8.

Generally, **signal to noise ratio** (SNR) refers to the distortion in transforming of signals from sender to receiver. By SNR, we mean signal to noise ratio. We can improve the sensitivity of performance by increasing the signal to noise ratio. SNR is used to evaluate the sensitivity performance of receiver.

Definition 2.5.9. (Robustness to Differential Cryptography)

The provision of robustness information on the strength of the S-box against the

differential attack is more correct. Differential uniformity is also used for this purpose, when more information about the power is needed than we considered the robustness. Robustness of S-boxes to differential cryptanalysis which is the most controlling cryptanalytic attack known to the data.

Definition 2.5.10.

The number of these **Fixed (F_p) and Opposite fixed points**(OFP) should kept as low as possible to avoid attacks in any statistic cryptanalysis [37].

Definition 2.5.11.

The Bit Independence Criterion (BIC) states that output bits will change independently when any single input bit changes. Measuring the coefficient of correlation among the couplings is the estimate of the degree of independence between pairs of avalanche variables. Presume in the S-box the Boolean functions are g_1, g_2, g_n. Studies have shown that $g_j \oplus g_k (j \neq k, 1 \leq j, k \leq n)$ can also satisfy the non-linearity and strict avalanche criterion if the two output bits g_j and g_k of Boolean functions satisfy BIC [21].

2.6 Software Tools for S-box Analysis

For studying the properties of S-box, some tools are available. A brief description of such tools is given below:

1. **Boolfun Package in R**

   R is a free, open source mathematical program used for statistical computing. It operates on different Windows, UNIX and Mac OS platforms, although the standard version of R does not support Boolean function evaluation, but

a package called Boolfun that provides features related to the cryptographic analysis of Boolean functions can be loaded [39].

2. **Sage Math**

   The Sage Math library is a free, open source mathematics tool that includes a Boolean function module and an S-box. With this method, we can check the algebraic properties and measure various cryptographic properties for S-boxes and Boolean functions related to the linear approximation matrix and difference distribution table [40].

3. **VBF**

   VBF stands for Boolean Function Library Vector. This tool was introduced by Alverez-Cubero and Zuf- ria [41] for the study of vector boolean functions used to test the cryptographic properties of S-boxes [41].

4. **SET**

   This method for evaluating the cryptographic properties of the Boolean function and S-boxes was proposed by Stjepan Picek [42] and his team. SET stands for S-box Evaluation Tool. It is a free tool for open source mathematics that is easy and convenient to use. It works in VS(visual studio) [42].

5. **SAMT**

   MATLAB software, is designed not only for the S-box but also for the Boolean functions that basically contruct the S-box. Any S-box defined as $S\colon GF(2^n) \rightarrow GF(2^n)$ for $2 \leq n \leq 20$ can be checked with this tool so it really provides a large space regarding Galois field for the analysis.

2.7 Chaos Theoy

Chaos is the study of the nonlinear and unexpected nature of surprises. Simply that is a way of predicting the unexpected. Chaos theory deals with nonlinear phenomena that are not easily stable or controllable, such as weather, volatility, stock market, our states of brain, etc. Nearly all chaos-based cryptographic algorithms use dynamic systems specified on the collection of real numbers, and hard for practical realization and implementation of circuits. When chaos is used in secret writing, it is called as chaos cryptography. Chaos cryptography is the study of fast designing of a secure system. The system dynamics have the capability to run under assured situation of chaos which is defined by the traffic model. It is the branch of mathematics which emphasize on the behaviour of dynamic system. Dynamic system is a system in which function rely on time dependent point in a geometrical space, i.e, moving pendulum, water flow in pipe etc. A map which present any kind of chaotic behavior is known as chaotic map. It may be discrete and continuous time parameter. Discrete maps are appropriate forms of iterated functions.

2.7.1 Application of Chaotic System

It is new and interesting field of abstract and complex mathematics. Until 1960 the world of science was relatively simple. Everything could be explained with formulas and had a predictable behavior. As the story goes, one day Edward Lorenz was working on weather forecasting machine, he decided to examine the past day sequence with more detail. He types the number from the previous day computer record and went to get a coffee. When he returned, he couldn't believe in his eyes. The new weather was nothing like the original. It was completely different. Then he realized that weather is a chaotic system. Actually chaos system is sensitive to initial condition. The application of chaos theory has been identified in many areas such as meteorology, sociology, physics, computer science, engineering, economics,

biology and philosophy. Chaos has applications like encryption, compression, and modulation in several functional blocks of the digital communication system. In the beginning, minor change leads to major change in future prediction. Chaos teaches us to believe on unexpected. It deals with the non linear things that are difficult to predict such as turbulence, weather, the stock market, brain states and so on. These phenomenons are often captured by the fractal mathematics. Almost all chaos based cryptographic algorithms use dynamical systems defined on the set of real numbers, and difficult for practical realization and circuit implementation.

2.7.2 Properties of Chaotic System

Chaos has been witnessed in many natural structures that cover a significant amount of technical and industrial areas. The phenomena of chaos can be found in almost all nonlinear deterministic systems. Chaos appears to exist when there is a continuous and disorganized progression in long term mathematical function. There are number of properties that summarize the characteristics observed in chaotic function.

1. **Self-Similarity:** It indicates the similar appearance at dissimilar scales of observation in an evolving systems with time or space.

2. **Non-Periodicity:** A chaotic system does have sequence of values for the evolving variable which repeat themselves resulting in periodic sequence beginning at any point in the sequence.

3. **Long-term Prediction:** Small changes in initial conditions, such as those caused by measurement errors or rounding error in numerical computation, can lead to significantly different outcomes for such dynamical systems, making long-term prediction difficult in general.

4. **Sensitivity to Initial Conditions:** Sensitivity to initial conditions means that the behavior of a system can diverge quickly by slightly different conditions, by making it unpredictable.

2.7.3 Lyapunov Exponent

The term "Lyapunov Exponent" has been widely used in the study of dynamical systems. They describe the rate at which neighboring trajectories in orthogonal directions converge or diverge. There are n exponents if the dynamic occur in an n-dimensional system.

$$\lambda_i = \sum_{i=1}^{n} \log |f'(x)|$$

2.7.4 Sine Map

The input of Sine function is $[0, \pi]$ and the output lies in $[0, 1]$. The chaotic sine map is derived from the sine function by transforming its input $[0, 1]$. The mathematical definition of the 1D sine map is:

$$y_{i+1} = \lambda \sin(\pi y_i); \quad y_0 \in [0, 1] \quad i \in \mathbb{Z}^+ \tag{2.4}$$

