iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** IV **Month of publication:** April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4239

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

Enhancement of Advanced Encryption Standard (AES) Cryptographic Strength via Generation of Cipher Key-Dependent S-Box

Marione Ver C. Suana[1], Dr. Ariel M. Sison[2], Dr. Cristina Aragon[3], Dr. Ruji P. Medina[4]
[1, 2, 3, 4] *Graduate School Department. Technological Institute of the Philippines – Quezon City*

Abstract*: The rapid increase of information transmitted electronically resulted to an increased reliance on cryptography and authentication. Cryptography is the science of using mathematics to encrypt and decrypt data to store sensitive information or transmit it across insecure networks. Advanced Encryption Standard block cipher (AES) is a widely used cryptographic block cipher system, the AES substitution box (S-Box) are generally inferred as static arrays and publicly known thus making it prone in Differential Fault Analysis (DFA) attacks (also often called side channel attacks). In this paper, a proposed method for constructing dynamic Cipher Key dependent S-box is introduced and implemented to encounter the possible attack on the fixed S -Box. Also, the qualities of the implemented S-boxes are examined based on the four major criteria of a good S-box. A comparative analysis is conducted that compared the randomness properties and simulation time of the proposed AES against the standard AES algorithm. The proposed dynamic key dependent S-box passed the Avalanche, bit independence, non-linearity and balance test which proven its security.

Keywords: Cryptography, Encrypt, Decrypt, Networks, Advanced Encryption Standard (AES), Block cipher, Differential Fault Analysis, Substitution box (S-Box).

I. INTRODUCTION

The protection of sensitive information against unauthorized access or fraudulent changes has been of prime concern throughout the centuries. Modern communication techniques, using computers connected through networks, make all data even more vulnerable for these threats [1]. Cryptography plays a critical role in the safety of information. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. While cryptography is the science of securing data, cryptanalysis is the science of analysing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called attackers [2].

The Encryption technique is employed in two ways, namely Asymmetric Encryption and Symmetric Encryption [3]. Asymmetric Encryption is a relatively new and complex mode of Encryption. Complex because it incorporates two cryptographic keys to implement data security. These keys are called a Public Key and a Private Key. The Public key is available to everyone who wishes to send a message. On the other hand, the private key is kept at a secure place by the owner of the public key [3] Popular asymmetric key encryption algorithm includes EI Gamal, RSA, DSA, Elliptic curve techniques and PKCS. Symmetric encryption is a conventional method of Encryption. It is also the simpler of two techniques. Symmetric encryption is executed by means of only one secret key known as 'Symmetric Key' that is possessed by both parties. This key is applied to encode and decode the information, Blowfish, AES, RC4, DES, RC5, and RC6 are examples of symmetric encryption. The most widely used symmetric algorithm is AES-128, AES-192, and AES-256.

The Advanced Encryption Standard (AES) is formal encryption method adopted by the National Institute of Standards and Technology (NIST), and is accepted worldwide [4].

The only non-linear component of AES block cipher systems is the Substitution box (S-Box) which are static in structure and do not depend on the undisclosed key, Numerous analysts have experimented and have revealed that there are some weak points in the configuration and construction of the current static S-Box since it is fixed and known to all.

Cryptanalyst managed to come up with a Differential Fault Analysis (DFA) attacks that can recover the secret key easier than anticipated[5], A cryptographically strong key dependent dynamic S-box for AES can provide security against this known weakness and resist side-channel attack.

II. REVIEW OF RELATED LITERATURE

The Advanced Encryption Standard is formal encryption method adopted by the National Institute of Standards and Technology (NIST) of the US Government, and is accepted worldwide [4]. AES is an encryption algorithm proposed by the Belgium cryptographers Joan Daeman and Vincent Rijmen for the competition held by NIST. Prior to selection Daeman and Rijmen used the name Rijndael (derived from their names) for the algorithm. Rijndael. Mars, RC2, Surpent and two fish are the final competitors. In 2000 NIST announced Rijndael algorithm was the winner by analyzing various security parameters and other characteristics. After adoption the Rijndael encryption algorithm was given the name Advanced Encryption Standard (AES) which is in common use today. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively [6]. The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor, the Data Encryption Standard (DES) [7]. The AES encryption has four transformations namely the Add Round Key, Sub Byte, Shift Row and Mix Column.

In recent years, many relevant researchers have constructed dynamic S-box for AES whose replacement is contacted with the cipher key. In this method, the more the choices of replacement are, the higher the encryption intensity of dynamic S-box will be. A kind of dynamic S-boxes has been put forward in and the analyses show that the proposed enhanced possesses better differential properties.

