International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 9, Issue.2, February 2021 E-ISSN: 2347-2693

Enhance Security of AES Algorithm Based on S-Box

Rawia Alkhamery[1*], Yasser Alahmadi[2], Mokhtar Alsorori[3], Saleh Alassali[4]

[1,2,3,4]Computer Science Dept., Sheba Region University, Marib - Yemen

[*]Corresponding Author: rawia00123@gmail.com, Tel.: +967-715880579

DOI: https://doi.org/10.26438/ijcse/v9i2.3945 | Available online at: www.ijcseonline.org

Received: 18/Feb/2021, Accepted: 22/Feb/2021, Published: 28/Feb/2021

***Abstract*—** Advanced Encryption Standard (AES) is an approved encryption algorithm that has been used so far in many applications. A strength of AES algorithm depends on substitution box (S-Box) that is the main component to provide nonlinearity operations. Although AES algorithm has been proven to be the most secure algorithm to date, the advances in computer processing speed nowadays and the attempts to break such algorithm through the linear and differential cryptanalysis made it vulnerable to obsolescence. Therefore, the development of the algorithm is still ongoing especially for modification of the static nature of its S-Box. This paper proposes a method to improve the security of AES algorithm by suggesting treatment in the Substitution Box which is used to generate nonlinear relationship. Experimental results showed that the proposed method can enhance security of AES algorithm in the same condition of efficiency.

***Keywords*—** Cryptography, symmetric key, block cipher, AES algorithm and dynamic S-Box.

I. INTRODUCTION

Due to the rapid grow in the field of technology, cryptography has become plays an important role in protect sensitive data against passive and active attacks, where most organizations need to protect its private data, especially data related to financial transactions. In these days, there are a lot of encryption algorithms used to protect data such as DES, 3DES, Blowfish, AES, RSA ...etc. each algorithm has advantages and disadvantages. AES is defined by the National Institute of Standards and Technology(NIST), where in 1997, the NIST published a call for a new Advanced Encryption Standard (AES) to replace the current Data Encryption Standard (DES) which has strength and security equal to or better than 3DES. NIST determined that AES should be symmetric block cipher with a length of 128 bits and support keys of lengths 128-192-256 bits. A set of proposals of 21 algorithms were presented and through nine months, 15 algorithms were chosen in the first round in 1998, in the second round, 5 algorithms were chosen in 1999. After several tests, NIST announced the selection of Rijndael as the proposed Advanced Encryption Standard (AES) [1], [2]. Its name is a combination of the names of its inventors Dr. Joan Daemen and Dr. Vincent Rijmen and published a final standard (FIPS PUB 197) in 2001[3], which is the most powerful and widely used in many cryptographic applications today. However, due to the rapid development in computer processing speed, this algorithm may be compromised, therefore, there are myriad studies have improved this algorithm. Majority of those studies have modified the static nature of S-box used in AES algorithm such as [4], [5], [6], [7], [8].

In this paper, the proposed method is presented to generate random start point of the S-Box using the value of the first byte of the secret key for improvement of the security of AES algorithm.

This paper is organized into seven sections, Introduction is in Section I. The structure of AES algorithm is in Section II, substitution box is in Section III, Related work is in Section IV, Proposed s-box generation is in Section V, Experimental results are in Section VI and Conclusion and Future work are in Section VII.

II. STRUCTURE OF AES ALGORITHM

AES is symmetric block cipher algorithm and support keys of lengths 128-192-256 bits. The input to the AES is divided into blocks of each 128-bit block, the input text is represented in hex and stored in a 4 × 4 matrix called State [9]. Similarly, the key is stored in a 4 × 4 matrix. This key is extended into an array of key schedule words, each word is four bytes, the total of words is 44 words for a 128-bit key, 52 words for a 192-bit key and 60 words for a 256-bit key.

