CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Review on an S-Box Design Algorithm Based on a New Compound Chaotic System

by

Shazia Ramzan

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

in the

Faculty of Computing

Department of Mathematics

2021

To my parents for their prayers and love, for giving me the determination to overcome many trying moments to pursue my dreams

CERTIFICATE OF APPROVAL

Review on an S-Box Design Algorithm Based on a New Compound Chaotic System

by

Shazia Ramzan

(MMT191012)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Shabieh Farwa	COMSATS, Wah Cantt
(b)	Internal Examiner	Dr. Dur e Shehwar Sagheer	CUST, Islamabad
(c)	Supervisor	Dr. Rashid Ali	CUST, Islamabad

Dr. Rashid Ali
Thesis Supervisor
December, 2021

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

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

Author's Declaration

I, **Shazia Ramzan** hereby state that my M.Phil thesis titled "**Review on an S-Box Design Algorithm Based on a New Compound Chaotic System**" 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.

Shazia Ramzan
Registration No: MMT191012

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**Review on an S-Box Design Algorithm Based on a New Compound Chaotic System**" 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.

Shazia Ramzan

Registration No: MMT191012

Acknowledgement

All praise be to Almighty ALLAH who has been bestowing me with his great bounties and enabled me to complete my dissertation.

I would like to thank my affectionate teachers, Dr. Muhammad Sagheer, Dr. Abdul Rehman Kashif, Dr. Shafqat Hussain, Dr. M. Afzal and **Dr. Rashid Ali** for their excellent teaching and support during these years. I would like to express my special gratitude to my supervisor **Dr. Rashid Ali** for his patience with me and guidance. A more supportive and considerate supervisor I could not have asked for. He was a big support and motivation in the difficult times as he encouraged and helped a lot during research and writing of thesis. I feel really blessed and proud to be his student.

I truly appreciate my mother for her love and continuous support, for without her, I would not have finished my degree. I am grateful to my father for all the prayers, who always supported me in achieving my targets. I would like to thank my sister and brother for their motivation.

I have also had the good fortune to study with some wonderful people: Aroush Fatima,M. Ibrahim Nazia Asif, Rahila Riaz and Khuzaima Nasir who helped and encouraged me throughout my studies. Their friendships are what I will miss the most and hope to keep forever. Especially, I would like to acknowledge Ms. Rahila Riaz and appreciate her friendship and contribution. My friend Nazia Asif, your support made all the difference and I cant thank you enough for driving me towards my goal always.

Finally, I am obliged to all the people who prayed for me, shared their knowledge during my degree program and supported me.

Shazia Ramzan

Abstract

Substitution-boxes (S-boxes) are important nonlinear components in block cryptosystem. The nonlinearity plays an important role in the security of cryptosystems. The S-boxes are used to increase the confusion ability of the cipher. Constructing S-boxes with a strong cryptographic feature is an important step in designing block cipher systems. A number of researchers proposed different methods for the construction of S-boxes based on chaotic maps. In this thesis, the new method for the construction of an S-box is reviewed. The method is based on compound chaotic system tent-logistic system, which has better chaotic performance and vast chaotic range than the tent map and logistic map. The main work is to construct simple and efficient S-box by using linear mapping and tent-logistic system, which can improve the efficiency of S-boxes. The scheme is implemented on MATLAB to construct the proposed S-box. The analysis of cryptographic strength of the constructed S-box is performed by using SAMT tool on MATLAB. Test comparison of constructed S-box with some old S-boxes shows that the obtained S-box by using the proposed procedure is better than other S-boxes.

Contents

List of Figures

List of Tables

Abbreviations

AC	Auto-Correlation
AES	Advanced Encryption Standard
ANF	Algebraic Normal Form
BIC-SAC	Bit Independence Criteria for SAC
BIC-NL	Bit Independence Criteria Nonlinearity
CI	correlation Immunity
DD	Dynamic Distance
DES	Data Encryption Standard
DP	Differential Probability
GF	Galois Field
IBM	International Business Machines
LP	Linear Probability
LE	Lyapunov Exponents
NL	Nonlinearity
PKC	Public Key Cryptography
RSA	Rivest-Shamir-Adleman
SAC	Strict Avalanche Criteria
SPN	Substitution-Permutation Network
S-box	Substitution Box
SET	S-box evaluation Tool
TT	Truth Table
TLS	Tent-Logistic System
VBF	Vector Boolean Function

Symbols

$\mathbb{C}$	Complex Number
$\mathbb{F}$	Field
$\lfloor \cdot \rfloor$	Floor Function of $\cdot$
$\mathbb{G}$	Group
m	Number of Input Bits
n	Number of Output Bits
$\mathbb{R}$	Real Number
R	Ring
$\mathbb{Z}$	Integers
Greek Letters	
δ	Control Parameter
λ	Lyapunov Exponent

Chapter 1

Introduction

Cryptology is the combination of two Greek words, 'kryptos' and 'logos', whose meanings are concealed and words respectively. In 1645, James Howell [1] invented the term cryptology. It is a branch of science concerned with secure communication of secret data. There are two primary branches, namely.

- Cryptography
- Cryptanalysis

Cryptography is the branch of cryptology, in which communication takes place in the secure fashion in such a way that no third party can read or change the information. Cryptography is used to hide the original information into coded form so that it cannot be read by anyone who is not intended for it. In cryptography, for the better understanding usually name of the two parties who share information with each other are considered as Alice and Bob. There are also some technical terms that are used for the secure communication between Alice and Bob over a public network. Alice converts the **plaintext** into **ciphertext** and sent it to Bob. The process of converting the plaintext into ciphertext is called **encryption** and process of converting ciphertext back into plaintext is called **decryption**. Obviously, there is an algorithm used to alter the plaintext into ciphertext such an algorithm is known as **Encryption algorithm**. A ciphertext can not be

understood as long as it is transformed back into the plaintext and the algorithm that is used to get the plaintext from ciphertext is called **Decryption algorithm**. There is a highly sensitive information used in encryption and decryption algorithm for conversion of plaintext and ciphertext, called **key**.
Cryptography is further classified in the following two categories:

- Symmetric Key Cryptography
- Public Key Cryptography

Symmetric key cryptography is also called the secret key cryptography. It was the only technique used for transmitting messages before the development of public key cryptography. In this method, only one key is used for both encryption and decryption by the sender and receiver and not known to the adversary (attacker). A secret key which can be a number, a word, or just a string of random letters is applied to the text or message to change the contents in a particular way. This might be as simple as shifting each letter by a number of places in the alphabet. As long as both sender and receiver know the secret key, they can encrypt and decrypt all messages. This was the only method which was used for secure communication until 1976 [2]. Examples of symmetric key encryption scheme includes Data Encryption Standard (DES) [3], RC4 [4] and Advanced Encryption Standard (AES) [5]. The drawbacks of symmetric key cryptography is key sharing and authentication.

To resolve the problems with symmetric key cryptography, Diffie-Helman proposed the idea of **public key cryptography** (PKC) in 1976 [2]. This concept is based on the one-way trapdoor function (it is easy to calculate in one direction but difficult to compute in the opposite direction without knowing the special information) for exchanging the key between two parties. The major drawback of the symmetric key is the management of the single key between sender and receiver. That is why the communication area needs a secure system that does not have security and key management issues.

Public key cryptography depends on two type of keys in which one is used for encryption (public key) and the other key is used for decryption (secret key).

Examples of such a system are ElGamal [6], RSA [7] etc.

A **stream cipher** is a symmetric key cipher that combines plaintext digits with a pseudorandom cipher digit stream (keystream). Each plaintext digit with the corresponding digit in a stream cipher is encrypted one by one to give one digit to the ciphertext stream.

A **block cipher** is a symmetric key cryptosystem that encrypts or decrypts one block of data at a time. IBM published the Data Encryption Standard (DES) in 1970 [3] as a symmetric block cipher. DES can encrypt 64-bit data with a block size of 8 bytes using a 56-bit key. Due to its limited key size, it could be broken by brute force in less than 24 hours [3]; for the removal of flaws and improvement purposes, 2DES and 3DES were developed. In 2001, Vincent Regimen and John Daemon presents a more difficult algorithm Called Rinjindael, known as Advanced Encryption Standard [5]. The major components of AES is S-box that provides the nonlinearrity. It is used in the field of cryptographic that helps to secure the system. Futher detail is given in the next section.

1.1 S-boxes in Cryptography

In cryptography, an S-box (substitution-box) is a fundamental component of symmetric key algorithms which performs substitution. S-boxes are essentially Boolean vectorial functions given as look-up tables. An S-box takes a small bit block and replaces it with another bit block. To make decryption effective, this substitution should be one-to-one. The S-box usually accepts m input bits and translates them to n output bits. An S-box $(m \times n)$ may therefore be viewed as a 2^m word n-bit look up table. An S-box should be constructed to make each output bit strongly dependent on every input bit.

In block ciphers, they are commonly used to conceal the relationship between the key and the ciphertext. S-boxes are designed on the basis of Shannon theory of confusion and diffusion and it is also implemented in substitution-permutation networks (SPN). SPN is a kind of block cipher that consists of multiple rounds, each of which includes a substitution, permutation, and key material addition.

The cipher blocks are created based on the idea of Confusion and Diffusion that is also implemented in the SPN [8]. Such networks consist essentially of a number of interconnected mathematical processes. Plaintext along with keys is taken as an input to obtain ciphertext by following the rounds of S-box. To obtain a plaintext, the inverse S-box with the same key is used for decryption. For example, DES [3] and AES [5] are examples of SPN cryptosystems.

1.2 Literature survey

Information security has become a popular issue due to the rapid expansion of communication networks and big data applications. Many scholars have suggested different methods to secure the information such as information encryption [9], watermarking [10], and privacy protection [11]. Cryptography is the fundamental technique in information security. Block encryption methods are commonly employed in symmetric cryptographic systems, such as the DES [3], AES [5] , and others. The substitution box is an crucial non-linear component of a block cipher scheme.

Considering the significance of S-box in block cipher systems, cryptosystem designers have long attempted to establish S-box with high cryptographic performance. There are many methods [12–14] are proposed for the construction of S-boxes. A byte conversion process is generated with an S-box to obtain a ciphertext block that corresponds to a plaintext block. Each element will be mapped using an S-box in the sub-byte process. The S-box is used to randomly modify the bit input. As a consequence, linear and differential attacks have a hard time for breaking the output bit sequence. S-boxes are built by using a variety of methods, including the analytical approach [15], algebraic techniques [16], Boolean function [17], and triangle groups [18]. S-box is the basic component of AES which is considered to be an effective cryptosystem. Since S-box has a major role in cryptography, it is essential to construct a cryptogrphically good S-box. Robustness of the S-box is improved by using a dynamic system rather than a static system.

The key schedule algorithm of RC4 is used to generate dynamic S-boxes, which is generated by changing the secret key in every round [19]. After this, the random S-box and inverse S-box are designed [20]. Later on, dynamic S-boxes are constructed by using chaotic maps [21, 22].

Most of the researchers believe that there is a close link between chaos and cryptography. A chaotic map is a map which contains some kind of chaotic behavior [14, 22, 23]. Their behavior may be continuous or discrete. Chaotic systems is very sensitive to initial conditions, so a little change in initial conditions will be able to design a very different maps from the same dynamical system. Maps are useful for the cryptograpic purposes therefore they are vastly used in the construction of S-box. It may be continuous or discrete. Chaotic maps address the discrete time-dynamic system represented by the equation.

$$y_{i+1} = f(y_i)$$

where f is a function that translates the current state y_i to the next state y_{i+1}.

Repeated iterations of map f, starting with initial condition y_o, produce a sequence of points

$$\{y_i : i = 1, 2, ..., \},$$

known as the orbit of a dynamical system. The nature of chaotic maps are deterministic, reproducible, uncorrelated and random like, which can be helpful to enhance the security of transmission in communication. Different techniques have been proposed in [13, 16, 24, 25] to construct the S-box on the basis of chaotic map.

Lambic [24] developed S-box by using chaotic map. In this method, for the construction of the S-box, a discrete chaotic map on the basis of the composition of permutation is used.

Lambic [25] proposed an algorithm based on chaotic maps to get random bijective

S-boxes . Lambic's method has the advantage of low complexity and the ability to achieve a large key space. A new method for creating cryptographically secure bijective substitution-boxes based on a 5D hyper-chaotic system was proposed in [26]. Belazi et al. [27] suggested an S-box approach that is based on the chaotic logistic-sine map that is both efficient and effective. Cavusoglu [13], used the chaotic scaled Zhongtang system to generate a robust S-Box. Ullah [28] used the chaotic system and linear fractional transformation to create S-box. A basic S-box approach based on the chaotic sine map was presented by Belazi and El-Latif [29]. The S-box is able to generate random integer sequences with highly efficient nonlinearity in the generated values. A new approach for creating cryptographically secure S-boxes on the basis of 5D hyper-chaotic system is proposed in [26]. A basic S-box based on the chaotic logistic-sine map, Belazi et al. [27] suggested an efficient S-Box approach. It is the simple and efficient S-box method was introduced to use the designed scheme in secure color image encryption technique. The major advantage of the proposed strategy is the dynamic aspect of keys used by chaotic map to generate strong S-boxes.

Jakimoshi and Koravec [30] have proposed two well known methods to create a S-box based on chaotic maps, one is logistic and other is exponential. Logistic chaotic map consists of four step method to generate S-box. This map includes a proper choice of parameters, discretization for designing a secure cryptosystem. Tang et al. [21] have proposed the method for designing 8 × 8 S-boxes using 2D chaotic baker map and analyzed their cryptographic properties. Chaotic baker map consists of two steps to generate S-box. Afterwards, Chen [31] proposed the method for designing S-boxes using 3D chaotic baker map. Their method was better than Tang et al method. Ozkaynak [32] have proposed the method for designing strong S-boxes using chaotic map. They choose a Lorentz system for chaotic map and analyzed that system was better for secure communication. Another 1D choatic map with a tent-like form was introduced by Zhou [33].

However, the above chaotic S-box construction approaches do not have a high linear probability (LP) or differential probability (DP) score, and their resistance to linear and differential attacks was not optimal. Furthermore, the earlier schemes' S-box creation procedure is extremely difficult and inefficient. Low-dimensional discrete chaotic systems can generate chaotic sequences more efficiently than high-dimensional continuous-time chaotic systems. Furthermore, several researchs demonstrate that discrete systems have a higher complexity than continuous systems [34, 35]. On the other hand, low-dimensional discrete mapping chaotic systems have a limited chaotic range and weak chaotic features. If such chaotic systems are used to construct S-boxes, the crucial space of cryptographic systems will be reduced, and cryptographic performance will be less than optimal. To tackle this problem, new discrete chaotic systems with improved performance must be designed.