A paper presented to propose a new s-box design generated by the well know RC4 stream cipher algorithm was introduced[6], RC4 is designed in 1987 by Ron Rivest, RC4 is variable key size stream cipher with byte oriented operation. The algorithm is based on the use of a random permutation of 256 bit state. It used in WEP and SSL/TLS (secure socket layer/transport layer security)[8]. In this design the proponents use the key expansion algorithm output which is 176, 208, 240 bytes long key is used to supply the permutation to generate the proposed S-box. On the other front, from the paper “Towards the Generation of a Dynamic Key-Dependent S-Box to Enhance Security” In author paper proposed scheme, for generating the Dynamic key dependent S-Box, the AES S-Box is considered as the standard and used as the look-up table[9]. For simplicity, the keys and the Dynamic key dependent S-Box function are represented in hex. With the key dependent Dynamic S-Box function, each of the 256 possible byte values is being transformed to a different byte based upon the code word generated from the key (a complete permutation). Every input gets changed, and all 256 possible elements are represented as the result of a change. Moreover, no two different bytes are mapped on to the same byte. The key used for encryption is considered to be 64 bits in length (eg., A451 B672 90F7 DE38). From the key, a codeword of 8 bits (C8C7C6C5C4C3C2C1) is generated at run-time based upon the Hamming Distance and Hamming Weight Another method to generate a dynamic S-box was proposed from the paper titled “Constructing Key Dependent Dynamic S-Box for AES Block Cipher System” the researchers aim to propose a novel strategy for improving the cryptographic strength of AES algorithm[10]. Also an attempt is made to enhance the complexity and resistance of AES by making its S-box to be uniquely key dependent. Pre-defined S-box suggests that the similar S-box will be utilized in every individual round while in the case of the proposed key-dependent random and dynamic S-box the new generated S-box retains from changing in every round with reference to expanded round key and number of working rounds. Predefined S-Box influence intruders to analyze S-box and discover the crucial facts but by utilizing key dependent S-Box approach, it makes it more intricate and troublesome for attackers to perform any offline inspection of an attack of single specific arrangement of S-boxes.

The comparison of RC4 algorithm, CodeWord Algorith and S-box Generator Algorithm shows that the RC4 algorithm is the best possible solution for the Weak implementations of AES and the enhancement provides resistance against side-channel attacks which is a significant security threat in AES cipher system. In Addition with that based from the paper “Randomness analysis and generation of key-derived s-boxes” have analysed S-boxes generated by RC4, Blowfish and Twofish, comparing them to pseudorandom s-boxes [11]. According to the study the RC4 Key derived S-boxes passes all of the Avalanche criteria, Bit Independence Criteria (BIC) and Balance Criteria. Which are important features for strong S-boxes to produce more confusion to the encryption process. The comparisons show that it has better results compared to Blowfish and Twofish. The Study also proved that key-derived s-boxes are essentially equivalent to random s-boxes, and can be equally employed as non-linear elements in many types of cryptosystems (stream and block ciphers, hash functions, etc.)

To deal with multiple vulnerabilities of RC4 algorithm and now over 30 years since published, Rivest and Schuldt published a paper ”Spritz- a spongy RC4-like stream cipher and hash function” to propose an improved variant, which call “Spritz”, a newer and stronger design and replacement for RC4. Spritz attempts to repair weak design decisions in RC4, while remaining true to its general design principles [12]. The Spritz design not only provides a “drop-in replacement” for RC4, with much improved security properties, but also provides a suite of new cryptographic functionalities based on its new “sponge-like” construction and interface [13].

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

III. AES WORKING PRINCIPLE

A. AES Rounds

The size of the key dictates how many rounds have to perform. With larger keys corresponding to more rounds and more secure but slower encryption, 128bits key performs 10 rounds cycle, 192bits have 12 performs cycle and 14 round cycle for 256bits key. Then the round processing occurs consisting of operations of the, SubBytes using Rijndael S-box, ShiftRow, MixColumn and a AddRoundKey. This is done for all rounds, with the exception of the MixColumn operation to the final round. The final result is the encrypted cipher block [14]. The AES block diagram is shown Fig. 1.

1) Substitution Bytes Transformation is a nonlinear byte substitution, using a substitution box (S-box), which is constructed by multiplicative inverse and affine transformation. It provides nonlinearity and confusion.

2) Shift Rows Transformation is a simple byte transposition, the bytes in the last three rows of the state are cyclically shifted. The offset of the left shift varies from one to three bytes. It provides inter-column diffusion.

3) Mix Columns Transformation is equivalent to a matrix multiplication of columns of the states. Each column vector is multiplied by a fixed matrix. It should be noted that the bytes are treated as polynomials rather than numbers. It provides inter-byte diffusion.