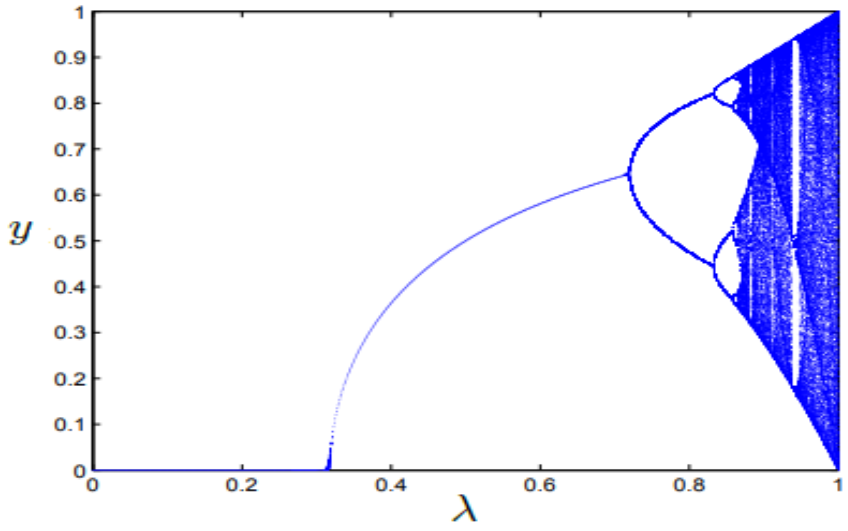

FIGURE 2.4: Bifurcation diagram of sine map

Where λ is the control parameter and the range is between $[0, 1]$. The bifurcation figure of sine map shows the chatoic behavior when $\lambda \in [0.87, 1]$. As the mathematical form of sine map and logistic map is totally different but their chaotic

behavior are quite similar which can be seen from the bifurcation figures. From the diagram of sine map clearly seen that sine map become chaotic when λ approaches to 1. The lyapunov exponent is widely used indicator for determining the chaotic behavior of a dynamical system.

Chapter 3

Construction of S-box based on Chaotic Map

In this chapter, the construction of S-box using one dimensional compound chaotic map [21] and encryption algorithm is discussed. An S-box is produced based on the compound chaotic map, linear congruence generator and sinusoidal map. The properties of S-box using SET [42] are also presented.

3.1 Compound Map

Consider chaotic piecewise functions T of having degree four and other is five. Where $t_1(u_m) = 16u_{m-1}^5 - 20u_{m-1}^3 + 5u_{m-1}$ and $t_2(u_m) = 8u_{m-1}^4 - 8u_{m-1}^2 + 1$ whose chaotic domain is $[-1, 1]$ [21]. The sinusoidal chaotic function $s(u) = \pi \sin(u)$, where the input range interval is $[0, \pi]$ and the output lies between $[0, 1]$.

$$T(u_m) = \begin{cases} 8u_{m-1}^4 - 8u_{m-1}^2 + 1 & u_m > 0 \\ 16u_{m-1}^5 - 20u_{m-1}^3 + 5u_{m-1} & u_m < 0 \end{cases} \tag{3.1}$$

The lyapunov exponent represents the sensitivity and signal randomness to the initial conditions. Entropy is the measure of uncertainty in the system. The entropy of the system can be used to calculate the level of disorder in the data. Higher entropy indicates higher uncertainty and a more chaotic system. Approximate entropy is a system complexity measurement method which represents the positive predictive power of data. As approximate entropy increases, with the period extension the relating time series becomes more complicated and odd. It can be seen, chosen maps are highly fragile for starting values and complication. So they will avail to construct chaotic sequences.

TABLE 3.1: Performances of chaotic mappings

	t_1	t_2	sin	compound
Lyapunov Exponents	1.6087	1.3862	0.68888	1.5087
Approximate Entropy	1.476666	1.2798	0.9594	1.4039

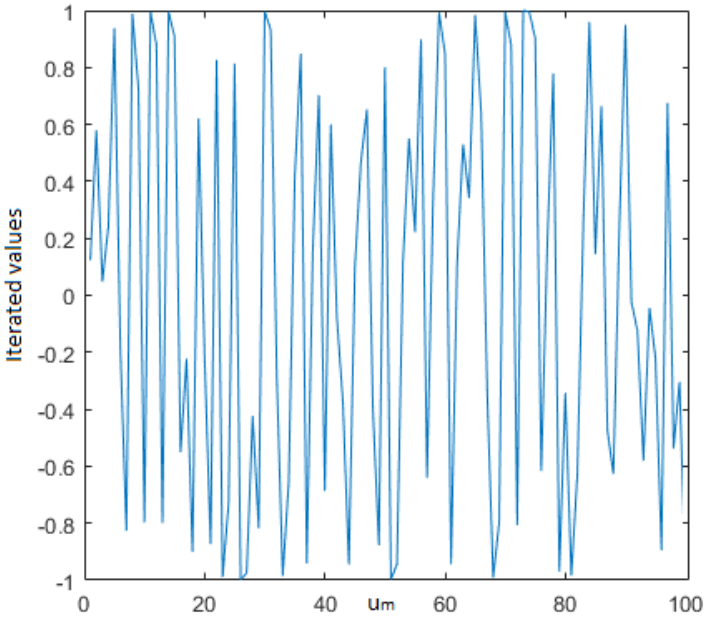

FIGURE 3.1: Key sensitivity of chaotic map.

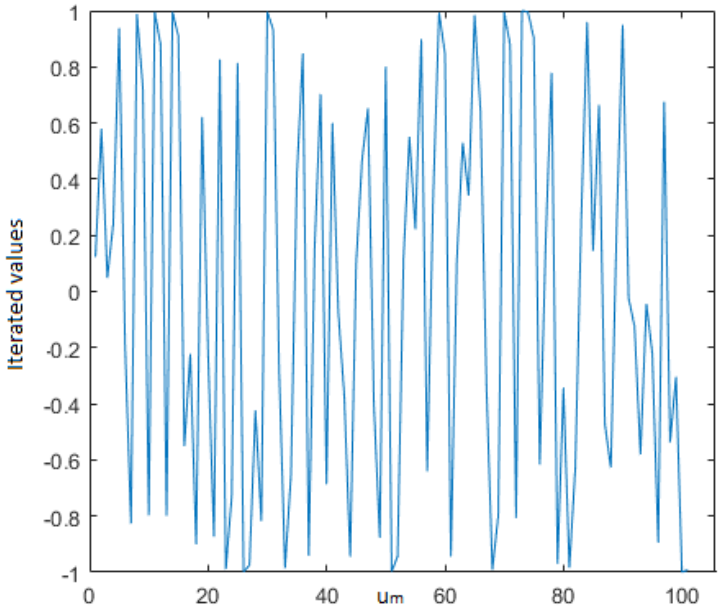

FIGURE 3.2: Key sensitivity of chaotic map.

3.1.1 Dynamic Compound Chaotic Sequences Generator

A strong structure that is chaotic must have considerable initial value sensitivity. For chaotic behavior $u_0 = 0.1234$ or $u_0 = 0.1234 * 10^{-11}$ as initial values are chosen. For any initial value $u_0 \in (0\ 1)$, its sequence is non-converging and non-periodic. The iterations are rendered 100 times under the same conditions, two chaotic sequences are acquired as presented in Figure 3.1 and 3.2 respectively. The system has good key, for the long term sensitivity actions. Iterate $T(u_m)$ from 1 to 100 we get a two sequences on two initial values which are non converging and non periodic. The chaotic orbits of 1D using $u_0 = 0.1234$ $u_0 = 0.1234 * 10^{-11}$ shown in Figure 3.1 and 3.2. Hence sequence u_m at starting values 0.1234 is

$$u_m = \{0.12340,\ 0.57987,\ 0.04868, \cdots\}.$$

When the initial value of u_m is $0.1234 * 10^{-11}$ the sequence is

$$u_m = \{0.00001234, ,\ 1.0,\ 0.9999,\ 0.987,\ \cdots\}$$

.

$$\begin{cases} t_0(u_{m-1}) = 16u_{m-1}^5 - 20u_{m-1}^3 + 5u_{m-1} \\ t_1(u_{m-1}) = 8u_{m-1}^4 - 8u_{m-1}^2 + 1 \\ u_m = T(u_{m-1}) = \begin{cases} t_0(u_{m-1}), & u_{m-1} < 0 \\ t_1(u_{m-1}), & u_{m-1} > 0 \end{cases} \end{cases} \tag{3.2}$$

1. Input two initial parameters u_0 and v_0.
2. Compute $u = (u_0 + v_0)/2$. If $u < 0$, to output an iterative value, select $t_0(u)$, otherwise select $t_1(u)$. Where u_0 is 0.123 and v_0 is 0.456.
3. If $i <= m$ (m is the sequence value number) then move to 2 or proceed to the last.