AES Cipher includes Nr rounds, the number of rounds depends on the key length, 10 rounds for a 128-bit key, 12 rounds for a 192-bit key and 14 rounds for a 256-bit key [2], [9], [10]. In AES algorithm each round consists of four different stages as follows:

1- **SubBytes**: Simple transformation operation that substitute every byte in the state matrix with a different value using a substitution box (S-Box). It adds nonlinearity and confusion.

2- **ShiftRows**: A shifting operation performed on the bytes of the last three rows of the state matrix, where each row is cyclically shifted to the right by a different number of bytes. This stage is linear and provides diffusion.

3- **MixColumn**: In this stage, each column vector of the state matrix is multiplied by a fixed matrix. Addition and multiplication operations are accomplished in polynomials. It provides inter-byte diffusion.

4- **AddRoundKey**: In this stage, the simple bitwise EXOR. operation is performed between the state matrix and appropriate roundkey. It provides confusion.

The encryption process begins with an initial round in which only Addroundkey stage is performed, followed by nine rounds that performs all four stages SubBytes, ShiftRows, MixColumn and AddRoundKey. In the last round only three stages are performed SubBytes, ShiftRows and AddRoundKey, Figure 1 illustrates the encryption process.

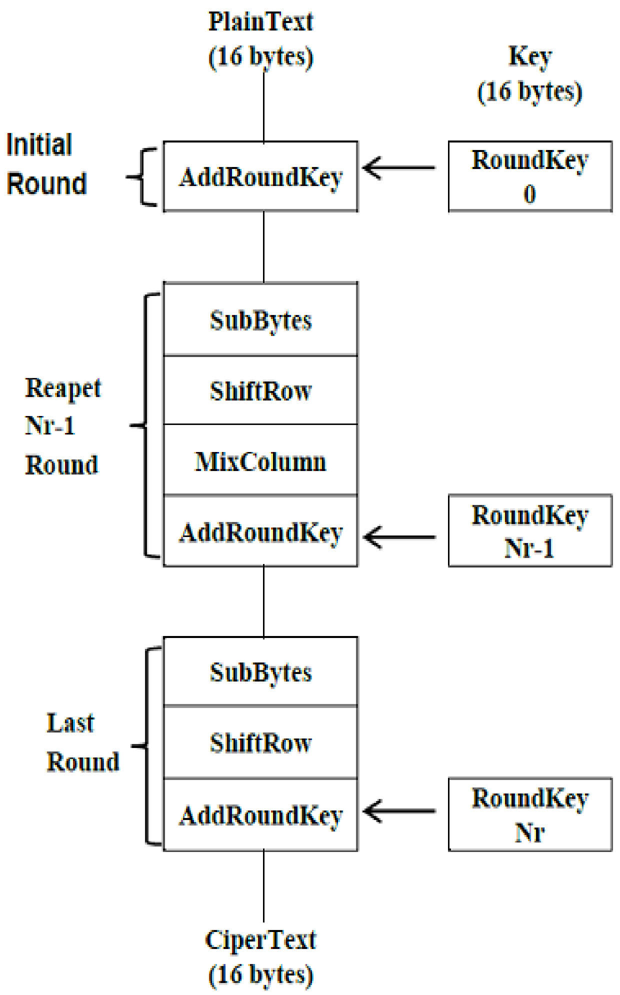

Figure 1: AES Encryption

The decryption process is performed in the inverse order of the processes used in the encryption process. Figure 2 illustrates the decryption process.

