A Key dependent dynamic S-Box using two Chaotic Logistic Equations based PRNG

A Dissertation Proposal
Submitted

By

CHANDRAKAR RAM KISHAN NANDARAM

(11301384)

to
Department of Computer Science and Technology

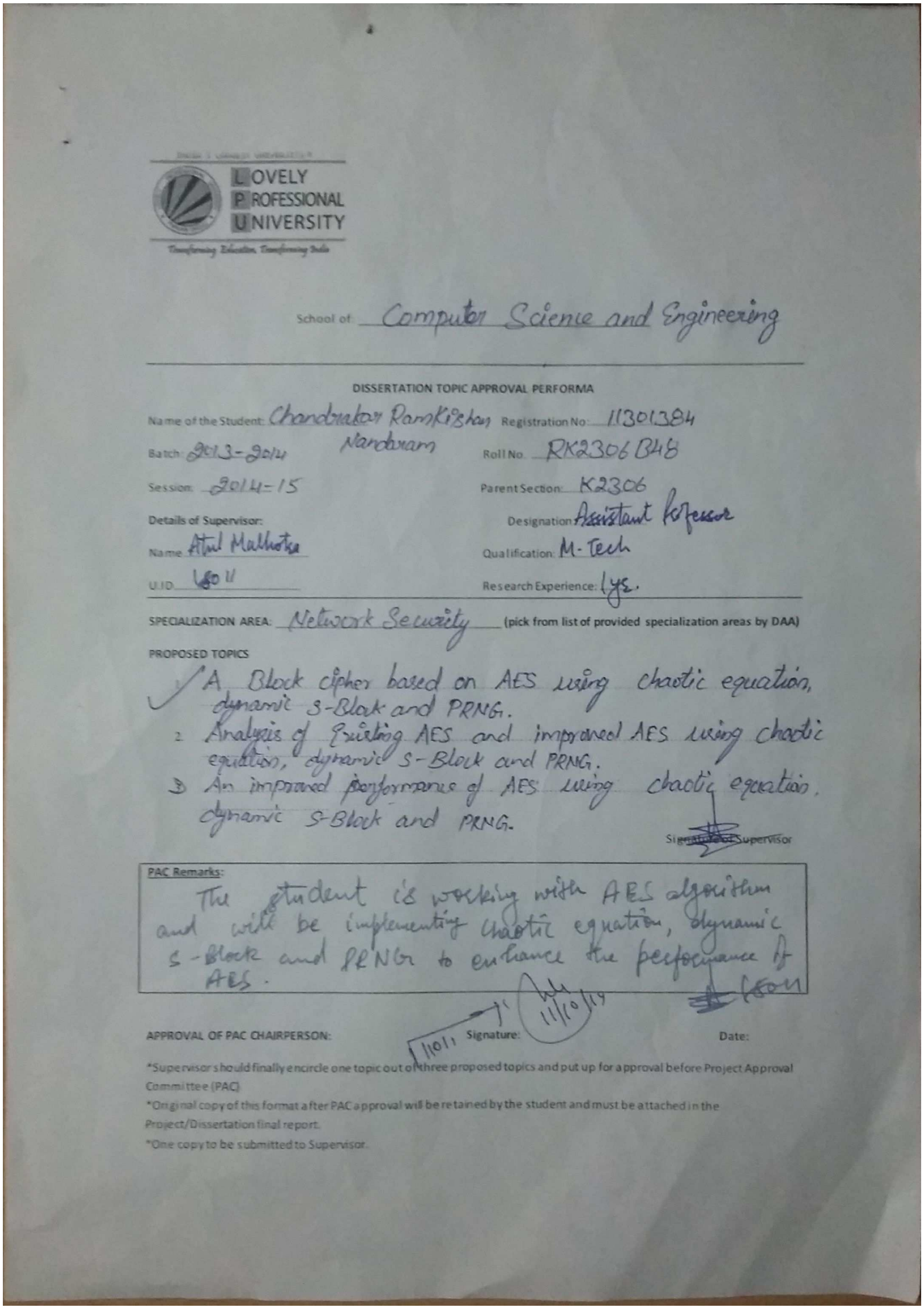
In partial fulfilment of the Requirement for the
Award of the Degree of

Master of Technology in
Computer Science & Engineering

Under the guidance of

Er. Atul Malhotra
(May,2015)

PAC APPROVAL FORM



LOVELY PROFESSIONAL UNIVERSITY

Transforming Education, Transforming India

School of: Computer Science and Engineering

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the Student: Chandrakar Ramkishan Nandaram   Registration No: 11301384

Batch: 2013-2014   Roll No. RK2306B48

Session: 2014-15   Parent Section: K2306

Details of Supervisor: Designation: Assistant Professor

Name: Atul Malhotra   Qualification: M-Tech

U.I.D. 16011   Research Experience: 1 yrs.

SPECIALIZATION AREA: Network Security (pick from list of provided specialization areas by DAA)

PROPOSED TOPICS

1. ✓ A Block cipher based on AES using chaotic equation, dynamic S-Block and PRNG.
2. Analysis of Existing AES and improved AES using chaotic equation, dynamic S-Block and PRNG.
3. An improved performance of AES using chaotic equation, dynamic S-Block and PRNG.

Signature of Supervisor

PAC Remarks:

The student is working with AES algorithm and will be implementing chaotic equation, dynamic S-Block and PRNG to enhance the performance of AES.

16011

11/10/14

APPROVAL OF PAC CHAIRPERSON: 11/10/14   Signature: Date:

*Supervisor should finally encircle one topic out of three proposed topics and put up for approval before Project Approval Committee (PAC)

*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.

*One copy to be submitted to Supervisor.

ABSTRACT

Advanced Encryption Standard (AES) block cipher system is widely practiced in many cryptographic applications. The main core of AES block cipher is the look up table or substitution S-Box. The Substitution S-Boxes are one of the major parts of the cryptographic system. It brings non-linearity to cryptosystem and also increases the security of the cryptographic system. In conventional AES S-Box is fixed, but if we generate dynamic S-Box then the security of the AES would be increased. The aim of this paper is to design a key dependent dynamic S-Box by using PRNG based on two chaotic logistic maps. The two chaotic logistic maps used in this paper are to generate the random numbers for dynamic S-Box in AES using its secret key. The proposed approach is to generate the dynamic S-boxes changing for every change of the secret key. The quality of the implemented S-box is tested and also compared with conventional AES on the basis of following criteria: avalanche effect, simulation time, key sensitivity, and randomness test. The generated S-boxes will increase the complexity and also has better results in security analysis. The key dependent dynamic S-Box will increase the security of the AES algorithm and makes the differential and linear cryptanalysis more difficult.

CERTIFICATE

This is to certify that **Chandrakar RamKishan Nandaram** has completed M.Tech dissertation titled **A Key dependent dynamic S-Box using two Chaotic Logistic Equations based PRNG** under my guidance and supervision. To the best of my knowledge, the present work is the result of his original investigation and study. No part of the dissertation has ever been submitted for any degree or diploma.

The dissertation is fit for the submission and partial fulfilment of the conditions for the award of M.Tech Computer Science & Engg.

Date :______________ Signature of Advisor

Name :

ACKNOWLEDGEMET

I am very grateful to my project guide **Mr. Atul Malhotra**, Asst. Prof. in Department of Computer Science & Technology, Lovely Professional University, for his extensive patience and guidance throughout my project work. Without his help I could not have completed my work successfully. I would like to thank **Mr. Dalwinder Singh**, Professor and Head, Department of Computer Science and Engineering, Lovely Professional University, for having provided the freedom to use all the facilities available in the department, especially the library. I also would like to thank my classmates for always being there whenever I needed help or moral support. Finally, I express my immense gratitude with pleasure to the other individuals who have either directly or indirectly contributed to my need at right time for the development and success of this work.

DECLARATION

I hereby declare that the dissertation entitled **A Key dependent dynamic S-Box using two Chaotic Logistic Equations based PRNG** submitted for the M.Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date:_____________ **Investigator**

Regn. No._____________

TABLE OF CONTENTS

Sr.No	Topic Name	Page No.
1.	INTRODUCTION………………………………………………..................	1
	1.1 Goals of cryptography...…………………………………….................	1
	1.2 Type of Security Attacks.……………………………....................	2
	1.3 Categories of Cryptography…………………………..................	2
	1.3.1 Secret Key Cryptography…………………………………….....	2
	1.3.2 Public Key Cryptography………………………………………	3
	1.3.3 Hash function …………………………………......................	4
	1.4 Advanced Encryption Standard (AES)………………..................	5
	1.4.1 The Substitute Bytes Transformation………………………...	7
	1.4.2 ShiftRows Transformation …………………………………..	10
	1.4.3 MixColumns Transformation…………………………….......	10
	1.4.4 AddRoundKey Transformation……………………………....	11
	1.4.5 AES Key Expansion ………………………..........................	11
	1.5 Chaotic Logistic Map ……………………………......................	14
2.	LITERATURE SURVEY ....…………………………....................	15
3.	PRESENT WORK .... …………………………….....................	20
	3.1 Problem Formulation	20
	3.2 Objectives …………………………..	20
	3.3 Methodology………………………..	20
	3.3.1 The proposed key dependent dynamic AES-128 S-BOX...	21
	3.3.2 Calculation of initial parameters using Key	24
	3.3.3 The proposed Pseudo Random Number Generator (PRNG)	25
4.	RESULTS AND DISCUSSIONS	27
5.	CONCLUSION AND FUTURE SCOPE	34
6.	LIST OF REFERENCES ……………………………......................	35
7.	APPENDIX …………………………………………....................	37

LIST OF TABLES

Sr.No	Table name	Page No.
1.	Look Up Table of S-Box	9
2.	Look up table of Inverse S-Box	9
3.	RC[j] values in Hex	13
4.	Key dependent dynamic S_Box	26
5.	Key dependent dynamic Inverse S_Box	26
6.	samples of Ciphertext due to one bit change in plaintext	29
7.	Dynamic S-Box1	31
8.	Dynamic S-Box2	32
9.	Designing Time of S-Boxes	32

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1.	Overview of cryptography	1
2.	Conventional encryption/decryption	3
3.	Public key Encryption/Decryption	4
4.	Representation of Block, State and Word	5
5.	Structure of AES	6
6.	The S-Box for ByteSub	7
7.	The S-box for Inv ByteSub	8
8.	Structure of the SubByte	9
9.	Structure of the ShiftRows	10
10.	Structure of MixColumn	10
11.	Structure of AddRound Key	11
12.	First round of the Expansion Key Algorithm	12
13.	Pseudo code for AES key expansion	12
14.	Behaviour of logistic map	14
15.	Flow chart for designing dynamic S-Box	22
16.	Structure of Key dependent AES S-Box	23
17.	Pseudo Random Number Generator	26
18.	Avalanche effects of standard AES	28
19.	Avalanche effects of dynamic AES with key dependent S-box.	28
20.	Percentage change in S-Boxes	30
21.	Two ciphered images, encrypted by the two slightly different keys.	31
22.	Original image and its Histogram	33
23.	Conventional AES Encrypted image and its Histogram	33
24.	Dynamic AES Encrypted image and its Histogram	33

CHAPTER 1

INTRODUCTION

In the recent years, with a growth of the internet, there were many applications based on internet has been developed such as online shopping, electronic card payment, stock trading, web-based banking, etc, and almost all the data is transferred over internet. The information transmitted over the internet is not only text, but also contains multimedia like image, audio etc. due to which important information are being exposed to illegal users. The security of transmission of information over the communication channel has become a major issue. For this reason, cryptography techniques were introduced. Most cryptography techniques have an easy implementation and are widely used in the field of information security. Cryptography is the artistic creation of combining some input data called the plain text, with a user specific secret password to get an encrypted output called cipher text which is human unreadable form by using certain algorithms. The cipher text can only convert into plain text by using the secret key shared by the sender. The algorithm that combines the key and the plaintext are called ciphers. The cryptographic algorithm consists of a number of mathematical operations and functions and that perform the encryption process by many ways. The key which is used in the encryption process usually used by the algorithm and its calculation, is a specific number.

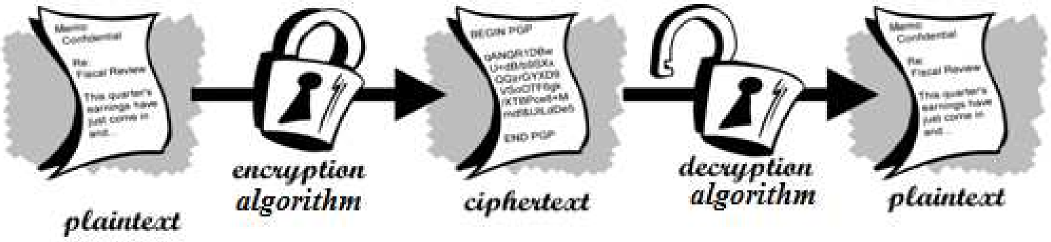


Figure 1: overview of cryptography

1.1 Goals of cryptography

There are main three goals of cryptography that are Confidentiality, Integrity, and Availability which are uniquely referred to be as CIA triad [15].

a. **Confidentiality:** Hiding information from unauthorized access. Only an authorized user can access the data where Information while exchange should remain secret.
b. **Integrity:** Preventing information from unauthorized modification, that is, information should be prevented from modification, by a person who is not authorized.
c. **Availability:** The information should be easily available to authorized users.

To achieve these above goals the cryptographic algorithm needs to be designed. The idea of the design algorithm is as follows that is everybody knows the algorithms, the algorithm exists in the public domain, but the key should not know to an attacker. So, therefore, the objective of any attacker is essential, to find out the key.

1.2 Types of Security Attacks

a. **Cryptanalytic Attack -** The attacker essentially tries to find out, or rather, applies mathematical techniques to find out the key and the weaknesses of existing cryptographic algorithms. Exhaustive key search or Brute-force attack is one of the cryptanalytic attacks.
b. **Non-cryptanalytic Attacks -** There are two kinds of attacks, a) passive attack, which are known as snooping attacks and traffic analysis, b) active attack, which includes masquerading, modification of messages, replying and denial of service. Some other type of attack which include in this Non-cryptanalytic attack is Man-in-the middle approach.

