International Journal of Pure and Applied Mathematics
Volume 119 No. 10 2018, 399-418
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

A Block Cipher Algorithm to Enhance the Avalanche Effect Using Dynamic Key-Dependent S-Box and Genetic Operations

[1]Balajee Maram and [2]J.M. Gnanasekar
[1]Department of CSE,
GMRIT, Rajam, India.
Research and Development Centre,
Bharathiar University,
Coimbatore.
balajee.m@hotmail.com
[2]Department of Computer Science & Engineering,
Sri Venkateswara College of Engineering,
Sriperumbudur Tamil Nadu.
jmg_sekar@yahoo.com

Abstract

In digital data security, an encryption technique plays a vital role to convert digital data into intelligible form. In this paper, a light-weight S-box is generated that depends on Pseudo-Random-Number-Generators. According to shared-secret-key, all the Pseudo-Random-Numbers are scrambled and input to the S-box. The complexity of S-box generation is very simple. Here the plain-text is encrypted using Genetic Operations and S-box which is generated based on shared-secret-key. The proposed algorithm is experimentally investigates the complexity, quality and performance using the S-box parameters which includes Hamming Distance, Balanced Output and the characteristic of cryptography is Avalanche Effect. Finally the comparison results motivates that the dynamic key-dependent S-box has good quality and performance than existing algorithms.

Index Terms:S-BOX, data security, random number, cryptography, genetic operations.

1. Introduction

In public network, several types of attacks[1] can be avoided by applying Data Encryption/Decryption[2]. There are two categories of Encryption methods are called Symmetric-key encryption and Asymmetric-key encryption. Symmetric-encryption algorithms are 1000 times faster than Asymmetric-encryption algorithms[3]. Still, Symmetric-key cryptography plays a vital for exchanging data over insecure communication channels[4]. The block ciphers such as DES (Data Encryption Standard)[5], AES (Advanced Encryption Standard)[6], and EES (Escrowed Encryption Standard)[7] are popular cryptography algorithms and are being used by many companies in worldwide. The Tiny Encryption Algorithm (TEA)[8,9,10] is one of the fastest and efficient algorithm which uses operations from orthogonal algebraic groups like XOR, ADD and SHIFT. But TEA suffers from equivalent keys, because its key size is only 126-bits[11].

Confusion and Diffusion is an important characteristic for Information Security. Confusion is being provided by different forms of existing Substitution Boxes (S-Box)[12]. S-Box is a mapping table which translates n-bits to m-bits. The design and analysis of a strong S-Box is a time consuming process, because it supports nonlinearity to the cryptosystems. But the design of S-Box leads to break easily [13,14]. The S-Box design is being suffered from two major challenges. They are S-Box Searching and Verification of S-Box against the desired properties for an S-Box[15]. The properties of S-Box are Avalanche Effect, Strict Avalanche (SAC) and Bit Independence Criteria (BIC), nonlinearity and maximum expected linear probability (MELP)[16,17] etc. These are the desirable properties for S-Box design[18,19].

Recently Random key-dependent S-Boxes are being generated for encryption process[16,20,21] and S-Boxes are generated and checked until a strong S-Box is found[22].

The proposed system is able to handle and solve the limitations in existing Symmetric-key cryptography algorithms. It is based on two large prime numbers for generating Pseudo-Random Numbers. For different seeds (different large primes), it can generate different Pseudo-Random Numbers. The proposed system can use very large primes, bitwise-XOR operation and some mathematical methods to scrambles the bits. It is able to handle 512-bit blocks and key. This paper works on the properties like Hamming-Distance, Balanced-Output and Avalanche-Effect.

1.1. Properties of S-box

Many design rules for S-box generation have been proposed. Some of the formal design criterions of S-boxes[27] for cryptographic purposes are as follows.

1.1.1. Strict Avalanche Criterion (SAC)

Strict Avalanche Criterion (SAC) is one of the important characteristic of cryptography. According to SAC, change in 1-bit of plain-text influences more

than half of the bits in cipher-text. Or, change in 1-bit of key then it should effects more than half of the bits in cipher-text. This is called Avalanche Effect.

1.1.2. Hamming Distance

The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different.

1.1.3. Balanced Output

Balanced output is the output which has the number of 1's & 0's should be matched approximately.

Finally, there are several other requirements. But they are beyond the scope of this paper.

2. Literature Survey

In this section, some of the existing algorithms are presented to discuss:

An algorithm for S-box generation "An Algorithm for Key-Dependent S-Box Generation in Block Cipher System" [23] has been proposed. Here the S-box generation is depends on secret-key. Here huge number of S-boxes is generated by changing the shared-secret-key. But it consumes more time for Encryption and Decryption.