4) Add Round Key Transformation is a simple XOR between each byte of the state and the round key. This transformation is its own inverse. It adds confusion [15].

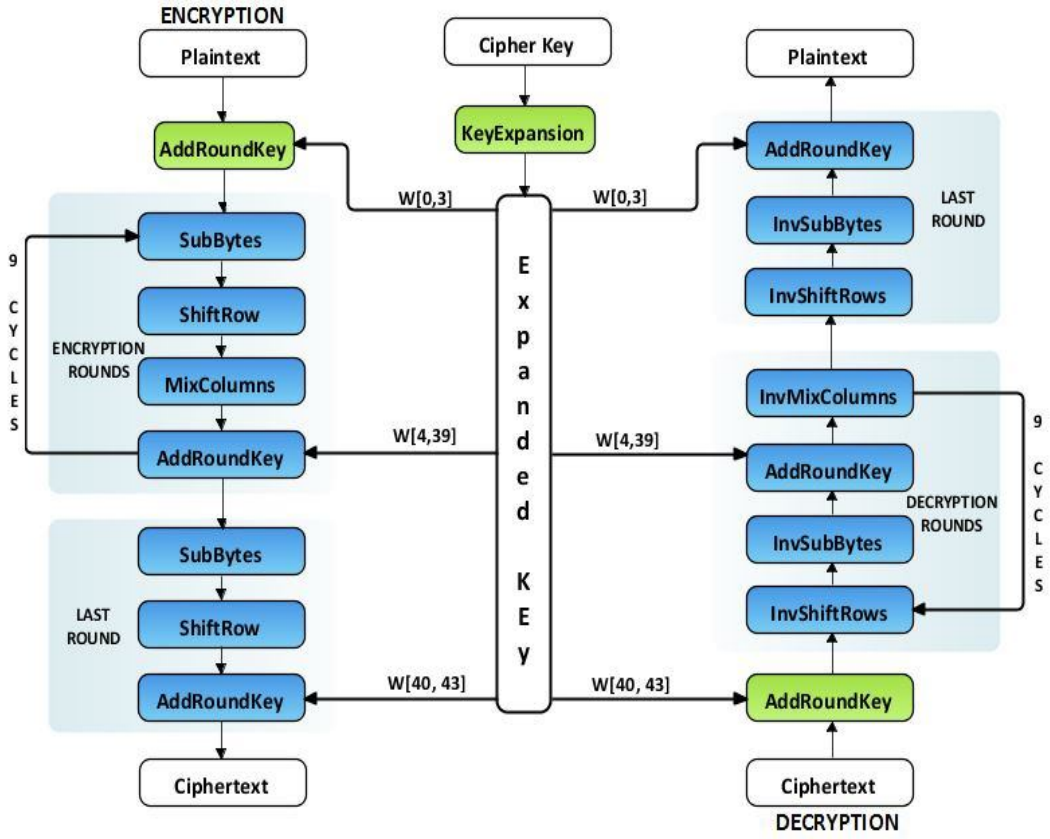


Figure 1: Advanced Encryption Standard 128bits Block Diagram.

B. AES Substitution Box (S-box)

The first step to a round is to do a byte by byte substitution with a lookup table called a Substitution-box or simply S-box. An S-box is a one to one mapping for all byte values from 0 to 255 in 16 x 16 array. Substitution is a nonlinear transformation which performs confusion of bits. A nonlinear transformation is essential for every modern encryption algorithm and is proved to be a strong cryptographic primitive against linear and differential cryptanalysis. The S-box is shown Table 1. All values are represented in hexadecimal notation[14].

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

C. AES Attack

A brute force attack on AES 128bits (Shortest key) requires at least 2^128 = 3.4 * 10^38 alternative keys and if an attacker have a powerful computer that can calculate 10^6 decryptions/µs then in worst case scenario it will take 5.4 * 10^18 years to find the correct key [16], Due to the large key space and high computational complexity the brute-force attacks are not threatening to the security of AES. However the other method to retrieve the key which can be exploited to break the system is to look at information that can be gained from any physical implementation of the algorithm this type of attack is called side channel. Side-channel attack is any attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms [17], one of the most common type of side channel attack is Differential Fault Analysis (DFA) attacks, the principle of these attack is to replace or modify the instructions and induce faults in the cryptosystem and the cryptanalysis studies the resulting output for useful information [18].