1.3 Categories of Cryptography

Cryptography algorithm can be broadly classified into three types depending on the number of keys that are used in the encryption and decryption process, and further defined by their application and use. Secret Key Cryptography, Public Key Cryptography and Hash Functions [15].

1.3.1 Secret Key Cryptography

Here same unique key is used for encryption and decryption process as shown in figure 2. Sender uses that unique key to convert the plain text into cipher text and sends it to the receiver. The receiver uses the same unique key and gets the plain text from the cipher text. As a single key is used for both, encryption and decryption functions, secret key cryptography is also known as Conventional or Symmetric Cryptography. In this cryptography the key must be known to both sender and recipient and it should be confidential.

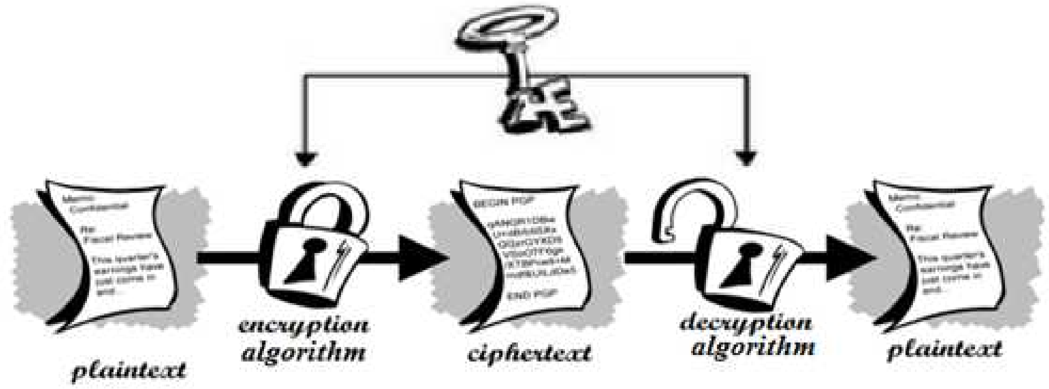


Figure 2: Conventional encryption/decryption

Advantages of conventional cryptography are:

- It is very fast.
- It is especially useful for encrypting data that is to be stored securely and not transmitted.

The biggest issue in this cryptography is the sharing of the key (Key Distribution). Secret key cryptography schemes are being divided as either stream ciphers or block ciphers. A stream cipher is functioning on a single bit at a time. A block cipher is functioning on one block of data at a time utilizing the same key on each block. Some of the secret key cryptography algorithms are DES (Data Encryption Standard), AES (Advance Encryption Stanadard), BLOWFISH etc.

1.3.2 Public Key Cryptography

The problems of key distribution are solved by public key cryptography. It is an asymmetric key cryptography which uses a pair of keys for encryption: “public key”- which is used in encryption process and a corresponding “private key”- which is usedin decryption process. Here the public key is available to the world but the private key is kept secret i.e., anyone having the copy of public key can encrypt information whereas decryption is only done with the known of private key. It is computationally infeasible to deduce the private key from the public key. Anyone who has a public key can encrypt information but cannot decrypt it. Only the person who has the corresponding private key can decrypt the information. The algorithms which come under these categories are RSA (named after its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman), and Digital Signature Algorithm.

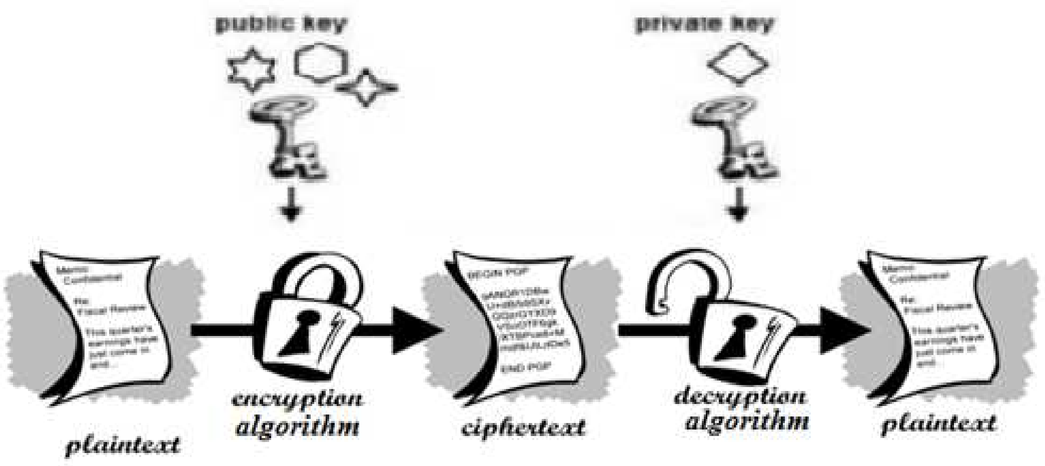


Figure 3: Public key Encryption/Decryption

Advantages of Public key cryptography are,

- The primary benefit of public key cryptography is that it allows people who have no pre-existing security arrangement to exchange messages securely.
- The need for sender and receiver to share secret keys via some secure channel is eliminated: all communications involve only public keys and no private key is ever transmitted or shared.

A major disadvantage of asymmetric ciphers is the key management. Both the sender and receiver communicating parties must, ideally, share a different key.

1.3.3 Hash function

Another name for the Hash functions is message digests and one-way encryption. The hash function does not use any key, in some sense. The hash value is computed based on the plaintext. Hash algorithms are typically applied to supply a digital fingerprint of a file's contents, often used to ensure that the file has not been modified by an intruder or virus. The Hash algorithms that are in common use today include: Message Digest algorithm, Secure Hash Algorithm, Tiger.

1.4 Advanced Encryption Standard (AES)

AES was presented by the National institute of standards and technology (NIST) [15] in the year of 2001. It's a symmetric block cipher which processes data blocks of 128 bits with key size of 128, 192, or 256 bits. In addition, the AES is an iterative algorithm: each iteration being called a round. Encryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Except for the last round in each case, all other rounds are alike. The AES algorithm's operations are performed on 4×4 matrix called the state, each cell of the matrix is filled with bytes. Figure4 shows the conversion of Bytes into State and then Words.

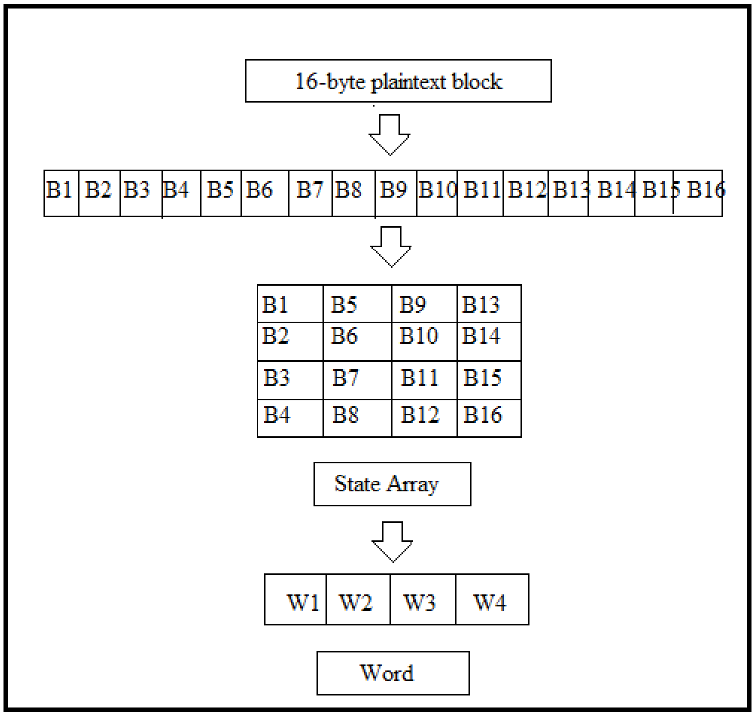


Figure 4: Representation of Block, State and Word

Most of the operations in the AES algorithm take place on bytes of data or on words of data (4 bytes long), which are represented in the field GF(2^8), called the Galois Field. These bytes are represented by the polynomial equation,

$$b_7x^7+ b_6x^6+ b_5x^5+ b_4x^4+ b_3x^3+ b_2x^2+ b_1x + b_0 = \sum b_ix^i$$

Where b_i € {0,1} and i=0,1,2,....7 there are 256 elements in GF(2^8).

For example, 0x 31(00110001) identifies the specific finite field x^5+x^4+1.

The initial state is the plaintext and the final state is the cipher text. As the block length is 128 bits, each row of the state consist 4-bytes. The four bytes in each column form a 32 bit word. In the intial phase the input data 128 bits are converted into state, then the initial Add round key is performed. A round function consists of four operations. First one is Substitution bytes, this function uses the S-Box and based on it Byte by Byte substitution will be performed. The Second operation is Shift Rows, in this function the circular shift operations are performed row by row. Third operation is Mix Columns, in this function the columns are substituted one by one using a substitution matrix. Fourth operation is the Add round key, in this operation the X-OR operation is performed bit wise on the present each byte of the state with the key. These four operations are performed on the input data. AES is having more efficiency compares with DES algorithm. DES uses Feistel structure for encryption of the data, but AES uses a substitution-permutation network. The overall encryption and decryption process of AES is shown in the figure5 [14].

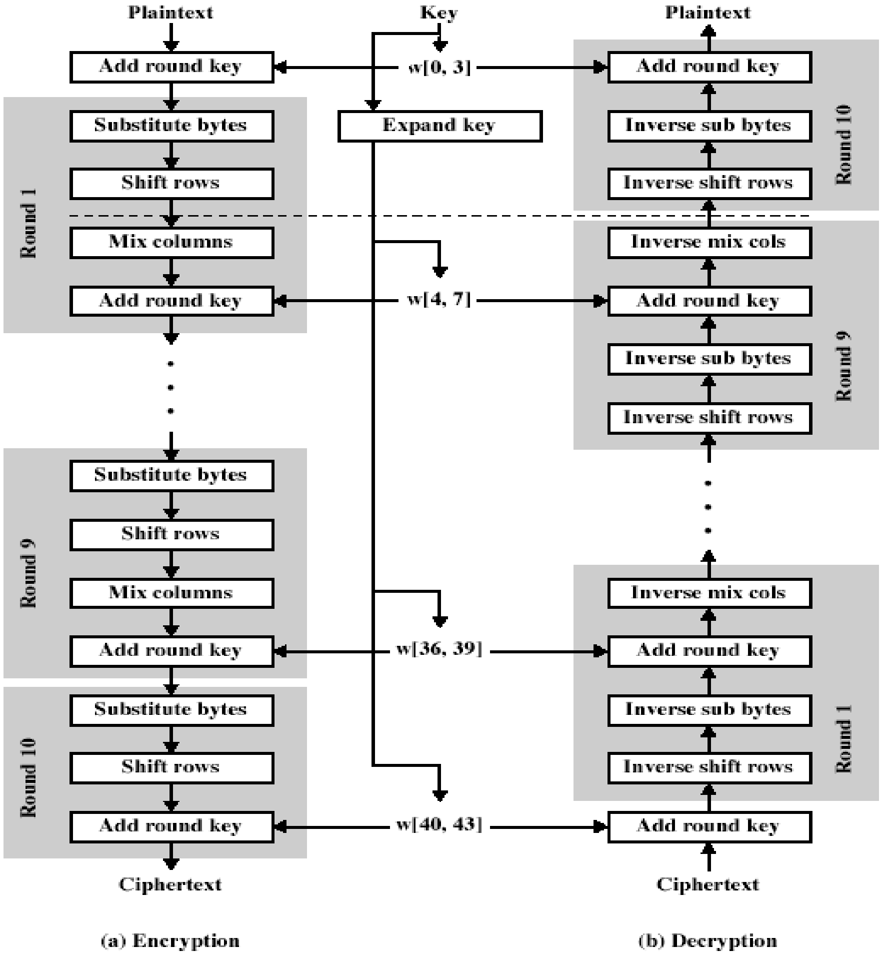


Figure 5: Structure of AES

1.4.1 The Substitute bytes transformation

The first operation performed on the state matrix is the substitution bytes transformation on every round of encryption in the main body. This substitution bytes transformation is also performed directly after the last round in the final part of the algorithm. The forward substitute byte transformation (SubBytes) is a simple table lookup. AES defines a 16 x 16 matrix of byte values, called an S-box that contains a permutation of all possible 256 8-bit values. The S-Box is constructed by composing two transformations. The lookup table of S-Box is generated by using the Galois fields GF(2) and GF(2^8), and affine transformation. The transformation S-Box find the multiplicative inverse of the byte in the field GF(2^8) and it is followed by second affine transformation [15]. The affine transformation is selected in order to make the substitution byte a complex algebraic expression while preserving the nonlinearity property.

i. Take the multiplicative inverse in the finite field GF(2); the element {00} is mapped to itself.

ii. Apply the following affine transformation over GF(2) which is defined as,

$b_i' = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus C_i$

Where $0 \leq I \leq 8$ and b_i is the ith bit of byte and C_i is the ith bit of byte C whose value is 0x63 or (01100011). In matrix form, the affine transformation element of the S-Box can be expressed as;

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

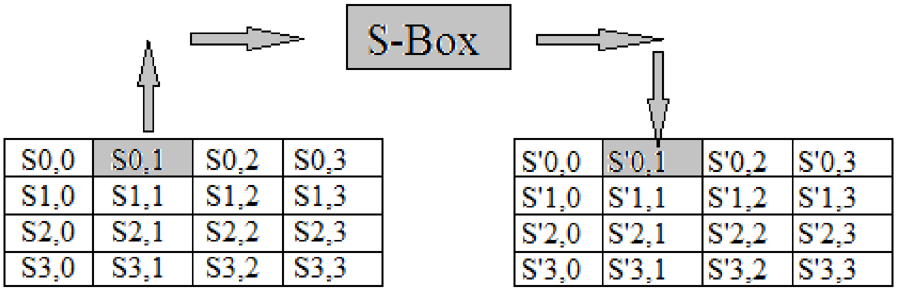
Figure 6: The S-Box for ByteSub

The Inverse ByteSub is used to reverse this operation in decryption process. The affine transformation for inverse ByteSub is as shown below;

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Figure 7: The S-box for Inv ByteSub Operation