To address the previous mentioned shortcomings of existing chaos based S-box construction methods, a novel and efficient S-box construction approach based on a new compound chaotic system is proposed by Lu et al. [36]. To improve the properties of LP and DP in S-boxes and improve the cryptosystem more resistant to linear analysis and differential attacks. The compound chaotic system TLS that has a broader chaotic range and higher chaotic performance than previous ones, making it more appropriate for cryptography applications. As a consequence, the generated S-box by using TLS has a higher score of LP and DP, so it helps in resisting the linear and Differential cryptanalysis Attack.

1.3 Thesis Objective

The objective of this thesis is to study the scheme of Lu et al. [36] for the construction of strong S-box. The proposed scheme is based on the compound chaotic system that is tent-logistic map. It is the combination of two chaotic maps that are tent map and logistic map. In this dissertation, the S-box is generated by using compound chaotic system (TLS). The properties of the generated S-box is performed and the analysis of cryptographic strength of constructed S-box is performed by using the SAMT tool on MATLAB. Test comparison of constructed

S-box with some old S-boxes shows that the obtained S-box by using the proposed procedure is better than other S-boxes is discussed in Section 4.3.

1.4 Layout of Thesis

The dessertation is composed as follow:

- **Chapter 2** gives the information about the basic definition that helps in the construction of Boolean function and their properties. It is required for the analysis of S-box. The properties of S-box are also presented with examples.
- **Chapter 3** describes the chaos theory and chaotic maps (tent map, logistic map and tent-logistic map). Some information of S-boxes generated by chaotic maps is also given in this section.
- **Chapter 4** is based on the design algorithm of the S-box, using tent-logistic map. After that the properties of constructed S-box are checked.
- **Chapter 5** gives the conculsion of the thesis.

Chapter 2

Substitution Boxes

Substitution box (S-box) is a bijective function that accepts an n-bit input and returns an m-bit output. It is the fundamental part of a symmetric encryption that conducts substitution. The S-box conceals the relationship between the ciphertext and the key. In this chapter, the definitions and basic concepts of group theory and algebra are explained that involve the construction and analysis of S-box. Some important cryptographic properties of S-box are also presented in this chapter that are regarded as unavoiable for the analysis of a S-box and analysis softwares of S-box are also defined.

2.1 Mathematical Background

To comprehend the explanation for the creation and success of the S-boxes, some fundamental principles of group theory are introduced first.

Definition 2.1.1.
Let $\mathbb{G}$ be a non empty set and * be a binary operation on $\mathbb{G}$. Then $(\mathbb{G}, *)$ is called a **Group**. If the following properties holds:

1. Closure: For all $b, c \in \mathbb{G}$, $b * c \in \mathbb{G}$.

2. Associative: For all $b, c, d \in \mathbb{G} \quad (b * c) * d = b * (c * d)$.

3. Identity: There exist an element $e \in \mathbb{G}$ such that $b * e = e * b = b$

4. Inverse: If $p \in \mathbb{G}$, then there exist an element $p_1 \in \mathbb{G}$ such that $p * p_1 = p_1 * p = e$

If the group $\mathbb{G}$ holds

$$b * c = b * c$$

for all $b, c \in \mathbb{G}$ then $\mathbb{G}$ is called an **Abelian Group**.

Example 2.1.2. Some examples of group and abelian group are given below:

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

2. Set of all invertible matrices of order $n \times n$ with ordinary matrix multiplication forms a group.

3. Set of real number $\mathbb{R}$ is a group under addition.

4. The set $\mathbb{R}$ and set of integers $\mathbb{Z}$ are the examples of abelian groups with respect to addition.

5. The set of $\mathbb{R} \setminus \{0\}$ is an example of an abelian group with respect to multiplication.

Definition 2.1.3.

A non-empty set R together with two binary operations '$+$' and '$*$' defined on R is said to be **Ring**, if the following axioms are satisfied.

1. $(R, +)$ is an abelain group.

2. $(R, *)$ is a semi-group.

3. Distributive property of multiplication over addition holds *i.e.*, for all $c, d, e \in R$.

$$c * (d + e) = c * d + c * e \quad \text{and}$$

$$(c + d) * e = c * e + c * e,$$

it is usually written as $(R, +, *)$ or simply R is a ring.

Example 2.1.4. some examples of ring are given below:

1. $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ all form ring under usual addition and multiplication.
2. $M_n(\mathbb{R})$ set of all $n \times n$ matrices over the ring R is also a ring under addition and multiplication .
3. If p is a prime number then the set $\mathbb{Z}_p$ of integer mod p is a ring with respect to the modulo addition and multiplication of integer.
4. Set of odd integer is not a ring because it does not satisfied closure property under multiplication.

Definition 2.1.5.
If a set $(\mathbb{F}, +, *)$ has all of the properties of a Ring $(\mathbb{F}, +.*)$ and $(\mathbb{F}\diagdown\{0\}, *)$ is an abelian group, then $\mathbb{F}$ is said to be a **Field**.

Example 2.1.6. Some examples of field are given below.

1. Set of $\mathbb{R}$ and $\mathbb{C}$ numbers are fields under usual addition and multiplication.
2. Set of $\mathbb{Z}$ is not a field as there are no multiplicative inverses in $\mathbb{Z}$.

Recall that a polynomial $P(z)$ of degree n in indeterminate z is an expression of the form

$$P(z) = a_n z^n + a_n z^{n-1} + \ldots + a_1 z + a_0$$

OR

$$P(z) = \sum a_i z^i \quad \forall \quad i = 0, 1, 2, \ldots, n,$$

where a_i are its coefficients, z^i are its variables. Degree of polynomial is highest power of z. Further, if a polynomial $m(z)$ with integer coefficients cannot be factorized as a product of two lower degree polynomials, then it is said to be an **Irreducible polynomial**.

Example 2.1.7. The polynomials $z^2 + 1$, $z^2 + z$ are reducible polynomials over $GF(2)$ and $z^2 + z + 1$, $z^3 + z + 1$ are the examples of irreducible polynomials over $GF(2)$ [37].

Example 2.1.8. Consider the two polynomial $(z^7 + z^2 + 1)$, $(z^6 + z^4 + z^2 + z + 1)$ and an irreducible polynomial $m(z) = (z^8 + z^6 + z^5 + z^4 + 1)$, then their product mod m is :

$$(z^7 + z^2 + 1)(z^6 + z^4 + z^2 + z + 1) \quad \text{mod} \quad (z^8 + z^6 + z^5 + z^4 + 1)$$

$$= (z^{13} + z^{11} + z^9 + z^7 + z^3 + z + 1) \quad \text{mod} \quad (z^8 + z^6 + z^5 + z^4 + 1)$$

$$= (z^5 + z^4 + z^3 + z) \quad \text{mod} \quad (z^8 + z^6 + z^5 + z^4 + 1)$$

For two polynomials $a(v)$ and $b(v)$, it is said that $b(v)$ is divided by $a(v)$ that is $b(v)/a(v)$ when $r(v) = 0$. Mathematically,

$$a(v) = q(v)b(v) + r(v)$$

There are 30 irreducible polynomials [37] of degree 8 with coefficients in $GF(2^8)$. Irreducible polynomials are essential for polynomial multiplication in $GF(q^n)$ when it is performed over modulo m on an irreducible polynomial.

Definition 2.1.9.

The **Galois Field** or finite field, is a field whose order is a prime power q^n. It is represented by GF (q^n). The elements of the Galois Field GF (q^n) elements are defined as [38]:

$$GF(q^n) = (0, 1, 2, \ldots, q - 1) \cup (q, q + 1, q + 2, q + 3, \ldots, q + q - 1)$$

$$\cup\, (q^2, q^2 + 1, q^2 + 2, \ldots, q^2 + q + 1) \cup \ldots$$

$$\cup\, (q^{n-1}, q^{n-1} + 1, q^{n-1} + 2, \ldots, q^{n-1}q - 1)$$

where $n \in \mathbb{Z}^+$. The order of the field is determined by q^n, and the characteristic of the field is defined by q. Each factor has a polynomial degree of at most $n - 1$.

From a cryptographic standpoint, one concentrate on the following cases:

- $GF(q), n = 1$
- $GF(2^n), q = 2$

All polynomials of degree less than n with coefficients from $GF(q)$ are the elements of $GF(q^n)$.

The finite field $GF(2^8)$ has 256 elements and is used in the advanced encryption standard **(AES)** [5], which was developed by using a fixed irreducible polynomial

$$m(v) = v^8 + v^4 + v^3 + v + 1.$$

Each element of $GF(2^8)$ has degree less than 8. In $GF(2^8)$ the polynomial multiplication is reduced by modulo $m(v)$.

TABLE 2.1: Elements of Finite Field $GF(2^8)$

Decimal	**Polynomials**	**Binary**	**Hexadecimal**
0	0	00000000	00
1	1	00000001	01
2	v	00000010	02
3	$v + 1$	00000011	03
4	v^2	00000100	04
5	$v^2 + 1$	00000101	05
6	$v^2 + v$	00000110	06
7	$v^2 + v + 1$	00000111	07
8	v^3	00001000	08
9	$v^3 + 1$	00001001	09
10	$v^3 + v$	00001010	0A
.	.	.	.
.	.	.	.
.	.	.	.
255	$v^7 + v^6 + v^5 + v^4 + v^3 + v^2 + v + 1$	11111111	FF

The elements of $GF(2^8)$ are equivalently represented by an 8-bit binary numbers, 2-digit hexadecimal numbers, or a positive integers between 0 and 255 inclusively. The polynomial and binary representations of the finite field $GF(2^8)$ are given in Table 2.1.

2.1.1 Addition and Subtraction in GF

In GF, the procedure of addition is very simple. If $h_1(y)$ and $g_1(y)$ any two polynomials in $GF(p^n)$ and $f_1(y) = h_1(y) + g_1(y)$ with the coefficients of $h_1(y)$, $g_1(y)$ and $f_1(y)$ are $C = c_{n-1}, c_{n-2}, \ldots, c_1, c_0$, $D = d_{n-1}, d_{n-2}, \ldots, d_1, d_0$ and $E = e_{n-1}, e_{n-2}, \ldots, e_1, e_0$ respectively. Let c_k, d_k and e_k are the coefficients of $h_1(y)$, $h_1(y)$ and $g_1(y)$ respectively then

$$e_k = c_k + d_k \mod p \quad \text{where,} \quad k = 0,\ 1, \ldots,\ n-1$$

Likewise if $f_1(y) = h_1(y) - g_1(y)$ is given as:

$$c_k = a_k - b_k \mod p$$

where $k \in \{0, 1,\ 2, \ldots,\ n-1\}$
Remember that in $GF(2^n)$ addition can be done using “XOR” operation.

Example 2.1.10. Suppose two polynomials $f(y)$ and $g(y)$ in $GF(2^4)$. The sum $f(y) + g(y)$ under the mod $m(y)$ where $f(y) = y^3 + y^2 + y + 1$, $g(y) = y^2 + 1$ and $m(y) = y^4 + y^3 + y + 1$, then

$$f(y) + g(y) = (y^3 + y^2 + y + 1) + (y^2 + 1)$$
$$f(y) + g(y) = (y^3 + y) \mod (y^4 + y^3 + y + 1)$$

Alternatively, from binary number system

$$f(y) = y^3 + y^2 + y + 1 = (1111)_2$$
$$g(y) = y^2 + 1 = (0101)_2$$

$$f(y) + g(y) = 1111 \oplus 0101$$
$$f(y) + g(y) = 1010 = y^3 + y$$

2.1.2 Multiplication and Multiplicative Inverse

In Galois Field, multiplication involves more attention. Suppose $f_1(z)$ and $g_1(z)$ be any two polynomials in $GF(p^n)$ and suppose $m_1(z)$ be irreducible polynomial. The degree of product of $f_1(z)$ and $g_1(z)$ should be less than n in $GF(p^n)$. If $h_1(z)$ represent the product of $f_1(z)$ and $g_1(z)$ then

$$h_1(z) = f_1(z).g_1(z) \mod p.$$

Suppose $a_1(z)$ represent the multiplicative inverse of $f_1(z)$ then

$$f_1(z).a_1(z) = 1 \mod p.$$

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

Example 2.1.11. Consider $f_1(z) = z^2 + 1$ and $g_1(z) = z^2 + z + 1$ are irreducible polynomial with $m_1(z) = z^3 + z^2 + 1$ in $GF(2^3)$. Then we have

$$\begin{aligned} f_1(z).g_1(z) &= (z^2+1).(z^2+z+1) \mod (z^3+z^2+1) \\ &= z^4+z^3+z^2+z^2+z+1 \mod (z^3+z^2+1) \\ &= z^4+z^3+2z^2+z+1 \mod (z^3+z^2+1) \\ &= 1 \mod (z^3+z^2+1). \end{aligned}$$

Definition 2.1.12.

A **Primitive Polynomial** is an irreducible polynomial of degree n over $GF(q)$ that divides any $a(x) = x^m + 1$ where $m = q^n - 1$, but not any $a(x)$ divided with smaller m [39].

Example 2.1.13. In $GF(2^3)$, the polynomial $m(z) = z^3 + z + 1$ with a degree 3 is primitive polynomial. If there is a smallest positive integer $t = 7$ such that $m(z) = z^3 + z + 1$ divides $z^t - 1 = z^7 + 1$ as

$$z^7 + 1 = (z^3 + z + 1)(z^4 + z^2 + z + 1)$$

if k is the root of $z^3 + z + 1$, then $k^7 = 1$. Table 2.2 lists the powers of k in $GF(2^3)$ in the form of polynomials :

Table 2.2: Roots of primitive polynomial in $GF(2^3)$

Decimal	**Roots**	**Polynomials**
0	k^0	1
1	k^1	k
2	k^2	k^2
3	k^3	$k+1$
4	k^4	k^2+k
5	k^5	k^2+k+1
6	k^6	k^2+1
7	k^7	1

2.2 Boolean Function

A function $f : GF(2^n) \to GF(2)$ is said to be a Boolean function if it accepts the n tuples $\{r_1, r_2, \ldots, r_n\} \in GF(2^n)$ as input and produces only one of the two output bits $\{0, 1\} \in GF(2)$ [40]. A Boolean function describes how Boolean output values can be determined by using logical calculations. Boolean function has two main possibilities one is true (on/ ones) and the second is false (off/ zero). These features are also beneficial in the construction of digital computer circuits, integrated circuits, and electronic circuits. Boolean functions are also extensively used in cryptography to design substitution boxes (S-boxes).

Example 2.2.1. For a mapping $GF(2^n)$ to $GF(2)$, for $n = 2$

$$f(r_1, r_2, r_3) = r_1 \oplus r_2.r_3,$$

with input bits r_1, r_2 and r_3 in Table 2.3.

TABLE 2.3: Truth table of Boolean function

r_1	r_2	r_3	$r_2.r_3$	$f = r_1 \oplus r_2.r_3$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

A Boolean function $f : GF(2^n) \to GF(2)$ can be expressed in two distinct ways.