Select one of the dynamically generating functions to construct a chaotic binary sequence. The device has more randomness with trajectory conversion process. Put the last value into the t_1 function, if we obtain a value less than 0 after iterations $m-1$. Otherwise bring the final value into t_2 function. This method can generate random sequences $\{u_m\}_{m=1}^n$. The inital value of u is 0.2895 and iterate the sequence using MATLAB from 1 to 65536 following sequence is obtained. $\{u_m\} = \{0.2895,\ 0.3857,\ -0.0131,\ -0.3221,\ -0.9976, ...\}$.

3.1.2 Linear Congruence Generator (LCG):

In order to produce random sequences (LCG) is the simplest procedure which is given as:

$$u_i = (Cu_{i-1} + D) \mod (N) \tag{3.3}$$

here the initial value u_0 must assure $0 \leq u_0 < N$ and $N > 0$. C is multiplier, $0 < C < N$. The increment is D, $0 \leq D < N$. If $D = 0$, the Eq.(3.3)

is known multiplicative congruence method or multiplicative addition congruence method. ($C = 12$, $D = 365$ and $N = 65537$). Random sequences can generated by introducing the LCG $\{LCG(r)\}_{r=1}^{n}$ using MATLAB iterate the sequence from 1 to 65535 split the sequence into two subsequences LCG_1, LCG_2. Hence the sequences $\{LCG_1\} = \{34,\ 773,\ 9641,\ 50520, ...\}$ and $\{LCG_2\} = \{47128,\ 41605,\ 40866,\ 31998, ...\}$.

3.2 Method for the S-box Generation:

In this section the S-box generation method by using dynamic chaotic map, the sinusoidal chaotic map and linear congruence generator (LCG) is presented . The steps of S-box construction are given below.

1. **Digitization of Compound Chaotic Sequences:**

   The starting value is u of dynamic chaotic compound system. Iterate the sequence $\{u_m\}_{m=1}^{n}$ from 1 to n to generate chaotic sequence, where n is 65535.

   Initial S-box $\{S_1(m)_{m=1}^{n}\}$ is obtained by the following equation.

$$S_1(m) = \begin{cases} \text{ceil}((1 - \frac{arccos(u_m)}{\pi} * 65535), & (u_m \in [-1, 1)) \\ 65535, & (u_m = 1) \end{cases} \tag{3.4}$$

   Where ceil is a ceiling function defined in definition 2.3.6. The sequence values of $S_1(m)$ are $S_1(m) = \{34215,\ 36461,\ 18813,\ 44502, ...\}$ the sequence is non periodic and non converging.

2. **Digitization of the Sinusoidal Chaotic Sequence:**

   Sine map is defined in definition 2.7.4 The mapping is described by $f(u) = \pi \sin(u)$ where the domain interval is $I = [0, \pi]$. Pseudo random sequence can make through the map $v_{(m)_{r=1}^{n}} = f(u)$, where n is 65535. $S_2(m)_{m=1}^{n}$ can

be obtained by following equation.

$$S_2(m) = [\lfloor (v_m \cdot 10^8) \rfloor] \bmod 65536 \tag{3.5}$$

where $\lfloor . \rfloor$ is a floor function of dot defined in definition 2.3.5.
The sequence $\{S_2(m)\} = \{59157,\ 28980,\ 45270,\ 38569, ...\}$.

3. **Design of Preliminary S-box:**
It is possible to obtain discrete chaotic sequences with the help of method indicated above as $S_1(m)_{m=1}^n$, $S_2(m)_{m=1}^n$, $LCG_1(m)_{m=1}^n$, $LCG_2(m)_{m=1}^n$. Then the S-box with the discrete chaotic sequences are generated as follows.

$$\begin{cases} S_3(m) = S_1(m) \oplus LCG_1(m) \\ S_4(m) = S_2(m) \oplus LCG_2(m) \\ S(m) = S_3(m) \oplus S_4(m) \end{cases} \tag{3.6}$$

$S_3(m)$ sequence is obtained by taking XOR of $S_1(m)$ sequence and sub sequence $LCG_1(m)$ that is

$$S_3(m) = \{34215,\ 36461,\ 18813,\ 44502,\ 4638, \cdots, \}.$$

Similarly $S_4(m)$ sequence is obtained by taking XOR of $S_2(m)$ sequence and sub sequence $LCG_2(m)$ that is

$$\{S_4(m)\} = \{59157.\ 28980,\ 45270,\ 38569,\ 64031, \cdots, \}.$$

$S(m)$ is obtained by taking XOR of $S_3(m)$ and $S_4(m)$ that is

$$\{S(m)\} = \{25266,\ 65369,\ 63915,\ 15231,\ 59393 \cdots, \}.$$

Then change $S(m)$ to an equivalent value of u through $\bmod (S(m), 256)$ in $[0, 255]$). The decimal representation of $S(m)$ are

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

Finally, convert them into the form 16×16.

Table 3.2: S-box

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

3.2.1 Properties of S-box

Boolean function and S-boxes plays an important role for non-linear elements in stream and block cipher. Non-linear elements and their properties are important for security. In case of boolean function, text file are defined in form of truth table. In case of S-boxes text file are defined in decimal or hexadecimal format. Every function is computed which help the researchers when they are examining the tool or adding new functionality. An extensive documentation containing information about all function and instruction about their use is a part of secure code package to comfort the user. Many softwares used for properties of S-boxes. S-box Evaluation Tool (SET) [42] is a tool used for the analysis of non-linear elements and their properties. For this, we firstly install the Microsoft visual studio, then create a text file. After this, we compile and run the program which give us the properties of S-boxes. Its properties like non-linearity, correlation immunity, absolute indicator, sum of the square indicator, algebraic degree, algebraic immunity, transparency order are discussed in chapter(2).

3.2.2 Analysis using Set Tool

By using SET (S-Box Evaluation Tool) [42], the analysis is given below.

- S-box is balanced.
- Absolute Indicator is 104.
- Non-Linearity is 92.
- Sum of Square Indicator is 277504.
- Co-Relation Immunity is 0.
- Algebraic Degree is 7.
- Algebraic Immunity is 4.
- Transparency Order is 7.800.

- Number of Fixed Points is 0.
- Number of Opposite Fixed Points is 0.
- Composite Algebraic Immunity is 4.
- Robustness to Differential Cryptoanalysis is 0.953.
- Delta Uniformity is 10.
- SNR (DPA) (F) is 9.893.
- Confusion Coefficient Variance is 0.083823.
- S-box does not fulfill SAC.