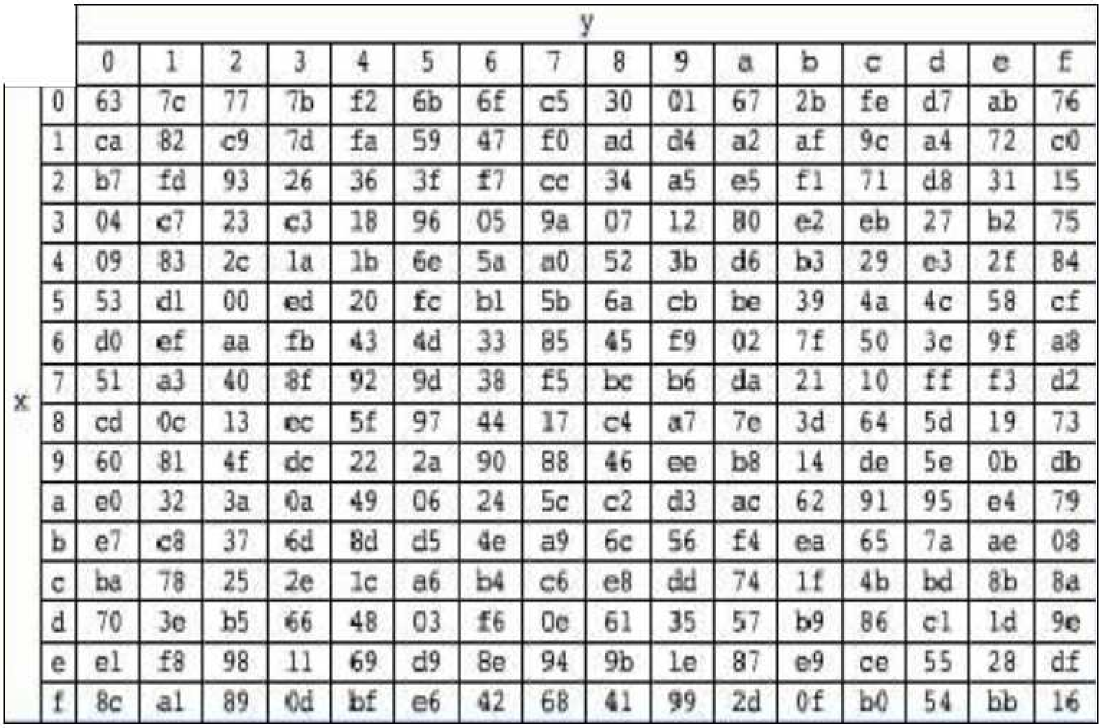
Each individual byte of state is mapped into a new byte in the following way: The first leftmost 4bits of the byte are used as a row value (x) and the last rightmost 4 bits are used as a column value (y). These row and column values serve as an index to the S-box to select a unique 8-bit output value. For example, consider the hexadecimal value {79} indicate row 7, column 9 of the S-box, which contains a value {B6}. By looking the lookup table the value {79} is mapped into the value {B6} [15].



S0,0	S0,1	S0,2	S0,3
S1,0	S1,1	S1,2	S1,3
S2,0	S2,1	S2,2	S2,3
S3,0	S3,1	S3,2	S3,3

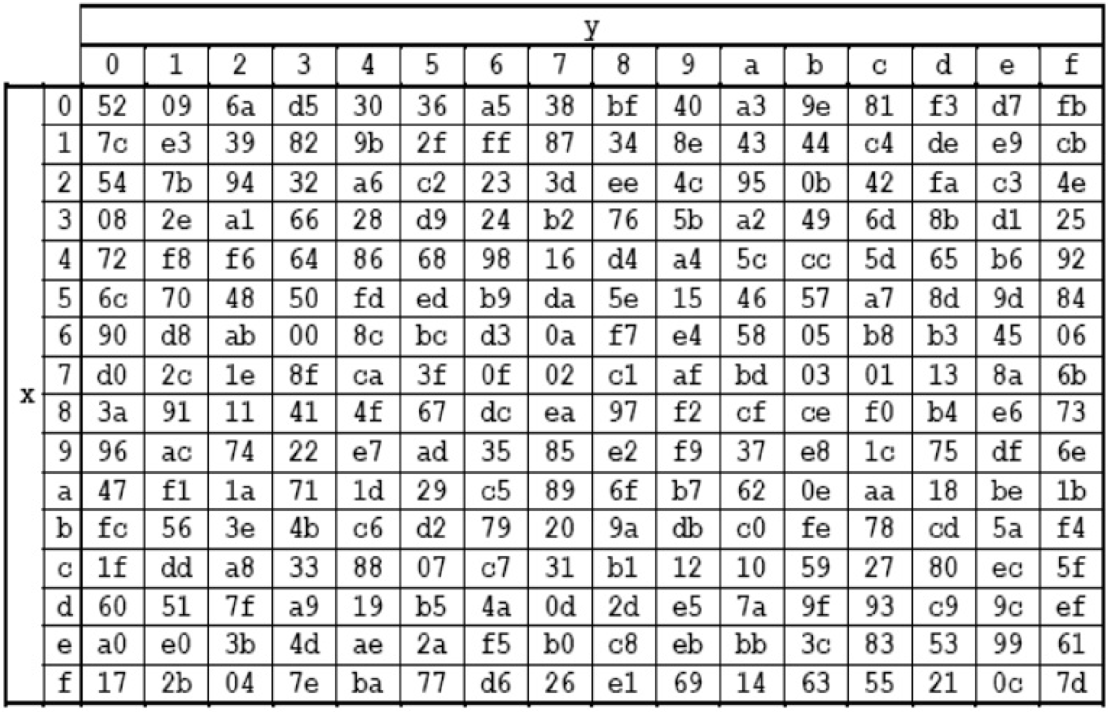
S'0,0	S'0,1	S'0,2	S'0,3
S'1,0	S'1,1	S'1,2	S'1,3
S'2,0	S'2,1	S'2,2	S'2,3
S'3,0	S'3,1	S'3,2	S'3,3

Figure 8: Structure of the SubByte



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

Table 1: Look Up Table of S-Box

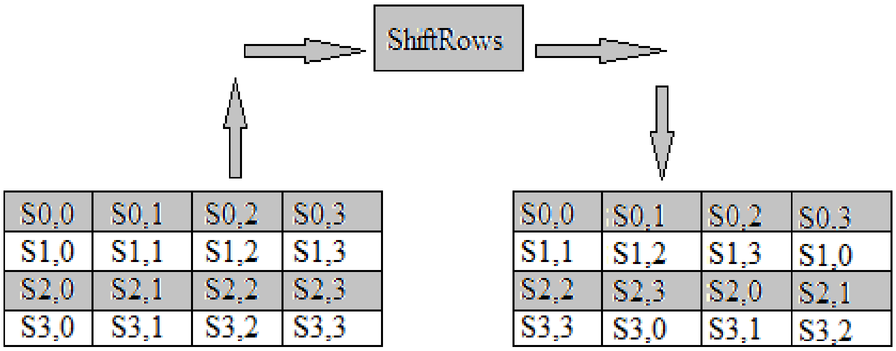


		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65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

Table 2: Look up table of Inverse S-Box

1.4.2 ShiftRows Transformation

The shift rows transformation takes every row of the state matrix and shifts it left. The first row of state is remains unchanged, the second row is circular shifted by one byte, the third row is circular shifted by two byte and the third row is shifted by three byte [15]. The structure of the Shiftrows is shown in the figure 9.



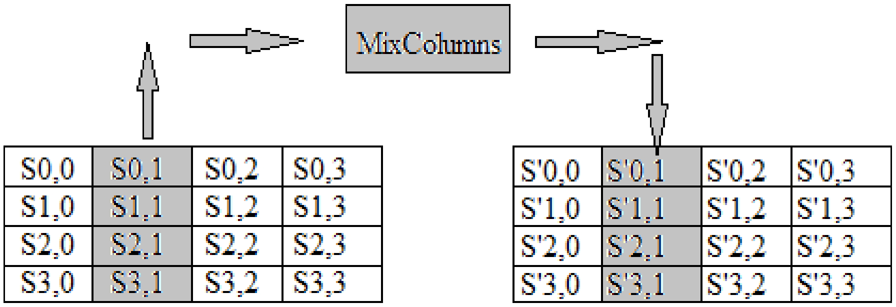
S0,0	S0,1	S0,2	S0,3
S1,0	S1,1	S1,2	S1,3
S2,0	S2,1	S2,2	S2,3
S3,0	S3,1	S3,2	S3,3

S0,0	S0,1	S0,2	S0,3
S1,1	S1,2	S1,3	S1,0
S2,2	S2,3	S2,0	S2,1
S3,3	S3,0	S3,1	S3,2

Figure 9: Structure of the ShiftRows

1.4.3 Mix Columns Transformation

In Mix Columns transformation, the four bytes of each column of the State are combined using an invertible linear transformation matrix as shown in in figure 10. The MixColumn function takes 4- bytes as input and output, where each input byte affects all four output bytes [15].



S0,0	S0,1	S0,2	S0,3
S1,0	S1,1	S1,2	S1,3
S2,0	S2,1	S2,2	S2,3
S3,0	S3,1	S3,2	S3,3

S'0,0	S'0,1	S'0,2	S'0,3
S'1,0	S'1,1	S'1,2	S'1,3
S'2,0	S'2,1	S'2,2	S'2,3
S'3,0	S'3,1	S'3,2	S'3,3

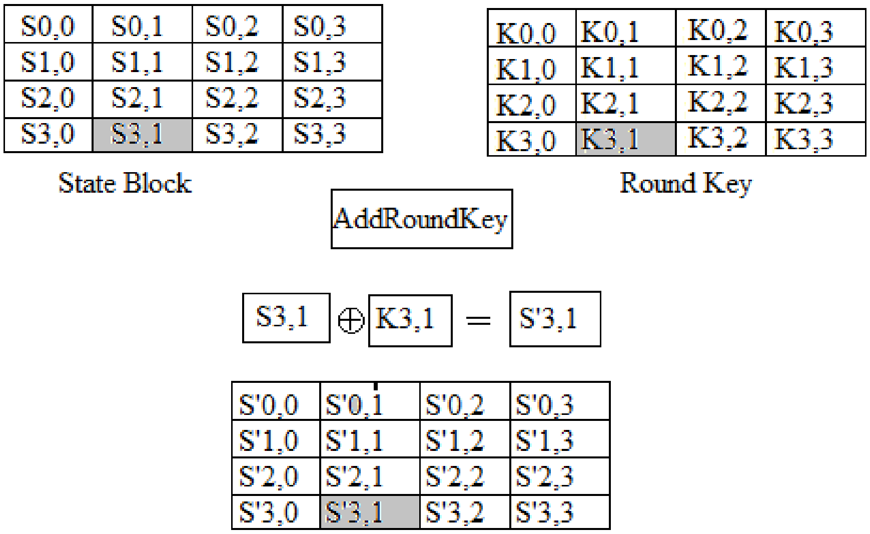
$$\begin{pmatrix} S'0,1 \\ S'1,1 \\ S'2,1 \\ S'3,1 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} S0,1 \\ S1,1 \\ S2,1 \\ S3,1 \end{pmatrix}$$

Tranformation matrix of MixColumns

Figure 10: Structure of MixColumn

1.4.4 AddRoundKey Transformation

In this transformation, the 128 bits of the round key are bitwise XORed with the 128 bits of State. As shown in figure 11, the function is viewed as a column wise function between the 4-bytes of a State column and one word of the round key [15].



S0,0	S0,1	S0,2	S0,3
S1,0	S1,1	S1,2	S1,3
S2,0	S2,1	S2,2	S2,3
S3,0	S3,1	S3,2	S3,3

State Block

K0,0	K0,1	K0,2	K0,3
K1,0	K1,1	K1,2	K1,3
K2,0	K2,1	K2,2	K2,3
K3,0	K3,1	K3,2	K3,3

Round Key

S'0,0	S'0,1	S'0,2	S'0,3
S'1,0	S'1,1	S'1,2	S'1,3
S'2,0	S'2,1	S'2,2	S'2,3
S'3,0	S'3,1	S'3,2	S'3,3

Figure 11: Structure of AddRound Key

1.4.5 AES Key Expansion

The AES key expansion algorithm is applied for generating the round key. This algorithm takes as input a four-word (16-byte) key and generates a linear array of 44 words (176 bytes). The first block of the AES key expansion is shown in the figure 12. It shows each group of 4 bytes in the key being assigned to the first 4 words, then the calculation of the next 4 words based on the values of the previous 4 words, which is repeated enough times to create all the necessary subkey information. The first word of the next round is calculated by using the function called f(g). Pseudo code for AES Expansion is given in figure 13. The key-expansion routine creates round keys word by word, where a word is an array of four bytes. The routines creates 4x(Nr+1) words. For Nk=4 words, Nr=10; this routine creates 44 words.