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

Truth Table (TT):

A tabular representation of the possible outcomes of a Boolean function, with the first two columns representing possible inputs and the last column displaying the result of executed function. Boolean function f can be represented as a binary vector of size $(2^n \times 1)$, with entries $f(r)$ indexed by the vectors $r \in GF(2^n)$.

Example 2.2.2. Consider the Boolean function $f = XOR$ of two variables r_1 and r_2 . The TT of $n = 2$ is shown in Table 2.4.

TABLE 2.4: Truth Table for Boolean function (XOR)

r_1	r_2	$r_1 \oplus r_2$
0	0	0
0	1	1
1	0	1
1	1	0

The Boolean function given as above can be written as $f = \lceil 0110 \rceil^T$.

Algebraic Normal Form (ANF):

ANF of boolean function is the most commonly used representation in cryptography. An ANF of a Boolean function $f : GF(2^n) \to GF(2)$ is a polynomial of the following form [41];

$$\begin{aligned} f(r_1, r_2, ...r_n) =& a_0 \oplus r_1 a_1 \oplus r_2 a_2 \oplus \ldots \oplus r_n a_n \oplus \\ & r_1 r_2 a_{1,2} \oplus \ldots \oplus r_{n-1} r_n a_{n-1,n} \oplus \ldots \\ & r_1 r_2 \ldots r_n a_{1,2,\ldots,n}, \end{aligned}$$

where $a_1, a_2 \ldots a_{1,2,\ldots,n} \in \{0,1\}^n$. Boolean functions are extensively used due to their outstanding properties. This ANF plays an essential role in the study of S-boxes and Boolean functions.

Example 2.2.3. Consider the two variables r_1 and r_2 and define the Boolean function "OR" on them. The ANF of 'OR' Boolean function is represented as:

$$g(r_1, r_2) = r_1 \oplus r_2 \oplus r_1 r_2.$$

It is written as:

Table 2.5: Truth Table of logical OR function

r_1	r_2	$r_1 \vee r_2$
0	0	0
0	1	1
1	0	1
1	1	1

2.2.1 Application of Boolean Function in S-boxes

Boolean functions are the important element of cryptography in constructing substitution boxes, therefore it is very critical that such crytographic characteristics are studied carefully. It is used for making difficult for adversaries to perform cryptanalysis [42]. The function S defined as:

$$S : GF(2^n) \longrightarrow GF(2^m),$$

takes n bits as input and outputs m bits. S-box can be defined as

$$S(v) = (f_1(v), f_2(v), \ldots, f_m(v),$$

Table 2.6: Truth table of $GF(2^4)$

u_1	u_2	u_3	u_4	$u_1u_2u_3$	$u_2u_3u_4$	$f(\alpha_i)$
1	1	1	1	1	1	1
1	1	1	0	1	0	0
1	1	0	1	0	0	1
1	1	0	0	0	0	1
1	0	1	1	0	0	1
1	0	1	0	0	0	1
1	0	0	1	0	0	1
1	0	0	0	0	0	1
0	1	1	1	0	1	1
0	1	1	0	0	0	0
0	1	0	1	0	0	0
0	1	0	0	0	0	0
0	0	1	1	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

where f_i $(i = 1, 2, \dots, m)$ are corresponding m-variables of Boolean functions. The Boolean functions are supposed to be the important components of S-boxes with the corresponding m vector.

Definition 2.2.4.

The sequence defined as $\{(-1)^{f(\alpha_0)}, (-1)^{f(\alpha_1))}, \dots, (-1)^{f(\alpha_{2^n-1})}\}$ is known as **Sequence of Boolean Function**. A balanced sequence has an equal number of ones and minus ones (actually zero), while an unbalanced sequence has an unequal number of ones and minus ones.

Example 2.2.5. Consider the Boolean function, which has four input bits u_1, u_2, u_3 and u_4.

$$f(u_1, u_2, u_3, u_4) = u_1u_2u_3 \oplus u_2u_3u_4 \oplus u_1$$

and$f(u_1, u_2, u_3, u_4)$ is calculated in Table 2.6.

The sequence of the Boolean function can be written as,

$$\begin{aligned}
&\{(-1)^{f(\alpha_0)},(-1)^{f(\alpha_1)},(-1)^{f(\alpha_2)},(-1)^{f(\alpha_3)},(-1)^{f(\alpha_4)},(-1)^{f(\alpha_5)},(-1)^{f(\alpha_6)},\\
&(-1)^{f(\alpha_7)},(-1)^{f(\alpha_8)},(-1)^{f(\alpha_9)},(-1)^{f(\alpha_{10})},(-1)^{f(\alpha_{11})},(-1)^{f(\alpha_{12})},(-1)^{f(\alpha_{13})},\\
&(-1)^{f(\alpha_{14})},(-1)^{f(\alpha_{15})}\}\\
=&\{(-1)^1,(-1)^0,(-1)^1,(-1)^1,(-1)^1,(-1)^1,(-1)^1,(-1)^1,(-1)^1,\\
&(-1)^0,(-1)^0,(-1)^0,(-1)^0,(-1)^0,(-1)^0,(-1)^0\}\\
=&\{-1,1,-1,-1,-1,-1,-1,-1,-1,1,1,1,1,1,1,1\}
\end{aligned}$$

Hence, the sequence of the Boolean function is balanced.

Definition 2.2.6.

A Boolean function $f : GF(2^n) \longrightarrow GF(2)$ is said to be **Linear** if and only if it can be expressed in the linear combination as

$$f(x_1, x_2, \ldots, x_n) = c_1x_1 \oplus c_2x_2 \oplus \ldots \oplus c_nx_n,$$

where $\oplus$ is the XOR operation [43] and the linear combination of two Boolean functions $f(x)$ and $g(x)$ is define as

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

There are exactly 2^n linear functions among the 2^{2^n} boolean functions of n variables.

Definition 2.2.7.

A mapping of Boolean function $f : GF(2^n) \longrightarrow GF(2)$ is said to be **Affine Function**, if the output of f has a linear combination along with constant [40], [44]. It can be described as follows:

$$f(x_1, x_2, \ldots, x_n) = c_1x_1 \oplus c_2x_2 \oplus \ldots \oplus c_nx_n \oplus c_0.$$

For an **Affine Cipher** a Boolean function over modulo d is used. It is a basic

Table 2.7: Conversion in affine cipher

A	0	N	13
B	1	O	14
C	2	P	15
D	3	Q	16
E	4	R	17
F	5	S	18
G	6	T	19
H	7	U	20
I	8	V	21
J	9	W	22
K	10	X	23
L	11	Y	24
M	12	Z	25

substitution cipher that is easy to crack due to the lack of security. This cipher performs addition and multiplication using the function given below

$$f(x) = (Ax \oplus C) \mod d.$$

The encryption key is made up of the letters A and C. The key will be added to an input, and then the modulus d is calculated. The following English alphabets are assigned to the numbers for encryption Table 2.7. For the explanation of Affine cipher example is given here.

Example 2.2.8. From the Table 2.7 to encrypt the message “LEO“ using the key $K = (5, 2) \mod 26$. The encryption function is

$$f(x) = (5x \oplus 2) \mod 26$$

then,

$$L = f(11) = 5 \mod 26 = F,$$

$$E = f(4) = 22 \mod 26 = W,$$

$$O = f(14) = 20 \mod 26 = U,$$

the obtained ciphertext is "FWU".

The decryption function is

$$x = [f(x) - 2] * 5^{-1} \mod 26,$$

$$5^{-1} = -5 \mod 26,$$

$$F = -5 * [5 - 2] \mod 26 = 11 = L,$$

$$W = -5 * [22 - 2] \mod 26 = 4 = E,$$

$$U = -5 * [20 - 2] \mod 26 = 14 = O,$$

hence the obtained plaintext is "LEO".

Definition 2.2.9.

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

Example 2.2.10. For a sequence $w(01100111)$ the number of zeroes is three and the number of ones is 5, The Hamming weight is define as

$$w(01100111) = H(01100111) = 5$$

the Hamming weight is 5

Definition 2.2.11.

Hamming Distance can be calculated using two Boolean function $h(v)$ and $k(v)$ as;

$$h(v), k(v) : GF(2^n) \longrightarrow GF(2),$$

is defined as [45]

$$d(h, k) = H(h(v) \oplus k(v)),$$

$$h(v) \oplus k(v) = h(v_0) \oplus k(v_0) \oplus h(v_1) \oplus k(v_1) \oplus ... \oplus h(v_{2^n-1}) \oplus k(v_{2^n-1}),$$

where

$$v = (v_0, v_1, \dots, v_{2^n-1}) \in GF(2^n).$$

It is take as the number of inputs where the functions how many bits need to be changed in truth table of h to get k [44].

Example 2.2.12. Consider the two Boolean function, with input bits r_1, r_2 and r_3

$$h(r) = r_1 r_2 r_3 \quad and \quad k(r) = r_1 \oplus r_2 \oplus r_3.$$

The Hamming distance of these Boolean functions is

$$d(h, k) = H(h(r) \oplus k(r))$$

$$d(h, k) = H(r_1 r_2 r_3 \oplus (r_1 \oplus r_2 \oplus r_3))$$

Table 2.8: Hamming distance of Boolean functions

r_1	r_2	r_3	$h = r_1 r_2 r_3$	$k = r_1 \oplus r_2 \oplus r_3$	$h(v) \oplus k(v)$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	1	1	0

The Hamming distance of $d(h, k) = 3$.

Example 2.2.13. For two simple Boolean functions $h(r)$ and $k(r)$ is defined as;

$$h(r) = 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1$$

$$k(r) = 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0,$$

then the Boolean function (XOR) between them is;

$$h(r) \oplus k(r) = 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1$$

$$H(h(r), k(r)) = 4$$

so, the Hamming distance $d(h, k) = 4$

Definition 2.2.14.

The correlation measurement between the Boolean function f and all the linear combinations is called the **Walsh Transform**. The Boolean function [46] of the Walsh transform is defined as:

$$WHT_g(b) = (-1)^{g(y) \oplus b.y} \quad \forall \quad b, y \in GF(2^n)$$

where g is the Boolean function, $b.y$ represents the dot product of b and y, and $\oplus$ is the (XOR) of g and $b.y$.

The Hamming distance is also used to measure the similarity of Boolean functions. Hamming distance is determine by counting the bits in the truth table of Boolean functions which are distinct, while the unintended distance is the sum by which this distance varies from expectation.

The predicted distanced of boolean function with affine function is define as:

$$ED = \frac{2^n}{2}$$

The discrepancy between these two values is referred to as unexpected distance. It is calculated as

$$\text{unexpected distance} = \text{Hamming distance} - ED$$

Example 2.2.15. Consider all affine and Boolean functions

$$g : GF(2^3) \longrightarrow GF(2),$$

then a TT for affine function is given in the Table 2.9.

Consider a Boolean function $g = [1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$. The predicted

Table 2.9: Calculation of hamming distances and unexpected distances of $g = [1\,0\,0\,1\,1\,0\,0\,1]$

Affine function	**Truth table** $=\ h$	$g \oplus h$	**Hamming distance**	**Expected distance**	**Unexpected distance**
1	11111111	01100110	4	4	0
y_0	01010101	11001100	4	4	0
y_1	00110011	10101010	4	4	0
y_2	00001111	10010110	4	4	0
$y_0 + y_1$	01100110	11111111	8	4	4
$y_0 + y_2$	01011010	11000011	4	4	0
$y_1 + y_2$	00111100	10100101	4	4	0
$y_0 + y_1 + y_2$	01101001	11110000	4	4	0

distances can be determined with respect to all affine functions in truth table by using the formula

$$ED = \frac{2^3}{2} = \frac{8}{2} = 4.$$

The discreparity between Hamming distance and ED to measure unexpected distances (g). The following table shows the calculated values. The Walsh transform of g is the maximum absolute value of all unexpected distances i.e., $WHT_g = 4$.

2.3 Substition Boxes

Consider a function $G : \mathbb{V}_2^n \longrightarrow \mathbb{V}_2^m$ for some positive integers n and m where $\mathbb{V}_2$ is the finite field with two elements. Such functions G with given Boolean functions $g_1, g_2, \ldots, g_m$ are defined as

$$G(v) = (g_1(v), g_2(v), \ldots, g_m(v))$$

at every $v \in \mathbb{V}_2^n$, called the coordinate functions of F. Such $(n \times m)$ functions are called S-boxes.

Thus a function $S : GF(2^n) \longrightarrow GF(2^m)$ which takes n-bits as input to produce m-bits as output is called an $(n \times m)$ S-box, defined as

$$v = S(u) = (g_1(u), g_2(u), \ldots, g_m(u)) \in GF(2^m), \quad \forall \quad u \in GF(2_n)$$

Example 2.3.1. Consider a 4 × 4 S-box, which takes four input bits, and returns four output bits. The elements of Galois field $GF(2^4)$ is given in the first column, while the elements of 4 × 4 S-box in vector from is shown in second column of Table 2.10.

Table 2.10: Boolean function in 4 × 4 S-box

$GF(2^4)$	$S-box$	g_1	g_2	g_3	g_4
0	9	1	0	0	1
1	13	1	0	1	1
2	10	0	1	0	1
3	15	1	1	1	1
4	11	1	1	0	1
5	14	0	1	1	1
6	7	1	1	1	0
7	3	1	1	0	0
8	12	0	0	1	1
9	8	0	0	0	1
10	6	0	1	1	0
11	2	0	1	0	0
12	4	0	0	1	0
13	1	1	0	0	0
14	0	0	0	0	0
15	5	1	0	1	0

The entries of an S-box are in binary form in Table 2.10, where each column represents a Boolean function g_i for $1 \leq i \leq 4$ of S-box. The properties of an S-box is determined by the Boolean functions that were earlier used to create it.

2.3.1 Essential Characteristics of S-box

S-boxes are made up of highly nonlinear Boolean functions. Without them, attackers might easily hack the system. There are two primary reason that tell the significant and the necessary characteristics of S-box design.

1. **Designing new Ciphers**

   The S-box design is the most important part for the design of a new cipher schemes as it is the sole nonlinear element of the system. This is essential component which depends on cipher strength. As cryptography advances, hackers also devise new attacking methods, so the design of S-box should be secured in advance to ensure the security of ciphers .

2. **S-box Development Design for private use**

   Trap-door are being used by adversaries to generate keys for certain ciphers such as AES [5], so every organisation, especially governments, wants a secure system that is only applicable to their organisation with an extra security layer, which is possible only if they design their own individual S-boxes for their specific system.

2.4 Classification of S-boxes

There are three sub classifications of S-boxes.

1. **Straight S-box**

   A straight S-box takes input and gives output of the same size. The well known AES [5] is an example of such S-box. This is the easiest and most common form of S-box design.

2. **Compressed S-box**

   S-box which takes more input bits/bytes but returns less output bits/bytes. DES is a good example of this type of S-box in which each block takes in 6 bits and outputs 4 bits block.

3. **Expanded S-box**

   This S-box takes in less input bits and gives back more bits. One can construct such S-box by duplicating some of the output or input bits.