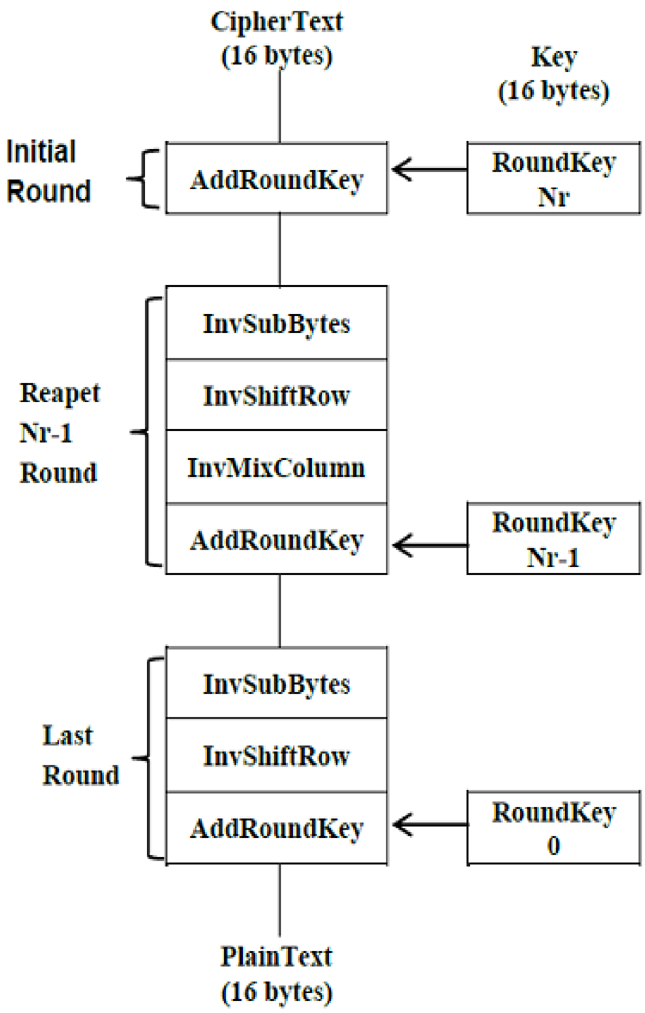

Figure 2: AES Decryption

III. SUBSTITUTION BOX

The S-Box is an essential component of any modern block cipher such as AES and plays a crucial role in making such cryptosystems resistant to various attacks because the implementation of S-Box is the only source of nonlinearity in block cipher which provides confusion to reveal the relationship between the input and the output [5]. AES algorithm describes a 16 × 16 matrix of byte values called an S-Box as given in Table 1, that includes a permutation of all possible 256 byte values used for substitution operation. The S-Box is constructed by first taking the multiplicative inverse in the finite field GF (2^8) and then affine transformation is performed over GF (2) [2], [5], [9]. In substitution operation each single byte of state matrix is mapped into a new byte in the s-box where 4-bit of left and right of one byte in the state matrix are used as indexes into the S-box to select a unique 8-bit output value. For example, during the encryption process the hexadecimal value {4E} in Table 1 references to row 4 and column E of the S-Box, which holds the value {2F} and during the decryption process the inverse operation will be performed where the hexadecimal value {2F} will be replaced by the original value {4E} as per Table 2.

Table 1: AES S-Box

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

Table 2: Inverse AES S-Box

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

IV. RELATED WORK

K. Kazlauskas and J. Kazlauskas[5], suggested a new algorithm to generating a dynamic key-dependent S-Box instead of the static S-Box used in the original AES algorithm in order to overcome linear and differential cryptanalysis. They also added a modification to the Key Scheduling algorithm and eliminating substitution of bytes from generating the round key. The advantage of this approach is the ability to generating a numerous dynamic S-Boxes by changing the secret key, but it consumes a significant amount of time to create dynamic S-Boxes.

H.M. El-Sheikh, O.A. El-Mohsen, S.T. Elgarf and A. Zekry [6], proposed a new approach for designing small S-Box instead of the S-Box used in traditional AES algorithm. The small S-Box is defined over GF (24) with different equations and different irreducible polynomials. The researchers used avalanche effect and strict avalanche criterion (SAC) to evaluate security of their algorithm, and also proved that the encryption time of their algorithm is lower than the encryption time of AES algorithm.

G.N. Krishnamurthy and V. Ramaswami[7], presented a new idea to modifying the structure of AES by the addition of a fifth stage called the S-Box rotation at the beginning of each round during the encryption process, while in the decryption process the four stages remained the same as in the original AES algorithm, where the inverse substitute bytes are modified to omitted offset used in the encryption process. They depicted that their algorithm does not consume a lot of implementation time and is resistance against linear and differential cryptanalysis.