The traditional AES S-Box are generally inferred as static arrays in software implementations for efficiency reasons , But if an attacker can gain access to the compiled code, i.e. the software binary, it is a trivial matter to load the binary using a tool such as Hex Editor and view the compiled source. If a lookup table has been statically defined it will be easily identifiable. For example, It is publicly known that the initial bytes of the AES S-box are[63 7C 77 7B] these can be located in the compiled source of an AES implementation by simple search [5]. One of the widely used software application that implements AES encryption is WinRAR, Figure 2 shows the actual scanning for AES S-box from the WinRAR software binary using Hex Editor. Once S-box has been found it can be changed to arbitrary values chosen by an attacker, resulting in potentially dangerous security breaks and can make the encryption key roll out [19]. This proves that the side-channel attacks are the most significant security threat in AES cipher system. It is very important that software developers takes these types security issues into account when implementing AES encryption[16].

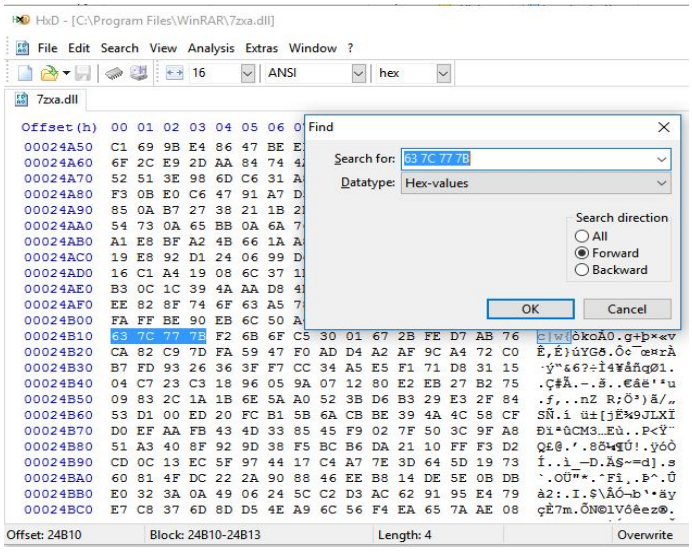


Figure 2: Locating the AES Sbox in WinRar software binary using Hex Editor.

In this paper, a new method for constructing cryptographically strong Cipher Key dependent S-box will be introduced. Although there have been many researchers on the Dynamic S-box, but most of the existing algorithms have several weaknesses either caused by low security level or largely increase the delay time due the design of the algorithm itself. The propose dynamic S-box aims replace the traditional Static AES S-box to enhance the complexity of the S-Box structure while addressing the threat of Differential Fault Analysis (DFA) attacks.

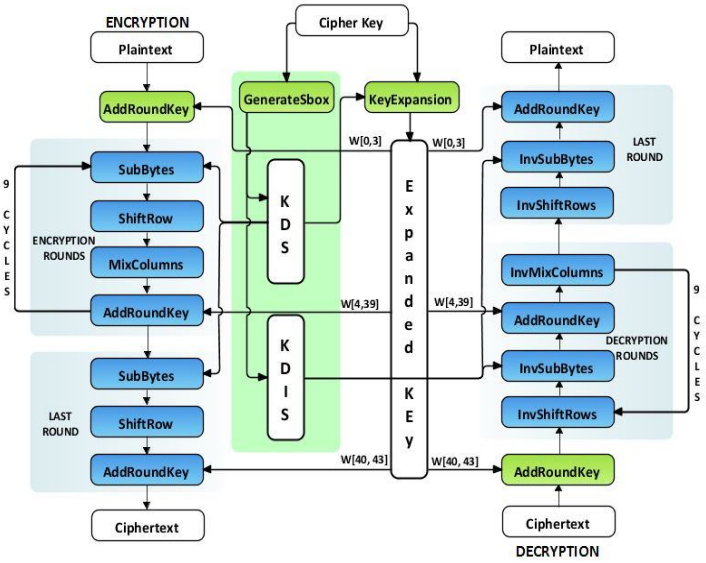


Figure 3: Proposed AES 128bit Algorithm Block Diagram.

The proposed method for deriving S-box from cipher key will be the initial process of AES cipher system this process will be called “Generate S-box”. An overall process of the proposed AES algorithm is shown in figure 3.

D. AES Enhancement

The algorithm that will use to generate key dependent s-box was inspired from Spritz stream cipher and hash function algorithm. Spritz Cipher is a sponge-based replacement for RC4 Cipher Stream, proposed by Rivest & Schuldt[13] the authors include a new permutation algorithm that was found to have the better statistical properties than RC4 after significant computational testing. This is meant to provide security against the known weaknesses and attacks on the original RC4 including the biases found in Key Scheduling (KSA) [13][11].