2.4.1 Properties of Strong S-box

Important properties of S-box are given below.

- S-box is balanced.
- S-box has high nonlinearity.
- All linear combinations of S-box are bent.
- All entries in the XOR table are 0 or 1.
- S-box satisfies bit independence criteria.
- S-box satisfies strict avalanche criteria.

2.4.2 Cryptographic Properties of Strong S-box

S-box satisfy the necessary properties for a cryptographically strong S-box. Since substitution boxes are an important part of many cryptosystem.

1. **Balanced**

   A mapping of Boolean function $S : GF(2^n) \longrightarrow GF(2)$ is said to be balanced if zero/one has equal in number in the truth table.

   Example 2.4.1. A comparison of balanced and unbalanced functions is provided in Table 2.11.

Table 2.11: Truth table of XOR and AND functions

r_1	r_2	r_3	$S_1 = r_1 \oplus r_2 \oplus r_3$	$S_2 = r_1 \cdot r_2 \cdot r_3$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Consider the Boolean functions XOR and AND, which are defined as follows:

$$S_1 = \oplus : GF(2^3) \longrightarrow GF(2),$$

$$S_2 = \cdot : GF(2^3) \longrightarrow GF(2).$$

The following truth table is defined for three variables r_1, r_2 and r_3. Fourth column shows the "XOR" function which has equal number of ones and zeros so, it is balanced while the fifth column represents the "AND" function which is not balanced as it has more zero's then onces.

2. **Bijective:**

   A mapping of Boolean function $S : GF(2^n) \longrightarrow GF(2)$ is called bijective iff all linear combinations of columns are balanced. A method [21] is introduced to verify the bijective property of a $(n \times n)$ S-box, which states that "The bijective property is fulfilled if for the Boolean functions $f_i(\text{for}1 \leq i \leq n)$ of S-box following condition holds.

   $$Hwt(\sum_{i=1}^{n} c_i f_i) = 2^{n-1} \tag{2.1}$$

   Where $c_i \in \{0, 1\}$ for $(c_1, c_2, ..., c_n) \neq (0, 0, ..., 0)$ and Hwt is the Hamming weight" [47].
   Equation (2.1) ensures that all Boolean functions f_i and their combinations are balanced.

Example 2.4.2. Consider the (4×4) S-box and demonstrate that 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, $S(0000) = 1\,1\,1\,1, S(0001) = 1\,0\,1\,1, ..., S(1111) = 1\,0\,0\,0$. 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.

3. **Nonlinearity**

Nonlinearity of a Boolean function (NL(g)) [48] $g(v) : GF(2^n) \longrightarrow GF(2)$ is defined as the minimum Hamming distance of g from any of its n-variable affine functions.

$$\text{NL}(g) = min\, d(g, h).$$

Example 2.4.3. Suppose r_1 and r_2 are input bits and $g(r)$ is a Boolean function: $g(r) = r_1 \oplus r_2$

Table 2.12: Truth table

r_1	r_2	$g(r)$	0	r_1	r_2	$r_1 \oplus r_2$	$g(r) \oplus 0$	$g(r) \oplus r_1$	$g(r) \oplus r_2$	$g(r) \oplus (r_1 \oplus r_2)$
0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	1	1	1	1	0	0
1	0	1	0	1	0	1	1	0	1	0
1	1	1	0	1	1	0	1	0	0	1

Where $0, r_1, r_2, r_1 \oplus r_2$ are the possible linear function of r_1 and r_2 and $d_1(g(r), 0) = 3, d_2(g(r), r_1) = 1, d_3(g(r), r_2) = 1, d_4(g(r), r_1 \oplus r_2) = 1$

So,

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

4. **Bent Function**

S-Boxes must be made up of Boolean functions that are strongly nonlinear. Different Boolean functions with great nonlinearity value can be define, but bent functions are a special kind of Boolean function with the highest nonlinearity.

Bent functions are defined by walsh transfrom as:

$$f(s) = \frac{1}{\sqrt{2}} \sum_{u \in GF(2^n)} (-1)^{f(u) \oplus s \cdot u},$$

where $f(u)$ is the Boolean function and $s.u$ is the dot product and $u \in GF(2)$. Thus a Boolean function f which contain to its maximum nonlinearity is called bent function [42].

Example 2.4.4. By maximum nonlinearity criterion, it is known that in $GF(2^2)$, the function with nonlinearity 1 is bent.

$$NL(f) = 2^n - 2^{\frac{n}{2}-1}$$

$$2^{2-1} - 2^{\frac{2}{2}-1} = 2 - 1 = 1$$

Clearly, bent functions are not linear or affine, but they are a form of Boolean functions with the highest Strict Avalanche Criterion (SAC) [49] and Bit Independence Criterion (BIC) [47].

5. **Dynamic Distance**

Dynamic Distance (DD) of order j for a Boolean function [50].

$$f : GF(2^n) \longrightarrow GF(2)$$

is defined as

$$DD_j(f) = \max_{1 \leq wt(d) \leq j} \frac{1}{2} \left| 2^{n-1} - \sum_{x \in GF(2^n)} f(x) \oplus d(f \oplus x) \right|$$

where f is the Boolean function, and $d(f \oplus x)$ shows the Hamming distance between them $d \in \{0,1\}^n$. It provides a measure for other dynamic properties such as SAC which will be satisfied if DD has small integral value and closer to zero.

Example 2.4.5. For calculating Dynamic distances DD1, we use a special matrix $d \in GF(2^n)$ with Hamming weight 1 for each entry.

Take the example of S-box present in Table 2.10 in $GF(2^4) \longrightarrow GF(2^4)$, and matrix d define as:

$$\text{d} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The DD of Boolean functions that have been calculated are $[4 \quad 4 \quad 4 \quad 2]$ S-box mapping $GF(2^8) \longrightarrow GF(2^8)$ of AES, then d is expressed as:

$$d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

For $S(u) = v$, first calculate $u \oplus d$ for each entry of all Boolean functions with each entry of this matrix d, then calculate

$$f(u) \oplus f(u \oplus d)$$

Finally, it can help to determine the DD for S-box.

6. **Correlation Immunity**

The correlation immunity [51] (CI) of a Boolean function denotes the scale of the independence between the linear combination of input bits and output bits. The relationship between the Walsh transform and the Hamming weight of its inputs can be used to determine its functional order. When $WHT_f(\beta) = 0$, and $1 \leqslant H(w) \leqslant p$, a Boolean function is said to have correlation immunity.

7. **Absolute Indicator and Sum of Square Indicator**

With respect to a point c , the derivative of a Boolean function mapping $f : GF(2^n) \longrightarrow GF(2)$ is defined as:

$$D_f(c) = f(x) \oplus f(x \oplus c).$$

Auto-correlation (AC) of a boolean function f can be defined on all $b \in GF(2^n)$ using the above defined derivative:

$$\triangle = \Sigma(-1)^{f(x) \oplus f(x \oplus c)} \quad \text{where} \quad x \in GF(2^n)$$

The absolute Boolean function f indicator is defined as the maximum absolute AC value excluding the origin that can be expressed as [51].

$$\triangle_f = \max_{c \in GF(2^n), c \neq 0} | \triangle_f \, (b)|$$

The Sum of square indicator [51] of Boolean function f is also derived from AC and can be expressed as:

$$\sigma_f = \sum_{c \in GF(2^n)} (\triangle_f(c))^2$$

8. **Algebric Immunity**

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

$$(f + 1)h = 0 \quad or \quad f \cdot h = 0$$

where a function h for which $f \cdot h = 0$ is called annihilator of f [48].

Example 2.4.6. Take the following two Boolean functions with input bits v_1 and v_2.

$$f(v) = v_1 + v_1 v_2 \quad \text{and} \quad h(v) = v_2$$

for the algebric immunity;

TABLE 2.13: Truth table of Algebric immunity

v_1	v_2	$f(v)$	$f \cdot h$	$(f+1)$	$(f+1) \cdot h$
0	0	0	0	1	0
0	1	0	0	1	1
1	0	1	0	0	0
1	1	0	0	1	1

According to the Table 2.13, $f \cdot h = 0$ and $(f+1) \cdot h = 0$.

9. **Algebraic Degree**

An algebraic degree is linked with the nonlinearity measures [48]. “For a Boolean function $f : GF(2^n) \longrightarrow GF(2)$, it is defined as the number of variables in highest order term with non-zero coefficients and can be expressed as

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

Higher algebraic degree is considered better than the lower.”

10. **Fixed and Opposite Fixed Points**

Take an S-box $S : GF(2^n) \longrightarrow GF(2^m)$ and for $u \in GF(2^n)$. A point is called fixed point of S-box if

$$S(u) = u$$

A point is said to be an opposite fixed point of an S-box if

$$S(u) = u'$$

and u' is the compliment of u.

S-boxes without fixed and opposite fixed points are preferable to those with fixed and opposite fixed points.

Example 2.4.7. Take (2×2) S-box with two Boolean functions.

There is a ‘2’ element that is a fixed point of S-box as shown in Table 2.14.

Example 2.4.8. A (2×2) S-box with two Boolean functions.

TABLE 2.14: S-box of fixed point

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

TABLE 2.15: S-box of opposite fixed points

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

and there is '1' opposite fixed point of S-box as shown in Table 2.15.

Any S-box that does not have fixed and opposite fixed points is thought to be good against differential cryptanalysis attack in comparison to those who has fixed point and opposite fixed points.

2.5 Software Tools in the Analysis of S-box

To check the properties of S-box, some different tools are available such as Boolfun Package in R, VBF, SageMath, SET and SAMT are describe below.

1. **Boolfun Package in R**

   R works on different windows like UNIX and Mac OS platforms but the standard version of R do not support the Boolean functions. It is possible to load a package name **Boolfun** [52], which gives functionality realted to cryptographic analysis of Boolean functions. R is the free open source Mathematical software used for computing statistics.

2. **VBF**

   VBF is abbreviation of Vector Boolean Function library. This tool is used for the analysis of cryptographic properties of S-boxes. It is introduced by Alverez - Cubero and Zuffiria [53].

3. **SageMath**

   SageMath library [48] is a free open source Mathematics tool which contains a module called Boolean functions and an S-box. From this tool, the algebraic properties and various cryptographic properties related to linear approximation matrix and difference distribution table for S-boxes and Boolean functions can be checked.

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

   Evaluation of cryptographic properties of Boolean function and S-boxes is presented by Picek [54] and his team. It is also a free open source mathematics tool which is easy to use. It works in VS (visual studio).

5. **SAMT**

   SAMT [55] is another tool for the evaluation of cryptographic properties of Boolean function and S-box. It works on MATLAB.

Chapter 3

Chaos Theory

Chaos theory is the branch of mathematics concered with the behaviour of dynamic systems. The term chaos refers to the science of unexpected events. It involves dynamic systems with nonlinear and unpredictable behavior. Chaos theory provides a method for dealing with unpredictable behavior such as turbulence, weather, stock market and so on. Chaos theory, properties (characteristic) of chaotic system, Lyapunov exponents and Bifurcation diagram are discussed in Section 3.1. Section 3.2 is about the chaos based cryptography. Section 3.3 is based on the chaotic map. Some chaotic maps are also explained in this section. The role of approximate entropy for the chaotic behaviour analysis is given in Section 3.4.

3.1 Chaos Theory

Chaos is derived from a Greek word 'Xaos', with the meaning as a state without order or predictability. Chaos theory was developed in the 1970 [56] and used in a scientific fields [57] such as physics, mathematics, engineering, and biology etc. The term chaos refers to the science of unexpected events. It study of the temporal development of the non-linear system is known as non-linear dynamics. A basic study of the mathematics and the random behaviour of the chaotic system gives a way to comprehend the importance of this theory in relation to the complexity of social processes. In 1963 Edward Lorenz [58] investigated chaos theory and presented a basic mathematical model to predict a weather. It is the first numerical method for identifying chaos in a dynamical system. A dynamic system is one in which the function is reliant on a time-dependent point in a geometrical space, such as a moving pendulum or water flowing through a pipe. In addition, chaotic system is highly sensitive to the initial parameters. Initial condition sensitivity means that every aspect of the chaotic system is very near to the other trajectory points. As a result, an insignificant change in the initial conditions can lead to a big variation in behaviour. The Butterfly effect [59] is defined as the sensitivity to the initial condition. In other words, chaos theory is the science of unexpected events.

3.1.1 Characteristics of Chaotic System

The phenomenon 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.. Chaotic systems include the following characteristics:

- **Apparently random but entirely deterministic behaviour**

  The chaotic system behaviour appears random but is, in fact, totally predictable. Therefore, if chaotic system iterate with the same initial conditions it gives the same output value set. Chaotic systems are also dynamic systems defined by differential equations (or iterative mappings) where every state rely on the previous state

  $$\frac{dy_j}{dt} = F_j(y_1, y_2, ..., y_n) \quad j = 1, 2, ..., n$$

  y_j are states depending on time.

- **Sensitivity dependence on the initial conditions**

  Chaotic systems evolve completely different when a small change occur in the original state throughout time.

- **Unpredictable**

  In chaotic system, one can understand the initial state of the chaotic system it doesn't mean that anyone can predict the next state of the system. In other words, for long term prediction of the future states is hard to obtain.

3.1.2 Lyapunov Exponents (LE)

The term 'Lyapunov Exponent' (LE) [60] has been widely used in the study of dynamical systems. The degree of divergence between two close trajectories of a dynamical system is described by LE. A positive LE indicates that, regardless of how close the two trajectories are, their divergence increases with each iteration, eventually causing them to be completely different. As a result, the LE of a chaotic dynamical system is positive. In a multidimensional dynamical system, there may be more than one LE. If it has more than one positive LE, its close trajectories exponentially diverge in several dimensions. This phenomenon is known as hyperchaotic behavior. A dynamical system with hyperchaotic behaviour performs extremely well in terms of chaos and its outputs are difficult to predict. LE can be defined as:

$$\lambda = \lim_{n \longrightarrow \infty} \frac{1}{n} \sum_{n=1}^{\infty} \ln |g^{'}(y_i)|$$

where $g(y_i)$ is the function of chaotic system. The Lyapunov exponent has three dynamics cases.

1. If the orbit attracts toward a stable point its mean that all lyapunov exponents are less than zero.

2. When the LE is zero, the system is neutrally stable. such system are conservative and in a steady state mode. They exhibit Lyapunove stability.

3. If the system is chaotic, then all the Lyapunov exponents are greater than zero.

3.1.3 Bifurcation Diagram

When the control parameter is altered, a bifurcation happens, which is a period-doubling, or a change from an M-point attractor to a 2M-point attractor. A bifurcation diagram is a graphic representation of the sequence of period-doublings that occurs as control parameter (δ) increases. The bifurcation diagram of any chaotic map is illustrated in Figure 3.1, with δ on the horizontal axis. Before plotting sequential values of y over a few hundred iterations, the system is allowed to settle down for each value of δ.