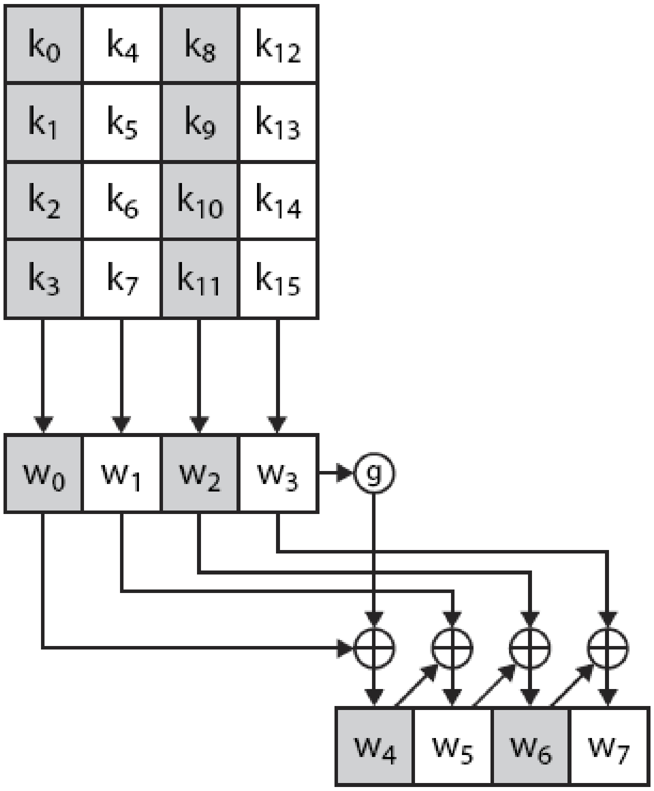
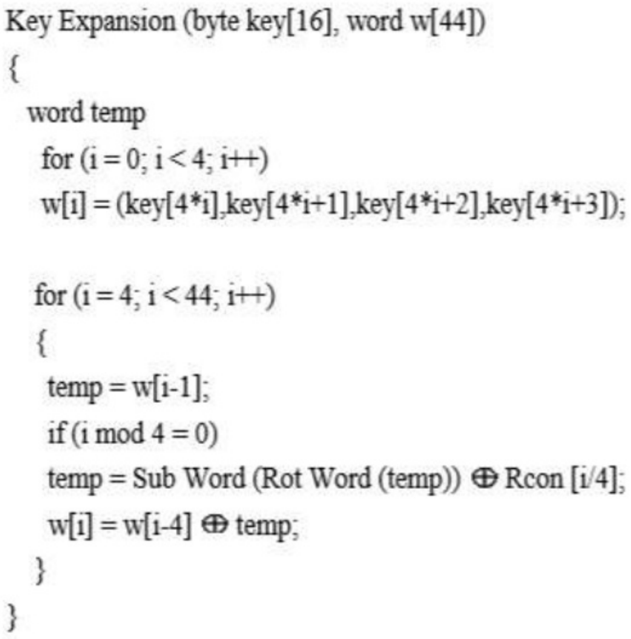


Figure 12: First round of the Expansion Key Algorithm



```
Key Expansion (byte key[16], word w[44])
{
  word temp
    for (i = 0; i < 4; i++)
    w[i] = (key[4*i],key[4*i+1],key[4*i+2],key[4*i+3]);

   for (i = 4; i < 44; i++)
   {
     temp = w[i-1];
     if (i mod 4 = 0)
     temp = Sub Word (Rot Word (temp)) ⊕ Rcon [i/4];
     w[i] = w[i-4] ⊕ temp;
   }
}
```

Figure 13: Pseudo code for AES key expansion.

Steps to find Wi when (i mod 4)=0

i. RotWord: perform one byte circular shift on Wi-j.

ii. SubWord :performs a byte substitution on each byte of its input word, using the S-Box.

iii. Thr result of step (i) and (ii) is XORed witha round constant Rcon[j] which is given by, Rcon[j]={RC[j],0,0,0}, where RC[j]=2*RC[j-1], with multiplication over GF(2^8).

J	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

Table 3: RC[j] values in Hex.

1.5 Chaotic Logistic Map

The relationship between cryptography and chaos are increasing day by day [4]. According to some of the properties of chaotic systems such as: sensitivity to initial conditions/system parameters, mixing property, deterministic dynamics and structural complexity can be considered analogous to the confusion, diffusion with small change in plaintext/secret key, diffusion with a small change within one block of the plaintext, deterministic pseudo randomness and algorithmic complexity properties of traditional cryptosystems [5]. As a result of this close relationship several chaos-based cryptosystems have been put forward since 1990. At present chaotic theory has been widely used in many fields, such as cryptography, mathematics, chemistry, biography, and so on. The most common used chaotic map is Logistic Map and it is given as $x_{n+1} = f(x_n) = rx_n(1 - x_n)$ Where x_n is a state variable, which lies in the interval [0,1] and r is called system parameter, which can have any value between 1 and 4.The system parameter value is calculate with the help of the bifurcation diagram as shown in figure 14 [7],[8],[9],[10],[11].

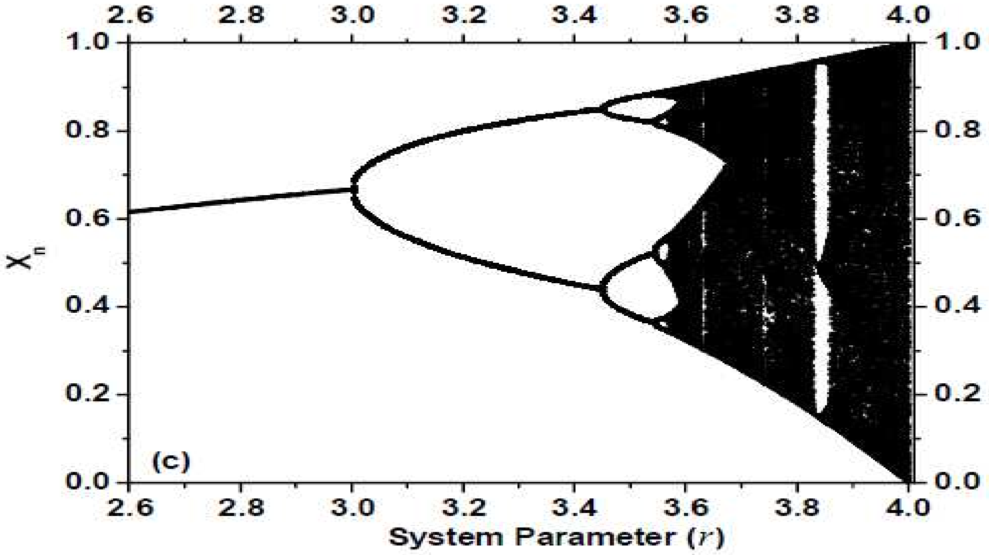


Figure 14: Behaviour of logistic map

CHAPTER 2

LITERATURE SURVEY

In this paper titled "A block encryption algorithm combined with the Logistic mapping and SPN structure" [1], researchers Huang, Jian-Hua, and Yang Liu had proposed a new block encryption algorithm by adding one dimensional discrete Chaotic Logistic Map (CLM) and SPN structure. Logistic map had been used for designing the S-Boxes and the P-Box and also the same logistic map had been used for generating the sub-keys. In this approach the sub-key had been created by performing an EX - OR operation between the user master key and the pseudo random sequence generated from chaos function. The one dimensional chaotic logistic map had been used as $x_{i+1} = ux_i(1 - x_i)$, where x_0 is the initial parameter and u is the bifurcation value i.e. u=4. The entire process of encryption contained 8 number of sub-keys, an initial transformation process, the transformation process with 8 rounds and a final output transformation process. The proposed algorithm contained the big key space, sensitivity to the key, strict avalanche effect, and well confusion and diffusion effect.

In this paper titled "An improved chaos-based stream cipher algorithm" [2], researchers Fu,Chong, and Zhi-Liang Zhu had proposed an improved chaos-based stream cipher algorithm to increase the security of an encryption system under limited conditions. With the help of three one dimension chaotic maps and their nonlinear transform, the key stream generator was formed. The balance and correlation properties of the key stream had been analyzed very effectively. The proposed algorithm also increased the key space, extended the period and improved the linear complexity of the key stream under precision restricted condition. The theoretical and practical results of the proposed algorithm provided huge anti attack ability against the adaptive parameter synchronization (APS) methods, which increased the security of the chaos based encryption system (CBES).

In this paper titled "Design and FPGA Implementation of a Pseudo Random Bit Generator Using Chaotic Maps" [3], researchers Khanzadi, Himan, Mohammad Eshghi, and Shahram Etemadi Borujeni had introduced a random bit sequence generator based on chaotic maps. In this method two chaotic map functions with two different keys were used. To calculate the initial state of the chaotic map, the Bifurcation diagram was used.

To generate the random bit sequence the initial state value was used and the output bits of two chaotic maps were EX-ORed. A computer simulation was used to justify the correctness of the algorithm. Finally, the proposed method was successfully passed all NIST and FIPS tests.

In this paper titled “Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography” [4], researchers Shujun, Li, Mou Xuanqin, and Cai Yuanlong presented a novel pseudo-random bit generator based on a couple of chaotic systems called CCS-PRBG to solve the problems in the digital Chaotic ciphers such as discrete dynamics, employed chaotic systems, encryption speed, practical security and realization. Some Cryptographic Properties of Digital CCS-PRBG such as balance on, long cycle-length, high linear complexity approximating to half of the cycle-length, δ-like auto-correlation, cross-correlation near to zero, chaotic-system-free were discussed. CSS-PRBG had the strong cryptographic properties and it was used to produce the stream ciphers with robust and large security than the chaotic ciphers.

In this paper titled “A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing” [5], researchers Patidar, Vinod, Krishan K. Sud, and Narendra K. Pareek had proposed a novel pseudo random bit generator (PRBG) based on two chaotic logistic maps iterated independently which were running side-by-side and initiating from random independent random initial conditions called a seed. There were two starting initial condition called seed given to the two logistic maps (X0, Y0 ∈[0, 1] and X0≠Y0). The outputs of both the chaotic logistic maps were compared which were generated from the pseudo random bit sequence. To generate the N-bit sequence continuously the logistic maps were iterated N times. The generated sequences were tested thoroughly using the NIST suite. NIST suite had sixteen independent statistical tests; work out to notice the exact characteristics expected of truly random bit sequences (TRBS). The results showed that the proposed PRBG had perfect cryptographic properties and it could be applied in the creation of new stream ciphers.

In this paper titled “A New Pseudo-Random Number Generator Based on Two Chaotic Maps” [6], researchers Francois, Michael, et. al. had proposed a new pseudo-random number generator (PRNG) using two chaotic maps to generate multiple key sequences. The rule of the proposed method was grounded on the structure of two chaotic maps produced by shuffling the position and permuting of an initial vector. The shuffling positions were calculated and indexed by a chaotic function based on linear congruence. The attacks like Distinguishing Attacks, Guess-and-Determine Attack, Differential Attack were analyzed. The advantages of the proposed PRNG were the adaptive key space size, the sensitivity to the initial inputs (keys), the quality of pseudo-random sequences, the security level against several attacks, the simplicity of implementation and the architecture portability.

In the paper titled “Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps” [7], researchers Jakimoski, Goce, and Ljupco Kocarev had proposed a class of block encryption ciphers based on chaotic maps. Two well-known chaotic maps were used, exponential x→a^x mod1 and logistic x→4x (1-x). Ciphers had satisfactory values of linear and differential approximation probabilities that were produced by map. The other way of generating S-boxes by using chaotic maps was also proposed.

In this paper titled “Implementation of AES and RSA using chaos system”[8], researchers Agrawal, Bhavana, and Himani Agrawal had proposed two cryptographic algorithms AES and RSA using chaos. The chaos system had been applied in AES to generate S-Box where as in RSA the chaos sequence was first mixed with the plaintext and then encryption and decryption was applied. A one dimensional chaotic map was used. This chaos based AES and RSA made the system fast and more complex as compared to conventional AES and RSA. The MATLAB tool was used for simulating the chaos AES and RSA, the result showed that by applying chaotic maps, speed of the system increased.

In this paper titled "Efficient and simple method for designing chaotic S-boxes" [9], researchers Asim, Muhammad, and Varun Jeoti proposed a four-step method based on the mixing property on the Piecewise Linear Chaotic Map (PCLM) for designing the S-Box in AES. The PLCM equation used was

$$F(x,p) = \begin{cases} x/p & 0 \le x \le p \\ (x-p)/\left(\frac{1}{2}-p\right), & p \le x \le \frac{1}{2} \\ F(1-x,p), & \frac{1}{2} < x \le 1 \end{cases}$$

Where $0 < p < 1/2$, x was a preliminary condition and p was the map F control parameter. In this method the PLCM equation iterated until all the possible values of S-Box were filled with the control parameter p=0.15 and Initial Condition x=0.76. The

Differential approximation probability (DAP) and linear approximation probability (LAP) were analyzed. The LAP of S-Box organized by this method was less than conventional AES S-Box. Although the DAP was comparatively higher.

In this paper titled “A Novel Method for Designing S-Boxes Based on Chaotic Logistic Maps Using Cipher Key” [10], researchers Mona Dara and Kooroush Manochehri had proposed a new method based on chaotic logistic map. The logistic map had been used to generate S-box for AES using its cipher key. The proposed system was to produce secure S-Box. The proposed method didn’t use the initial value and system condition in a logistic map. But the logistic map uses the cipher key to generate initial value (x_0). The dynamic S-box was generated by iterating the logistic map. The generated S-box had increased the complexity and had better result in security analysis. In the proposed algorithm, lot of S-boxes were produced by using change in cipher key and that was the main advantage of the proposed system. The generated S-boxes were analyzed and also passed all of the important criteria such as avalanche effect, strict avalanche effect, bit independent criterion, nonlinearity and key sensitivity. The generated S-box was more secure and robust in favour of attacks such as differential and linear cryptanalysis.

In this paper titled “Implementation of strong AES by using dynamic S-Box dependent of master key”[11], researchers Khamlich, Eddine had proposed an algorithm for designing the dynamic S-Box which depended on master key. The main advantages of this proposed algorithm was S-Box and the key expansions were dependent on entire initial key, so that the AES would be much stronger. A one dimension chaotic logistic map was used to generate the dynamic S-Box in AES. The initial value (x_0) of the logistic map ($x_{i+1} = ux_i(1 - x_i)$) was calculated by using the cipher key length. A dynamic S-Box was generated as the XOR operations performed between the one byte of master key with the each byte of the conventional AES S-Box. The round keys were used for finding a value which was used to spin the S-Box in each and every round, this process had taken more time. This proposed modification of AES and modified S-Box were created in a Cyclone II device with the help of VHLD language.

In this paper titled “Key-Dependent S-Box Generation in AES Block Cipher System” [12], researchers Kazlauskas, Kazys, and Jaunius Kazlauskas had presented a new approach for generating random S-Box which was depended on the secrete key. In this approach for every change of secret key a new random S-Box was generated. A random key-dependent S-Box and inverse S-Box was generated by using two transformations that were, first one was key pseudo-expansion and the other was key-dependent S-Box and inverse S-Box. In key pseudo-expansion, transformation the secret key was used. The efficiency of this method wass tested by the changing one bit of the secret key to produce new S-Box. Researchers also introduced the independent measure ratio based on correlation method. The advantage of this proposed algorithm was a vast number of S-Boxes were generated by altering secret key.