Chapter 4

Encryption Scheme Based on 3D Chatoic Logistic Map

In this chapter, a new method is proposed to construct the S-box [22]. The comparison of S-boxes based on 3D and 1D chaotic map using SET tool [42] are also presented. In the second section the algorithms for the text encryption based on Feistal structure and 3D S-box is presented.

4.1 Logistic Map

Logistic map is a one dimensional chaotic map that has simple structure but complex chaotic behavior. The mathematical definition of logistic map is defined as:

$$p_{n+1} = rp_n(1 - p_n)$$

where $0 < p_n < 1$ and $r = 4$ is the condition to make this equation chaotic. where r is bifurcation parameter which lies in the interval $[0, 4]$. To begin the iteration, the initial value p_0 is set to be greater than 0 and less than 1. The resulting sequences is then non-periodic and non-converging. The logistic map is a discrete dynamic system, exhibiting chaotic behavior for its parameter values r. The bifurcation diagram is a numerical tool for illustrating the logistic map asymptotic

behavior for different parameter values r. On the horizontal axis of the map, the bifurcation parameter r is shown and the vertical axis displays the set of values of the asymptotically visited logistic function from almost all initial conditions. There are many chaotic maps in literature one of them is logistic map which we used to construct a new S-box.

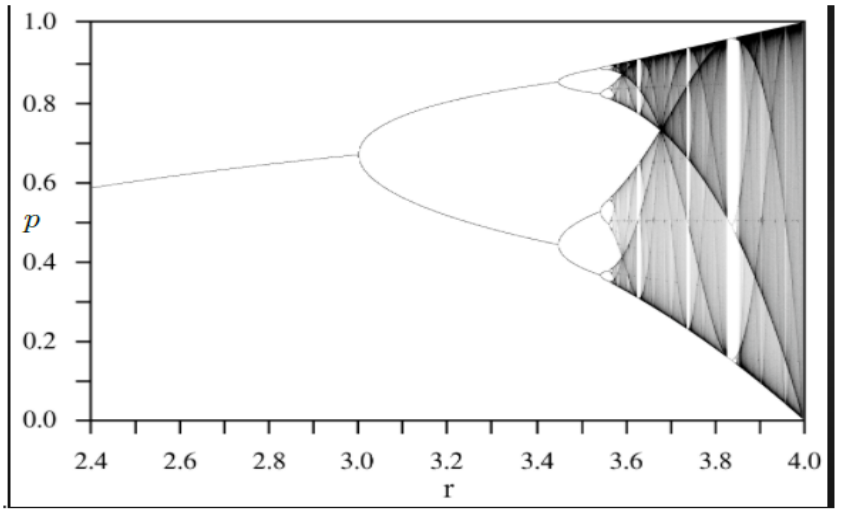

FIGURE 4.1: Bifurcation diagram of logistic diagram

Liu et al. proposed the 2D logistic map given by the formula:

$$p_{n+1} = \mu_1 p_n(1 - p_n) + \gamma_1 q_n^2 p_n \tag{4.1}$$

$$q_{n+1} = \mu_2 q_n(1 - q_n) + \gamma_2(p_n^2 + p_n q_n) \tag{4.2}$$

The formulas given above improve quadratic coupling p_n^2, q_n^2, $p_n q_n$ and provide the more security to the system. When $2.75 < \mu_1 < 3.4, 2.7 < \mu_2 < 3.45, 0.15 < \gamma_1 < 0.21$, and $0.13 < \gamma_2 < 0.15$, the system comes into chaotic state and can generate a chaotic sequence in the region $(0, 1]$.

4.1.1 Generation of 3D Chaotic Maps

By using the following formula, Pawan N.Khade [22] expand the 2D logistic map concept to 3D: The logistic map is given by an equation.

$$p_{n+1} = \lambda p_n(1 - p_n) + \beta q_n^2 p_n + \alpha l_n^3 \tag{4.3}$$

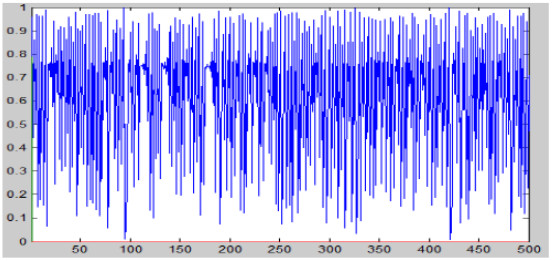

FIGURE 4.2: Plot of p component of 3D logistic map

$$q_{n+1} = \lambda q_n(1 - q_n) + \beta l_n^2 q_n + \alpha p_n^3 \tag{4.4}$$

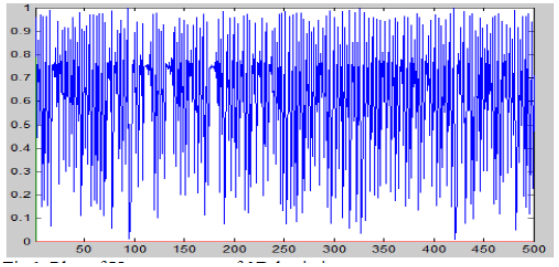

FIGURE 4.3: Plot of q component of 3D logistic map

$$\ell_{n+1} = \lambda \ell_n(1 - \ell_n) + \beta p_n^2 l_n + \alpha q_n^3 \tag{4.5}$$

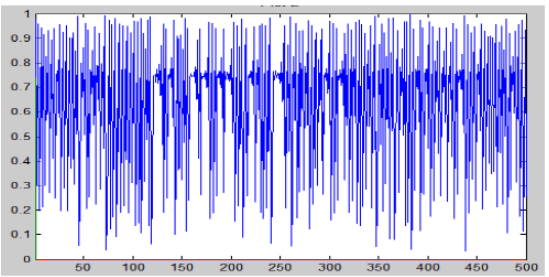

FIGURE 4.4: Plot of ℓ component of 3D logistic map

These equations exhibit the chaotic behavior for $3.53 < \lambda < 3.81, 0 < \beta < 0.022, 0 < \alpha < 0.015$. The initial values of p, q, l are any value between 0 and 1. Presence of cubic quadratic coupling and 3 constant terms make the 3D logistic map even more complicated sequences. The initial value of $p_1 = 0.2350$, $q_1 = 0.3500$, $\ell_1 = 0.7350$, $\alpha = 0.0125$, $\beta = 0.0157$, $\lambda = 3.7700$.

The presence of cubic, quadratic and 3 constants couplings conditions make the 3D logistics map much harder and more stable.

1. By using the MATLAB code we have computed 100000 values of p, q and ℓ.
2. The initial values for the system is taken as $p_1 = 0.2350$, $q_1 = 0.3500$, $\ell_1 = 0.7350$.
3. Iterate these sequences to 100000 and discard 30000 values.
4. After ignoring the starting values of sequences values are multiplied with 100001 to convert them in whole number.
5. After that apply floor function and then taking minimum value of p, q and sequence ℓ which is 34275.
6. The sequence p=$\{p_n\} = \{66386,\ 85316,\ 48360,\ 95419, ...\}$,
   q=$\{q_n\} = \{44949,\ 53949,\ 36732,\ 61463, ...\}$ and
   ℓ=$\{\ell_n\} = \{53953,\ 36786,\ 61477,\ 15345, ...\}$

1. **Linear Congruence Generator (LCG)**:
   For obtaining random sequence LCG is the common method which is defined as $u_i = (Cu_{i-1} + D) \bmod (N)$, where u_i is the sequence of pseudorandom values. N is the modulus, C is the multiplier and D is the increment. $C = 12$, $D = 365$, $N = 65537$, i varies from 1 to 65519 initial value of u_i is 500. Using MATLAB values have been computed few of them is given below. $LCG = \{500,\ 6365,\ 11208,\ 3787,\ 45809,\ 25777, ...\}$ the sequence is chaotic. The minimum value of u is 34275.