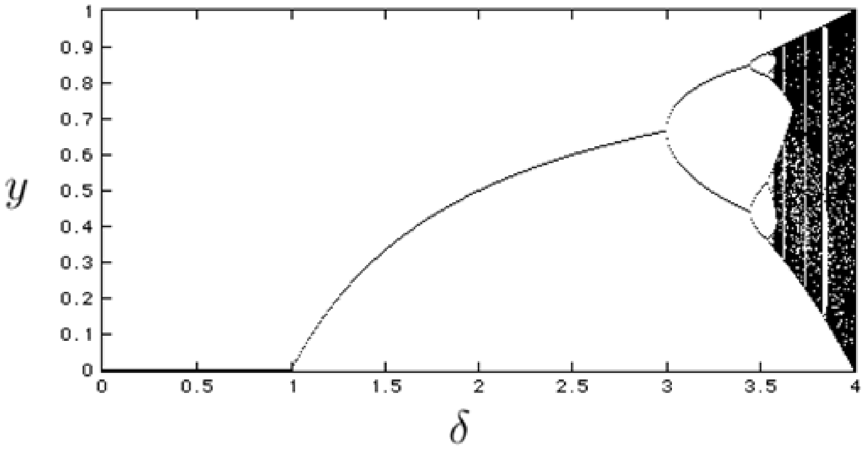

Figure 3.1: Bifurcation diagram

It is clear from the Figure 3.1, when $\delta \leq 1$ every point are plotted at zero. So for $\delta \leq 1$ there is only one point attractor. Now when $\delta \in (1, \ 3)$, there is still one point attractors but the attracted value of y increases as δ increases. Bifurcation occurs at $\delta = 3$, 3.45, 3.54, 3.564, 3.569, (approximately) etc. Until just beyond 3.57, where the system become chaotic. However the system is not chaotic for all value of $\delta \in [3.57, \ 4]$, even there are some point in which it show three point attractors.

3.2 Chaos-based Cryptography

It is difficult to stop an unauthorised user monitoring in the communication network, used for satellite, mobile phones, and the internet. For the secrete communication over the public networks, certain cryptographic technique is used. In the content of security, videos and images are important in many applications such as medical imaging, industrial imaging, military imaging systems, and private multimedia messages.

Chaos theory plays an essential role to enhance the security of cryptosystem. Chaotic systems and cryptographic technique methods have similar characteristics such as sensitivity to changes in the parameters, unpredictability over long durations, and random-like behaviour [61–63]. However, relationship between cryptographic methods and chaos based cryptography is essential [64]. In [64–67] the differences between cryptographic methods and chaotic systems are shown in Table 3.1.

Table 3.1: Comparison between cryptographic algorithms and chaotic systems

Properties of chaotic systems	**Properties of cryptographic scheme**
Parameters (Real)	Key (Boolean)
Sensitive to change the initial parameters	Diffusion
Ergodicity	Confusion
Iterations	Rounds
Deterministic dynamical	Deterministic Pseudorandom
Using set of real numbers	Finite set of integers
Structure complexity	Algorithm complexity

The difference between cryptographic system and chaotic is that, cryptographic algorithms based on finite set of integers and chaotic system is defined on floating points numbers [66]. The parameters of chaotic maps are valuable if they are real numbers that may be utilised as encryption and decryption keys in cryptography methods. In cryptographic scheme, if one bit change in plaintext/key change the ciphertext throughout. In other side, iterations of chaotic systems are utilized to expand the initial area in the chaotic systems. The aim of cryptographic properties (confusion/diffusion) is to complicate the relationship between secret message and the key and also between the plain message. The chaotic property (sensitive to initial condition) are close to the diffusion property of the encryption system of cryptography. The ergodicity property shows that it is very hard to predict the behavior of the system on the basis of initial conditions, similar to confusion property of cryptography.

3.3 Chaotic Maps

According to Alligood et al. [68], a chaotic map is a domain and a range function in the same space, and the starting-point of the trajectory is the initial condition. Chaotic dynamics have a unique features that can be seen clearly by imagining the system starting twice under different initial conditions. Chaos theory tries to explain the results of a system that is sensitive, complicated, and unexpected. Chaotic dynamical systems enhance the communication security with higher dimensions and more than one positive Lyapunov exponent [69]. Lyapunov exponent is help to select the initial parameters of chaotic maps that fall in chaotic areas. In the study of dynamical systems, a chaotic system is the system that shows some chaotic behaviour. Some chaotic maps are explain in this sections: Logistic map, tent map and after that logistic and tent map is combine together that is called tent-logistic map (compound map).

3.3.1 Logistic Map

The logistic map is a quadratic mapping (or recurrence relation). it is simple non-linear dynamical map. The map was made popular in a study by biologist May [70], in part as a discrete-time demographic model comparable to Pierre Franois Verhulst's logistic equation. One of the most well-known 1D chaotic maps is the logistic map. It has simple mathematical structure but complicated chaotic behaviour. Logistic map is defined as:

$$y_{n+1} = \delta \, y_n \, (1 - y_n) \tag{3.1}$$

where δ is a system parameter with a value between $\delta \in [0, 4]$. It is characterised, when a small change occurs in the parameter δ it brings a change in the qualitative behavior. Where y_n has a value between 0 and 1 that indicates the current population to the greatest population conceivable. The goal of this nonlinear difference equation is represent two effects.

- When the population size is modest, reproduction causes the population to grow at a rate proportionate to the existing population.
- Starvation (density-dependent mortality), in which the growth rate is proportional to the value derived by subtracting the theoretical “carrying capacity” of the environment from the present population.

The chaotic behavior of logistic map exhibits for $\delta \in [3.57, 4]$. Figure 3.2 shows the Lyapunov exponent of the logistic map. Figure 3.3 shows the bifurcation diagram of logistic map, the horizontal axis depicts the values of the parameter δ while the vertical axis shows the values of y.

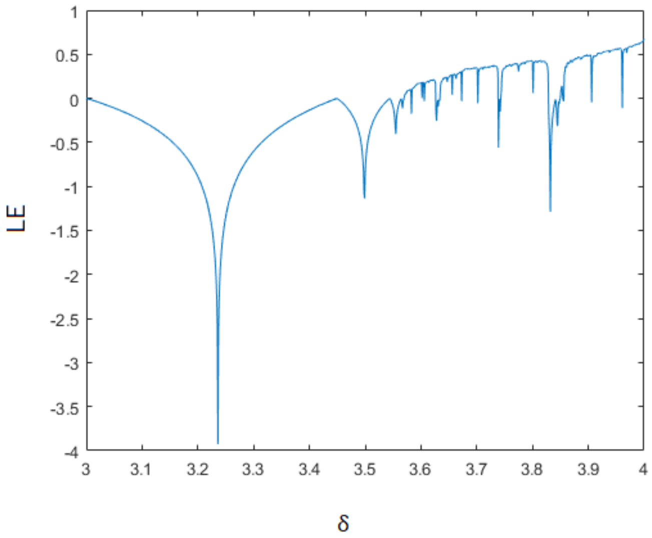

FIGURE 3.2: Lyapunove exponent of logistic map

In Figure 3.4, the horizontal axies shows the values of y_n and vertical axis shows its frequency distribution.

Drawbacks:

In the logistic map, there are three flaws. One is that the system’s chaotic range is restricted to $\delta \in [3.57, 4]$. In side the given range [3.57, 4], few factors give rise a change in the logistic map to behave in predictable manner. The non-unifrom dispersal of the state value in the given range [0, 1], is another disadvantage. When $\delta = 3.9$, the logistic map shows aperiodic behaviour, according to the authors of [71]. However, instead of utilising the range of $3.57 \leq \delta \leq 4$, the result of this

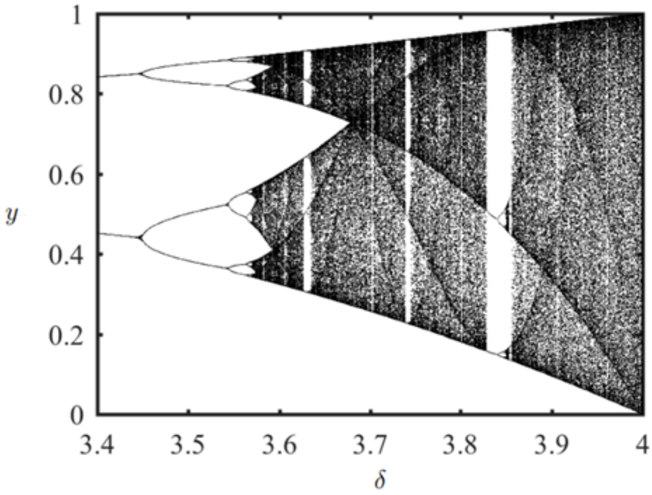

FIGURE 3.3: Bifurcation diagram of logistic map

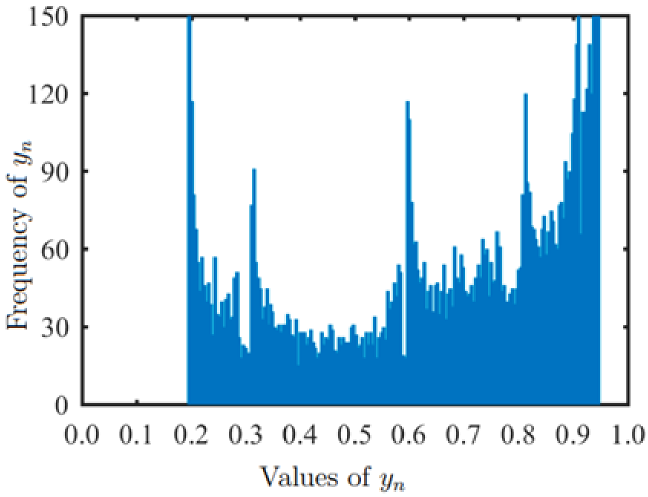

FIGURE 3.4: State distribution of δ

rang is to limited the key space. The logistic map's application value is reduced as a result of these flaws.

3.3.2 Tent Chaotic Map

A Tent map is an iterated function of a dynamical system governed by equation (3.2) and exhibit chaotic behaviour. It has a similar shape to the logistic map. It is another discrete 1D chaotic map. This chaotic map has tent-like shape in its bifurcation diagram that is why its name is tent. The mathematical model of tent map is [33].

$$y_{n+1} = \begin{cases} \dfrac{\delta}{2}\, y_n & y_n < 0.5 \\ \dfrac{\delta}{2}\,(1 - y_n) & y_n \geq 0.5 \end{cases}, \tag{3.2}$$

where δ is a system parameter in the [0, 4] range. Figure 3.5 shows the bifurcation diagram of tent map and Figure 3.6 depicts the state distribution of δ.

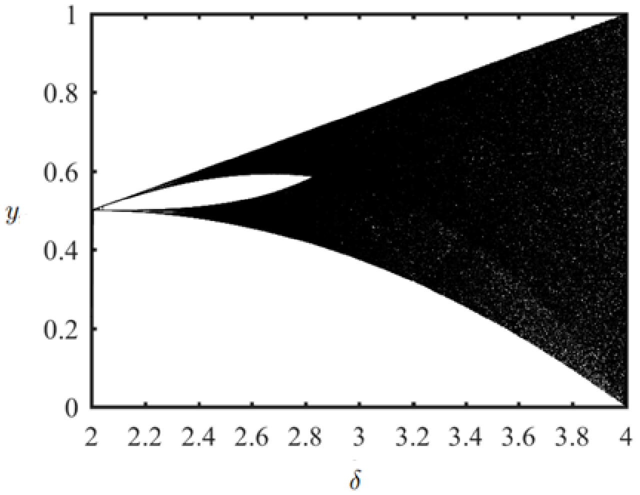

FIGURE 3.5: Bifurcation diagram of tent map

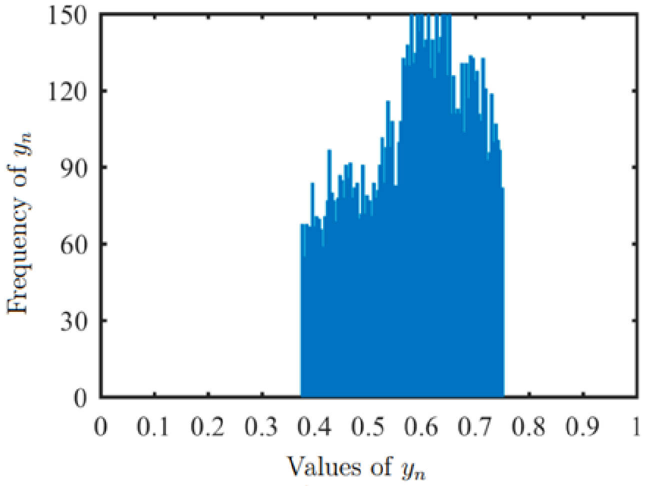

FIGURE 3.6: State distribution of δ

Drawbacks

In tent map, three defects have been discovered. The very first defect is the chaotic range, which is specified in the system. Another defect is the irregular dispersal of the state value in the given range [0, 1]. It is also has a limited key space.

3.3.3 Tent-Logistic System

The new compound chaotic system citelu2019novel is obtained by combine the logistic map (3.1) and tent maps (3.2), which is said to be **tent-logistic system (TLS)**. It is obtained to overcome the problem of tent and logistic maps. It has the following mathematical model:

$$y_{n+1} = \begin{cases} \frac{4}{9}(9-\delta)\, y_n\, (1-y_n) + \frac{2\delta}{9}\, y_n, & y_n < 0.5 \\ \frac{4}{9}(9-\delta)\, y_n\, (1-y_n) + \frac{2\delta}{9}\, (1-y_n), & y_n \geq 0.5 \end{cases} \tag{3.3}$$

where δ is a system parameter with a value between 0 and 9. When $\delta = 0$ and $\delta = 9$, (3.3) degenerates into the chaotic logistic map and the chaotic tent map, respectively. As a result, the finest logistic and tent maps may both be considered special instances of (3.3).