J. Juremi, R. Mahmod and S. Sulaiman[8], proposed a new technique for generating a dynamic key-dependent S-Box instead of the static S-Box used in traditional AES algorithm. The researchers used measures of randomness to test their algorithm and obtained results is satisfactory. However, their technology completely replaced the original S-Box of AES algorithm with the new dynamic S-Box and eliminated the Inverse S-Box, which was a violation of the AES design.

J. Juremi, R. Mahmod, S. Sulaiman and J. Ramli[11], have designed a new algorithm based on the S-Box rotation property. The researchers clearly showed how to create S-Boxes that depend on the round key. The structure of the proposed algorithm is very similar to that of the original AES algorithm with the addition of a key-dependent S-Box without changing its values.

R. Hosseinkhani and H.H.S Javadi[12], presented a new algorithm for creating key-dependent S-Boxes. Their algorithm was resistance against linear and differential cryptanalysis. The researchers performed some experiments on their algorithm to conclude that it is capable of creating multiple S-Boxes, and that it improved the security of the original algorithm without modification that violates the original design standards.

H. M. Azzawi[13], suggested generating a dynamic S-Box by combining the output of three keys using a simple EXOR process. The author used avalanche effect criterion to evaluate security of his algorithm and demonstrated that the proposed method is able to prevent cryptanalysis and brute-force attacks.

K. Kazlauskas, G. Vaicekauskas and R. Smaliukas[14], Proposed a key-dependent S-box generation algorithm. The algorithm was examined for randomness testing. Experimental results proved that the generated Cipher-Text sequences are random. Moreover, the proposed algorithm is faster in terms of execution speed compared to the algorithm presented by the same researchers as in [5] and also it is resistance against linear and differential cryptanalysis.

V. PROPOSED S-BOX GENERATION

As we discussed previously in section III, AES describes a 16 × 16 table called an S-Box. This S-Box has the static nature which in turn will allow the attacker to analyse the S-Box and discover its flaws [8]. So, a dynamic S-Box should be constructed at a time, which will make it difficult task for the attacker to analyse the S-Box.

There are many studies that have proposed enhancement of AES using the S-Box rotation depending on the secret key such as [4], [5], [8], [11], [12], [14]. In the most of these studies the S-Box rotation is performed in each AES round using a value that is calculated from the round key. Thus, a fifth stage is added to the original AES called the S-Box rotation stage. Since building and implementation of the S-Box consumes a lot of execution time [6], performing this process in each round will consume an excessive amount of time.

This paper proposed a method to enhance AES by performing the S-Box rotation only once, using the value of the first byte of the secret key, called here a seed value. In the proposed method, the S-Box rotation will be performed only at initial round of both encryption and decryption processes as shown in Figure 3 and 4 respectively. So, the S-Box rotation stage in each round will be eliminated and thus the execution time will be low. The transformation stages in the proposed algorithm will remain the same as it is in AES algorithm.