2. **Digitization of the Chaotic Sinusoidal Arrangement:**

The sinusoidal mapping is defined by $f(t) = \pi \sin t$. A pseudo-random sequence $v_{(k)}{}_{k=1}^{n}$ can be generated through the chaotic map, where n is 54681 and $v_{(k)} = f(k)$. $S_1(k)_{k=1}^{n}$ can be obtained by following equation. The initial value of k is 7691 and computed 54681 using MATLAB.

$$S_1(k) = [floor(v_k \cdot 10^8)] \bmod 65521 \tag{4.6}$$

Hence the sequence values of $S_1(k) = \{7691,\ 49511,\ 21515,\ 50902,\ 13610,\ 24710....\}$ is non converging and non periodic. The minimum value of sequence is 34275.

3. **Preliminary S-box Configuration:**

It is possible to obtain discrete chaotic sequences by using the method mentioned above, expressed as $LCG(u)_{u=1}^{n}$, $S_1(k)_{k=1}^{n}$. Then the S-box with the discrete chaotic sequences.

$$\begin{cases} S_2(k) = S_1(k) \oplus p \\ S_3(k) = q \oplus u \\ S_4(k) = S_2(k) \oplus S_3(k) \\ S_5(k) = S_4(k) \oplus \ell \\ S_5(k) = S_5 \bmod(256) \end{cases} \tag{4.7}$$

The $S_2(k)$ is obtained by taking the bitxor of sequence p and $S_1(k)$ that is $S_2(k) = \{60305,\ 105500,\ 55458,\ 106423,\ 41525,\ 81732,\ 31947,\ 105885,\ \cdots\}$ and $S_3(k)$ is the bitxor of the sequences q and u such that $S_3(k) = \{60305,\ 105500,\ 55458,\ 106423,\ \cdots\}$. Similarly $S_4(k)$ is bitxor of sequence $S_2(k)$ and $S_3(k)$ such that $S_4(k) = \{113576,\ 79328,\ 98860,\ 114699,\ 58945\ \cdots\}$. $S_5(k)$ is obtained by taking the bitxor of sequences $S_4(k)$, ℓ that is $S_5(k) = \{64909,\ 123987,\ 60863,\ 105230, 123150\ \cdots\}$. Convert $S_(k)$ to mod $(S_5(k), 256)$ with the corresponding value (valued at $[0, 255]$) such that

the decimal representation is $\{141,\ 62,\ 95,\ 179,\ 59,\ 214,\ 71,\ 184,\ 0,\ 171,\ 75,\ \cdots\}$. Finally, transcribe them to the 16×16 form. Using the above method we get the following S-box.

TABLE 4.1: S-box

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

4.1.2 Analysis of S-box using Set tool

By using SET (S-box Evaluation Tool) [42], the analysis is given below.

- S-box is balanced.
- Non-Linearity is 96.
- Absolute Indicator is 104.
- Number of Opposite Fixed Points is 1.
- Sum of Square Indicator is 264832.
- Co-Relation Immunity is 0.

- Transparency Order is 7.806.
- Algebraic Degree is 7.
- Composite Algebraic Immunity is 4.
- Robustness To Differential Crypto-analysis is 0.953.
- Delta Uniformity is 12.
- SNR (DPA) (F) is 9.353.
- Confusion Coefficient Variance is 0.124787.
- S-box does not fulfill SAC.

4.1.3 Comparison of the Properties of S-box

Using SET tool the properties of S-box are analyzed and compare the properties of both S-box based on 1D and 3D chaotic map. Where S_A represent S-box based on 1D and S_B represent S-box based on 3D logistic map.

TABLE 4.2: Comparision of the Properties of S-box

Properties of S-box	S_A	S_B
Balanced	yes	yes
Bijective	yes	yes
Fixed points	0	0
Opposite fixed points	0	0
Non-linearity	92	96
Sum of square indicator	271744	269056
Absolute indicator	96	96
SAC	not satisfied	not satisfied
Confusion Coefficient Variance is	0.10114	0.124787
SNR (DPA) (F) is	9.893	9.249
Robustness To Differential Crypto-analysis is	0.961	0.953
Delta Uniformity is	10	12
Composite Algebraic Immunity is	4	4
Transparency Order is	7.797	7.806
Co-Relation Immunity is	0	0

4.2 Design Algorithm for Encryption

A block cipher is used for the text encryption is the classical structure of Feistel. This algorithm use the length of input and output as 32-bit and 128-bit key length.

1. **Structure of Encryption Algorithm**

   Feistel network is an overall, popular block cipher structure. Confusion and diffusion are the key concepts of cipher design. The structure is shown in Figure 4.5. The formulation on encryption is as follows:

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus F(R_{i-1}, \quad P_{i-1}) \end{cases} \tag{4.8}$$

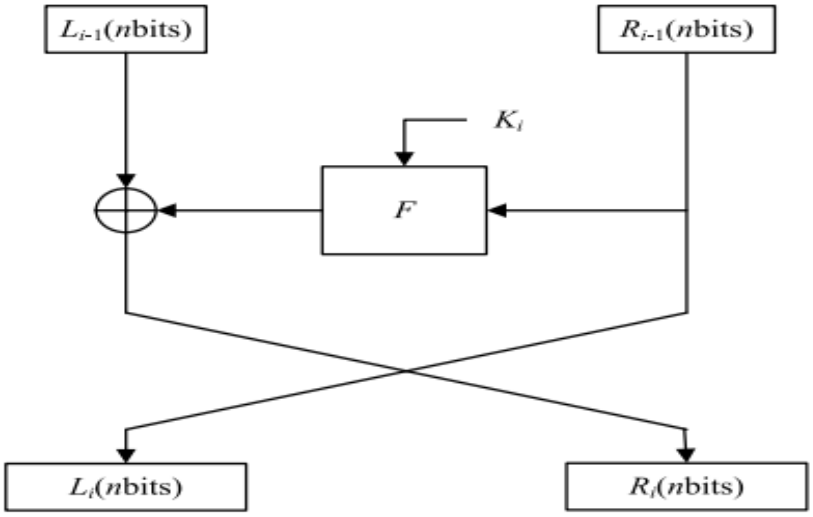

FIGURE 4.5: Encryption Scheme Structure.