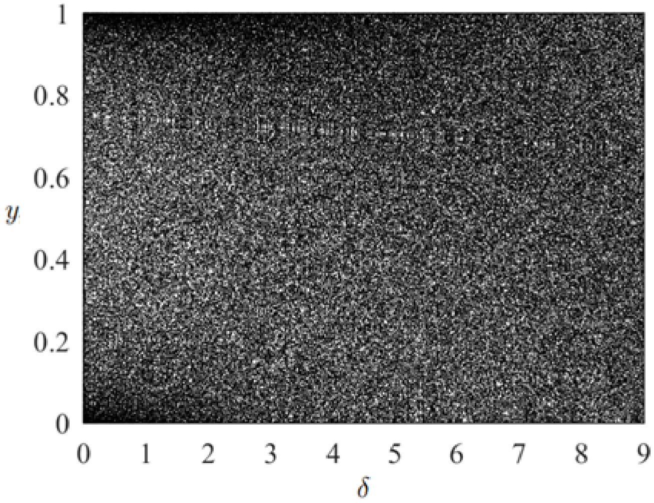

Figure 3.7: Bifurcation diagram of tent-logistic map

Figure 3.7 depicts the TLS bifurcation diagram, which shows the chaotic range covered the range [0, 9] and Figure 3.8 shows the state distribution diagram. Its output sequences are evenly dispersed throughout the range of [0, 1] (see Figure

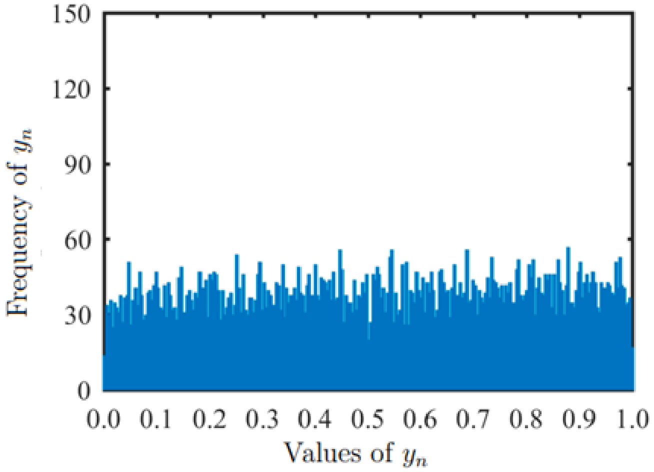

FIGURE 3.8: State distribution of δ

3.8).

On comparing logistic/tent maps with Tent logistic map. The tent-logistic method offers two benefits, first the chaotic range is greater than the logistic and tent maps. If the system parameter δ is used as the secret key, the keyspace of a cryptosystem employing the new compound system will be considerably increased. Second, the output sequence by using the tent-logistic system is uniformly spread throughout the whole value range between 0 and 1. On the basis of the benefits, the TLS is more suitable for cryptography applications.

Proposition 3.3.1. For $\delta \in [0,\ 9]$, system (3.3) in all its range is a map

$$g : y_m \in (0,\ 1) \longrightarrow y_{m+1} \in (0,\ 1)$$

1. Equation (3.3) degenerates to the chaotic logistic map when $\delta = 0$.

$$g_L : y_m \in (0, 1) \longrightarrow y_{m+1} \in (0, 1)$$

2. Equation (3.3) degenerates to the chaotic tent map when $\delta = 9$.

$$g_T : y_m \in (0, 1) \longrightarrow y_{m+1} \in (0, 1)$$

3. When $0 < \delta < 9$ and $y_n < 0.5$, then

$$g_1[y_n < 0.5] < g_1[0.5] = 1.$$

4. When $0 < \delta < 9$ and $y_n \geq 0.5$, then

$$g_2[y_n > 0.5] < g_2[0.5] = 1.$$

Proof. (1)

Set $\delta = 0$ in Equation (3.3)

$$y_{n+1} = \begin{cases} \frac{4}{9}(9-0)\ y_n\ (1-y_n) + \frac{2(0)}{9}\ y_n, & y_n < 0.5 \\ \frac{4}{9}(9-0)\ y_n\ (1-y_n) + \frac{2(0)}{9}\ (1-y_n), & y_n \geq 0.5 \end{cases}$$

$$y_{n+1} = \begin{cases} \frac{4}{9}(9)\ y_n\ (1-y_n) + \frac{0}{9}\ y_n, & y_n < 0.5 \\ \frac{4}{9}(9)\ y_n\ (1-y_n) + \frac{0}{9}\ (1-y_n), & y_n \geq 0.5 \end{cases}$$

$$y_{n+1} = \begin{cases} (4)\ y_n\ (1-y_n) & y_n < 0.5 \\ (4)\ y_n\ (1-y_n) & y_n \geq 0.5 \end{cases}$$

Now assume that $\delta = 4$, then the above equation becomes the logistic system

$$y_{n+1} = \begin{cases} f_1[y_n] = \delta\ y_n\ (1-y_n) & y_n < 0.5 \\ f_2[y_n] = \delta\ y_n\ (1-y_n) & y_n \geq 0.5 \end{cases}$$

both interval $y_n < 0.5$ and $y_n \geq 0.5$, the Equation (3.3) gives the logistic map. □

Proof. (2)

Set $\delta = 9$ in Equation (3.3)

$$y_{n+1} = \begin{cases} \frac{4}{9}(9-9)\ y_n\ (1-y_n) + \frac{2(9)}{9}\ y_n, & y_n < 0.5 \\ \frac{4}{9}(9-9)\ y_n\ (1-y_n) + \frac{2(9)}{9}\ (1-y_n), & y_n \geq 0.5 \end{cases}$$

$$y_{n+1} = \begin{cases} \frac{4}{9}(0)\ y_n\ (1-y_n) + \frac{18}{9}\ y_n, & y_n < 0.5 \\ \frac{4}{9}(0)\ y_n\ (1-y_n) + \frac{18}{9}\ (1-y_n), & y_n \geq 0.5 \end{cases}$$

$$y_{n+1} = \begin{cases} 2\ y_n y_n < 0.5 \\ \\ (1-y_n) y_n \geq 0.5 \end{cases}$$

here assume that $\delta = 2$, its gives the tent map

$$y_{n+1} = \begin{cases} f_1[y_n] = \delta\ y_n & y_n < 0.5 \\ \\ f_2[y_n] = \delta\ (1-y_n) & y_n \geq 0.5 \end{cases}$$

Proof. (3) □

When $0 < \delta < 9$ and $y_n < 0.5$, the first part of system (3.3) is

$$g_1[y_n] = \frac{4}{9}(9-\delta)\ y_n\ (1-y_n) + \frac{2\delta}{9}\ y_n$$
$$g_1[y_n] = \frac{1}{9}\{(36-4\delta)\ (y_n - y^2(n)) + 2\delta\ y_n\}$$
$$g_1[y_n] = \frac{1}{9}\{36y_n - 36y^2(n) - 4\delta y_n + 4\delta y^2(n) + 2\delta y_n\}$$
$$g_1^{'}[y_n] = \frac{1}{9}\{36 - 72y_n - 2\delta + 8\delta y_n\}$$
$$g_1^{'}[y_n] = \frac{1}{9}\{(36-2\delta) - (72-8\delta)\ y_n\}$$

if $y_n = 0.5$ then,

$$\frac{1}{9}\{(36-2\delta) - (72-8\delta)\ y_n\} > \frac{1}{9}\{(36y-2\delta) - (72-8\delta)\ 0.5\}.$$

□

So that, $g_1[y_n < 0.5] < g_1[0.5] = 1$.

Proof. (4)

When $0 < \delta < 9$ and $y_n \geq 0.5$, the second part of system (3.3) becomes,

$$g_2[y_n] = \frac{4}{9}(9-\delta)\ y_n\ (1-y_n) + \frac{2\delta}{9}\ (1-y_n)$$

$$g_2[y_n] = \frac{1}{9}\{(36-4\delta)\ (y_n - y^2(n)) + 2\delta\ (1-y_n)\}$$

$$
\begin{aligned}
g_2[y_n] &= \frac{1}{9}\{36y_n - 36y^2(n) - 4\delta y_n + 4\delta y^2(n) + 2\delta - 2\delta y_n\} \\
g_2^{'}[y_n] &= \frac{1}{9}\{36 - 72y_n + 8\delta y_n - 6\delta y_n\} \\
g_2[y_n] &= \frac{1}{9}\{(36y - 6\delta) - (72 - 8\delta)\ y_n\}
\end{aligned}
$$

if $y_n = 0.5$,then
$\frac{1}{9}\{(36y - 6\delta) - (72 - 8\delta)\ y_n\} \leq \frac{1}{9}\{(36y - 6\delta) - (72 - 8\delta)\ 0.5\}$.

So that, $g_2[y_n > 0.5] < g_2[0.5] = 1$. □

3.4 Approximate Entropy

There are a variety of methods for determining system complexity from time sequence.

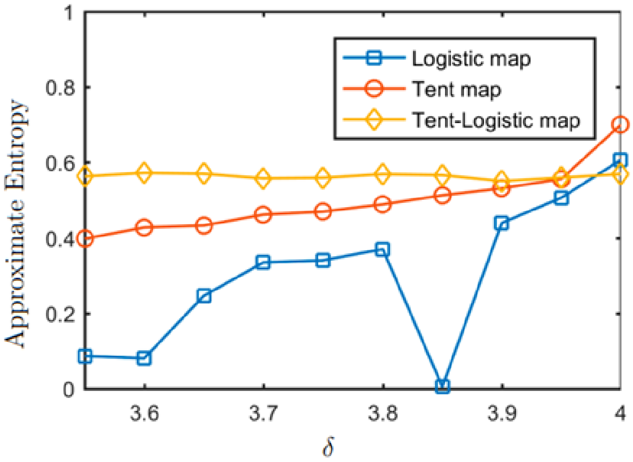

FIGURE 3.9: State distribution of δ

Approximate Entropy [72] is one of the most well-known techniques. The temporal sequence (time sequence) becomes more complicated as the approximate entropy increases. To quantify the complexity of sequences generated by distinct chaotic systems, the predicted entropy values of the sequence created by the three chaotic maps (logistic, tent, and tent-logistic maps) are computed [36] and displayed in Figure 3.7 [36]. The estimated entropy levels of the sequence generated by the tent-logistic map are the greatest among the three chaotic maps in the circumstances with the largest δ values. The sequence formed by the tent-logistic map is verified to be more complex than the sequence generated by the tent and logistic maps.

Chapter 4

Construction of S-box by Using Tent-Logistic Map

S-box play an important role in cryptography. Recently a method for the construction of strong chaotic S-boxes is proposed by Lu et al. [36]. In this chapter, an S-box construction method by using compound chaotic map is discussed. The introduction and the use of S-boxes is given in Section 4.1. The generation mechanism of S-box using a novel compound chaotic system (TLS) is presented in the Section 4.2 and Section 4.3 is based on the performance evolution tests of S-box by using SAMT (tool) on MATLAB.

4.1 Introduction of S-boxes

An S-box is the nonlinear component in block cipher system. In symmetric cryptosystem, the use of S-box is very essential. It helps to convert the plaintext block into ciphertext block, which may causes a confusion effect between plaintext and ciphertext. A $q \times r$ S-box is described as:

$$S : [0,1]^q \longrightarrow [0,1]^r,$$

where q is the input bit and r is the output bit. For the case when $q = r$ the data is neither compressed nor extended during the encryption transformation, the S-box is completely rely on reversible transformation. Figure 4.2 depicts the function and basic idea of $q \times r$ S-box.

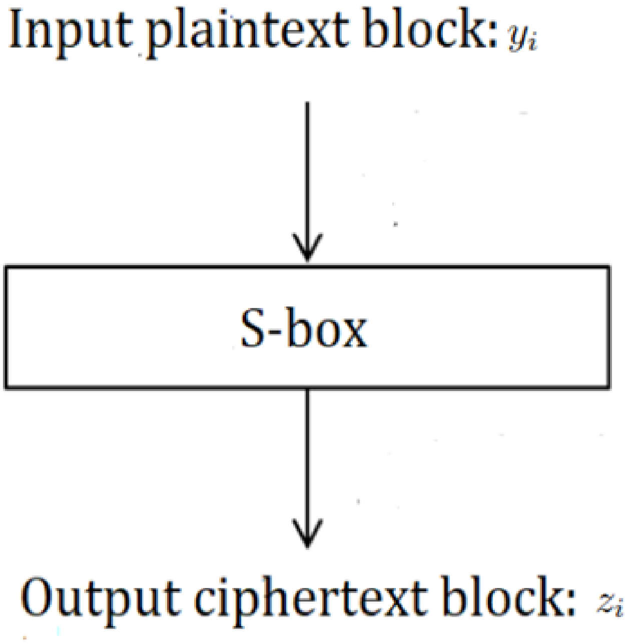

FIGURE 4.1: Function and the basic principle of $q \times r$ S-box

TABLE 4.1: The matrix S_b of 8×8 S-box

j/k	1	2	3	4	$\cdots$	15	16
1	$S_b(1,1)$	$S_b(1,2)$	$S_b(1,3)$	$S_b(1,4)$	$\cdots$	$S_b(1,15)$	$S_b(1,16)$
2	$S_b(2,1)$	$S_b(2,2)$	$S_b(2,3)$	$S_b(2,4)$	$\cdots$	$S_b(2,15)$	$S_b(2,16)$
3	$S_b(3,1)$	$S_b(3,2)$	$S_b(3,3)$	$S_b(3,4)$	$\cdots$	$S_b(3,15)$	$S_b(3,16)$
4	$S_b(4,1)$	$S_b(4,2)$	$S_b(4,3)$	$S_b(4,4)$	$\cdots$	$S_b(4,15)$	$S_b(4,16)$
.	.	.	.	.	$\cdots$	.	.
.	.	.	.	.	$\cdots$	.	.
.	.	.	.	.	$\cdots$	.	.
.	.	.	.	.	$\cdots$	.	.
15	$S_b(15,1)$	$S_b(15,2)$	$S_b(15,3)$	$S_b(51,4)$	$\cdots$	$S_b(15,15)$	$S_b(15,16)$
16	$S_b(16,1)$	$S_b(16,2)$	$S_b(16,3)$	$S_b(16,4)$	$\cdots$	$S_b(16,15)$	$S_b(16,16)$

Where $y_i \in \{0,1\}^q$, $z_i \in \{0,1\}^r$ and $j = 1, 2, ..., n$. The most common type of S-box is an 8×8 matrix, which is notably used in digital image encryption systems

[73]. The algorithm of an 8 × 8 S-box is the subject of this work. An 8 × 8 S-box is a collection of numbers ranging from 0 to 255 that is represented by 16 × 16 matrix $S_b(j,k)$, $j,k = 1,2,...,16$ as illustrated in Table 4.1. There are a total of (2^8!) various types of variation for 8 × 8 S-box. As a consequence, the simplest 8 × 8 S-box is obtained by:

$$S[y] = S_b(j,k) = (j-1) \times 16 + k - 1. \tag{4.1}$$

The process of converting input byte y into output byte z through an S-box with matrix S_b is represented by the function $S[y]$ as:

$$\begin{cases} j = \lfloor y/16 \rfloor + 1 \\ k = \text{mod}(y,16) + 1 \\ z = S[y] = S_b(j,k) \end{cases} \tag{4.2}$$

where $\lfloor c \rfloor$ is the floor function of c i.e., $\lfloor c \rfloor$ returns the closest integer less than or equal to c. The residual after dividing c by n is returned by mod(c, n), where c is the dividend and n is the divisor.