In this report titled “Dynamic AES-128 with Key-Dependent S-box”[13] researchers Mahmoud, Eman Mohammed, et al had proposed a new method to design dynamic S-box which depends on the cipher key. The private key were calculated and then serves to PN generator as initial state. The PN generator and AES secret key were used to produce sequence of 16 hexadecimal values and these sequences were employed to arrange S-box vectors. The proposed algorithm leads to boost the complexity and difficult to make the linear and differential cryptanalysis. In order to achieve secure communications this algorithm is suitable to exchange the keys on insecure communication channels. In this method if any change in cipher key, the organization of the s-box will be change fundamentally. Security analysis test such as Randomness tests, Avalanche effect, Correlation factor and Simulation time, for quantifying the durability of the dynamic AES-128 with key dependent S-box had been applied. The operation of this method was tested, for that purpose the correlation ratio between standard AES and modified one was calculated.

In this paper titled “Performance Comparison of AES and AES key dependent S-Box Simulation using MATLAB”[14] researchers Shivkumar, S., and G. Umamaheswari had presented AES-RC4 method that uses the key scheduling algorithm to produce key dependent S-Box. The output of this algorithm produces 256 values which were depended on input key. Dissimilar round keys were producing by using RC4 key expansion algorithm. The round keys were used for generating a value and it was used to spin the S-Box in each and every round. The strength of the AES-RC4 S-Box was tested using the parameters such as avalanche criteria, randomness test and bit independence criteria (BIC). The result showed that the security of the AES was improved.

CHAPTER 3

PRESENT WORK

3.1 Problem Formulation

On the basis of above literature survey some problem have been identified, which have been left by the researchers in the previous research work. Substitution Boxe (S-Box) is the major component of any cryptography algorithm. The main issue is that the S-Box which is used in many cryptography algorithm is static, and this will leads an bruteforce attack. Logistic chaotic maps had beed used to desiging the dynamic S-Box, but the issue is finding the initial parameter of logistic map. Another main issue is that simulation time for desiging the Dynamic S-Box is very complex and it takes more time. A new method has been proposed to overcome the problems left by the researchers.

3.2 Objective

On the basis of above literature survey some problem have been identified, which have been left by the researchers in the previous research work. A new method has been proposed to overcome the problems left by the researchers

- To design a key dependent dynamic AES S-Box by using two chaotic logistic maps based Pseudo Random Number Generator (PRNG), which has been increasing the security level of AES algorithm.
- To create a key dependent S-Box where the secret key supplied by the user would be used for calculating the initial condition (X0 and Y0) of the logistic maps.
- To enhace the key sensitivity to the S-Box such that by changing the one bit of secret key a different dynamic S-Box would be generated.
- To increase the avalanche effect and to minimize the simulation time taken for desiging dynamic S-Box.

3.3 Methodology

AES is a block cipher algorithm which is widely used in many applications. The key idea of the methodology would be to design a key dependent AES S-Box which uses two chaotic logistic maps based Pseudo Random Number Generator (PRNG) [5]. The major steps used in AES would be the “Substitute Byte Transformation”. This function would perform a non-linear substitution, which would be performed independently on each input byte. The 16×16 matrix used in the AES would be called S-Box and meet the following criteria: key sensitivity, non-linearity, avalanche effect, order diffusion, the dynamic criteria. Therefore, to meet the above requirements a new key dependent dynamic S-Box would be designed**.** The structure of the proposed AES would be same as the conventional AES structure, but in proposed AES structure the Substitution box (S-Box) would be dynamic. Pseudo Random Number Generator (PRNG) based on two Chaotic Logistic map would be used to generate S-Box for AES using its secret key [11]. The initial parameter values X_0 and Y_0 of logistic map would be calculated from the secret key. In our proposed method the whole implementation is same as the conventional AES except the Substitution S-Box. Figure 11 shows the structure of our proposed method. Pseudo Random Number Generator (PRNG) has been used to generate the dynamic S-Box. Secret key has been used to calculate the values of initial parameter X_0 and Y_0 of logistic map which is act as a seed values for PRNG. Random number is generated by PRNG by iterating the chaotic logistic map. The output of the PRNG has been used to generate the S-Box.

3.3.1 The proposed key dependent dynamic AES-128 S-BOX

The below mentioned pseudo code is used for generating the Key dependent dynamic AES S-box:

1. Initialization: The entries of S-Box are set equal to the values from 0 through 255 in ascending order; that is S[1] = 0, S[2] = 1, . . . , S[256] = 255. Thus, the operations can be written as;

```
for j=1:256
    s_box(j)= j-1;
end
```

2. Calculate initial parameters value X0 and Y0 of chaotic logistic map: The method uses AES secret key (K), that has 128/192/256 bits length , the value X0 and Y0 is calculated as follows

$$\mathrm{K} = \mathrm{K}_1, \mathrm{K}_2, \dots, \mathrm{K}_{32} (\mathrm{Hexadecimal}) \text{for 128 bit key}$$

$$\mathrm{K} = \mathrm{K}_1, \mathrm{K}_2, \dots, \mathrm{K}_{48} (\mathrm{Hexadecimal}) \text{for 192 bit key}$$

$$\mathrm{K} = \mathrm{K}_1, \mathrm{K}_2, \dots, \mathrm{K}_{64} (\mathrm{Hexadecimal}) \text{for 256 bit key}$$

$$\mathrm{X}_0 = \mathrm{K}_1 \times 2^{(\mathrm{keysize}/2)-1} + \mathrm{K}_3 \times 2^{(\mathrm{keysize}/2)-2} + \dots + \mathrm{K}_{(\mathrm{keysize}-1)} \times 2^0$$

$$\mathrm{Y}_0 = \mathrm{K}_2 \times 2^0 + \mathrm{K}_4 \times 2^1 + \dots + \mathrm{K}_{\mathrm{keysize}} \times 2^{(\mathrm{keysize}/2)-1}$$

Here, Ki is the part of secret key in hexadecimal mode and '*keysize*' is the length of the key also in hexadecimal mode.

3. Random number Generator: The random number is generated by iterating the logistic maps.

$$X_{n+1} = uX_n(1 - X_n)$$
$$Y_{n+1} = uY_n(1 - Y_n)$$

4. Now we use random number to generate the initial permutation of S-Box. This involves starting with S[1] and going through S[256] and for each S[i], swapping S[i] with S[rand_num]. This operation is shown below

```
for i=1:256
    rand_num = rand_num_gen();
     if(rand_num==0)
         rand_num=1;
     end
     swap(S[j],S[rand_num]);
end
```

The flow chart of the entire process of designing of dynamic S-Box using PRNG based on two chaotic logistic maps is shown in the figure15.

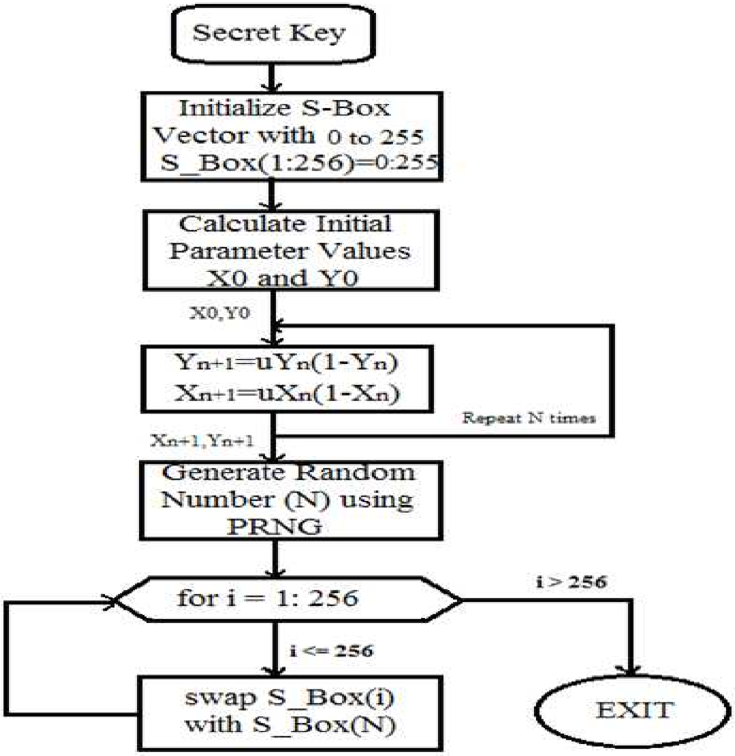

Secret Key
Initialize S-Box Vector with 0 to 255 S_Box(1:256)=0:255
Calculate Initial Parameter Values X0 and Y0
X0,Y0
Yn+1=uYn(1-Yn)
Xn+1=uXn(1-Xn)
Repeat N times
Xn+1,Yn+1
Generate Random Number (N) using PRNG
for i = 1: 256
i > 256
i <= 256
swap S_Box(i) with S_Box(N)
EXIT

Figure 15: Flow chart for Dynamic S-Box Generator

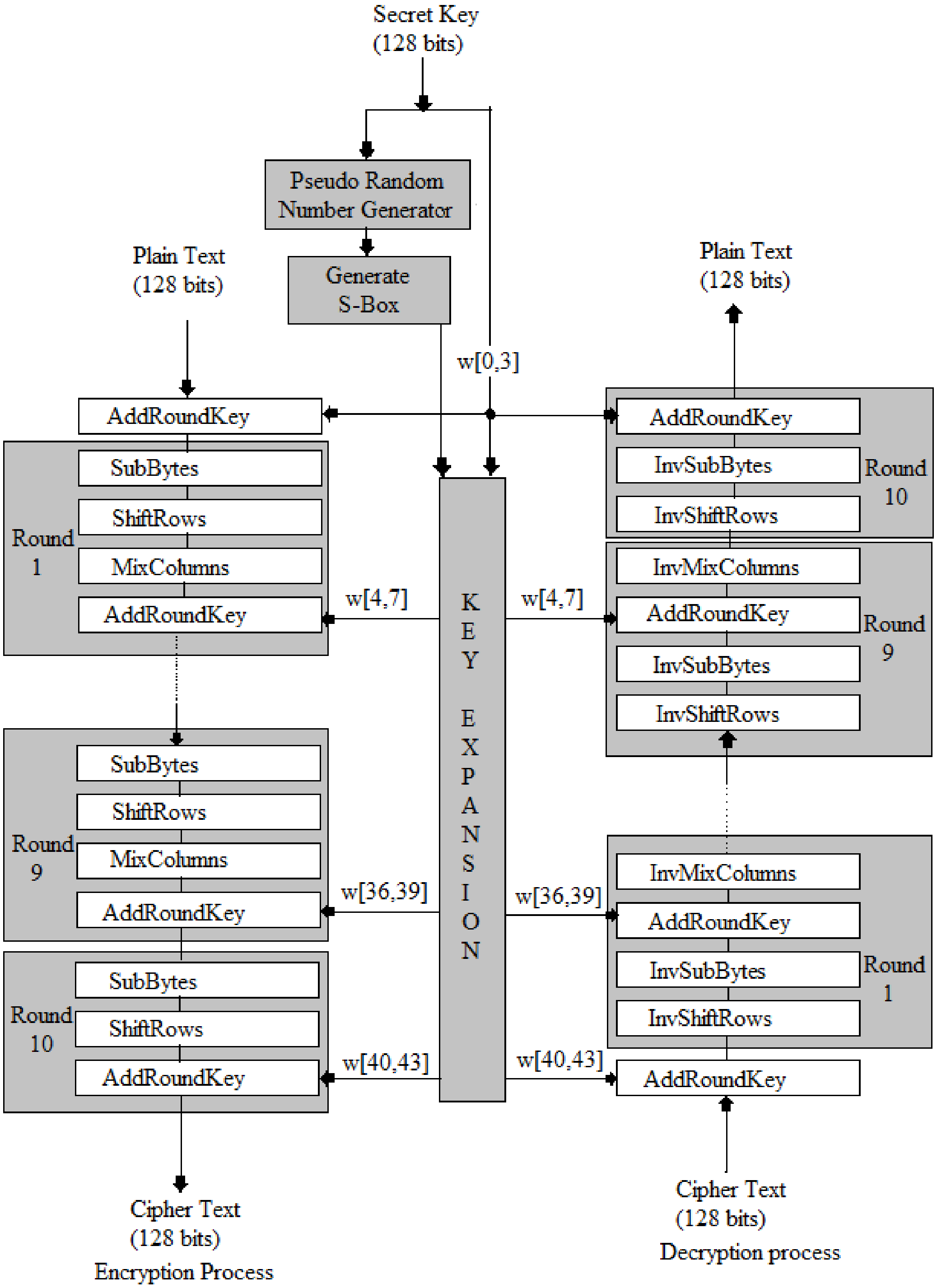


Figure 16: Structure of Key dependent AES S-Box

3.3.2 Calculation of initial parameters using Key

The values of the initial parameter $X_0\ and\ Y_0$ are calculated by using the AES secret key supplied by the user. Initially the secret key is divided in to two parts, first part is used for calculating X_0 value and the second part is used for calculating Y_0. Before calculating the initial parameter values the secret key is converted in to hexadecimal mode.

$$\mathrm{K} = \mathrm{K}_1, \mathrm{K}_2, \dots, \mathrm{K}_{32} (\text{Hexadecimal}) \text{for 128 bit key}$$
$$\mathrm{K} = \mathrm{K}_1, \mathrm{K}_2, \dots, \mathrm{K}_{48} (\text{Hexadecimal}) \text{for 192 bit key}$$
$$\mathrm{K} = \mathrm{K}_1, \mathrm{K}_2, \dots, \mathrm{K}_{64} (\text{Hexadecimal}) \text{for 256 bit key}$$

Here K_i is the part of the key in hexadecimal mode. Next the real numbers $X_0\ and\ Y_0$ values are calculated as follows:

$$\mathrm{X}_0 = \mathrm{K}_1 \times 2^{\mathrm{keysize}/2-1} + \mathrm{K}_3 \times 2^{\mathrm{keysize}/2-2} + \cdots + \mathrm{K}_{\mathrm{keysize}-1} \times 2^0$$
$$\mathrm{Y}_0 = \mathrm{K}_2 \times 2^0 + \mathrm{K}_4 \times 2^1 + \cdots + \mathrm{K}_{\mathrm{keysize}} \times 2^{\mathrm{keysize}/2-1}$$

Where K_i defines the i^th bit of secret key and *keysize* is the size of the key in hexadecimal mode. The calculated values of X_0 and Y_0 are in decimal number then change them into the real numbers from the decimal because X_0 and Y_0 lies between 0 and 1.