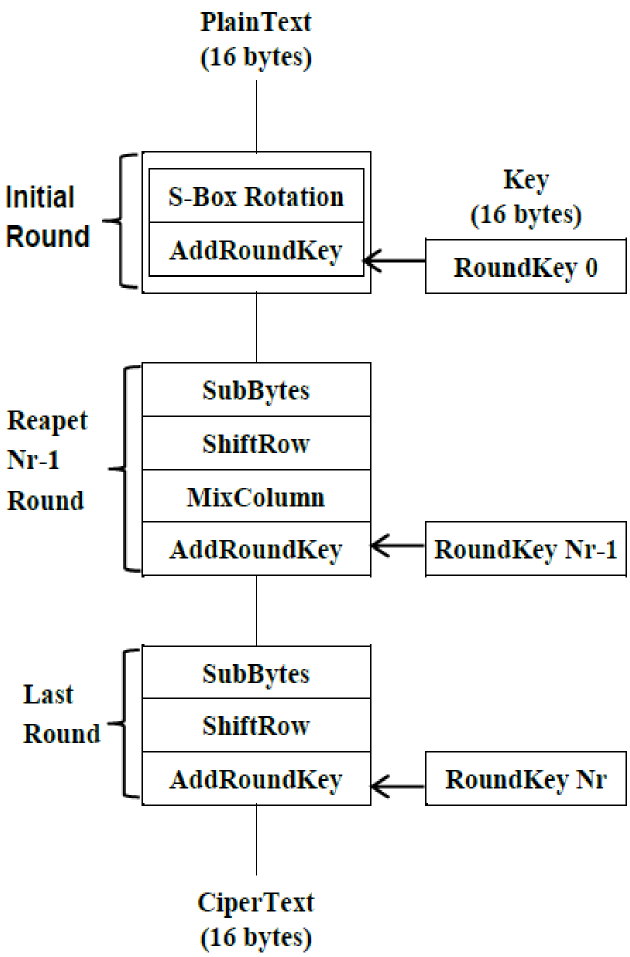

Figure 3: Proposed Encryption Algorithm

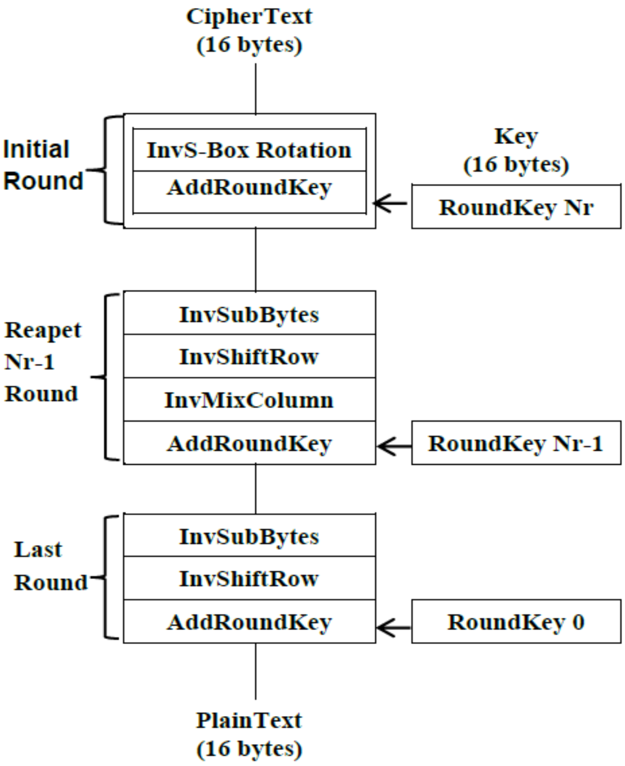

Figure 4: Proposed Decryption Algorithm

Steps of the suggested method are as follows:

Step 1: Assume the secret key agreed upon between the sender and the receiver is :

2A 07 AC B4 7A 58 8D 9E D8 37 A5 3E 92 3C 40 55

Step 2: Compute the value of the first byte of the secret key (first one byte) and use it as a seed value for the S-Box rotation. Here **2A** is the value of the first byte of the secret key, so the Seed value = **2A** and Cipher Key = **2A 07 AC B4 7A 58 8D 9E D8 37 A5 3E 92 3C 40 55**

Step 3: Perform the s-box rotation using the seed value **2A**. For the encryption, the original S-Box will be rotated to the left by the seed value as shown in Table 3. For decryption, Perform the reverse operation using Inverse S-Box, then subtract the result with the seed value used to rotate the S-Box. For example: assume the byte of state is **3F**, for encryption, the corresponding value in the rotated S-Box is **F**9. For decryption, we perform the following :

(InvS-Box(F9) – 2A) mod 256 = (69 – 2A) mod 256 = 3F

Step 4: Perform the steps of AES algorithm from start to end.