Example 4.1.1. To find the element of S-box by using Equation (4.2) and (4.3). an input bit $y = 55$ is considered, it is used in Equation (4.3)

$$j = \lfloor (55/16) \rfloor + 1$$

$$j = 3 + 1 = 4$$

$$k = \text{mod}(55,\ 16) + 1$$

$$k = 7 + 1 = 8$$

$$\text{Put} \quad j = 4 \quad \text{and} \quad k = 8 \quad \text{in Equation} \quad 4.2$$

$$S_b(4,8) = 55 \quad \text{Consequently,}$$

$$z = S[y] = S_b(j,k) = 55$$

$$z = S_b(4,8) = 55.$$

Similary the method is performed to obtain all the elements of S-box.

It is an easy way to find a simple 8×8 S-box. $S[0] = S_b(1,1) = 0$, $S[1] = S_b(1,2) = 1$, ..., $S[255] = S_b(16,16) = 255$. In $z = S[y] = y$, It is self-evident. The simple 8×8 S-box does not change any input value after applying, therefore it can not used in the encryption system. The inverse transformation is defined as:

$$y = S^{-1}[z],$$

in the decryption method. The $S^{-1}[z]$ is determine by:

$$y = S^{-1}[z] = (j-1) \times 16 + k - 1. \tag{4.3}$$

4.2 The Proposed S-box Generation Algorithm

Many researchers [15–17] have proposed the design approaches of S-boxes generation with various cryptographic strengths. However, because most of these approaches are complicated and inefficient that is time consuming. Lu et al. [36] proposed a novel S-box generated method by using the compound chaotic map and novel linear mapping. This approach takes advantage of the tent-logistic map have strong chaotic properties. The algorithm for creating S-box is explained below.

Algorithm 4.2.1.

Input: Chaotic map (3.3), initial parameters y_0, control parameter δ, integer $L = 65536$ and $c > 0$.

Output: S-box.

Step 1: Take an integer c, s.t $c > 0$ and $c \neq k \times 257$ where $k = 1, 2, \cdots$

Step 2: Initiate an array D as $D = [0,\ 1, \cdots, 255]$.

Step 3: Use the following linear mapping to generate a new array E_i:

$$E_i = c \times (D(i-1)+1) \mod 257, \quad i = 1, 2, \cdots, 256 \tag{4.4}$$

Step 4: Transform 1D array E_i into 2D array Eb_1 and consider Eb_1 as a initial S-box.

Step 5: Use the tent-logistic map (3.3), with the control parameters δ, initial parameter y.

Step 6: Iterate (3.3) L-times, to create a chaotic sequence of length L.

Step 7: To remove the transient effect, discard the first (L-256) numbers of the chaotic sequence, then create a new chaotic sequence of length 256, which is represented by Y.

Step 8: Sort the chaotic sequence Y, after that generate a positive index array F and $F = \{F(1), F(2), \ldots, F(256)\}$. Due to chaotic behaviour they will ultimately lead to $F(i) \neq F(j)$ as long as $i \neq j$.

Step 9: Calculated 1D array B_1 by using the index array F.

Step 10: The suggested S-box is created by transforming the 1D array B_1 into a 2D matrix S_b.

The implementation of Algorithm 4.2.1 is performed on the PC with MATLAB R2017a having operating system window 8.1 pro 64 bit, Core i5-4300M with 2.60 GHz CPU and 8GB Ram. Using the parameters $y_0 = 0.66$, $\delta = 4.5$, $c = 56$, $L = 65536$ on MATLAB. The constructed S-box is shown in Table 4.2.

TABLE 4.2: S-box.

85	22	149	69	231	165	224	83	207	54	232	188	200	43	211	16
239	253	215	3	31	199	1	237	70	139	78	161	55	51	99	123
61	63	245	97	137	48	67	59	4	92	154	121	66	136	57	74
86	88	169	23	52	40	204	112	98	173	32	250	0	30	5	252
240	75	223	254	44	192	210	183	35	81	234	19	228	230	71	21
62	124	203	37	106	179	56	131	24	235	80	145	28	151	208	95
87	115	209	26	125	104	201	146	157	249	91	130	49	217	202	8
114	186	14	226	116	42	167	20	255	160	138	182	41	158	222	12
185	17	72	64	187	50	196	251	118	219	53	111	39	247	168	147
177	143	6	172	120	135	141	122	127	33	45	263	47	171	244	132
133	164	10	248	18	29	15	162	108	155	107	82	193	225	214	176
148	198	58	159	144	229	73	189	46	90	126	174	109	25	102	184
13	191	100	218	129	150	94	101	38	79	117	93	180	36	190	113
9	178	212	89	181	216	7	213	27	233	156	140	128	2	197	60
241	103	242	152	110	246	11	243	227	153	34	195	119	68	163	238
96	134	206	76	194	170	175	221	65	77	220	166	105	205	142	84

4.3 Performance Tests

The properties of constructed S-box in Table 4.2 is performed by using the software tool (MATLAB). Some cryptographic properties such as SAC, BIC-SAC, BIC-NL, and nonlinearity are briefly describe and some properties are also given which is described in Chapter 2 is presented in this section.

4.3.1 Strict Avalanche Criterion

SAC [12], is an essential component for cryptographic S-box. This criterion indicates, if one input bit is change then in the result of each output bit is changed with the probability 0.5. The probability of $P(j, k)$ is 0.5 for each jth and kth, where jth is input bit and kth is output bit, $j, k = 1, 2, \cdots, n$. In order to fulfill the requirement, the Boolean function must be 50 percent dependent on each of its input bits. The values of SAC of S-box rely on the dependency matrix. Table 4.3 shows the dependency matrix of the constructed S-box 4.2 for SAC. The value of jth row and kth column of the table shows the $P(j, k)$ values. The values of $P(j, k)$ shows that the constructed S-box is close to 0.5.

Table 4.3: Dependency matrix of S-box for (SAC)

j/k	1	2	3	4	5	6	7	8
1	0.5625	0.5781	0.5313	0.5625	0.5313	0.4063	0.5156	0.4219
2	0.4688	0.6250	0.4531	0.5469	0.5313	0.4844	0.5625	0.5781
3	0.5156	0.3906	0.5000	0.5000	0.5000	0.4375	0.5156	0.4531
4	0.5156	0.4531	0.5313	0.5313	0.5625	0.4375	0.4531	0.4688
5	0.4688	0.4688	0.5000	0.5156	0.5156	0.4844	0.4688	0.4063
6	0.4219	0.5000	0.5000	0.4531	0.5156	0.5156	0.5313	0.5000
7	0.5000	0.5156	0.5625	0.4531	0.4375	0.4844	0.4219	0.4219
8	0.5156	0.5313	0.4063	0.5000	0.5313	0.5625	0.4844	0.5156

4.3.2 Bit Independence Criterion for SAC

In accordance with the Bit Independence Criterion (BIC) [15] criteria, the ith and jth bit of the data block changes independently, when the kth bit of the data block is altered. This means that the output bit value of S-box and the input bits are changed without disturbing each other. To determine this property of S-box, the BIC-SAC is introduced. The result of BIC-SAC is determine by:

$$(S_i[y] \oplus S_j[z] - S_i[y] \oplus S_j[y]), \quad \forall \quad y \in \{0, 1, ..., 255\},$$

where y and z are one bit different each time. If the average of BIC-SAC is close to 0.5, then any S-box meet this property. The results of BIC-SAC for constructed S-box 4.2 is given in Table 4.4. According to the results, the constructed S-box satisfied the requirements of BIC-SAC.

Table 4.4: BIC for SAC

Boolean Function	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_1	0	0.4785	0.4707	0.4941	0.5098	0.4902	0.5137	0.5117
S_2	0.4785	0	0.5215	0.4902	0.5254	0.5039	0.4902	0.5098
S_3	0.4707	0.5215	0	0.5215	0.4980	0.4961	0.4980	70.5020
S_4	0.4941	0.4902	0.5215	0	0.4727	0.4941	0.5117	70.4961
S_5	0.5098	0.5254	0.4980	0.4727	0	0.4766	0.5156	0.5098
S_6	0.4902	0.5039	0.4961	0.4941	0.4766	0	0.4805	0.5059
S_7	0.5137	0.4902	0.4980	0.5117	0.5156	0.4805	0	0.4941
S_8	0.5117	0.5098	0.5020	0.4961	0.5098	0.5059	0.4941	0

4.3.3 Bit Independence Criterion for Nonlinearity

Bit independence criterion for nonlinearity (BIC-NL) is another important feature for the strong S-box. To find the results of BIC for nonlinearity, compute the nonlinearity values of all output bit values by $(y_i \oplus y_j)$, where $i, j = 1, 2, ..., n$ and $y \in \{0, 1, ..., 255\}$. The results of BIC-NL represented in Table 4.5, shows the average value is 104 so the constructed S-box 4.2 satisfied the BIC-NL and $S_1, S_2, \ldots, S_8$ is the Boolean functions of S-box.

Table 4.5: BIC-NL

Boolean Function	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
y_1	0	102	104	104	104	104	104	102
y_2	102	0	100	104	104	100	102	104
y_3	104	100	0	102	102	104	106	100
y_4	104	104	102	0	104	106	102	104
y_5	104	104	102	104	0	104	96	106
y_6	104	100	104	106	104	0	106	102
y_7	104	102	106	102	96	106	0	104
y_8	102	104	100	104	106	102	104	0

4.3.4 Nonlinearity

An S-box may alternatively be represented in nonlinear format as:

$$z = z_1, z_2, \cdots, z_n = S[t] = S_1[t], S_2[t], \cdots, S_n[t] \tag{4.5}$$

where $z_j = S_j[t] \in \{0, 1\}$, $S_j[t]$ is an n-bit Boolean functions with $j = 1, 2, \cdots, n$ and y is the input bit. To reduce cryptographic attacks, an S-box should have a high nonlinear relationship between input/output values. The nonlinearity (NL) of Boolean functions $S_j[t]$ is used to measure the nonlinear strength of an $n \times n$ S-box, which is computed as:

$$(NL)_j = \frac{1}{2}(2^n - \max_{t\in\{0,1\}^n} \Big|WS - S_j[t]\Big|). \tag{4.6}$$

Here, $WS - S_j[t]$ denotes the Walsh spectrum of function $S_j[t]$, which is computed as:

$$WS - S_j[t] = \sum_{u\in\{0,1\}^n} (-1)^{S_j[t]\oplus t.u}, \tag{4.7}$$

where $t \cdot u$ is the dot product of t and u, computed as follows:

$$t.u = (t_1 \times u_1) \oplus (t_2 \times u_2) \oplus \ldots \oplus (t_n \times u_n). \tag{4.8}$$

Where $\oplus$ refers to the XOR of modulo 2. In an S-box, $(NL)_j$ is the nonlinearity value of the jth constituent of Boolean function. The higher the NL, better the performance against linear cryptanalysis. Table 4.6 shows the NL values of all eight component Boolean functions in the proposed S-box. The average value of nonlinearity is 105.2500, with a minimum of 102, a maximum of 108. Table 4.6 also shows the nonlinearity values of the initial S-box (s_I), which are significantly lower than the nonlinearity values of the final S-box (s_F). The results shows that the nonlinearity of final S-box is enhanced by using the pseudo random chaotic sequence of initial S-box, where the s_i shows the number of Boolean functions of S-box and $i = 1, 2, \ldots, 8$.

Table 4.6: NL of Boolean function of the generated S-box

s_i	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	Average
s_I	54	54	54	54	54	54	54	54	54
s_F	108	106	106	106	108	102	102	104	105.2500

4.3.5 Linear Probability

High confusion and diffusion effects ensure the security of a cryptosystem. S-boxes helps to achieve cryptosystems, considerable confusion/diffusion effects by providing a nonlinear mapping between input/output data. When the nonlinearity of S-box is increase then it is hard to cryptanalysis attack. LP is used to compute the resistance of linear cryptanalysis, which is estimated as:

$$\text{LP} = \max_{\alpha_y, \beta_y \neq 0} \left| \frac{P\{y \in M | y.\alpha_y = S(y).\beta_y\}}{2^n} - \frac{1}{2} \right| \tag{4.9}$$

where $M = 0, 1, ..., 255$, α_y and β_y are the respective input, output bits ($\alpha_y \in M, \beta_y \in M$), '.' represents the dot product and $P\{y \in M|Y\}$ denotes the number of y that fulfills the condition Y. Maximum LP of suggested S-box is just 0.125, indicating that it is resistant to linear cryptanalysis.

4.3.6 Differential Probability

Another successful approach for deciphering ciphertext is differential cryptanalysis [74]. This approach is used to discover plaintext pairings with the same differentials as their associated ciphertext pairs. Attackers can get a portion of the key by using these plaintext pairings and matching ciphertext pairs. Performance against differential cryptanalysis is measured using the DP, which is computed as follows:

$$\mathrm{DP} = \max_{\Delta_y \neq 0, \Delta_z} \left(\frac{P\{y \in M | S(y) \oplus S(y \oplus \Delta y) = \Delta z\}}{2^n} \right) \tag{4.10}$$

where $\Delta y = y \oplus y^{'}$ and $\Delta z = z \oplus z^{'}$ are differentials corresponding to the input $(y, y^{'})$ and output $(z, z^{'})$ pairs. An S-box with a lower DP can withstand differential cryptanalysis better. DP value of generated S-box is 0.039. Because of the small value, the recommended S-box is very resistant to differential cryptanalysis attacks.

4.3.7 Others Properties of S-box

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

- S-box is bijective.
- The number of fixed point and opposite fixed point of S-box is 3.
- Nonlinearity of constructed S-box is 108.
- Dynamic distance of all Boolean functions of S-box is defined in the table below.

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
8	10	4	8	4	12	2	10
4	16	6	6	4	2	8	10
2	14	0	0	0	8	2	6
2	6	4	4	8	8	6	4
4	4	0	2	2	2	4	12
10	0	0	6	2	2	4	0
0	2	8	6	8	2	10	10
2	4	12	0	4	8	2	2

- Perfect nonlinearity (PN) value of all Boolean functions of S-box (s_i) are given below:

s_i	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
PN	148	128	112	124	132	120	116	132

- Avalanche Criterion Percentage of S-box is 59.3750.

- Desired SAC value of S-box is 1024.

- Results for the Boolean functions which satisfy Avalanche criterion is 38.

- Sum of Squares Indicator of Boolean functions of S-box (s_i) are:

s_i	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
SSI	167296	172672	176896	169984	176896	208000	206848	197632

- Absolute Indicator (AI) of Boolean functions of S-box (s_i) are:

s_i	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
AI	64	64	64	56	72	80	64	64

- Bent nonlinearity value of S-box is 116.6863.

- Dynamic Distances (DDF) of Boolean functions of S-box (s_i) are:

s_i	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
DDF	10	16	12	8	8	12	10	12

- Differential Branch Number of S-box is 3.

- Hamming weight (HW) of all Boolean functions of S-box are given below:

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

4.3.8 Performance Comparison

Cryptographic performance of generated S-box is compared with some recently suggested S-boxes. Table 4.7 & 4.8 shows the different performance analysis results of S-boxes.