Example:

Consider the AES cipher key : “LOVELYUNIVERSITY” convert this cipher key into hexadecimal format, now the cipher key written in hexadecimal format as follows

Cipher key: 4C 4F 56 45 4C 59 55 4E 49 56 45 52 53 49 54 59

Now the keysize of the cipher key is 32 in hexadecimal format. Now calculate the initial parameter values X0 and Y0 by using the above equation.

Calculating X0 value

$$\mathrm{X}_0 = \mathrm{K}_1 \times 2^{\mathrm{keysize}/2-1} + \mathrm{K}_3 \times 2^{\mathrm{keysize}/2-2} + \cdots + \mathrm{K}_{\mathrm{keysize}-1} \times 2^0$$
$$\mathrm{X}_0 = 4 \times 2^{32/2-1} + 4 \times 2^{32/2-2} + 5 \times 2^{32/2-3} + \cdots + 5 \times 2^0$$
$$\mathrm{X}_0 = 4 \times 2^{15} + 4 \times 2^{14} + 5 \times 2^{13} + \cdots + 5 \times 2^0$$
$$\mathrm{X}_0 = 3417639$$

Calculating Y0 value

$$\mathrm{Y}_0 = \mathrm{K}_2 \times 2^0 + \mathrm{K}_4 \times 2^1 + \cdots + \mathrm{K}_{\mathrm{keysize}} \times 2^{\mathrm{keysize}/2-1}$$
$$\mathrm{Y}_0 = \mathrm{C} \times 2^0 + \mathrm{F} \times 2^1 + 6 \times 2^2 + \cdots + 9 \times 2^{32/2-1}$$

$$\mathrm{Y}_0 = \mathrm{C} \times 2^0 + \mathrm{F} \times 2^1 + 6 \times 2^2 + \cdots + 5 \times 2^{15}$$
$$\mathrm{Y}_0 = 3610463$$

Convert both the decimal values in to real number, now the X0 and Y0 value is changed to 0.3417639 and 0.3610463 respectively.

3.3.3 The proposed Pseudo Random Number Generator (PRNG)

The dynamic S-Box would be generated by using the Pseudo Random Number Generator (PRNG) based on two Chaotic Logistic map (CLM). In PRNG, it would be using two chaotic logistic map which would be running side-by-side. The initial parameter of CLM would be calculated from the secret key which is act as a seed value for PRNG. Inside the PRNG the two chaotic logistic map which is running side-by-side which uses the X_0 and Y_0. The values of X_0 and Y_0 are real numbers, every time logistic map generate real numbers, but the output of the PRNG should be an integer number and to achive this both the chaotic logistic map has to be iterating eight times in order to get the eight bit binary sequence. The 8-bit binary sequence is converted into decimal number. To generate the next integer random number this whole process is repeated. The output of the PRNG that is random number lies between 0 and 255. This generated random number has been use to generate the initial permutation of S-Box. The two chaotic equations used in the PRNG are given as.

$$X_{n+1} = uX_n(1 - X_n)$$
$$Y_{n+1} = uY_n(1 - Y_n)$$

Where X_0, Y_0 are the initial parameter and u is the control parameter, u=4, (X_0,Y_0) $\in (0,1)$ and $X_0 \neq Y_0$. The output bit sequence is generated by comparing the outputs of both the chaotic logistic maps in the following way:

$$g(X_{n+1}, Y_{n+1}) = \begin{Bmatrix} 1 \ \ if\ X_{n+1} > Y_{n+1} \\ 0 \ \ if \ \ X_{n+1} \leq Y_{n+1} \end{Bmatrix}$$

The set of initial parameter (X_0 Y_0)$\in (0,1)$ and $X_0 \neq Y_0$ serves as the seed for the PRNG, if we supply the exactly same seed to the PRNG, it will generate the same bit sequence [5].

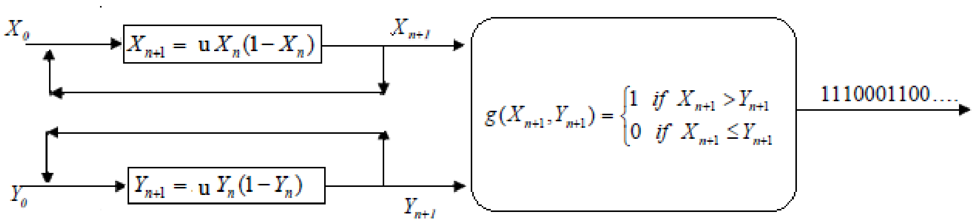
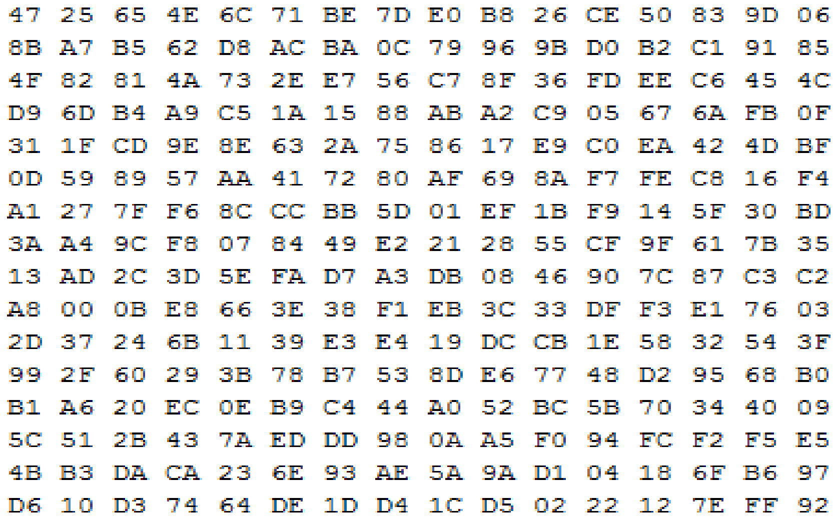

X_0
$X_{n+1} = uX_n(1-X_n)$
X_{n+1}
$g(X_{n+1}, Y_{n+1}) = \begin{cases} 1 & \text{if } X_{n+1} > Y_{n+1} \\ 0 & \text{if } X_{n+1} \le Y_{n+1} \end{cases}$
1110001100....
Y_0
$Y_{n+1} = uY_n(1-Y_n)$
Y_{n+1}

Figure 17: Pseudo Random Number Generator

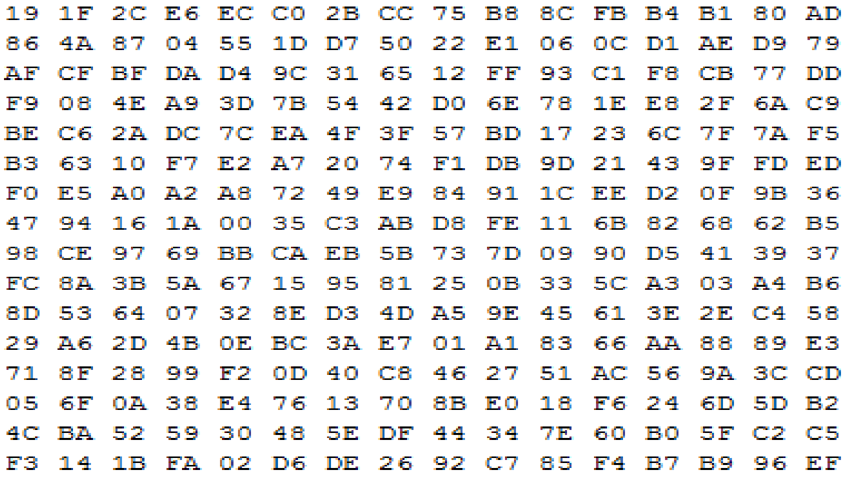
Table represent the proposed key dependent dynamic S-box and its corresponding inverse S-box respectively when secret key is equal to:

"0F 15 71 C9 47 D9 E8 59 0C B7 AD D6 AF 7F 67 98"



47	25	65	4E	6C	71	BE	7D	E0	B8	26	CE	50	83	9D	06
8B	A7	B5	62	D8	AC	BA	0C	79	96	9B	D0	B2	C1	91	85
4F	82	81	4A	73	2E	E7	56	C7	8F	36	FD	EE	C6	45	4C
D9	6D	B4	A9	C5	1A	15	88	AB	A2	C9	05	67	6A	FB	0F
31	1F	CD	9E	8E	63	2A	75	86	17	E9	C0	EA	42	4D	BF
0D	59	89	57	AA	41	72	80	AF	69	8A	F7	FE	C8	16	F4
A1	27	7F	F6	8C	CC	BB	5D	01	EF	1B	F9	14	5F	30	BD
3A	A4	9C	F8	07	84	49	E2	21	28	55	CF	9F	61	7B	35
13	AD	2C	3D	5E	FA	D7	A3	DB	08	46	90	7C	87	C3	C2
A8	00	0B	E8	66	3E	38	F1	EB	3C	33	DF	F3	E1	76	03
2D	37	24	6B	11	39	E3	E4	19	DC	CB	1E	58	32	54	3F
99	2F	60	29	3B	78	B7	53	8D	E6	77	48	D2	95	68	B0
B1	A6	20	EC	0E	B9	C4	44	A0	52	BC	5B	70	34	40	09
5C	51	2B	43	7A	ED	DD	98	0A	A5	F0	94	FC	F2	F5	E5
4B	B3	DA	CA	23	6E	93	AE	5A	9A	D1	04	18	6F	B6	97
D6	10	D3	74	64	DE	1D	D4	1C	D5	02	22	12	7E	FF	92

Table 4: Key dependent dynamic S_Box



19	1F	2C	E6	EC	C0	2B	CC	75	B8	8C	FB	B4	B1	80	AD
86	4A	87	04	55	1D	D7	50	22	E1	06	0C	D1	AE	D9	79
AF	CF	BF	DA	D4	9C	31	65	12	FF	93	C1	F8	CB	77	DD
F9	08	4E	A9	3D	7B	54	42	D0	6E	78	1E	E8	2F	6A	C9
BE	C6	2A	DC	7C	EA	4F	3F	57	BD	17	23	6C	7F	7A	F5
B3	63	10	F7	E2	A7	20	74	F1	DB	9D	21	43	9F	FD	ED
F0	E5	A0	A2	A8	72	49	E9	84	91	1C	EE	D2	0F	9B	36
47	94	16	1A	00	35	C3	AB	D8	FE	11	6B	82	68	62	B5
98	CE	97	69	BB	CA	EB	5B	73	7D	09	90	D5	41	39	37
FC	8A	3B	5A	67	15	95	81	25	0B	33	5C	A3	03	A4	B6
8D	53	64	07	32	8E	D3	4D	A5	9E	45	61	3E	2E	C4	58
29	A6	2D	4B	0E	BC	3A	E7	01	A1	83	66	AA	88	89	E3
71	8F	28	99	F2	0D	40	C8	46	27	51	AC	56	9A	3C	CD
05	6F	0A	38	E4	76	13	70	8B	E0	18	F6	24	6D	5D	B2
4C	BA	52	59	30	48	5E	DF	44	34	7E	60	B0	5F	C2	C5
F3	14	1B	FA	02	D6	DE	26	92	C7	85	F4	B7	B9	96	EF

Table 5: Key dependent dynamic Inverse S_Box

CHAPTER 4

RESULTS AND DISCUSSIONS

To simulate the key dependent dynamic AES S-box, in MATLAB (2010 version) script has been implemented for both AES and dynamic AES. In order to ensure that implemented key dependent dynamic AES S-Box is effective, some cryptographic tests have been applied such as, Avalanche effect, Key sensitivity, simulation time of S-Box etc, to check the strength of the S-box in comparison to its earlier implementation. A careful analysis has been performed to check the all parameters, the experimental results are shown as under:

4.1 Avalanche Effect :

It is an important property of any encryption algorithm. In general avalanche effect means that a small variation in either of the input plaintext or the cipher key should reflect many changes in the cipher texts. Any one bit change in the plaintext or ciphertext should change atleast half of the ciphertext.

4.1.1 Avalanche effect due to one bit change in plaintext

It is a most important design objective to apply avalanche effect on the implemented algorithm. Figure 18 shows the avalanche effect of the standard AES S-Box algorithm and the proposed dynamic AES with key dependent S-box respectively due to one bit change in plaintext. The graph shown in figure 19 has been generated when the plaintext and the cipher key is taken as follows

Plaintext : 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10

Cipher Key: 0F 15 71 C9 47 D9 E8 59 0C B7 AD D6 AF 7F 67 98

Bit location change in the plaintext has been taken in the X-Axis and the percentage in the avalanche effect due to one bit change in plaintext has been taken in the Y-Axis. Both the graphs clearly indicate that how much minimum and maximum percentage change in the avalanche effect is observed when one bit of the plaintext is modify. The standard AES algorithm shows that avalanche effect lies between 36% and 59%, where as our implemented algorithm shows that avalanche effect lies between 39% and 61%, which means that it is more difficult to perform cryptanalysis over the proposed method of encryption.

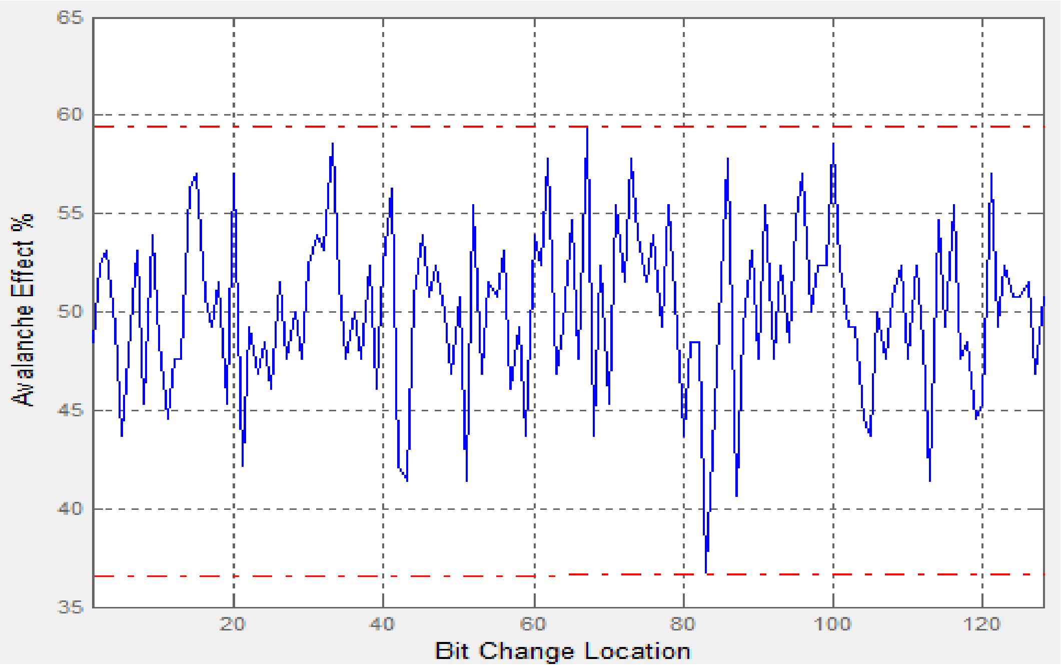


Figure 18: Avalanche effects of standard AES.