Table 3: Rotated S-Box by **2A** to Left

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

VI. EXPERIMENTAL RESULTS

In this paper, the original and enhanced algorithm are implemented in C++ language. The aim of this experiment is for checking confusion and diffusion property used avalanche effect test besides performance test.

A. Avalanche Criteria

In order to considering a block cipher algorithm to be robust, it should satisfy the avalanche effect property. That is, if one bit is changed in the input (plain-text or key), half of the output should be changed [6], [15], [16]. In this experment, 128-

bit encryption is used with plain-text 00000000000000000000000000000000 (hex) and the key 06ACB47D588A9ED837D50E923C4055B5 (hex) as in [8]. Table 4 shows the partial results of experiment that carried out for one bit change in the plain-text.

Table 4. Partial Result of Experiment

No	Plaintext	Ciphertext	Bit variance
1	00000000000000000000000000000000	4E40333F92630F7881CDC183219DAA3B	70
2	00000000000000000001000000000000	CB21D5605591378E2C3498FEA314577B	
3	00000000000000000023000000000000	4018F753EA69F8C7EDBD25B45499539F	67
4	00000000010000000023000000000000	A933895B454D7231862DDAD5BBB97908	
5	00000000BE0000000023000000000000	0B74156E4334223DF24964C5B9DD57D4	71
6	00000000BE0000000023000001000000	D685DC6BC572576E8D8A0F1C24A7CC9D	
7	00000000BE00000000230000E1000000	ACCBDE32996C1D442BF9891520CB3960	69
8	00000000BE00000000230000E1000100	FAF9EF9F06DDD7C89EA657F0227CFF5C	

In this experiment 4000 samples are used for the avalanche effect test. The key is Kept as constant, changing one bit of plain-text in every experiment. Table 5 shows the final results of this experiment.

Table 5. Avalanche Effect Test

Number of samples	Number of samples that satisfy Avalanche test	
	AES Algorithm	Proposed Algorithm
4000	2120	2150

B. Performance Test

The time function in C++ language is used for measurement of the execution time of both original and proposed algorithm during impelemnt all experiments (4000 samples), the results are shown in Table 6.

Table 6. The Execution Time Comparison

Algorithm	Execution Time (Sec)
AES Algorithm	49
The Proposed Algorithm	51

VII. CONCLUSION AND FUTURE WORK

In this paper, a proposed method is presented in order to improve the security of AES algorithm by making its S-Box generation randomly based on a random value which computed from the secret key. In the proposed method, the S-Box rotation process is only performed one time in order to avoid increasing of the execution time, as it is shown in Table 6, the difference between the execution time of original and proposed algorithm is only 2 (Sec). Furthermore, performing of the S-Box rotation one time as such will be sufficient to generate a random and unknown S-Box which in turn will increase the number of possibilities that the attacker will encounter, because the S-Box is unrecognizable for the attacker.
In the future, the researchers will improve AES algorithm by S-Box expansion.

REFERENCES

[1] J. Daemen, and V. Rijmen, "*AES Proposal:Rijndael*", Version 2. Submission to NIST, **1999**.

[2] J. Daemen, and V. Rijmen,"*The block cipher Rijndael*", Proceedings of the Third International Conference on smart card Reseacrh and Applications, CARDIS'98, 1820, pp.**277-284**, **Berlin: Springer, 2000**.

[3] Advanced Encryption Standard (AES). Federal Information Processing Standards Publication **197**, **2001**.

[4] K. Anurupam, "*Dynamic S-box implementation in PRESENT Cipher*", International Journal of Computer Sciences and Engineering, Vol.**6**, Issue.**9**, pp.**426-431**, **2018**.

[5] K. Kazlauskas and J. Kazlauskas, "*Key-Dependent S-Box Generation in AES Block Cipher System*", INFORMATICA, Vol.**20**, No.**1**, pp.**23-34**, **2009**.

[6] H.M. El-Sheikh, O.A. El-Mohsen, S.T. Elgarf and A. Zekry, "*A New Approach for Designing Key-Dependent S-Box Defined over GF (2^4) in AES*", International Journal of Computer Theory and Engineering Vol.**4**, No.**2**, pp.**158-164**, **2012**.