An algorithm for S-Box generation "Enhancing Advanced Encryption Standard S-Box Generation Based on Round Key" [24] has been proposed. The rotation value is depends on round key. According to rotation value, the values in S-box are rotated. It is a time consuming process. "Efficient Implementation of AES By Modifying S-Box"[25] has been proposed. The performance of the original AES algorithm using S-BOX which is based on polynomial is very good. In this paper, S-box and Inv S-box have been modified by swapping each word of S-box and Invs-box generated by new polynomial. The result shows the modified AES yields better results.

In "IMPLEMENTATION OF STRONGER AES BY USING DYNAMIC S-BOX DEPENDENT OF MASTER KEY" [26], a master key dependent S-box has been proposed. Here each entry in S-Box is XOR with master-key. This calculation part yields higher Encryption/Decryption time.

In "Implementation Of Stronger S-Box For Advance Encryption Standard" [27], a two stage pipeline combinational logic based S-box is presented. It is better than a typical ROM based lookup table implementation which access time is fixed and unbreakable.

Key-dependent S-box has been proposed in "Key-Dependent S-Box Generation in AES Block Cipher System" [28]. The static s-boxes are vulnerable. So the dynamic s-boxes are resistant to different attacks. The proposed system is used to generate large number of s-boxes by changing the secret-key.

In this paper, a new algorithm "KEY-DEPENDENT S-BOX IN LIGHTWEIGHT BLOCK CIPHERS" [29] to generate S-BOX based on the secret key is proposed. This research showed the intensive analysis for cost and security for 1bit, 2bits, 4bits and more than 4 bits. Depending on the application, the suitable method can be selected.

Cryptographically strong S-Box is proposed in "Construction of Cryptographically Strong 8x8 S-boxes" [30]. The construction of S-box is the action of PGL (2, GF (2^8)) group on GF(2^8); The proposed S-box is comparable with AES S-box, Affine Power Affine S-box, Gray S-box. And it is better than Prime S-box.

In "Analyses of SKIPJACK S-Box"[31], the performance of S-box has been done. It is better than X_{yi} and Residue prime S-box. It is suitable for secure communication.

It compares the nonlinearity of SKIPJACK S-BOX with existing algorithms. Its result is almost similar to existing optimal values.

A new strategy has been proposed in "A novel design for the construction of safe S-boxes based on TDERC sequence" [32] for developing cryptographically strong 8X8 S-boxes. These proposed S-boxes satisfy bijective and nonlinearity properties.

A quasi groups based S-box has been explained in "A NEW APPROACH INTO CONSTRUCTING S-BOXES FOR LIGHTWEIGHT BLOCK CIPHERS"[33]. Linearity test is not conducted on this approach. But theoretically a new S-box has been proposed, but not yet tested.

A novel S-box has been introduced in "Performance Efficiency of Modified AES Algorithm Using Multiple S-Boxes"[34]. Here a new S-box AES-2SBoxXOR is explained. But it consumes more Encryption/Decryption time.

In "S-box Optimization Technique with a Primitive Irreducible Polynomial"[35], a new design of S-box has been explained. Here the S-box design is depends on the strength of the user key. If the elements in the key are same then it reduces the strength of the S-box. In "Generating Pseudorandom S-Boxes – a Method of Improving the Security of Cryptosystems Based on Block Ciphers"[36], new Pseudorandom S-boxes have been introduced. In [37], the generation of key dependent S-box has been explained. The algorithm "Designing an algorithm with high Avalanche Effect "[38] proposes the algorithm that gives good Avalanche Effect and it is able to incorporate in the process of encryption of any plain text. But the proposed algorithm is tested correctly and the results are not clear. The difference between modern cryptography and classical cryptography algorithms have been analyzed in [39,48]. It concludes that the Avalanche-Effect of modern cryptography algorithms is good. In [40,41,42], explains different method to improve Avalanche-Effect in cryptography. In

"Symmetric Key Encryption using Genetic Algorithm"[43], a new approach is explained to use crossover and mutation processes in symmetric encryption/decryption. In [44], explains the use of the Genetic operators plays a vital role in key formation which are resistant from different attacks. The generation of Pseudo Random Sequence and cryptography algorithm using operations of genetic algorithm is explained in "A Cryptography Algorithm Using the Operations of Genetic Algorithm & Pseudo Random Sequence Generating Functions"[45]. In [46,47,] the cryptography properties Space complexity, computation time and memory usage have been discussed. The generation and use of key dependent S-box is discussed in [49,50]. Comparatively the algorithms in [49,50] are better than both classical and modern cryptography algorithms.