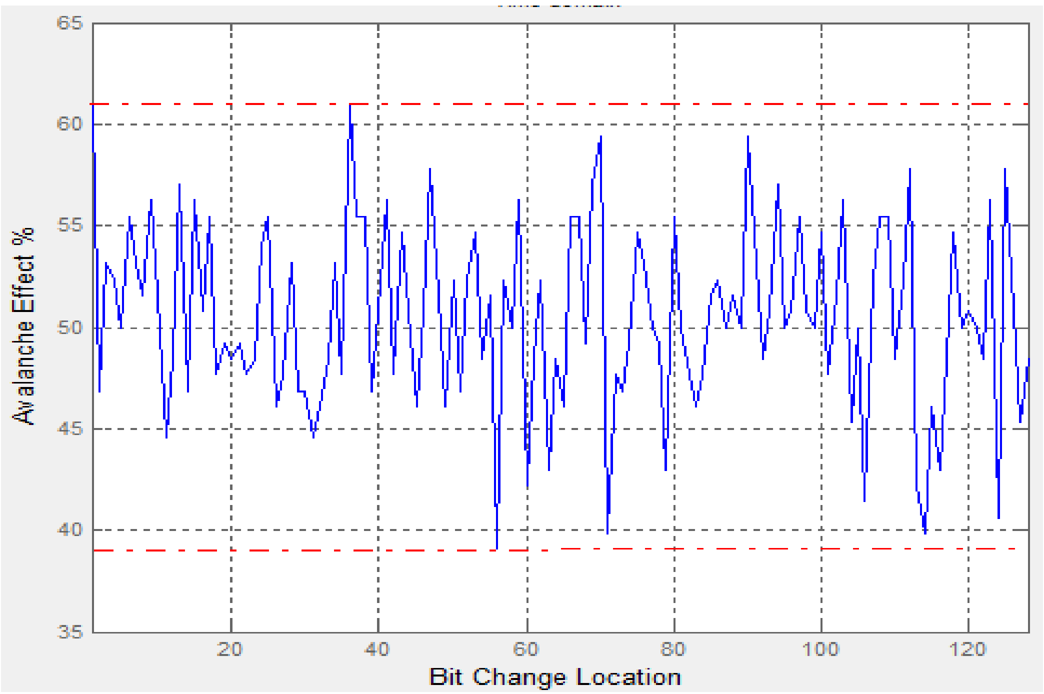


Figure 19: Avalanche effects of dynamic AES with key dependent S-box.

Table 6 Shows samples of the standard AES S-Box and the proposed AES-128 with Dynamic S-Box ciphertext due to one bit change in the plaintext.

Bit variation index	Plaintext	Ciphertext of standard AES	Ciphertext of dynamic AES
0	01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10	FF 0B 84 4A 08 53 BF 7C 69 34 AB 43 64 14 8F B9	7D AC 73 FE 38 1A D9 EB 8E 21 67 18 32 92 EB 2E
1	81 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10	E6 78 28 55 CA 75 23 88 98 D2 EF 9F 22 DE F9 B1	0C 07 0E 03 2F E3 81 EA 28 43 99 CD 4A 67 16 90
2	41 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10	C7 85 BE BA 57 80 69 DE 56 75 6B EC C1 A7 41 2D	E7 88 4B 0F 28 A7 B0 F5 E9 E8 67 77 A8 75 A9 84
4	11 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10	68 E8 8A 7A 56 31 7F 4D E4 25 C6 95 F9 BB 29 01	CC 1B A3 BC A5 5F 1D 14 9D 04 BE 7C 88 40 06 FA
8	00 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10	61 2B 89 39 8D 06 00 CD E1 16 22 7C E7 24 33 F0	69 3B F3 7B 60 3F 36 F6 20 B1 CE 6F FA 5D 51 F1
16	01 22 C5 67 89 AB CD EF FE DC BA 98 76 54 32 10	95 22 23 FA A6 18 A5 9D CA 9C A9 BD C9 CD 96 07	79 A8 1F 4D E3 D7 54 10 01 50 04 5D 12 69 23 E9
32	01 23 45 66 89 AB CD EF FE DC BA 98 76 54 32 10	AF 20 28 D5 80 B6 51 93 E3 F1 E5 95 3D 1E 3C D7	F0 D9 FC 7C B8 95 FE B0 B4 84 92 AA 82 C7 C3 26
64	01 23 45 67 89 AB CD EE FE DC BA 98 76 54 32 10	E7 9C 01 F9 34 3E 12 98 2D D7 3D 5E 8F 85 3F 30	8A AC 8B D2 83 2D 00 A3 88 2B 64 FE 41 87 06 FF
70	01 23 45 67 89 AB CD EF FA DC BA 98 76 54 32 10	88 5A 0C E0 29 F5 9F 9C 5A 05 92 4C 51 4B 21 71	3E 7F 12 23 CD 95 65 2E 86 C6 5C FB 4D 64 16 6C
76	01 23 45 67 89 AB CD EF FE CC BA 98 76 54 32 10	7A 10 83 F5 11 2B 35 90 B0 0F 6C 49 98 CC F6 12	8F CA B0 DF BC E0 87 55 76 2B 71 84 9C 4A 76 C9
100	01 23 45 67 89 AB CD EF FE DC BA 98 66 54 32 10	32 58 FD 21 FB 06 C3 07 99 C8 81 A7 1C DE 51 7A	06 C1 0A E2 E0 EC C3 A2 F3 15 B5 70 E9 1F D0 FA
126	01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 14	D3 57 5C 09 AE EE B3 97 B1 87 B6 25 FC 49 E8 0B	3D 78 E3 85 D4 86 2D 87 D0 08 70 85 6F 23 C2 CB
127	01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 12	A7 A6 DD 08 6F 88 1E 8E 46 3C CD 27 B8 01 EF 51	18 E8 B3 F1 FA D7 2D 60 95 89 3D 69 F7 F3 20 8E
128	01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 11	0E 99 9A FE CA 61 75 FC 25 0B 46 DA FE 79 3B 0E	CF 2C 71 3D F1 77 7C 8D A5 0B B7 E4 41 9F 41 51

Table 6: samples of Ciphertext due to one bit change in plaintext.

4.1.2 Avalanche effect due to one bit change in secret key

Figure 20 shows that how much variation is observed in the S-Boxes due to change in one bit of the secret key. The graph clearly represent that the percentage change in the different S-Boxes lies between 95.2 and 99.6.

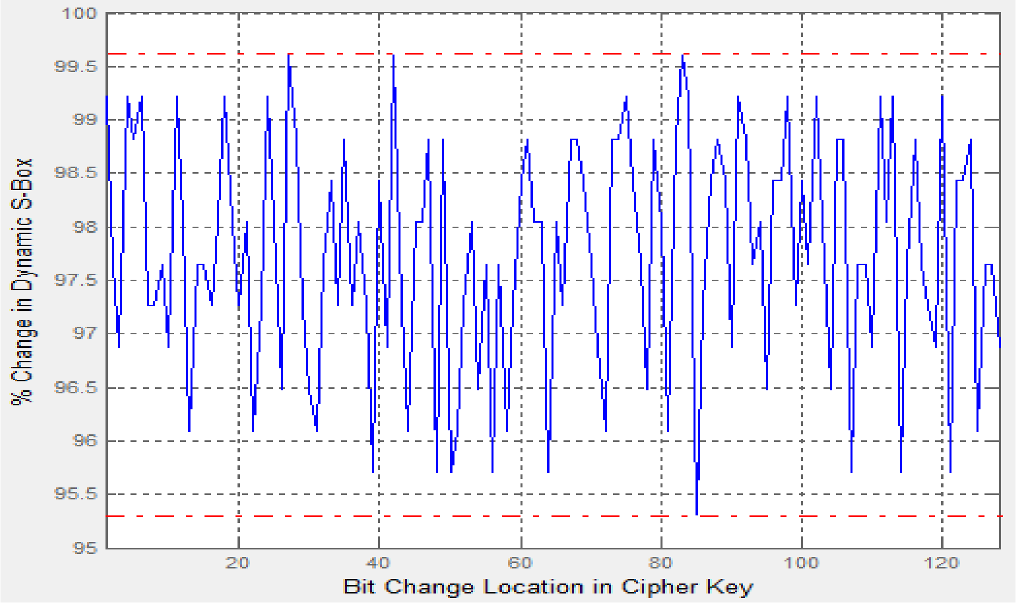


Figure 20: Percentage change in S-Boxes

4.1.3 Key Sensitivity Test:

The following two steps have been performed to test the Key Sensitivity.

First, the original image is encrypted by using the secret key (K1),

K1= " 1f'1571c947d9e8590cb7add6af7f6798 " (Hex).

Then, try to decrypt the encrypted image by using the one bit modified secret key (K2)

K2= " 0f1571c947d9e8590cb7add6af7f6798 " (Hex)

The strength of the algorithm is that even for a single bit change in the secret key value the image is not decrypted as shown in figure 21. Since different S-Boxes have been generated by changing one bit change in secret key. The difference of dynamic S-box1 and dynamic S-box2 elements is illustrated in Table 8 and 9. This clearly indicates that even when one bit of secret key is modified; it has its effect on all the pixels.

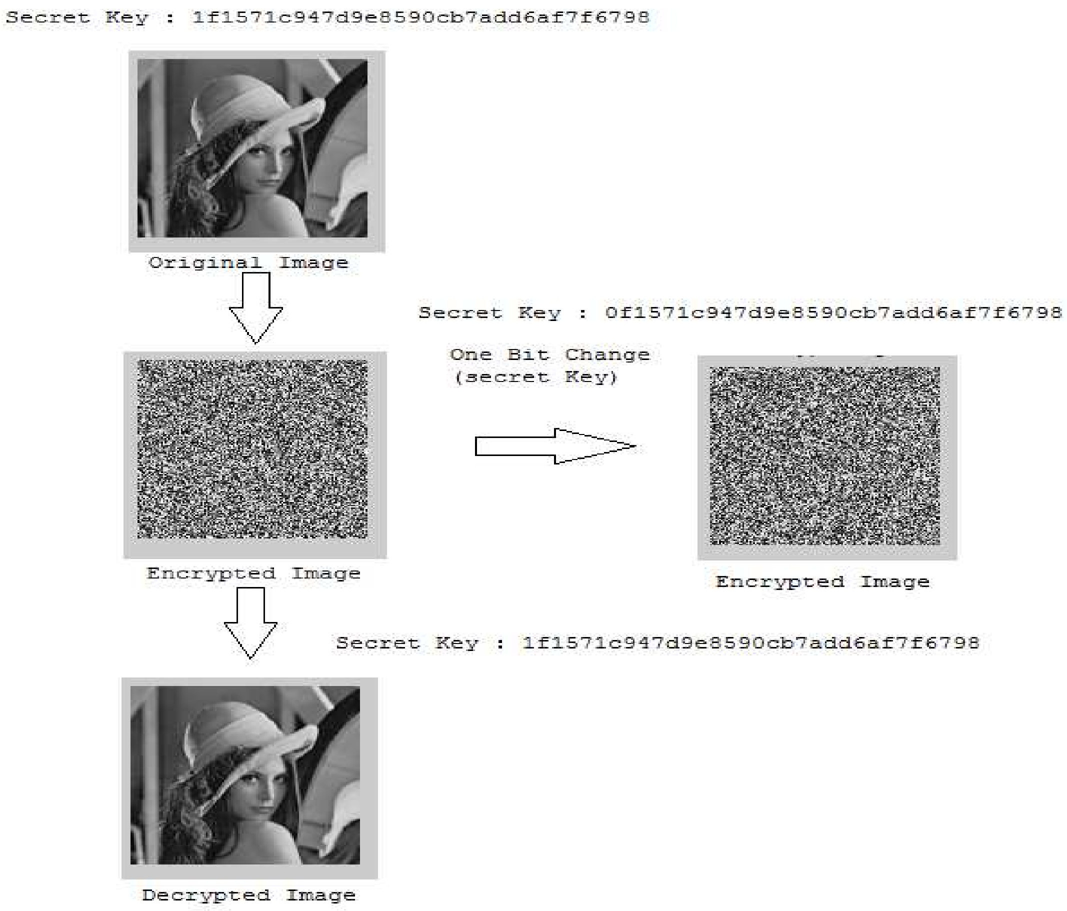
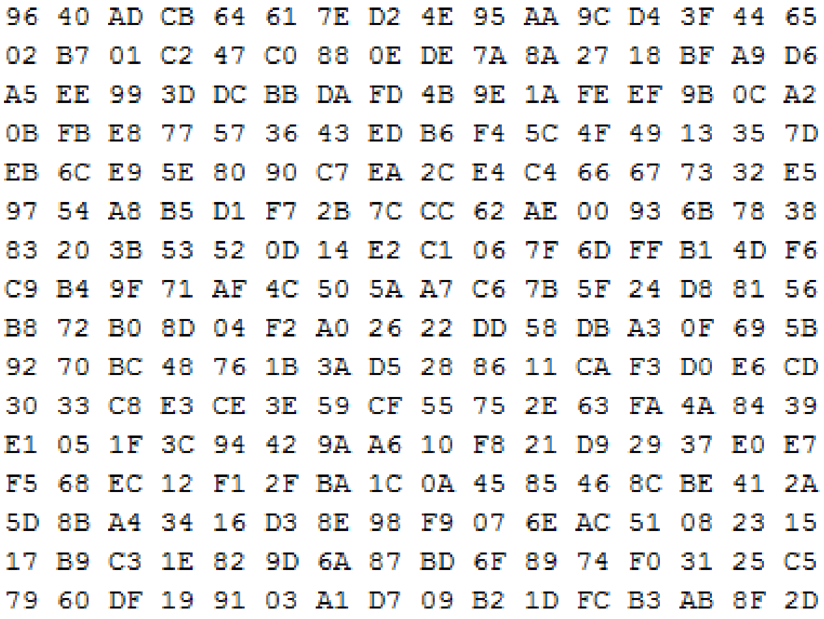
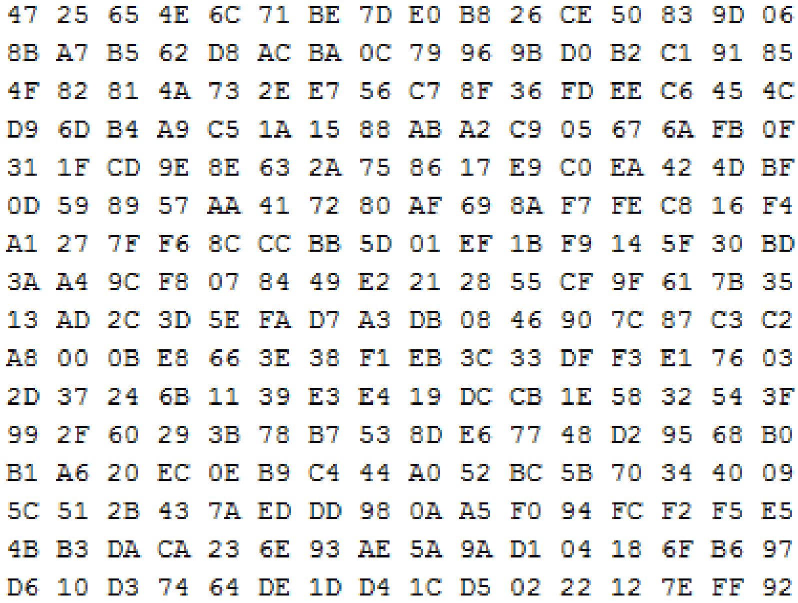