Plaintext and ciphertext consist of fixed size blocks. Ciphertext is obtained from plaintext by iterating a round function. Input to round function consists of key and the output of previous round usually implemented in software. Plaintext block is split into left and right P=(L_0, R_0) halves and for i_{th} round named as L_i and R_i. For each round i=$(1, 2, \cdots n)$, compute

$L_i = R_{i-1}$ where F is a round function P_{i-1} is a subkey and ciphertext: C=(L_n, R_n). On the right half a round function is applied and this will used a subkey generated from a master key. The output of this function is XORed with the left half and then their halves are exchanged. Input bit block data has 32 bits length. Moreover, there is no need to exchange the final round of network for high and low byte. When the number of rounds is 3 the strength of diffusion and confusion are very strong in the experiment. Therefore implement the algorithm with a good level security the encryption round should be 32.

2. **Nonlinear Operation ($\tau(\cdot)$)**

   The nonlinear operation used in this technique is S-box. The S-box is denoted by $\tau(\cdot)$. Description of nonlinear operation is given as follows. For input is:
   $D = (d_0, d_1, d_2, d_3) \in (Z_2^8)^4$ and output $C = (c_0, c_1, c_2, c_3) \in (Z_2^8)^4$ then there is $(c_0, c_1, c_2, c_3) = \tau(D) = (S\_(d_0), S\_(d_1), S\_(d_2), S\_(d_3))$. S-box is applied on four bytes d_0, d_1, d_2, d_3.

3. **Linear Operation**

   The output data of nonlinear operation is the input data of linear operation, here XOR is linear operator. Where the input data is C, taking XOR of C with right circular shift of 13 bits of C and 23 bits of C.
   $L(C) = C \oplus (C <<< 13) \oplus (C <<< 23)$.

4. **Key Schedule**

   The subkeys are produced by the key scheduling algorithm at each encryption round in this encryption algorithm. The process of key scheduling decreases the device parameters and fixed parameters, depending on key extension algorithm of $SM4$. The $SM4$ algorithm is a block cipher, with block size of 128 bits and key length of 128 bits. The structure of encryption and

decryption are same, except that the round key schedule which has its order reversed during decryption. This technique decreases the potential of storage, but does not impact the security. Two operations, linear L and non linear $(\tau(\cdot))$ are performed. The 128 bits master key of algorithm is specified as $MP_0, MP_1, MP_2, MP_3 \in Z_2^{32}$. Z_2^{32} shows a word of 32 bits and Z_2^8 shows a byte. Initially,

$$(P_0, P_1, P_2, P_3) = (MP_0, MP_0 \oplus MP_1, MP_0 \oplus MP_2, MP_0 \oplus MP_3),$$

where $r = 0, 1, 2...$, rounds. Then do

$$rp_i = P_{i+4} = P_i \oplus V(P_i \oplus P_{i+1} \oplus P_{i+2} \oplus P_{i+3})$$

.

Where $V: Z_2^{32} \to Z_2^{32}$, $V(\cdot) = L(\tau(\cdot))\cdot$ Two operations are included in the V function, linear $L(\cdot)$ and nonlinear $\tau(\cdot)$. Depending on these operations, round subkeys rp_i can be generated to design the encryption algorithm.

4.2.1 Producing the Round Subkeys

A 32 bit block data is divided into 8 bits to generate 4 round subkeys as $rp_i = (rp_{i_0}, rp_{i_1}, rp_{i_2}, rp_{i_3}) \in (z_2^8))^4$ are the round subkeys.

The technique of generating the round subkeys are

$$\begin{cases} P_{i_0} = rp_{i_0} \\ P_{i_1} = rp_{i_0} \oplus rp_{i_1} \\ P_{i_2} = rp_{i_0} \oplus rp_{i_2} \\ P_{i_3} = rp_{i_0} \oplus rp_{i_3} \end{cases}$$

P_{i_0}, P_{i_2}, P_{i_3} are obtained by taking the bitxor with round subkeys. The algorithm for key set up is as follows:

Input: The 128 bits master key as $MP_0, MP_1, MP_2, MP_3 \in Z_2^{32}$

Output:

The round subkeys : $(P_{i_0}, P_{i_1}, P_{i_2}, P_{i_3})$

1. Take bitwise XOR of MP_0, MP_1, MP_2, MP_3 to get (P_0, P_1, P_2, P_3) where $(P_0, P_1, P_2, P_3) = (MP_0, MP_0 \oplus MP_1, MP_0 \oplus MP_2, MP_0 \oplus MP_3)$
2. while $i <$ Round do
3. $(d_0, d_1, d_2, d_3) \leftarrow P_i \oplus P_{i+1} \oplus P_{i+2} \oplus P_{i+3}$, $(d_0, d_1, d_2, d_3) \in (Z_2^8)^4$
4. $(c_0, c_1, c_2, c_3) \leftarrow (S\_(d_0), (S\_(d_1), S\_(d_2), S\_(d_3))$
5. $C \leftarrow (c_0, c_1, c_2, c_3)$, $C \in Z_2^{32}$
6. $P_{i+4} = P_i \oplus (C \oplus (C <<< 13) \oplus (C <<< 23))$
7. $rp_i \leftarrow P_{i+4}$
8. $(rp_{i0}, rp_{i1}, rp_{i2}, rp_{i3}) \leftarrow rp_i$
9. $(P_{i0}, P_{i1}, P_{i2}, P_{i3} \leftarrow (rp_{i0}, rp_{i0} \oplus rp_{i1}, rp_{i0} \oplus rp_{i2}, rp_{i0} \oplus rp_{i3})$
10. end while

4.2.2 Round Function F

The round function F, used in the block cipher algorithm is an essential component. If the round function is complex, it becomes harder to decode the cipher. The structure of this function is shown in 4.6. In Figure 4.6, $\oplus$ is a bitwise XOR and $+$ shows addition operation for module 256. The round subkeys used in this function are Kr_1, Kr_2, Kr_3, Kr_4. The technique uses simple rotation and nonlinear operations such as modules plus, cyclic shift, XOR in the function F. It improves algorithmic performance.

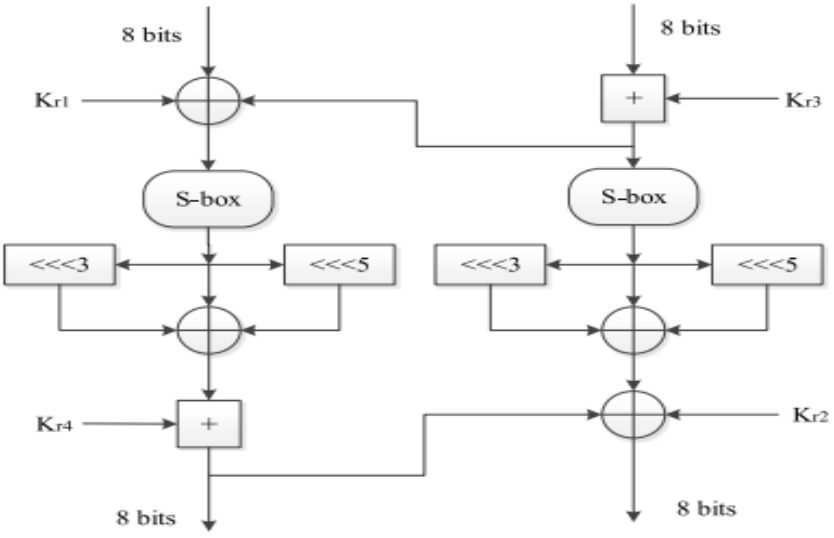

FIGURE 4.6: Round function F

1. Two inputs of size 8 bits each is given names a_1 and a_2 . On one side addition is applied and on other side XOR is taken place. The sum of the input a_1 is taken place with subkey Kr_3 and on other side XOR of a_2 is taken by Kr_1. Now XOR the both of their outputs.