[7] G.N. Krishnamurthy and V. Ramaswami, "*Making AES Stronger: AES wich Key - Dependent S-Box*", International Journal of Computer Science and Network Security, Vol.**8**, No.**9**, pp. **388-398**, **2008**.

[8] J. Juremi, R. Mahmod and S. Sulaiman, "*A Proposal for Improving AES S-box with Rotation and Key-Dependent*", in the proceedings of the 2012 IEEE International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), Kuala Lumpur, **Malaysia**, pp.**38-42**. **2012**.

[9] W. Stallings, "Cryptography and Network Security", principles and practices,7th Edition, Pearson Prentice Hall, pp.172-189, 2017.

[10] A. Karki, "*A Review on Advanced Encryption Standard*", International Journal of Computer Sciences and Engineering (ICSE), Vol.**6**, Issue.**8**, pp.**551-556**, **2018**.

[11] J. Juremi, R. Mahmod, S. Sulaiman and J. Ramli, "*Enhancing Advanced Encryption Standard S-Box Generation Based on Round Key*", International Journal of Cyber-Security and Digital Forensics, Vol.**1**, No.**3**, pp.**183-188**, **2012.**

[12] R. Hosseinkhani and H.H.S Javadi, "*Using Cipher Key to Generate Dynamic S-Box in AES Cipher System*", International Journal of Computer Science and Security, Vol.**6**, No.**1**, pp.**19-28**, **2012**.

[13] H. M. Azzawi, "*Enhancing The Encryption Process of Advanced Encryption Standard (AES) By Using Proposed*

Algorithm to Generate S-Box", *Journal of Engineering and Development*, Vol.**18**, No.**2**, **2014**.

[14] K. Kazlauskas, G. Vaicekauskas and R. Smaliukas,"*An Algorithm for Key-Dependent S-Box Generation in Block Cipher System*", INFORMATICA, Vol.**26**, No.**1**, pp.**51-56**, **2015**.

[15] N. Tiwari and A. Kumar, "*Security Effect on AES in Terms of Avalanche Effect by Using Alternate S-Box*", in Proc. Int. Conf. Intell. Data Commun. Technol. Internet Things, pp.**1-14**, **201**8.

[16] A. Datta, D. Bhowmik and S. Sinha, "*A Novel Technique for SAC Analysis of S-Boxes for Boomerang-Style Attacks*", International Journal of Computer Sciences and Engineering (ICSE), Vol.**7**, Issue.**5**,pp.**7-13**, **2019**.

AUTHORS PROFILE

Rawia Mohammed Alkhamery pursed B.Sc. electrical engineering (computers &control) from Sana'a University, from Sana'a University, Yemen. She is currently pursuing Master of Computer Science, Department of Computer Science, Sheba Region University, Yemen.

Mr. Yasser Ali Alahmadi, pursed B.Sc. CS from Sana'a University, Yemen. He is currently pursuing Master of Computer Science, Department of Computer Science, Sheba Region University, Yemen. His interest research area: Information Security and C# Programming.

Mr. Mokhtar Alsorori, pursed B.Sc. CS from Al-Neelain University, Sudan in 2003, M.Sc. IT from Mysore University, India in 2007 and Ph.D. in Computer Science from Kakatiya University, India in 2020. He is currently working as Assistant Professor in Department of Information Systems, Sheba Region University, Yemen. His interest research area: Cyber Security and Digital Image Watermarking.

Mr. Saleh Alassali, pursed B.Sc. CE from KSU University, Saudi Arabia in 1988, M.Sc. CS from Pune University, India in 2000 and Ph.D. in Information Security from SRTM University, India in 2005. He is currently working as Associated Professor in each of Department of Computer Sciences, Sheba Region University, Science and Technology University, Yemen. He has published more than 8 research papers in reputed international journals. His main research work focuses on Cryptography Algorithms and Random Number Generators. WhatsAPP: 770317665, Yemen.