The Generation of AES key Dependent S-box using Spritz Cipher Stream will be as follow:

1) Initialization of array S and temporary vector after given the Cipher key.

```
For i = 0 to 255
        S [i] = i
        T [i] = CK [i mod CK.length]
```

The entries of S will be an array set equal to the values from 0 through 255 in ascending order; that is S[0] = 0, S[1] = 1, . . . , S[255] = 255.

Temporary vector T is created to makes use of the variable length of Cipher Key (EK) to initialize a 256 Bytes array.

2) Now T will use to produce the permutation of S. This involves starting with S[0] and going through S[255] and for each S[i] swapping S[i] with another byte in S according to a scheme dictated from sum of j, K, S[i] and T[i] mod by 256 .

```
Initial value for J and K is 0
For i = 0 to 255
    j = (k + j + S[i] + T[i]) mod 256
    k = (k + i + S[j]) mod 256
    swap S[i], S[j]
```

The output of previous steps gives 256 values, all of which depends on the input key. This means that if one byte was change from the cipher key, the output will get another different set of 256 values. This Algorithm constructs the 256 value of S-box depending on the cipher key.

IV. RESULTS AND DISCUSSION

In order to ensure that proposed dynamic AES-128bits key dependent S-box is secure and cryptographically strong, some tests and analysis must be applied. The investigation is divided into two phases: comparative analysis, and security analysis.

A. Comparative Analysis

In Comparative Analysis will test the randomness properties and simulation time of the proposed modified AES compared to standard AES. Randomness test in data evaluation are used to analyse the distribution of a set of data to see if it is random (pattern less) and simulation time to record the time required by the algorithm to process a complete encryption and decryption.

B. Security Analysis

The proposed scheme will be analyze based on the features that any S-box must exhibit in order to be an effective s-box. Each of these features contributes to the security of the S-box. Those features are Strict Avalanche Criteria, Bit Independence Criteria, Nonlinearity and Balance [20] [21] [22].

The following are a brief description of the four criteria:

1) Strict avalanche criterion (SAC) occurs if one input bit i is changed, each output bit will change with probability of one half. Strict avalanche requires that if there are any slight changes in the input vector, there will be a significant change in the output vector. To achieve this effect, The S-box will need a function that has a 50% dependency on each of its n input bits.

2) Bit independence criterion (BIC) or correlation-immunity requires that output bits act independently from each other. In other words, there should not be any statistical pattern or statistical dependencies between output bits from the output vectors.

3) Nonlinearity requires that the S-box is not a linear mapping from input to output. This would make the cryptosystem susceptible to attacks. If the S-box is constructed with maximally nonlinear Boolean functions, it will give a bad approximation by linear functions thus making a cryptosystem difficult to break.

4) Balance means that each Boolean vector responsible for the S-box has the same number of 0's and 1's.

C. Comparative Analysis Result

A Runs Test was conducted using MATLAB r2017b to check the randomness property of the generated s-box vs. standard S-box. The result in runs test is based on the number of runs of consecutive values above or below the mean of the test value. Too few runs indicate a tendency for high and low values to cluster. Too many runs indicate a tendency for high and low values to alternate. MATLAB Runs test also uses a test statistic which is the difference between the number of runs and its mean, divided by its standard deviation. The test statistic is approximately normally distributed when the null hypothesis is true. Null hypothesis is the hypothesis that there is no significant difference between specified set of values, if If hypothesis test = 1, then runs test rejects the null hypothesis And if hypothesis test = 0 then runs test fails to reject the null hypothesis. The test result returned value of h = 0 indicates that runs test does not reject the null hypothesis that the values in x are in random order at the default 5% significance level. which declares that the proposed key derived AES S-box passed the run test therefore the generated S-box produces independent, uniformly distributed and unpredictable values but with different test result values compared with the standard S-box. The test result is shown at Table 3.

TABLE 2
RANDOMNESS TESTS RESULT OF PROPOSED AES S-BOX AND STANDARD AES S-BOX.

	Standard AES S-box	Proposed AES S-box
Number of runs	127	120
The number of 0's	128	128
The number of 1's	128	128
Hypothesis test result	0	0
Test statistic	-0.1879	-1.0646

Figure 4 represents the value of Key Derived S-box generated from "ThisisaSamplekey" Cipher key, Figure 5 is the line graph plots the S-box value from index 0 to 256, This visual observation appears that the transition pattern from one index to another are different, not correlated and clearly random.

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

Figure 4: Generated S-box using "ThisisaSamplekey" as a Cipher Key.