2. S-box is applied on the result of step 1.

3. Left rotate the output of step 2 by 3 bits that is $<<<$ 3, and 5 bits such that $<<<$ 5.

4. XORed step 2 with step 3.

5. Sum is taken with the result of step 4 and subkey kr_4.

6. The output of above step is then XORed with subkey kr_3 and step 4.

4.3 Structure of Decryption Algorithm

The Feistel structure in cryptography, is a symmetrical system that is used in block ciphers. The benefit of this structure is a symmetric cryptosystem, i.e processes of data encryption and decryption are quite similar or even identical. Instead of

starting with a block of plaintext, the ciphertext block is fed into start of feistal structure and the process thereafter the same. The process is said be almost similar but not the same. In case of decryption, the only difference the subkeys use in encryption are used in reverse order. The final swapping of 'L' and 'R' is essential. The formula for decryption is as follows:

$$\begin{cases} R_{i-1} = L_i \\ L_{i-1} = R_i \oplus F(L_i, \quad F_{i-1}) \end{cases} \tag{4.9}$$

Where i is the number of round.

4.4 Properties of Encryption and Decryption Algorithm and Key Schedule

The properties of above algorithm is discussed in this section.

4.4.1 Implementing Encryption Algorithm and Decryption

We choose the plaintext "Temperature" and to implement the Figure 4.5 and its 128 bits master key is "A quick brown fox jumps over the lazy dog". In the decryption process if the key is changed the original data cannot be decrypted correctly, the cipher algorithm therefore has the outstanding efficiency of sensitive dependence on initial conditions.

4.4.2 Key Schedule Analysis

The algorithm improve the method of key expansion by applying simple rotation and nonlinear operations to produce the round subkeys. This increases the performance of the algorithm, but it has not reduced the safety. A master key of 128 bits is used in this algorithm, that and the key space can be extended upto

2128. But it is temporarily impossible to decode successfully given the current computing power by a powerful attack on the key.

4.4.3 Structure Analysis

These considerations are of great importance in developing cryptographic algorithms. When used in hardware-based applications, the most critical factor in terms of implementation is the space requirements. Linear operations and the S-box are chosen to build the round function F. Though the method uses more key space, but the performance has improved considerably. The Feistel layout is a reasonable choice to minimize the difficulty of the encryption algorithm as well as number of S-boxes.

4.4.4 Ciphertext Statistical Analysis

In general, there are many aspects to a analytical study of a block cipher such as: "0-1" balance, distribution of ASCII values and entropy analysis in the cipher text.

1. **"0-1" Balance**

   The balance "0-1" means the 0 and 1 numerical relationship. It has the following formula:

   $$\epsilon = \left| \frac{e_1 - e_2}{n} \right| \tag{4.10}$$

   where e_1, e_2 depict the number of 0 and 1 respectively, n is the total number of zero and ones. If value of ϵ close to 0, the better the balance is in the ciphertext. The ciphetext balance is 0.081.

2. **Statistical Character Frequency**

   In cryptographic analysis, statistical character frequency is an efficient attack technique. Owing to the high frequency and the low frequency of certain

characters statistical analysis of the characters can be used to accurately analyse the details.

3. **Information Entropy**

   The probability of discrete random events occurring is expressed by information entropy. Cipher sequence is choosen as a discrete random event.

$$H(S) = \sum_{S} P(s_i) log_2 \frac{1}{P(s_i)} \tag{4.11}$$

   where $P(s_i)$ is a probability of s_i. Ciphertext is split into many bytes that is 8 bits in this test. The data entropy value is 8 if the sequence dispersion is preferably random and uniform. The scheme become more chaotic if the value of entropy is close to 8. If the value of entropy is less than 8 the system would be unstable.

4. **Confusion and Diffusion Analysis**

   The completeness, avalanche effect, strict avalanche effect in the block cipher algorithm can also in some way explain the algorithm's security. These results are assign to together as non linearity.

   L and $L^{(i)}$ are the binary input vectors and their corresponding output vector are denoted by $T(L)$ and $T(L^{(i)})$. Assume T: $T_2^n \rightarrow T_2^m$ is a multiple output function, this implies that there is an m-bit output corresponding the n-bit input. Suppose the input vector is $L = (l_1, l_2, l_3, ...l_n)(l_k \in \{0, 1\}$, and $k = 1, 2, ...)$. Let $L^{(i)}$ for $i = 1, 2, ...n$ denote i_{th} bit change of X. Where the T vector is derived from F ($F \subset Z_2^n$), and the input vector number is represented as $\neq$F. The Hamming distance of L is the non-zero vector number of distinct bits referred as WH(L).

4.4.4.1 Ciphertext Analysis

Randomness of the sequence is found by $SP800_{22}$ suit test which provides 16 test as seen in table below. P values of 16 tests are greater than significant value

0.01. The test shows that our chaotic sequence is random and gives the values of frequency, block frequency, non overlapping, serial and linear complexity. The values of these element exceed 0.1 which proves the sequence is random.

TABLE 4.3: Test Result for the Ciphertext

Stastical Test	**P-Value**	**Result**
Frequency (monobit)	0.2112995473337106	Random
Frequency test within a block	0.2112995473337106	Random
Run Test	0.64806177602518	Random
Longest Run of Ones in a Block	0.9720587005536493	Random
Binary Matrix Rank Test	0.48124763401363657	Random
Discrete Fourier Transform (Spectral) Test	0.4912971242158921	Random
Non-Overlapping Template Matching Test	1.0	Random
Overlapping Template Matching Test	0.8865885582114529	Random
Maurer's Universal Statistical test	-1.0	Non-Random
Linear Complexity Test	0.49853075529672125	Random
Approximate Entropy Test	1.0	Random
Cummulative Sums (Forward) Test	0.3382072791980029	Random
Cummulative Sums	0.3382072791980029	Random
Serial test(P-value1)	0.773030538391796	Random
Serial test(P-value2)	0.6032393115402785	Random
Random Excursions Variant Test(x=-3)	0.11752486809663916	Random
Random Excursions Test(x=-4)	0.03763531378731428	Random

[h!]

Chapter 5

Conclusion

An S-box has a major role in symmetric key cryptography. Thus, it is very important to design a strong S-box. To design a suitable S-box is a difficult job, but several criteria have been proposed which provide protection against attacks. In this chapter, to conclude our work, we will discuss the security analysis of proposed S-box. As per calculations, we conclude that the non linearity of proposed S-box is 96 which is greater than compound chaotic S-box whose non linearity is 92. S-boxes also satisfy (BIC). So in current study these two properties are satisfied by proposed S-box and S-box based on 1D. It is concluded that, the S-box of this study meets the required properties, which proves that our proposed S-box is cryptographically strong and can resist against cryptanalysis attacks. Encryption algorithm is based on Feistal structure and S-box. A proposed S-box is based on three dimensional chaotic logistic map, sinusoidal map and linear congurence generator. Analysis shows that the proposed S-box has good result than 1D compound chaotic map. The presence of three quadratic equation made S-box more random and secure. A Feistel structure is a convenient option to reduce the encryption algorithm's complexity as well as the number of S-boxes. The S-box and linear operations are chosen in order to build the round function F. The algorithm enhanced the process of key expansion by simple rotation and nonlinear implementation. There is a 128-bit master key for the algorithm, so it is possible to increase key space to 2128 bits. It is not possible to decrypt the ciphertext by

a forceful attack on a key. The values of frequency (monobit) is 0.2112 HZ and entropy is greater than the limiting value.