TABLE 4.7: Comparison of chaotic S-boxes

S-boxes	SAC	NL			BIC-SAC	BIC-NL	LP	DP
		Min	Max	Avg				
Zahid. [75]	0.507	104	108	106.8	0.507	103.9	0.140	0.054
Belazai et al. [27]	0.496	102	108	105.3	0.499	103.8	0.156	0.039
Khan et al. [76]	0.502	102	108	103.5	0.501	103.0	0.133	0.039
Rijindal. [77]	0.504	112	112	112	0.504	112	0.062	0.016
S_I	0.495	54	54	54	0.501	77.1	0.289	1.000
S_F	0.495	102	108	105.25	0.499	104	0.125	0.039

TABLE 4.8: Comparison of non-chaotic S-boxes

S-boxes	SAC	NL			BIC-SAC	BIC-NL	LP	DP
		Min	Max	Avg				
Cavusoglu et al. [22]	0.520	104	110	106.3	0.501	104.2.8	0.133	0.039
Wang et al. [78]	0.495	104	110	106.5	0.498	103.8	0.141	0.039
Liu et al. [14]	0.498	102	108	104.5	0.508	104.6	0.125	0.047
Lambic. [24]	0.503	106	108	106.8	0.502	103.8	0.133	0.039
D.Lambic [25]	0.501	108	112	109.3	0.506	108.2	0.094	0.031
S_I	0.495	54	54	54	0.501	77.1	0.289	1.000
S_F	0.495	102	108	105.25	0.499	104	0.125	0.039

From Table 4.7 & 4.8, it is clear that the generated S-box has a smaller values of LP and DP than other S-boxes. This comparison demonstrate our method give good performance. SAC value of the constructed S-box is 0.505, which is quite near to the ideal SAC value (0.5). The suggested S-box, BIC value is fairly well. It is noticed that the initial S-box that is generated by linear mapping (4.4) has low nonlinearity and the nonlinearity of final S-box is increased that is constructed by using compound chaotic map (3.3).

Chapter 5

Conclusions

In this thesis, the compound chaotic system and the linear mapping is used for the construction of S-box. The innovations of this work are as follows:

- The S-box is constructed by using a simple and effective method. The initial S-box is obtained by using a linear mapping (4.4). After that, TLS (3.3) is used to scramble the initial S-box and obtain final S-box.
- The constructed S-box has small value of linear probability (LP) and differential probability (DP) than the some old S-boxes, so the suggested S-box strongly resists differential cryptanalysis attacks and linear cryptanalysis.

SAMT [55] tool is used for the test analysis of the S-box. In comparison to previous S-boxes, the created S-box has extremely small values of LP and DP and a good average value of nonlinearity, according to the test results and performance analysis. This indicates that the suggested S-box is resistant to both linear and differential cryptanalysis and can be used in the block cryptosystem.

As a future work, it is possible to optimized this S-box applying metaheuristics, similarly optimazition performed by continous chaotic map. The proposed S-box can be used in designing the image encryption schemes. By using the same algorithm S-box can be designed by using hyper chaotic maps. It may increase the nonlinearity and effectiveness of S-box.

Bibliography

[1] J. L. Massey, "An introduction to contemporary cryptology," *Proceedings of the IEEE*, vol. 76, no. 5, pp. 533–549, 1988.

[2] F. Bao, R. H. Deng, W. Geiselmann, C. Schnorr, R. Steinwandt, and H. Wu, "Cryptanalysis of two sparse polynomial based public key cryptosystems," in *International Workshop on Public Key Cryptography*, pp. 153–164, Springer, 2001.

[3] J. Thakur and N. Kumar, "Des, aes and blowfish: Symmetric key cryptography algorithms simulation based performance analysis," *International journal of emerging technology and advanced engineering*, vol. 1, no. 2, pp. 6–12, 2011.

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

[5] S. D. Sinha and C. P. Arya, "Algebraic construction and cryptographic properties of rijndael substitution box," *Defence Science Journal*, vol. 62, no. 1, p. 32, 2012.

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

[7] 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.

[8] H. M. Heys and S. E. Tavares, “Substitution-permutation networks resistant to differential and linear cryptanalysis,” *Journal of cryptology*, vol. 9, no. 1, pp. 1–19, 1996.

[9] X.-P. Zhang, R. Guo, H.-W. Chen, Z.-M. Zhao, and J.-Y. Wang, “Efficient image encryption scheme with synchronous substitution and diffusion based on double s-boxes,” *Chinese Physics B*, vol. 27, no. 8, p. 080701, 2018.

[10] C. Zhu, G. Wang, and K. Sun, “Improved cryptanalysis and enhancements of an image encryption scheme using combined 1d chaotic maps,” *Entropy*, vol. 20, no. 11, p. 843, 2018.

[11] S. Zhang, G. Wang, M. Z. A. Bhuiyan, and Q. Liu, “A dual privacy preserving scheme in continuous location-based services,” *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4191–4200, 2018.

[12] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A compact rijndael hardware architecture with s-box optimization,” in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 239–254, Springer, 2001.

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

[14] L. Liu, Y. Zhang, and X. Wang, “A novel method for constructing the s-box based on spatiotemporal chaotic dynamics,” *applied sciences*, vol. 8, no. 12, p. 2650, 2018.

[15] C. Adams and S. Tavares, “The structured design of cryptographically good s-boxes,” *Journal of cryptology*, vol. 3, no. 1, pp. 27–41, 1990.

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

[17] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *Computers, IEEE Transactions on*, vol. 100, no. 8, pp. 677–691, 1986.

[18] A. Rafiq and M. Khan, "Construction of new s-boxes based on triangle groups and its applications in copyright protection," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15527–15544, 2019.

[19] K. Kazlauskas and J. Kazlauskas, "Key-dependent s-box generation in aes block cipher system," *Informatica*, vol. 20, no. 1, pp. 23–34, 2009.

[20] H. Noura, L. Sleem, M. Noura, M. M. Mansour, A. Chehab, and R. Couturier, "A new efficient lightweight and secure image cipher scheme," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 15457–15484, 2018.

[21] 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.

[22] X. Wang, A. Akgul, U. Cavusoglu, V.-T. Pham, D. Vo Hoang, and X. Q. Nguyen, "A chaotic system with infinite equilibria and its s-box constructing application," *Applied Sciences*, vol. 8, no. 11, p. 2132, 2018.

[23] M. Khan, T. Shah, and S. I. Batool, "Construction of s-box based on chaotic boolean functions and its application in image encryption," *Neural Computing and Applications*, vol. 27, no. 3, pp. 677–685, 2016.

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

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

[26] E. Al Solami, M. Ahmad, C. Volos, M. N. Doja, and M. M. S. Beg, "A new hyperchaotic system-based design for efficient bijective substitution-boxes," *entropy*, vol. 20, no. 7, p. 525, 2018.

[27] A. Belazi, M. Khan, A. A. Abd El-Latif, and S. Belghith, "Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 337–361, 2017.

[28] A. Ullah, S. S. Jamal, and T. Shah, "A novel construction of substitution box using a combination of chaotic maps with improved chaotic range," *Nonlinear Dynamics*, vol. 88, no. 4, pp. 2757–2769, 2017.

[29] A. Belazi and A. A. Abd El-Latif, "A simple yet efficient s-box method based on chaotic sine map," *Optik*, vol. 130, pp. 1438–1444, 2017.

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

[31] G. Chen, Y. Chen, and X. Liao, "An extended method for obtaining s-boxes based on three-dimensional chaotic baker maps," *Chaos, solitons & fractals*, vol. 31, no. 3, pp. 571–579, 2007.

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

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

[34] S. Ke-Hui, H. Shao-Bo, H. Yi, and Y. Lin-Zi, "Complexity analysis of chaotic pseudo-random sequences based on spectral entropy algorithm," *Acta Physica Sinica*, vol. 62, no. 1, 2013.

[35] H. Shao-Bo, S. Ke-Hui, and Z. Cong-Xu, "Complexity analyses of multi-wing chaotic systems," *Chinese Physics B*, vol. 22, no. 5, p. 050506, 2013.

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

[37] V. Shoup, "New algorithms for finding irreducible polynomials over finite fields," *Mathematics of computation*, vol. 54, no. 189, pp. 435–447, 1990.

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

[39] E. R. Berlekamp, "Factoring polynomials over finite fields," *Bell System Technical Journal*, vol. 46, no. 8, pp. 1853–1859, 1967.

[40] M. Stanek, "On cryptographic properties of random boolean functions," *Journal of Universal Computer Science*, vol. 4, no. 8, pp. 705–717, 1998.

[41] M. M. Wong, M. D. Wong, A. K. Nandi, and I. Hijazin, "Composite field gf (((2 2) 2) 2) advanced encryption standard (aes) s-box with algebraic normal form representation in the subfield inversion," *IET circuits, devices & systems*, vol. 5, no. 6, pp. 471–476, 2011.

[42] S. Picek, D. Jakobovic, J. F. Miller, L. Batina, and M. Cupic, "Cryptographic boolean functions: One output, many design criteria," *Applied Soft Computing*, vol. 40, pp. 635–653, 2016.

[43] J. A. Clark, J. L. Jacob, S. Maitra, and P. Stănică, "Almost boolean functions: The design of boolean functions by spectral inversion," *Computational Intelligence*, vol. 20, no. 3, pp. 450–462, 2004.

[44] J. Seberry, X.-M. Zhang, and Y. Zheng, "Systematic generation of cryptographically robust s-boxes," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 171–182, 1993.

[45] Y. Wei and E. Pasalic, "On the approximation of s-boxes via maiorana-mcfarland functions," *IET Information Security*, vol. 7, no. 2, pp. 134–143, 2013.

[46] J. Manz, "A sequency-ordered fast walsh transform," *IEEE Transactions on Audio and Electroacoustics*, vol. 20, no. 3, pp. 204–205, 1972.

[47] Y. Wang, K.-W. Wong, C. Li, and Y. Li, "A novel method to design s-box based on chaotic map and genetic algorithm," *Physics Letters A*, vol. 376, no. 6-7, pp. 827–833, 2012.

[48] Y. V. Tarannikov, "On resilient boolean functions with maximal possible nonlinearity," in *International Conference on Cryptology in India*, pp. 19–30, Springer, 2000.

[49] I. Hussain and T. Shah, "Literature survey on nonlinear components and chaotic nonlinear components of block ciphers," *Nonlinear Dynamics*, vol. 74, no. 4, pp. 869–904, 2013.

[50] S. Mister and C. Adams, "Practical s-box design," in *Workshop on Selected Areas in Cryptography, SAC*, vol. 96, pp. 61–76, Citeseer, 1996.

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

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

[53] H. Ochi, N. Ishiura, and S. Yajima, "Breadth-first manipulation of sbdd of boolean functions for vector processing," in *Proceedings of the 28th ACM/IEEE Design Automation Conference*, pp. 413–416, 1991.

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

[55] R. Wieland, M. Voss, X. Holtmann, W. Mirschel, and I. Ajibefun, "Spatial analysis and modeling tool (samt): 1. structure and possibilities," *Ecological Informatics*, vol. 1, no. 1, pp. 67–76, 2006.

[56] Z. Su, G. Zhang, and J. Jiang, "Multimedia security: a survey of chaos-based encryption technology," *Mutimedia: A Multidisiplinary Approach to Complex Issues, Ed. I. Karydis, InTech*, pp. 99–124, 2012.

[57] F. Su, "The chaos theory and its application," in *Journal of Physics: Conference Series*, vol. 2012, p. 012118, IOP Publishing, 2021.

[58] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[59] É. Ghys, "The butterfly effect," in *The Proceedings of the 12th International Congress on Mathematical Education*, pp. 19–39, Springer, Cham, 2015.

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

[61] K.-W. Wong, S.-W. Ho, and C.-K. Yung, "A chaotic cryptography scheme for generating short ciphertext," *Physics Letters A*, vol. 310, no. 1, pp. 67–73, 2003.

[62] J. Amigo, L. Kocarev, and J. Szczepanski, "Theory and practice of chaotic cryptography," *Physics Letters A*, vol. 366, no. 3, pp. 211–216, 2007.

[63] Z. Hong and D. Ji-xue, "Chaos theory and its application in modern cryptography," in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, vol. 7, pp. V7–332, IEEE, 2010.

[64] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International journal of bifurcation and chaos*, vol. 16, no. 08, pp. 2129–2151, 2006.

[65] M. Gotz, K. Kelber, and W. Schwarz, "Discrete-time chaotic encryption systems. i. statistical design approach," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, no. 10, pp. 963–970, 1997.

[66] F. Dachselt and W. Schwarz, "Chaos and cryptography," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 12, pp. 1498–1509, 2001.

[67] L. Kocarev, "Chaos-based cryptography: a brief overview," *IEEE Circuits and Systems Magazine*, vol. 1, no. 3, pp. 6–21, 2001.

[68] J. A. Yorke and K. T. Alligood, “Period doubling cascades of attractors: a prerequisite for horseshoes,” *Communications in mathematical physics*, vol. 101, no. 3, pp. 305–321, 1985.

[69] A. Skrobek, “Cryptanalysis of chaotic stream cipher,” *Physics letters A*, vol. 363, no. 1-2, pp. 84–90, 2007.

[70] A. Díaz-Méndez, J. Marquina-Pérez, M. Cruz-Irisson, R. Vazquez-Medina, and J. L. Del-Río-Correa, “Chaotic noise mos generator based on logistic map,” *Microelectronics journal*, vol. 40, no. 3, pp. 638–640, 2009.

[71] J. Ahmad and S. O. Hwang, “Chaos-based diffusion for highly autocorrelated data in encryption algorithms,” *Nonlinear Dynamics*, vol. 82, no. 4, pp. 1839–1850, 2015.

[72] S. M. Pincus, “Approximate entropy as a measure of system complexity.,” *Proceedings of the National Academy of Sciences*, vol. 88, no. 6, pp. 2297–2301, 1991.

[73] S. Zhu, G. Wang, and C. Zhu, “A secure and fast image encryption scheme based on double chaotic s-boxes,” *Entropy*, vol. 21, no. 8, p. 790, 2019.

[74] C. Li, S. Li, K.-T. Lo, and K. Kyamakya, “A differential cryptanalysis of yen–chen–wu multimedia cryptography system,” *Journal of Systems and Software*, vol. 83, no. 8, pp. 1443–1452, 2010.

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

[76] F. A. Khan, J. Ahmed, J. S. Khan, J. Ahmad, M. A. Khan, and S. O. Hwang, “A new technique for designing 8× 8 substitution box for image encryption applications,” in *2017 9th Computer Science and Electronic Engineering (CEEC)*, pp. 7–12, 2017.

[77] J. Daemen and V. Rijmen, “The design of aes,” *The Advanced Encryption Standard*, 2002.

[78] X. Wang, Ü. Çavuşoğlu, S. Kacar, A. Akgul, V.-T. Pham, S. Jafari, F. E. Alsaadi, and X. Q. Nguyen, "S-box based image encryption application using a chaotic system without equilibrium," *Applied Sciences*, vol. 9, no. 4, p. 781, 2019.