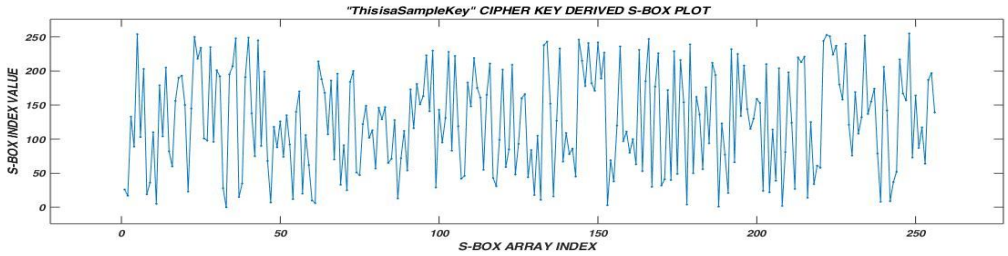


Figure 5: A line graph of generated S-box using "ThisisaSamplekey" as a Cipher Key.

simulation time taken by two different algorithms is recorded in nanosecond shown in table 4-4. It is clear that Standard AES is faster than the proposed AES. The difference in simulated time is the result of proposed S-box dynamic change which is anticipated since the proposed method consist more cryptographic algorithm.

TABLE 3
SIMULATION TIME

Standard AES		Proposed AES	
Encryption	Decryption	Encryption	Decryption
1100446 ns	247556 ns.	1220779 ns.	247556 ns.
Total Time = 1348002 ns.		Total Time = 1468335 ns.	
Difference = 120333 ns. / 0.000120333 s			

D. Security Analysis Result

1) Strict avalanche criterion (SAC) – the avalanche effect test was applied to generated S-box. Only one bit is changed to perform this test, for Generated key dependent S-box avalanche test the first value of cipher key 00,00,00,00,00,00,00,00 was changed from 0 to 1. The generated key derived S-Box obtained 51.4648% avalanche effect as shown in Table 5. The test demonstrates that proposed AES algorithm features strict avalanche effect.

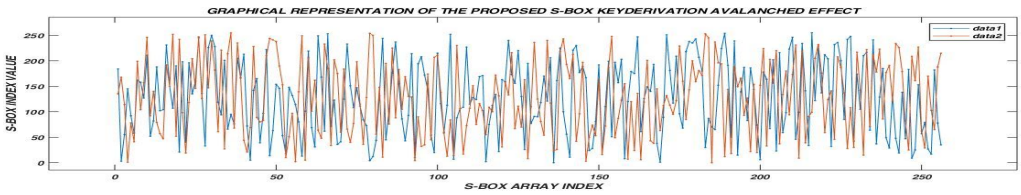


Figure 6: A line graph of generated S-boxes Avalanched effect.

Figure 6 illustrate the avalanche effect of the proposed S-box value that a change in one bit of the cipher key produces a huge or more than a half change in generated S-box. The blue line graph represents the generated S-box when the key is 00,00,00,00,00,00,00,00 and the red line graph represents the generated S-box when the key is 10,00,00,00,00,00,00,00 the changes of the output vectors are clearly visible.

2) *Bit independence criterion (BIC)* - In order to measure the degree of independence between a pair of avalanche variable, A MATLAB 'corr' test were applied to calculate their correlation coefficient, if its zero it means that the variable are independent, if its 1 that means stronger positive correlation and -1 is stronger negative correlation. The value obtained from bit independence test of AES key derived S-box is -0.0293 as shown in Table 5, The results shown that the generated s-boxes features a strong negative correlation which means the output bits act independently from each other.

TABLE 4
AVALANCHED AND BIT INDEPENDECE TEST RESULT

	Cipher Key	Avalanche TestResult	BIC Test Result
1st Run	00,00,00,00,00,00,00,00	51.4648 %	-0.0293
2nd Run	10,00,00,00,00,00,00,00		

3) *Nonlinearity* - Key derived S-box features Nonlinearity since the dynamic key dependent S-Boxes are generated from key in sufficiently random fashion stated on Randomness test shown in Table 3 and in addition to that the generated S-box passed the Nonlinearity test of SET (S-box Evaluation Toolbox) shown in fig. 6, SET is a tool for the analysis of cryptographic properties of Boolean functions and S-boxes that is written in ANSI C code [23].

4) *Balance* - An S-box with n input bits and m output bits, $m \leq n$, is balanced if each output occurs 2^{n-m} times. For the S-box to be balanced it should have the same number of 0's and 1's. As per the result from SET (S-box Evaluation Toolbox) balance calculation the generated key derived S-box is also Balance as shown in fig. 7.