Secret Key : 1f1571c947d9e8590cb7add6af7f6798
Original Image
Secret Key : 0f1571c947d9e8590cb7add6af7f6798
One Bit Change
(secret Key)
Encrypted Image
Encrypted Image
Secret Key : 1f1571c947d9e8590cb7add6af7f6798
Decrypted Image

Figure 21: Two ciphered images, encrypted by the two slightly different keys.



96	40	AD	CB	64	61	7E	D2	4E	95	AA	9C	D4	3F	44	65
02	B7	01	C2	47	C0	88	0E	DE	7A	8A	27	18	BF	A9	D6
A5	EE	99	3D	DC	BB	DA	FD	4B	9E	1A	FE	EF	9B	0C	A2
0B	FB	E8	77	57	36	43	ED	B6	F4	5C	4F	49	13	35	7D
EB	6C	E9	5E	80	90	C7	EA	2C	E4	C4	66	67	73	32	E5
97	54	A8	B5	D1	F7	2B	7C	CC	62	AE	00	93	6B	78	38
83	20	3B	53	52	0D	14	E2	C1	06	7F	6D	FF	B1	4D	F6
C9	B4	9F	71	AF	4C	50	5A	A7	C6	7B	5F	24	D8	81	56
B8	72	B0	8D	04	F2	A0	26	22	DD	58	DB	A3	0F	69	5B
92	70	BC	48	76	1B	3A	D5	28	86	11	CA	F3	D0	E6	CD
30	33	C8	E3	CE	3E	59	CF	55	75	2E	63	FA	4A	84	39
E1	05	1F	3C	94	42	9A	A6	10	F8	21	D9	29	37	E0	E7
F5	68	EC	12	F1	2F	BA	1C	0A	45	85	46	8C	BE	41	2A
5D	8B	A4	34	16	D3	8E	98	F9	07	6E	AC	51	08	23	15
17	B9	C3	1E	82	9D	6A	87	BD	6F	89	74	F0	31	25	C5
79	60	DF	19	91	03	A1	D7	09	B2	1D	FC	B3	AB	8F	2D

Table 7: Dynamic S-Box1



```
47 25 65 4E 6C 71 BE 7D E0 B8 26 CE 50 83 9D 06
8B A7 B5 62 D8 AC BA 0C 79 96 9B D0 B2 C1 91 85
4F 82 81 4A 73 2E E7 56 C7 8F 36 FD EE C6 45 4C
D9 6D B4 A9 C5 1A 15 88 AB A2 C9 05 67 6A FB 0F
31 1F CD 9E 8E 63 2A 75 86 17 E9 C0 EA 42 4D BF
0D 59 89 57 AA 41 72 80 AF 69 8A F7 FE C8 16 F4
A1 27 7F F6 8C CC BB 5D 01 EF 1B F9 14 5F 30 BD
3A A4 9C F8 07 84 49 E2 21 28 55 CF 9F 61 7B 35
13 AD 2C 3D 5E FA D7 A3 DB 08 46 90 7C 87 C3 C2
A8 00 0B E8 66 3E 38 F1 EB 3C 33 DF F3 E1 76 03
2D 37 24 6B 11 39 E3 E4 19 DC CB 1E 58 32 54 3F
99 2F 60 29 3B 78 B7 53 8D E6 77 48 D2 95 68 B0
B1 A6 20 EC 0E B9 C4 44 A0 52 BC 5B 70 34 40 09
5C 51 2B 43 7A ED DD 98 0A A5 F0 94 FC F2 F5 E5
4B B3 DA CA 23 6E 93 AE 5A 9A D1 04 18 6F B6 97
D6 10 D3 74 64 DE 1D D4 1C D5 02 22 12 7E FF 92
```

Table 8: Dynamic S-Box2

4.2 Simulation Time

Simulation Time is the time required by the algorithm for processing completely a particular length of data. It depends on the processor speed, complexity of the algorithm, space available in RAM etc. Table 7 shows the simulation time of Fixed S-Box and Dynamic S-Box. Simulation time for designing dynamic S-Box is very less as compared to simulation time for designing fixed S-Box.

Type of S-Box	Designing Time of S_Box (seconds)
Fixed S_Box	2.4648158
Dynamic S_Box	0.343202200000007

Table 9: Designing Time of S-Boxes

4.3 Randomness tests

Randomness tests are used to ensure randomness properties of the outputs corresponding to the implemented algorithm. Image Histogram test can be used for that purpose.

Image Histogram test

In this test (lena.jpg) image is encrypted using the same key for different S-boxes. The histogram for these encrypted images is plotted. Figure 22 shows the original image and its histogram. Figure 23 represents the encrypted image using conventional AES algorithm within its histogram. Figure 24 represents the encrypted image using key dependent dynamic AES S-box algorithm and its histogram. The encrypted images represent the randomness properties of both algorithms. The generated histogram of both the encrypted image is nearly the identical and extensively different from the original image, therefore, it does not provide any clue to employ statistical attack.

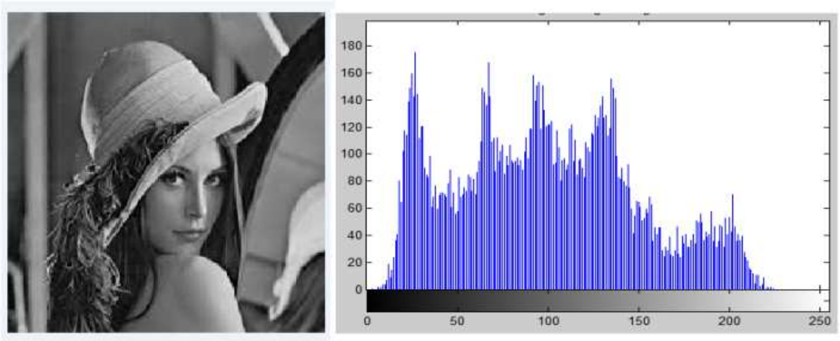


Figure 22: Original image and its Histogram

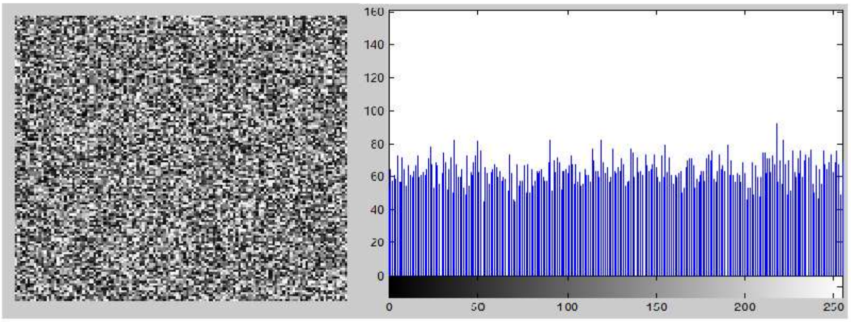


Figure 23: Conventional AES Encrypted image and its Histogram

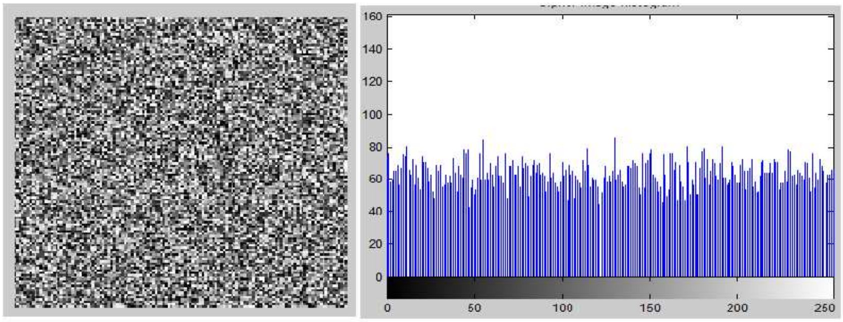


Figure 24: Dynamic AES Encrypted image and its Histogram

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

The proposed new method enables the construction of key dependent dynamic AES S-Box using two chaotic logistic maps based PRNG. This method solves the problem of fixed structure S-box of Conventional AES. The performance of the proposed technique has been tested by changing only one bit of the secret key to generate different S-boxes. The main advantage of the proposed algorithm is that various S-boxes can be generated for every one bit change in cipher key. This will increase the security of the AES. The results show that all the testing parameters namely, the Avalanche effect, Key sensitivity Analysis, Simulation time of S-Box and randomness test are approximately fulfilled and the comparison shows that the proposed method has better performance than the conventional AES. The security of the cryptographic algorithm could further be improved by using different transformations of chaotic logistic map in future work.

CHAPTER 6

REFERENCES

[1] Huang, Jian-hua, and Yang Liu. "A block encryption algorithm combined with the Logistic mapping and SPN structure." Industrial and Information Systems (IIS), 2010 2nd International Conference on. Vol. 2. IEEE, 2010.

[2] Fu, Chong, and Zhi-liang Zhu. "An improved chaos-based stream cipher algorithm." Natural Computation, 2007. ICNC 2007. Third International Conference on. Vol. 3. IEEE, 2007.

[3] Khanzadi, Himan, Mohammad Eshghi, and Shahram Etemadi Borujeni. "Design and FPGA Implementation of a Pseudo Random Bit Generator Using Chaotic Maps." IETE Journal of Research (Medknow Publications & Media Pvt. Ltd.) 59.1 (2013).

[4] Shujun, Li, Mou Xuanqin, and Cai Yuanlong. "Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography." Progress in Cryptology—INDOCRYPT 2001. Springer Berlin Heidelberg, 2001. 316-329.

[5] Patidar, Vinod, Krishan K. Sud, and Narendra K. Pareek. "A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing." Informatica (Slovenia) 33.4 (2009): 441-452.

[6] Francois, Michael, et al. "A new pseudo-random number generator based on two chaotic maps." Informatica 24.2 (2013): 181-197.

[7] Jakimoski, Goce, and Ljupco Kocarev. "Chaos and cryptography: block encryption ciphers based on chaotic maps." IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 48.2 (2001): 163-169.

[8] Agrawal, Bhavana, and Himani Agrawal. "Implementation of AES and RSA Using Chaos System." International Journal of Scientific & Engineering Research 4.5 (2013): 1413-1417.

[9] Asim, Muhammad, and Varun Jeoti. "Efficient and simple method for designing chaotic S-boxes." ETRI journal 30.1 (2008): 170-172.

[10] Dara, Mona, and Kooroush Manochehri. "A Novel Method for Designing S-Boxes Based on Chaotic Logistic Maps Using Cipher Key." World Applied Sciences Journal 28.12 (2013): 2003-2009.

[11] Khamlich, Eddine. "Implementation of stronger AES by using Dynamic S-Box dependent of Master Key." Journal of Theoretical and Applied Information Technology 53.2 (2013).

[12] Kazlauskas, Kazys, and Jaunius Kazlauskas. "Key-dependent S-box generation in AES block cipher system." Informatica 20.1 (2009): 23-34.

[13] Mahmoud, Eman Mohammed, et al. "Dynamic AES-128 with key-dependent S-box." (2013).

[14] Shivkumar, S., and G. Umamaheswari. "Performance Comparison of Advanced Encryption Standard (AES) and AES key dependent S-box-Simulation using MATLAB." Process Automation, Control and Computing (PACC), 2011 International Conference on. IEEE, 2011.

[15] William, Stallings, and William Stallings. Cryptography and Network Security. 4/E. Pearson Education India, 2006.

CHAPTER 7

APPENDIX

A

AES	-	Advance Encryption Standard
APS	-	Adaptive Parameter Synchronization

C

CBES	-	Chaos Based Encryption System
CIA	-	Confidentiality, Integrity and Authentication
CLM	-	Chaotic Logistic Map

D

DAP	-	Differential approximation probability
DES	-	Data Encryption Standard

G

GF	-	Galois Fields

L

LAP	-	linear approximation probability

N

NIST	-	National institute of standards and technology (NIST)

P

PLCM	-	Piecewise Linear Chaotic Map
PRNG	-	Pseudo Random Number Generator

S

SPN	-	Substitution Permitution Network

T

TRBS	-	Truly Random Bit Sequences