In future research, we think that it is possible to optimize this S-box based on logistic map similar to the optimization performed for continuous chaotic system. We could apply the proposed S-box presented in chapter 4 for image encryption.

Bibliography

[1] O. Abraham and G. O. Shefiu, "An improved caesar cipher (icc) algorithm," *International Journal Of Engineering Science & Advanced Technology (IJESAT)*, vol. 2, pp. 1198–1202, 2012.

[2] S. Som, M. Kundu, and S. Ghosh, "A simple algebraic model based polyalphabetic substitution cipher," *International Journal of Computer Applications*, vol. 975, pp. 8887, 2012.

[3] T. Jakobsen, "A fast method for cryptanalysis of substitution ciphers," *Cryptologia*, vol. 19, no. 3, pp. 265–274, 1995.

[4] H. Panduranga, "Advanced partial image encryption using two-stage Hill cipher technique," *International Journal of Computer Applications*, vol. 60, no. 16, pp. 1102–1117, 2012.

[5] B. Acharya, G. S. Rath, S. K. Patra, and S. K. Panigrahy, "Novel methods of generating self-invertible matrix for Hill cipher algorithm," *International Journal of Computer Applications*, vol. 44, no. 6, pp. 1102–1117, 2007.

[6] W. Stallings, "Cryptography and network security, 4/E". *Pearson Education India*, vol. 12, no. 3, p. 74, 2006.

[7] N. Nedjah, L. de Macedo Mourelle, and C. Wang, "A parallel yet pipelined architecture for efficient implementation of the advanced encryption standard algorithm on reconfigurable hardware," *International Journal of Parallel Programming*, vol. 44, no. 6, pp. 1102–1117, 2016.

[8] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (blowfish)," *In International Workshop on Fast Software Encryption.* Springer, vol. 12, no. 3, pp. 191–204, 1993.

[9] M. M. Rahman, T. K. Saha, and M. A.-A. Bhuiyan, "Implementation of RSA algorithm for speech data encryption and decryption," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 12, no. 3, pp. 74, 2012.

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

[11] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[12] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography.* Springer Science & Business Media, vol. 10, no. 6, pp. 19–22, 2006.

[13] L. R. Knudsen and M. Robshaw, "The block cipher companion" *Springer Science & Business Media*, vol. 10, no. 6, pp. 74–84, 2011.

[14] W. Diffie and M. E. Hellman, "Special feature exhaustive cryptanalysis of the nbs data encryption standard," *Computer*, vol. 10, no. 6, pp. 74–84, 1977.

[15] V. Rijmen and J. Daemen, "Advanced encryption standard," *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, vol. 10, no. 6, pp. 19–22, 2001.

[16] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson, T. Kohno, and M. Stay, "The twofish teams final comments on AES selection," *AES round*, vol. 2, no. 1, pp. 1–13, 2000.

[17] N. T. Courtois, "An improved differential attack on full gost," *In The new codebreakers. Springer*, vol. 2, no. 1, pp. 282–303, 2016.

[18] G. Krishnamurthy and V. Ramaswamy, "Making AES stronger: AES with key dependent S-box," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no. 9, pp. 388–398, 2008.

[19] 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.

[20] X. Zhang, Y. Mao, and Z. Zhao, "An efficient chaotic image encryption based on alternate circular S-boxes," *Nonlinear Dynamics*, vol. 78, no. 1, pp. 359–369, 2014.

[21] L. Yi, X. Tong, Z. Wang, M. Zhang, H. Zhu, and J. Liu, "A novel block encryption algorithm based on chaotic S-box for wireless sensor network," *IEEE Access*, vol. 7, pp. 537–5390, 2019.

[22] P. N. Khade and M. Narnaware, "3D chaotic functions for image encryption," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 3, pp. 323, 2012.

[23] M. S. Iqbal, S. Singh, and A. Jaiswal, "Symmetric key cryptography: Technological developments in the field," *International Journal of Computer Applications*, vol. 117, no. 15, pp. 325, 2015.

[24] W. G. Barker, *Introduction to the analysis of the Data Encryption Standard (DES)*. Aegean Park Press, vol. 117, no. 15, pp. 5307–5390, 1991.

[25] M. A. Musa, E. F. Schaefer, and S. Wedig, "A simplified AES algorithm and its linear and differential cryptanalyses," *Cryptologia*, vol. 27, no. 2, pp. 148–177, 2003.

[26] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[27] W. Stallings, L. Brown, M. D. Bauer, and A. K. Bhattacharjee, "Computer security: principles and practice" *Pearson Education Upper Saddle River, NJ, USA*, vol. 21, no. 2, pp. 120–126, 2012.

[28] J. J. Rotman," A first course in abstract algebra" *Pearson College Division*, vol. 16, no. 6, pp. 1070–1078, 2000.

[29] Z. Zhang, H. Wang, A. V. Vasilakos, and H. Fang, "Ecg-cryptography and authentication in body area networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1070–1078, 2012.

[30] J. W. Ceaser, "Presidential selection: Theory and development," *Proceedings of the IEEE*, vol. 79, no. 5, pp. 598–620, 1979.

[31] C. J. Monico, "Semirings and semigroup actions in public-key cryptography," *Ph.D. dissertation, University of Notre Dame Notre Dame*, vol. 1, no. 1, pp. 1–11, 2002.

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

[33] J. Kerl, "Computation in finite fields," *Arizona State University and Lockheed Martin Corporation*, vol. 1, no. 1, pp. 1–84, 2004.

[34] P. Jovanovic and M. Kreuzer, "Algebraic attacks using sat-solvers," *Groups Complexity Cryptology*, vol. 2, no. 2, pp. 247–259, 2010.

[35] C. Carlet, Y. Crama, and P. L. Hammer, "Vectorial Boolean functions for cryptography" vol. 1, no. 1, pp. 1–84, 2010.

[36] J. Daemen and V. Rijmen, "The design of Rijndael: AES-the advanced encryption standard springer science & business media," vol. 23, no. 2 pp. 362–366, 2013.

[37] K. Mohamed, M. N. M. Pauzi, F. H. H. M. Ali, "Study of S-box properties in block cipher," *In International Conference on Computer, Communications, and Control Technology (I4CT). IEEE*, vol. 24, no. 3, pp. 362–366, 2014.

[38] 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.

[39] F. Lafitte, "The boolfun package: Cryptographic properties of Boolean functions," vol. 26, no. 3, pp. 365–369, 2012.

[40] W. Stein, "Sage mathematics software," *http://www. sagemath. org/*, vol. 21, no. 4, pp. 141–148, 2007.

[41] J. A. Alvarez-Cubero and P. J. Zufiria, "A C++ class for analysing vector Boolean functions from a cryptographic perspective," in *International Conference on Security and Cryptography (SECRYPT) IEEE*, vol. 23, no. 2, pp. 1–9 2010.

[42] 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,* vol. 20, no. 4, pp. 140–149, 2014.