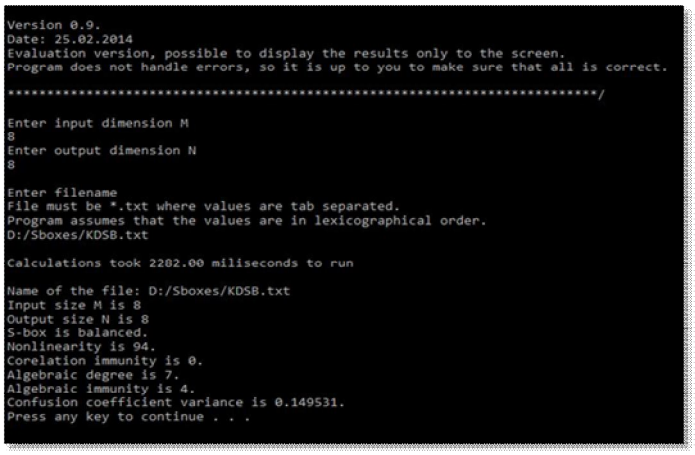


Figure 7: key derived S-box test result via SET (S-box Evaluation Toolbox)

V. CONCLUSION

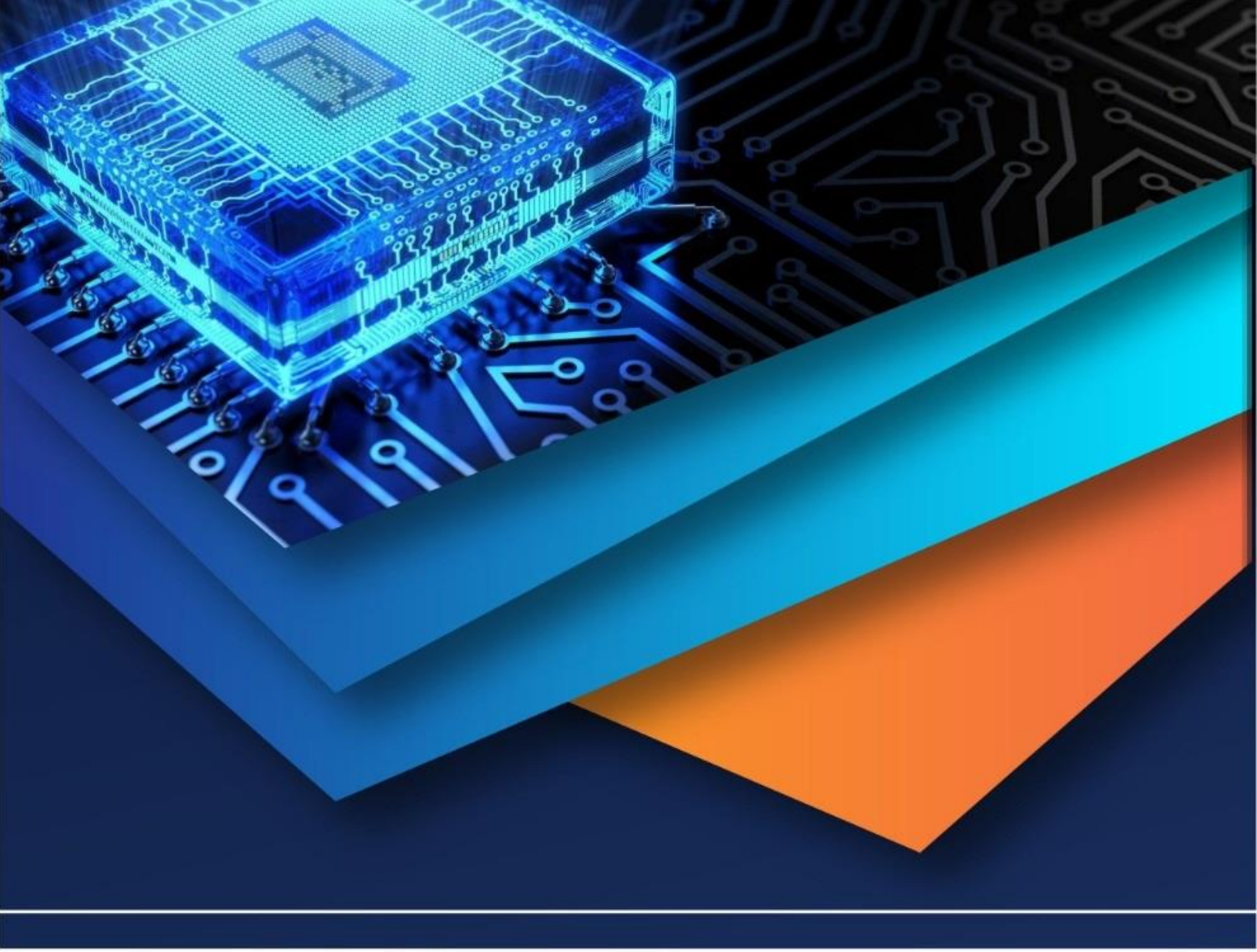
In this paper a new method for constructing cryptographically strong key dependent S-box was introduced. The proposed S-box passed the randomness, avalanche, bit independence, nonlinearity and balance tests which are important features of S-boxes to produce more confusion to the AES encryption process. The proposed method can integrate higher degree of security and can be adopted to enhance the complexity of the S-Box structure and thereby resist side-channel attack.