3. S-Box and Inverse S-Box Construction

3.1. Pseudo-Code for Generation of Pseudo-Random Numbers[49,50]

Here Pseudo-Random Numbers[49,50] are generated that are needed for Encryption/Decryption process. The generation process of Pseudo-Random Numbers is as follows:

1: Take two large prime number p and q.
2: for all i=0, 1, …, 255 do
3: p=(p*q+1) mod 256
4: a(i)←p
5: end for

3.2. Pseudo-Code for Generation of Pseudo-Random Numbers for positions in S-Box[49,50]

Pseudo-Random Numbers plays an important role for generation of S-box in this algorithm. Pseudo-Random Numbers required for positions to keep the Pseudo-Random Numbers in S-box for Encryption/Decryption process. The generation process of Pseudo-Random Numbers for position is as follows:

1: Calculate next primes of p and q are p1 and q1
2: for all i=0, 1, …, 255 do
3: p1=(p*q1+1) mod 256
4: p(i)←p1
5: end for

3.3. Algorithm for Generation of Key-Dependent Dynamic S-Box[49,50]

Input:

a) The secret key key[i], i= 1, 2, …, n is the vector of n integer numbers from the interval [0..255].
b) Array of Pseudo-Random Numbers i.e. a() for S-Box values
c) Array of Pseudo-Random Numbers i.e p() for positions
d) Number of rounds 'rounds'

Output:

a) The key-dependent substitution box S-Box(i)(j), i= 0,1,…,15 and j=0,1,…,15 is the 2-Dimensional vector of the different integer numbers(bytes) from the range [0,255].
b) The key-dependent inverse substitution box invS-Box(i)(j), i= 0,1,…,15 and j=0,1,…,15 is the 2-Dimensional vector of the different integer numbers(bytes) from the range [0,255].

Algorithm:

Step 1: Make S-Box(i)(j) from the array a(),i= 0,1,…,15 and j=0,1,…,15 is the 2-Dimensional vector of the different integer numbers(bytes) from the range [0,255].

Step 2: Each row in S-Box()() is circularly shifted to Left/Right according to the values of Key()

Step 3: for round=1 .. rounds do

Step 4: for all row=0, 1, …, 16 do

Step 5: if key(index) is even then row in S-Box()() is circularly shifted to Left of key(index) mod 16 positions

Step 6: if key(index) is odd then row in S-Box()() is circularly shifted to Right. Of key(index) mod 16 positions

Step 7: end of Step 4 for loop

Step 8: end of Step 3 for loop

Step 9: All the elements in S-Box()() are permuted according to values in p() array.

Step 10: Now S-Box()() is ready.

Using Pseudo-Random-Number generation algorithm, 256 numbers are generated in the range [0,255]. These 256 numbers are placed in 16X16 2-Dimensional matrix is called S-box. Each row in S-box is circularly shifted to Left/Right according to the values of Shared-Secret-Key. If the key (index) is even then the row in S-box is circularly shifted to left of key(index) mod 16 positions. If the key(index) is odd then the row in S-box is circularly shifted to right of key(index) mod 16 positions. The above process will be continued for the given number of rounds. Again Pseudo-Random-Number generation algorithm, 256 numbers are generated in the range [0,255]. These 256 numbers are placed in 1-Dimensional array p(). All the elements in S-Box are permuted according to the values in the array p(). Now the key-dependent S-box is generated and ready to use in Encryption/Decryption process.

3.4. Algorithm for Inverse S-Box[49,50]

Step 1: Arrange all the elements from S-Box()() to a()

Step 2: for all i=0,1,…,255 do

Step 3: Calculate inva(a(i))←i

Step 4: end for

Step 5: Arrange all the elements from inva() to invS-Box()()

Step 6: Inverse of S-Box()() is ready.

All the elements in S-box are rearranged from 2-dimensional array to 1-dimensional array ‘a’. The inverse of ‘a’ is calculated using the following formula: Inva(a[index])=index. Then all 256 elements are rearranged from 1-dimensional to 2-dimensioal array is called invS-Box.

4. Proposed System

4.1. Algorithm for Data Encryption

Step 1: Take the Plain-text

Step 2: Convert Plain-text into binary form

Step 3: Divide the binary form data into 512-bit blocks

Stage 1

Step 4: From 1 to 512

→No change in first bit. This is the first bit in Result.

→Pre-cipher2$_i$= 1, Pre-cipher1$_i$ = Pre-cipher2$_{i-1}$

0, Pre-cipher1$_i$ # Pre-cipher2$_{i-1,}$ Where i is 2 to 512

Stage 2

Step 5: take z=0

Step 6: Take next two numbers from Pseudo-Random-Number array p()

Let Cross-Over1(co1)=p(z)

Cross-Over2(co2)=p(z+1)

Step 7: Apply two-point Cross-Over on first 32-character block and second 32-character block.

Step 8: Take next two number from Pseudo-Random-Number array p()

Let Mutation1(m1)=p(z+2)

Mutation2(m2)=p(z+3)

Step 9: Apply Mutation with the values m1 & m2 on first 32-character block and second 32-character block.

Step 10: z=(z+4); if z is equal to 256 then z=0

Step 11: Apply DRDP method on 64-character block

Step 12: For i=1 to 64 step by 2

i) Take two characters
ii) Append binary form of second number to first.
iii) Apply bitwise XOR on Step ii and Random-Number
iv) Quotient=(Result of Step iii)/256
v) Reminder=(Result of Step iii) mod 256
vi) Quotient and SBOX[Reminder/16][Reminder MOD 16] are the cipher characters of Step i (plain-characters).
vii) Go to Step i

Step 13: if more Plain-text blocks are there, then go to Step 4.

Step 14: Cipher-text is ready

Step 15: Stop

4.2. Algorithm for Data Decryption

Step 1: Take the Cipher-text

Step 2: Convert into binary form

Step 3: Divide the binary form into 512-bit blocks, let z=0

Step 4: Take 512-bit block and convert it into ASCII

Step 5: Take 2 characters

Step 6: pass the 2nd character through Inverse S-Box.

Step 7: Append binary form of Step 6 to binary form 1st Character

Step 8: Apply bitwise XOR on output of Step 7and Random-Number

Step 9: if(characters in current 512-bit block) then goto Step 5

Step 10: Apply DRDP on Current block of 64 characters and Key(z)

Step 11: Apply Mutation on current block of 64-characters.

→Take next two number from Pseudo-Random-Number array p()

Let Mutation1(m1)=p(z+2)

Mutation2(m2)=p(z+3)

Step 12: Apply Cross-Over on current block of 64-characters.

→Take next two numbers from Pseudo-Random-Number array p()

Let Cross-Over1(co1)=p(z)

Cross-Over2(co2)=p(z+1)

Step 13: Apply two-point Cross-Over on first 32-character block and second 32-character block.

Step 14: From 1 to 512

→No change in first bit. This is the first bit in Result.

→From 2nd bit, From 2nd bit, compares current & previous bit. If both are same then output is '1' otherwise '0'.

Step 15: increment z by 1, if(blocks are available) then goto Step 4

Step 16: Stop

5. Experimental Setup of the Proposed System

In this section the results of analysis are given. The analysis includes the affect of S-box on the proposed algorithm. The proposed algorithm has been checked through JAVA code.

The shared secret prime numbers are p=1300193,q=1299869 for s-box and p=9901,q=10037 for position of elements in s-box.

5.1. Example1

key[]={65,177,99,34,60,189,222,200,187,155,23,9,13,68,14,0,35,161,171,201,229,254,49,52,90,111,119,117,131,137,129,163,217,229,246,65,177,99,34,60,189,222,200,187,155,23,9,13,68,14,0,35,161,171,201,229,254,49,52,90,111,119,117,131,137};

Table 1: Elements in S-BOX of 5.1

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

From Table 1, there is no relationship between any two consecutive rows. In each row, there is no relationship between any two consecutive elements. Hence it is not possible to predict the elements in an S-box.

Table 2: Elements in Inverse S-Box of 5.1

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

From Table 2, there is no relationship between any two consecutive rows. In each row, there is no relationship between any two consecutive elements. Hence it is not possible to predict the elements in a Inverse S-box.

6. Discussion on Security Parameters, Avalanche-Effect and S-Box Properties

6.1. Security Analysis

The privacy of data is measured by the variance between the actual and the perturbed values which is given by the following formula:

$$A = \frac{VAR(A - A')}{VAR(A)}$$

Table 3: Relation between plain-text and Cipher-text of alphabets (a..z)

	97	98	99	100	101	102	103	104	105	106	107	108	109
Plain-text(a..z)	110	111	112	113	114	115	116	117	118	119	120	121	122
	233	20	230	235	17	15	237	227	25	23	22	224	225
Cipher-text	28	222	243	238	248	6	240	241	12	238	248	249	4

Table 3shows the security performance of the proposed system. It compares the data before applying the proposed algorithm and after applying the proposed algorithm. Each element is scrambled through Stage-1, Stage-2 and S-box. It has been analyzed that the privacy or the security level of the confidential data is improved a lot by the proposed method.

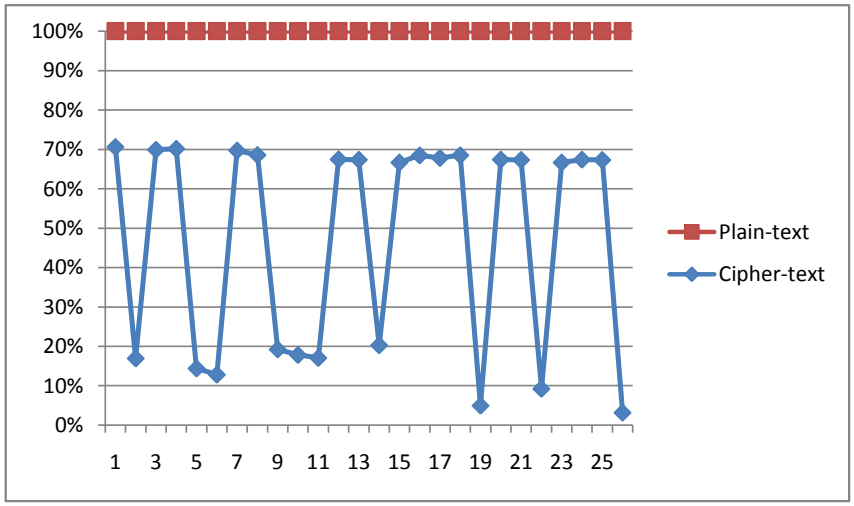

Figure 1: Relation between Plain-text and Cipher-text of Alphabets (a..z)

From the Figure 1, the plain-text characters 113 and 119 have the same cipher-text characters as shown in the following Figure 2. So the proposed algorithm supports many-to-one relation between plain-text and cipher-text. It concludes it is very difficult to get the plain-text from known cipher-text.

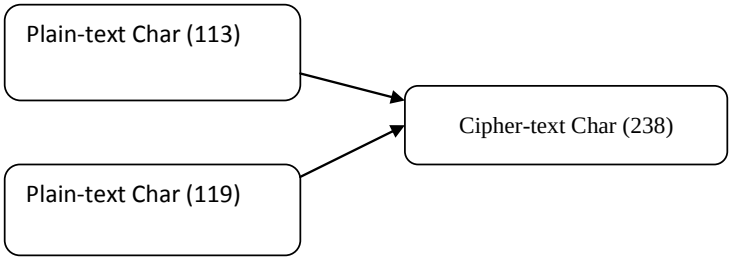

Figure 2: Mapping of Plain-Chars(113 & 119) with Cipher-Char(238)

6.2. Performance of the Proposed Algorithm

Table 4: Performance of the proposed system using Avalanche Effect

S.No	Plain-text	A	B	C	D
1	This is GMR Inst. of Tech. It is in Rajam 532129 Srikakulam Dist	490	95	374	73
	Thks is GMR Inst. of Tech. It is in Rajam 532129 Srikakulam Dist				
2	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	506	98	392	76
	caaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa				
3	234# js b Mtmbfr235# js b Mtmbfr234# js b Mtmbfs234# js b Mtmbfr	505	98	374	73
	434# js b Mtmbfr235# js b Mtmbfr234# js b Mtmbfs234# js b Mtmbfr				
4	123456789abcdefghijk11111111111122222222zzzzzzzzyyyyyyyyy#@!^&*%	506	98	379	74
	323456789abcdefghijk11111111111122222222zzzzzzzzyyyyyyyyy#@!^&*%				
5	1~~~~~~~~~~~~~~~~~~~~~~~~~~~~____________________ ``````` ______	506	98	375	73
	3~~~~~~~~~~~~~~~~~~~~~~~~~~~~____________________ ``````` ______				
6	AAAAABBBBBaaaaa11111CCCCC%%%%%*****(((((>>>>><<<<<:::::""""""----	498	97	376	73
	ACAAABBBBBaaaaa11111CCCCC%%%%%*****(((((>>>>><<<<<:::::""""""----				
7	12345!@#$%12345^&*()2345$%^&*90987^%$#21^&*$#@!";.,/?+_)9876534@	481	93	370	72
	12355!@#$%12345^&*()2345$%^&*90987^%$#21^&*$#@!";.,/?+_)9876534@				

A→Avalanche Effect after Stage-1, B→Avalanche Effect (%) after Stage-1

C→Avalanche Effect after Stage-2, D→Avalanche Effect (%)after Stage-2

From Table4, Avalanche-Effect of the proposed algorithm after Stage-1 and Stage-2 are compared.

After Stage-1 of the proposed algorithm, the Avalanche-Effect is more than 96% (approx). After Stage-2, the Avalanche-Effect is more than 73% (approx). After stage-1, The Avalanche-Effect of the proposed algorithm is better than the existing algorithms[43,44,45,46].

6.3. Hamming Distance

The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different.

Table 5: Hamming Distance of Pi, LCG and Proposed Algorithms

S.No	**Plain-text**	**Hamming Distance**
1	This is GMR Inst. of Tech. It is in Rajam 532129 Srikakulam Dist	49
2	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	47
3	234# js b Mtmbfr235# js b Mtmbfr234# js b Mtmbfs234# js b Mtmbfr	50
4	123456789abcdefghijk11111111111122222222zzzzzzzzyyyyyyyyy#@!^&*%	50
5	1~~~~~~~~~~~~~~~~~~~~~~~~~~~__________________ ``````` _____	48
6	AAAAABBBBBaaaaa11111CCCCC%%%%%*****(((((>>>>><<<<<:::::""""""----	49
7	12345!@#$%12345^&*()2345$%^&*90987^%$#21^&*$#@!";.,/?+_)9876534@	51

6.4. Balanced Output

Balanced output is the output which has the number of 1's & 0's should be nearly same.

Table 6: Balanced Output of Pi, LCG and Proposed Algorithms

S.No	**Plain-text**	**0's**	**1's**
1	This is GMR Inst. of Tech. It is in Rajam 532129 Srikakulam Dist	273	240
2	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	249	264
3	234# js b Mtmbfr235# js b Mtmbfr234# js b Mtmbfs234# js b Mtmbfr	253	260
4	123456789abcdefghijk11111111111122222222zzzzzzzzyyyyyyyyy#@!^&*%	252	261
5	1~~~~~~~~~~~~~~~~~~~~~~~~~~~____________________ ``````` ___	262	251
6	AAAAABBBBBaaaaa11111CCCCC%%%%%*****(((((>>>>><<<<<:::::""""""----	281	232
7	12345!@#$%12345^&*()2345$%^&*90987^%$#21^&*$#@!";.,/?+_)9876534@	264	249

6.5. Comparison of Proposed Algorithm with Existing Algorithms

The proposed algorithm is compared with existing algorithms. There are two stages in the proposed algorithm.

After Stage-1, the proposed algorithm's Avalanche Effect is 96% (Average). After Stage-2, the proposed algorithm's Avalanche Effect is 73% (Average).

Table 7: List of algorithms with Avalanche Effect (%)

Encryption Technique	**Avalanche Effect (%)**
Proposed system after Stage-1(Average case)	96.71429
Proposed system after Stage-2(Average case)	73.42857
"A Block Cipher Having a Key on One Side of the Plan Text Matrix and its Inverse on the Other Side"	68.5
Algorithm in Ref 38	56
DES	54.68
Algorithm in Ref 40	53
AES-RC4	52.34
Algorithm in Ref 51	52.34
Algorithm in Ref 41	49.21
Original AES	46.88
Algorithm in Ref 24	46
MTEA	32
Blowfish	28.71
TEA	25
Ref 36	24.219
Playfair Cipher	6.25
Vigenere Cipher	3.13
Caesar Cipher	1.56

From Table 7, The Architecture of the proposed system in [38] is not simple and its Avalanche Effect is 56%. According to Table 7, the Avalanche Effect of classical cryptography algorithms is very less. The Table 5concludes the Avalanche Effect of proposed system is the best. After Stage-1, the Avalanche Effects are 96%, 58%, 75% in best-case, worst-case and average-case respectively in proposed system. After Stage-2, the Avalanche Effects are 76%, 37.5%, 57.69% in best-case, worst-case and average-case respectively in proposed system.

7. Conclusion

The properties of S-Box and Avalanche-Effect are very important for cryptography. Till now many algorithms have been introduced for S-box generation and to improve the Avalanche-Effect. The performance of dynamic S-box is better than static S-box. In this research paper, a new Block-Cipher algorithm has been proposed which enhances the Avalanche-Effect using dynamic key-dependent S-box.

The proposed algorithm produces better results for the S-box parameters Hamming-Distance and Balanced-Output. It produces best Avalanche-Effect after Stage-1 as well as Stage-2 of the proposed algorithm. The best value of Stage-1 Avalanche-Effect is 98% and the best value of Stage-2 Avalanche-Effect is 76%. Both are better than the existing algorithms. The proposed algorithm is also applicable to images, audio, video and UNICODE files. Finally it concludes that the proposed algorithm is easy to implement and embed into any existing crypto-systems.

References

[1] Stallings W, Network Security Essentials (Applications and Standards), Pearson Education: USA (2004), 2-80.

[2] Pfleeger, C.P., Pfleeger, S.L., Security in Computing, Pearson Education: USA (2004), 642-66.

[3] AviKak Lecture Notes on Computer and Network Security, (2015).

[4] Dragos T, Sequential and Parallel Cascaded Convolution Encryption with Local Propagation: Toward future Directions in Cryptography, Proceedings of The third International Conference on Information Technology-New Generations (2006), 464-69.

[5] Data Encryption Standard, (2015).

[6] Advanced Encryption Standard, (2015).

[7] Escrowed Encryption Standard, (2015).

[8] Hernández J.C., Isasi P., Ribagorda A., An application of genetic algorithms to the crypto analysis of one round TEA, Proceedings of the Symposium on Artificial Intelligence and its Application, 2002.

[9] Hernández J.C., Sierra J.M., Isasi P., Ribargorda A., Finding efficient distinguishers for cryptographic mappings, with an application to the block cipher TEA, Computational Intelligence 20(3) (2004), 517-25.

[10] Hernández J.C., Sierra J.M., Ribagorda, A., Ramos B., Mex-Perera J.C., Distinguishing TEA from a Random Permutation: Reduced Round Versions of TEA Do Not Have the SAC or Do Not Generate Random Numbers, Cryptography and Coding, Springer-Verlag: Berlin Heidelberg (2001), 374-77.

[11] Kelsey J., Schneier B., Wagner D., Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X NewDES, RC2, and TEA, Lecture Notes in Computer Science (1997), 233–46.

[12] Mar P.P., Latt K.M., New analysis methods on strict avalanche criterion of S-boxes, World Academy of Science, Engineering and Technology 2(12) (2008), 899-903.

[13] Adams C., Tavares S., The structured design of cryptographically good S-boxes, Journal of Cryptology 3(1) (1990), 27-41.

[14] Hussain I., Shah T., Mahmood H., Afzal M., Comparative analysis of S-boxes based on graphical SAC, International Journal of Computer Applications 2(5) (2010), 1-7.

[15] Ahmed N., Testing an S-Box for Cryptographic Use, International Journal of Computer and Electrical Engineering (2016), 1-5.

[16] Keliher L., Meijer H., Tavares S., A new substitution-permutation network cryptosystem using key-dependent s-boxes, In: Proc. SAC'97, Canada (1997), 13–26

[17] Keliher L., Refined analysis of bounds related to linear and differential and linear cryptanalysis for the AES, (In: H. Dobbertin et al., eds. Advanced Encryption Standard AES Š04, Bonn (2004), 42–57.

[18] Vergili I., Yücel M.D., On Satisfaction of Some Security Criteria for Randomly Chosen S-Boxes, in Proc. 20th Biennial Sympisum on Communications, Kingston (2000), 1-5.

[19] Vergili I., Yücel M.D., Avalanche and Bit Independence Properties for the Ensembles of Randomly Chosen n ×n S-Boxes, Turk Journal of Electrical Engineering 9(2) (2001), 137-45.

[20] Kazlauskas K., Key-dependent S-box generation in AES block cipher system, Informatica 20(1) (2009), 23–34

[21] Schneier B., Description of a new variable-length, 64-bit block cipher (Blowfish), (In: Proc.Fast Software Encryption, Springer-Verlag: Berlin Heidelberg (1994), 191–204.

[22] Mroczkowski P., Generating Pseudorandom S-Boxes– a Method of Improving the Security of Cryptosystems Based on Block Ciphers, Journal of Telecommunications Information Technology (2009), 74–79.

[23] Kazys Kazlauskas, Gytis Vaicekauskas, Robertas Smaliukas, An Algorithm for Key-Dependent S-Box Generation in Block Cipher System", Informatica 26(1) (2015), 51–65.

[24] Juremi J., Mahmod R., Sulaiman S., Ramli J., Enhancing advanced encryption standard S-box generation based on round key, International Journal of Cyber-Security and Digital Forensics (IJCSDF) 1(3) (2012), 183-188.

[25] Vijay L.H., Basavaraj P.H., Veena V.D., Efficient Implementation of Aes By Modifying S-Box, IOSR Journal of Computer Science (IOSR-JCE) (2015), 35-39.

[26] Sliman Arrag, Abdellatif Hamdoun, Abderrahim Tragha, Salah Eddine Khamlich, Implementation of Stronger Aes By Using Dynamic S-Box Dependent Of Master Key, Journal of Theoretical and Applied Information Technology 53 (2) (2013).

[27] Bharath kumar B., Rajesh kumar R., Implementation of Stronger S-Box for Advanced Encryption Standard, The International Journal Of Engineering And Science (IJES) 3(12) (2014), 39-47.

[28] Kazys Kazlauskas, Jaunius Kazlauskas, Key-Dependent S-Box Generation in AES Block Cipher System, INFORMATICA 20(1) (2009), 23–34.

[29] Sufyan Salim Mahmood Aldabbagh, Imad Fakhri Taha Al Shaikhli, muhammad Reza Zaba, Key-Dependent S-Box In Lightweight Block Ciphers, Journal of Theoretical and Applied Information Technology 62(2) (2014).

[30] Iqtadar Hussain, Tariq Shah, Muhammad Asif Gondal and Waqar Ahmad Khan, Construction of Cryptographically Strong 8x8 S-boxes, World Applied Sciences Journal 13(11) (2011), 2389-2395.

[31] Iqtadar Hussain, Tariq Shah, Muhammad Asif Gondal, Yong Wang, Analyses of SKIPJACK S-Box, World Applied Sciences Journal 13 (11) (2011).

[32] Hussain Alkhaldi A., A novel design for the construction of safe S-boxes based on TDERC sequence, Alexandria Eng. J. (2015).

[33] Hristina Mihajloska, Danilo Gligoroski, A New Approach Into Constructing S-Boxes For Lightweight Block Ciphers, 8th Conference on Informatics and Information Technology with International Participation (2011).

[34] Felicisimo V.W., Performance Efficiency of Modified AES Algorithm Using Multiple S-Boxes, International Journal of New Computer Architectures and their Applications (IJNCAA) 5(1) (2015), 1-9.

[35] Dhrubajyoti Sikdar, S-box Optimization Technique with a Primitive Irreducible Polynomial, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) 3(3) (2014).

[36] Piotr Mroczkowski, Generating Pseudorandom S-Boxes – a Method of Improving the Security of Cryptosystems Based on Block Ciphers, Journal of Telecommunications and Information Technology, (2009).

[37] Sandy Harris, Carlisle Adams, Key-Dependent S-Box Manipulations, LNCS, Springer-Verlag Berlin Heidelberg (1999), 15–26.

[38] Sriram Ramanujam and Marimuthu Karuppiah, Designing an algorithm with high Avalanche Effect, IJCSNS International Journal of Computer Science and Network Security 11(1) (2011).

[39] Ganesh Patidar, Nitin Agrawal, Sitendra Tarmakar,A block based Encryption Model to improve Avalanche Effect for data Security, International Journal of Scientific and Research Publications 3(1) (2013).

[40] Parvez khan Pathan, Basant Verma, Hyper Secure Cryptographic Algorithm to Improve Avalanche Effect for Data Security, International Journal of Computer Technology and Electronics Engineering (IJCTEE) 1 (2) (2015).

[41] Jayant P.B., Dr. Prashant N.C., Avalanche Effect of AES Algorithm, (IJCSIT) International Journal of Computer Science and Information Technologies 5(3) (2014), 3101 – 3103.

[42] Ajeet Singh, A New Approach to Enhance Avalanche Effect in Aes to Improve Computer Security, Information Technology & Software Engineering, JITSE 5(1) (2014)

[43] Dr. Poornima G. Naik,Girish R. Naik, Symmetric Key Encryption using Genetic Algorithm, Sinhgad Institute of Management and Computer Application (SIM CA), (2014).

[44] Amritha Thekkumbadan Veetil, An Encryption Technique Using Genetic Operators, International Journal Of Scientific & Technology Research 4(7) (2015).

[45] Suvajit Dutta, Tanumay Das, Sharad Jash, Debasish Patra, Dr.Pranam Paul, A Cryptography Algorithm Using the Operations of Genetic Algorithm & Pseudo Random Sequence Generating Functions, Dutta et al., International Journal of Advances in Computer Science and Technology, 3(5) (2014).

[46] Vikendra Singh, Sanjay Kumar Dubey, Analysing Space Complexity of Various Encryption Algorithms, International Journal of Computer Engineering and Technology (IJCET) 4(1) (2013), 414-419.

[47] Shashi Mehrotra Seth, Rajan Mishra, Comparative Analysis Of Encryption Algorithms For Data Communication, International Journal of Computer Science and technology 2(2) (2011).

[48] Balajee Maram K., Gnanasekar J.M., Light Weight Cryptographic algorithm to Improve Avalanche Effect for Data Security using Prime Numbers and Bit Level Operations, International Journal of Applied Engineering Research 10 (21) (2015), 41977-41983.

[49] Balajee Maram K., Gnanasekar J.M., Generation of a Dynamic Random 16X16 S-Box for Unicode Text Using Prime Numbers and Secret-Key, Australian Journal of Basic and Applied Sciences 9(36) (2015), 140-149.

[50] Balajee Maram K., Gnanasekar J.M., Evaluation Of Key Dependent S-Box Based Data Security Algorithm Using Hamming Distance and Balanced Output", Tem Journal 5(1) (2016).

[51] Hasan M.A., Enhancing The Encryption Process of Advanced Encryption Standard (AES) By Using Proposed Algorithm To Generate S-Box, published by Journal of Engineering and Development 18(2) (2014).