VI. ACKNOWLEDGMENT

The researcher would like to take this opportunity to thank those who contribute their valuable time, support, comments, suggestions and persuasion. And to acknowledge the financial support provided by the Commission on Higher Education (CHED), Kto12 Project Management Unit, Philippines.

REFERENCES

[1] N. Smart, “Cryptography: An Introduction,” J. Appl. Clin. Med. Phys., vol. 12, no. 1, p. 3428, 2010.
[2] J. Hoffstein, J. Pipher, and J. H. Silverman, An Introduction to Cryptography, vol. XVI. 2008.
[3] Obaida Mohammad and A. Al-Hazaimeh, “A New Approach for Complex Encrypting and Decrypting Data,” Int. J. Comput. Networks Commun., vol. 5, no. 2, p. 88, 2013.
[4] A. Ferah, “AES Encryption AES Encryption and Related Concepts,” pp. 0–4, 2016.
[5] J. Kremers, “Practical hacking AES using the S-box weakness,” 2011.
[6] S. V. Radhakrishnan and S. Subramanian, “An analytical approach to s-box generation q,” Comput. Electr. Eng., vol. 39, no. 3, pp. 1006–1015, 2013.
[7] H. D. H. Knebl, “1. Introduction to Cryptography,” Springer-VerIag Berlin Heidelb. 2015, pp. 1–10, 2002.
[8] M. Panda, “Performance Analysis of Encryption Algorithms for Security,” pp. 840–844, 2016.
[9] G. Jacob, A. Murugan, and I. Viola, “Towards the Generation of a Dynamic Key-Dependent S-Box to Enhance Security.,” IACR Cryptol. ePrint Arch., vol. 2015, p. 92, 2015.
[10] G. Manjula and H. S. Mohan, “Constructing key dependent dynamic S-Box for AES block cipher system,” Proc. 2016 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. iCATccT 2016, no. 10, pp. 613–617, 2016.
[11] R. Álvarez and A. Zamora, “Randomness analysis and generation of Key-Derived S-Boxes,” Log. J. IGPL, vol. 24, no. 1, pp. 68–79, 2014.
[12] S. Banik and T. Isobe, “Cryptanalysis of the Full Spritz Stream Cipher RC4 Stream Cipher,” pp. 1–25, 2016.
[13] R. L. Rivest and J. C. N. Schuldt, “Spritz — a spongy RC4-like stream cipher and hash function,” 2014.
[14] J. Daemen and V. Rijmen, The Rijndael Block Cipher: AES Proposal. 2003.
[15] A. Shireen, S. Mukhtar, G. Farheen, and S. Mukhtar, “An Introduction of Advanced Encryption Algorithm : A Preview,” vol. 3, no. 12, pp. 2012–2015, 2014
[16] J. Mönttinen, “The Security of Advanced Encryption Standard,” 2015.
[17] H. Bar-El, “Whitepaper on Introduction to Side Channel Attacks,” Secur. Integr. Circuits Syst., 2010.
[18] I. Verbauwhede, D. Karaklajić, and J. M. Schmidt, “The fault attack jungle - A classification model to guide you,” Proc. - 2011 Work. Fault Diagnosis Toler. Cryptogr. FDTC 2011, pp. 3–8, 2011
[19] T. Kerins and K. Kursawe, “A Cautionary Note on Weak Implementations of Block Ciphers,” 1st Benelux Work. Inf. Syst. Secur., 2009.
[20] C. Easttom, “A Guideline for Designing Cryptographic S-Boxes by Chuck Easttom Keywords: Cryptography, S-Box, Block Cipher,” pp. 1–8
[21] J. M. Cheung, “The Design Of S-Boxes,” San Diego State Univ., 2010.
[22] M. Ahmad, F. Ahmad, Z. Nasim, Z. Bano, and S. Zafar, “Designing chaos based strong substitution box,” 2015 8th Int. Conf. Contemp. Comput. IC3 2015, pp. 97–100, 2015.
[23] S. Picek, L. Batina, and D. Jakobovi, “A Toolbox for S-box Analysis,” pp. 140–149, 2014.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY