Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 9 (2014) pp. 213-221
© MIR Labs, www.mirlabs.net/jias/index.html

Statistical Analysis of S-Box in Rijndael-AES Algorithm and Formulation of an Enhanced S-Box

Juliet N. Gaithuru[1] **and Majid M. Bakhtiari**[2]

[1]Universiti Teknologi Malaysia, Faculty of Computing,
Skudai, Johor 81310, Malaysia
julietgaithuru@yahoo.com

[2]Universiti Teknologi Malaysia, Faculty of Computing,
Skudai, Johor 81310, Malaysia
bakhtiari.majid@gmail.com

Abstract: **The internet is widely used to support services ranging from education, medicine, entertainment to e-commerce. Cryptography helps to safeguard this transmitted information using authentication, digital signatures and encryption algorithms. In this paper we focus on AES, a symmetric block cipher algorithm which implements the Rijndael algorithm. We present a statistical analysis of the Rijndael-AES S-Box so as to evaluate the weaknesses present in the S-Box. These tests evaluate susceptibility of the AES S-Box to algebraic and statistical attacks. Using the obtained results, a technique of formulating a more non-linear S-Box is suggested. This technique uses the incursive congruential method, which produces highly non-linear output with a lower degree of correlation than the current AES S-Box.**

Keywords: Cryptography, Advanced Encryption Standard (AES), Rijndael, substitution box (S-Box), cryptanalysis, frequency, serial, correlation, non-linearity, pseudo-random number generator (PRNG).

I. Introduction

Information is a fundamental part of an organization or individual. Often times, this information needs to be transmitted between two parties over a public channel, which exposes the information to threats such as eavesdropping and unauthorized modification by attackers. In order to safeguard the information from these threats, cryptography is employed [1].
Cryptography has become a necessity with the rising popularity of the use of the internet for daily aspects of life such as communication, e-commerce, entertainment, education, medicine and electronic banking transactions. Cryptography helps to ensure that the information being transmitted and received is secure and that the fundamental principles of information security are met; confidentiality, integrity, non-repudiation and authentication. This is achieved through the use of authentication, digital signatures and encryption algorithms [2].
Encryption algorithms are broadly categorized into two: symmetric key algorithms and asymmetric key algorithms. Symmetric algorithms use a single shared secret key for both encryption and decryption of messages transmitted between two parties. Examples of symmetric key algorithms include AES, 3DES, CAST, RC6 and Blowfish. However, symmetric key algorithms have one major shortcoming of key distribution because the key must be transmitted to the communicating parties over the public channel prior to communication [1, 3]. On the other hand, asymmetric key algorithms use a public key for encryption by the sender and a private key for decryption by the recipient. Examples of asymmetric key algorithms include ECC, RSA, ElGamal, Knapsack, Rabin and the more recent NTRU [4].
AES is one of the most common symmetric encryption algorithms being implemented. AES replaced 3DES and in terms of resource consumption, AES is faster and more efficient than 3DES thus resulting in less resource consumption. This replacement was done following an inquiry by NIST into a replacement following a collaboration between the Electronic Frontier Foundation and distributed.net resulted in a publicized breaking of the DES key in 22 hours and 15 minutes [5].When the performance of AES was compared to that of 3DES, DES and RC2 on a computer with a P-4 2.4GHz CPU processor, the results showed that AES had better performance than 3DES, DES and RC2 [4].
Rijndael algorithm was selected as the encryption algorithm for the Advanced Encryption Standard (AES) in October 2001 after a rigorous 3-year evaluation process and published as FIPS 197 in the Federal Register in December 2001 [6]. NIST selected Rijndael as the standard symmetric key encryption algorithm in AES over four other candidate algorithms; MARS, RC6, Serpent and Two-fish. Its comparative performance in relation to the other four finalists ranked as the best in terms of speed, code compactness, design simplicity and resistance against attacks.
Rijndael is a byte-oriented substitution-linear transformation network with 10, 12 or 14 rounds, depending on the key size (128, 192 or 256 bits respectively). It processes data blocks of 128-bits which is partitioned into an array of bytes. Rijndael‘s round function consists of four layers or transformations; SubBytes, ShiftRows, MixColumns and AddRound-

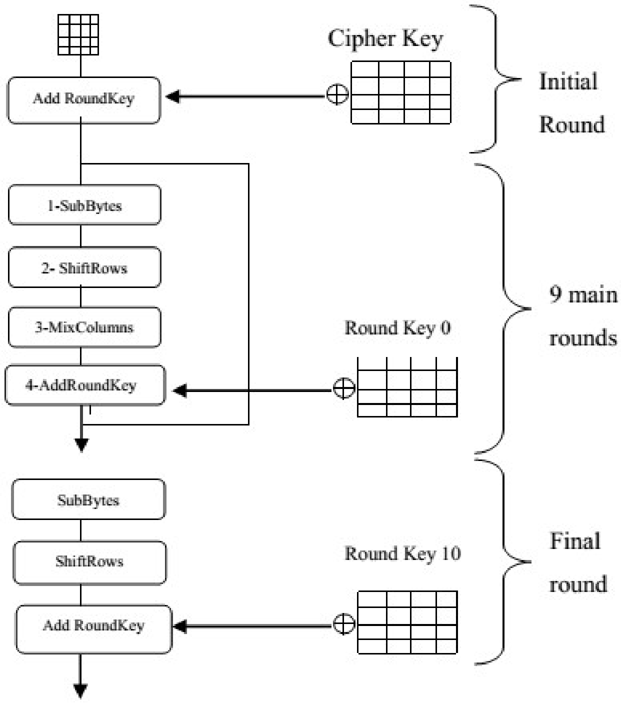


Figure. 1: Structure of the Rijndael-AES algorithm

Key as illustrated in Figure 1.

This paper evaluates the weaknesses present in the AES S-Box and proposes a technique of generating a more non-linear S-Box with low correlation thus a more secure AES algorithm. A more secure AES algorithm will ensure its even more secure implementation by the US government at the NSA, secure peer transactions and user authentication on the cloud.

In this paper, the operation of the Rijndael AES (Advanced Encryption Standard) algorithm will be discussed, followed by a statistical analysis of the S-Box in Section III. The formulation of a more non-linear S-Box will then be discussed in Section IV. This will be followed by a qualitative analysis of the formulated S-Box in Section V and the conclusion in Section VI.

II. Preliminaries

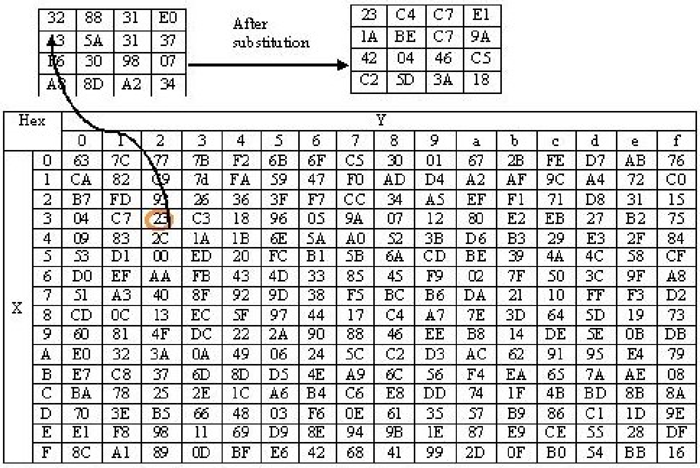
AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. The key size can be 128, 192 or 256 bits for 10, 12 or 14 rounds respectively. The key size depends on the number of rounds. In most ciphers, the round transformation has the Feistel structure. However, in Rijndael the round transformation is composed of four distinct invertible uniform transformations called layers, as illustrated in the Figure 1.

The choices of the layers are based on the Wide Trail Strategy which provides resistance against linear and differential cryptanalysis. Using the most common AES application of a 128-bit key, 10 rounds will be implemented in the AES encryption cipher. Each round goes through four transformations: substitution, permutation, mixing and key adding. In addition, before the first round, one AddRound Key is applied and in the 10th and last round, mixing is not implemented.

The plaintext block of data to be encrypted is first converted into a 4×4 matrix of bytes, referred to as a state. Before the first round, a key addition layer is applied so that any layer after the last key addition in the cipher or before the first round can simply be peeled off without knowledge of the key. This helps to protect it from known plaintext attacks. The four transformations in Rijndael-AES proceed as follows:

A. *SubBytes transformation*

In the SubBytes transformation, a non-linear substitution is first done for each byte. The transformation is defined by a look-up process or a mathematical calculation in $GF\left(2^8\right)$ in which the transformation is non-linear. For the look-up process, the byte is interpreted as two hexadecimal digits where the MSB defines the row and the LSB defines the column of the substitution table whose output is determined by the output of the substitution box (S-Box) as illustrated in Figure 2.



32	88	31	E0
43	5A	31	37
F6	30	98	07
A8	8D	A2	34

After
substitution

23	C4	C7	E1
1A	BE	C7	9A
42	04	46	C5
C2	5D	3A	18

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

Figure. 2: SubBytes Transformation.

B. *ShiftRow transformation*

In the shift row transformation, the rows of the state are cyclically shifted over different offsets. Row 0 is not shifted; Row 1 is shifted over 1 byte, row 2 over 2 bytes and so on. The shift offsets depend on the length of the block.

C. *MixColumn Transformation*

The MixColumns transformation involves multiplying the columns of the state by a fixed matrix called the multiplication matrix. During multiplication, the columns of the state are considered as polynomials over $GF\left(2^8\right)$ and multiplied modulo x^4+1 with a fixed polynomial $c(x)$ given by Equation 1

$$c(x) = 03x^3 + 01x^2 + 01x + 02 \qquad (1)$$

In order to make the cipher and its inverse more similar in structure, the linear mixing of columns is not done in the last round. However, this does not affect the security of the cipher in any way.

D. *AddRoundKey transformation*

In this transformation the Round Key is applied to the generated state by a simple bit-wise XOR. The RoundKey is derived from the secret key by means of the key schedule. The Round Key length is equal to the block length [7].

This paper focuses on the SubBytes transformation of AES, since it is the only non-linear part of AES thus has direct influence on performance and security.

III. Statistical Analysis of Rijndael

NIST selected Rijndael as the standard symmetric key encryption algorithm in AES over four other candidate algorithms; MARS, RC6, Serpent and Two-fish. It is used to encrypt sensitive American federal information. Rijndael was selected based on the following three criteria: resistance against all known attacks; speed and code compactness on a wide range of platforms; and design simplicity.

In the last decade, there have been many researchers who have tried to unearth weaknesses in AES. However, no practical weaknesses have been found so far. Research has shown that attacks on the Rijndael AES algorithm are dependent upon the generation of 2119-2128 plaintext-ciphertext pairs [8].

It has been discovered that an attack is possible by exploiting weaknesses in its structure as well as weaknesses present during implementation.

1. *Weaknesses in the structure* are prone to:

   (a) *Saturation attacks*- also referred to as the square attack. It is the most powerful cryptanalysis of AES and can break down a 7-round reduced version.

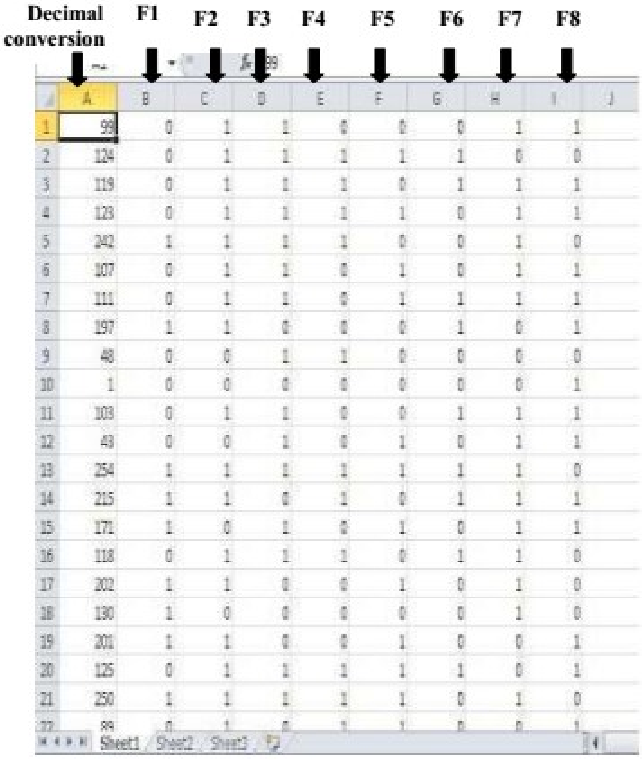
   (b) *Algebraic attacks*- which try to exploit the round transformation following the revelation by W.Millan and J.Fuller that the S-Box can be described by a single Boolean function [9]. This boolean function is given by:

$$y = 05x^{fe} + 09x^{fd} + f9x^{fb} + 25x^{f7} + f4x^{ef} + 01x^{df} + b5x^{bf} + 8fx^{7f} + 63 \quad (2)$$

2. *Weaknesses in non-structural properties* are prone to the exhaustive key search attack-which is possible for 56-bit key DES, but not for 128-bit AES due to the significant financial investment in hardware required [10]. However, if the cost of processing power is halved every six months according to the current trend, then this attack may be plausible at the end of the century.

3. *Implementation weaknesses* which pave the way for side-channel attacks- which could be in the form of timing analysis, differential & simple power analysis. AES has shown resistance due to its binary XOR and table look-up process.

These flaws could pave the way for discovery of new attack strategies on the AES algorithm, therefore motivating research efforts into developing a way to improve the security of the Rijndael algorithm by enhancing the S-Box structure.



	A	B	C	D	E	F	G	H	I
1	99	0	1	1	0	0	0	1	1
2	124	0	1	1	1	1	1	0	0
3	119	0	1	1	1	0	1	1	1
4	123	0	1	1	1	1	0	1	1
5	242	1	1	1	1	0	0	1	0
6	107	0	1	1	0	1	0	1	1
7	111	0	1	1	0	1	1	1	1
8	197	1	1	0	0	0	1	0	1
9	48	0	0	1	1	0	0	0	0
10	1	0	0	0	0	0	0	0	1
11	103	0	1	1	0	0	1	1	1
12	43	0	0	1	0	1	0	1	1
13	254	1	1	1	1	1	1	1	0
14	215	1	1	0	1	0	1	1	1
15	171	1	0	1	0	1	0	1	1
16	118	0	1	1	1	0	1	1	0
17	202	1	1	0	0	1	0	1	0
18	130	1	0	0	0	0	0	1	0
19	201	1	1	0	0	1	0	0	1
20	125	0	1	1	1	1	1	0	1
21	250	1	1	1	1	1	0	1	0
22	89	0	1	0	1	1	0	0	1

Figure. 3: Arrangement and conversion of the Rijndael S-Box output to decimal and then binary form.

This paper explores the inherent weaknesses in the Rijndael-AES algorithm by discussing the results of the statistical analysis of the AES S-Box used in the SubBytes transformation of the algorithm. Since the S-Box is the only source of non-linearity of Rijndael, it follows that this part is the key determinant of the performance and security of the algorithm.

Applications that produce random sequences should possess the following attributes: unpredictability, irreproducibility and uniform distribution. A statistical analysis of the S-Box output is carried out to establish to what degree AES S-Box output meets the desired attributes of randomness [11]. There are several test suites available for carrying out statistical randomness tests [12]. For instance, the DIEHARD test suite has limitations in test sequences and sample size.

However, the collection of tests specifically chosen for this study are the ones most suited for testing block ciphers which produce short sequences. These tests are: monobit/frequency test, serial test, correlation test and non-linearity test. These tests are used to uncover any existing patterns in the S-Box output, which could be exploited via statistical analysis techniques thus proving the existence of weaknesses in the Rijndael-AES algorithm.

The process of statistical testing involves obtaining the 256 pairs of hexadecimal output of the S-Box and arranging them sequentially in a single column, starting from the first hexadecimal pair on the top left-hand corner of the S-Box (63_{16}) up to the bottom right-most hexadecimal pair (16_{16}) . Then each pair of hexadecimal values, in each column, is converted into its decimal equivalent and then finally into its binary equivalent. This process is as illustrated in Figure 3

The result is a string of 256 rows which are arranged in 8 columns. For the purpose of this study, each of the eight columns, are taken to represent a single function. The functions are numbered F1, F2,...,F8. Therefore, each function F1 to F8 has a length of 256 binary bits. These 8 strings of 256 bits in length are then subjected to each of the four statistical tests and their relative performance evaluated in comparison to the standard expected results. This analysis helps to pin-point the presence of weaknesses in the AES S-Box.

A. *Monobit/ Frequency Test Results*

The monobit test, also called the frequency test, develops the frequency distribution from a sequence of discrete integers over a specified domain. It focuses on the number of 0s or 1s in a stream and assesses the closeness of the fraction of 1s to 0.5. It is a basic test and is recommended as the first test to be run before running other tests, because if the data stream fails this test then there will be a problem when running other tests as well [13]. This is because failure of this test indicates a lack of uniformity of distribution of the output.

The monobit test results of the AES S-Box show no deviation from the baseline thus proving that there is uniformity of the output, which is a desirable attribute of an application that produces a random output.

B. *Serial Test Results*

The serial test is a test for the randomness or pseudo-randomness of sequences, especially binary sequences. The serial test checks the number of occurrences of 2^n n-bit overlapping patterns to see if they are as expected for a random sequence. Random sequences display uniformity; such that, every n-bit pattern has an equal chance of occurrence. The probability of seeing any given pair is given by $\frac{1}{2^n}$.

The function call implemented in this test is of the form serial(m,n) , where m is the length of each block in bits (in this case 256 bits) and n is the length of the bit string being evaluated or tested [14, 15].

This paper evaluates patterns of 2-bits, 3-bits and 4-bits expressed as serial$(256,2)$, serial$(256,3)$ and serial$(256,4)$ respectively. The serial test for $(256,1)$ is equivalent to the monobit frequency test. The results obtained are then compared to the expected values, based on the probability of seeing any given pair $\frac{1}{2^n}$.

The 2 bit serial test evaluates the number of occurrences 00, 01, 10 and 11 bits in the 256-bit string for each of the eight functions. After obtaining the number of occurrences, the average number of occurrences is computed. For instance, for F1 the average number of occurrences of the 2-bit serial test will be $(59+68+69+54)\times\frac{1}{4}=63.75$. This process is repeated for serial$(256,3)$ and serial$(256,4)$. The results of the serial test for 1-bit, 2-bit, 3-bit and 4-bit serial tests is illustrated in Table 1.

The value of baseline and deviation are obtained by computing:

$$\text{Baseline = Total no. of bits (256)} \times \frac{1}{2^n}$$

$$\text{Deviation}= \left|\left(\frac{\text{serial patterns obtained}-\text{Baseline}}{\text{Baseline}}\right)\times 100\%\right| \qquad (3)$$

Table 1: The 1-bit, 2-bit, 3-bit and 4-bit Serial test results.

Functions	1-bit frequency matches	Average 2-bit serial patterns	Average 3-bit serial patterns	Average 4-bit serial patterns
	0, 1	00,01,10,11	000,001,..., 111	0000,0001,0010, ..., 1111
F1	128	63.75	31.75	15.8125
F2	128	63.75	31.75	15.8125
F3	128	63.75	31.75	15.8125
F4	128	63.75	31.75	15.8125
F5	128	63.75	31.75	15.8125
F6	128	63.75	31.75	15.8125
F7	128	63.75	31.75	15.8125
F8	128	63.75	31.75	15.8125
Baseline	128	64	32	16
Deviation	0%	0.390625%	0.78125%	1.17188%

The results of the 2-bit serial test shown in Table 1 show that the number of serial matches of all the eight functions have an equal degree of deviation of 0.390625%. The 3-bit serial test results show a deviation of 0.78125% for all the eight functions. While the results of the 4-bit serial test show a deviation of 1.17188%. These results indicate that there is no perfect uniformity in terms of groups of 4 bits. This provides a margin for the existence of weakness which creates possibility of the improvement of the structure of the S-Box in AES.

C. *Correlation Test Results*

The correlation test is used to compute the correlation coefficient between adjacent pairs of values. The process of correlation testing involves selecting a pattern of bits, which can be viewed as a frame, starting from the beginning and shifting the bits right till the 256^{th} bit in each function as illustrated in Figure 4. Then the selected pattern of bits (frame) is used to evaluate the number of matches which are then recorded. The frame is then shifted right by 1 bit and the testing for matches (correlation) repeated until the 256^{th} bit. The the size of the frame is then progressively increased by 1 bit and the number of matches recorded. In this paper, the evaluation is done up to a length of 4 bits.

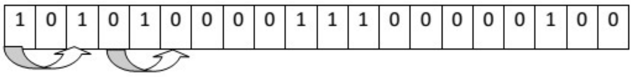

1 0 1 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0

Figure. 4: Right-shift mechanism in the correlation test.

In the correlation test, as opposed to the frequency test, only one pattern is evaluated for the number of matches. However, in the frequency test, all the patterns are evaluated. For instance; for 2-bit frequency test,the test will evaluate 00,01,10 and 11 while in the correlation test it will pick the first 2 bits in the 256-bit string followed by the next two (by shifting right) till the end. For example for F1=0000100100001...The 2-bit correlation test picks the first 2 bits (00) and correlates them up to the 256^{th} bit, by sliding the frame right all the way up to the end. Then the frame is slid onto the next two bits on the right and the process is repeated. This process is repeated for each of the eight functions. Then the frame size is increased to the first 3 bits

Table 2: The 2-bit, 3-bit and 4-bit Correlation test results.

Functions	2-bit correlation test		3-bit correlation test		4-bit correlation test	
	2-bit matches	Deviation	3-bit matches	Deviation	4-bit matches	Deviation
F1	68	6.25%	41	28.125%	22	37.5%
F2	66	3.125%	35	9.375%	17	6.25%
F3	62	3.125%	34	6.25%	21	31.25%
F4	69	7.8125%	37	15.625%	19	18.75%
F5	67	4.6875%	31	3.125%	18	12.5%
F6	65	1.5625%	32	0%	17	6.25%
F7	65	1.5625%	32	0%	19	18.75%
F8	64	0%	31	3.125%	16	0%
Baseline	64	-	32	-	16	-

and finally 4 bits and the process repeated for each of the eight functions

The results of the correlation test are as shown in Table 2. The measure of the deviation is obtained by computing

$$\text{Deviation} = \left| \left(\frac{\text{Correlation or no. of matches obtained} - \text{Baseline}}{\text{Baseline}} \right) \times 100\% \right| \tag{4}$$

The results of the correlation test for 2 bits show that there is deviation from the baseline, ranging from 7.8% for F4, to 0% for F8. The results of the correlation test for 3 bits show that there is deviation from the baseline, ranging from 28.125% for F1, to 0% for F6 and F7. The results of the correlation test for 4 bits show that there is deviation from the baseline, ranging from 37.5% for F1, to 0% for F8.

This implies that function F1 has the highest repetition of adjacent pairs of bits followed by F4, thus showing a high likelihood of predictability via a statistical analysis. On the other hand, function F8 is the least predictable due to the low level of repetition of adjacent bits.

D. Non-linearity Test Results

Nonlinearity is defined as the distance between a reference function under evaluation and group of all possible affine functions, which gives an indicator of the strength of invertible substitution tables and thus the strength of the encryption. In order to reach the closest affine function, then the bits require a change in the configuration [16]. The value of non-linearity is given by Equation III-D

$$N_f = 2^{n-1} - \frac{1}{2} max|w_f(a)| \qquad a \in \{0,1,2,\dots,2^{n-1}\}$$

For n = 8,

$$N_f \le 2^{n-1} - 2^{\frac{n}{2}-1} \qquad \Longrightarrow N_f = 120$$

$$N_f = 2^{8-} - 2^{4-1} = 2^7 - 2^3 = 128 - 8 = 120 \tag{5}$$

The optimal value of N_f is given as 120 (for a completely non-linear function). The non-linearity test involves analyzing the constituent Boolean functions of the S-Box so as to measure their cryptographic strength. The result of the non-linearity test is shown in Table 3.

The non-linearity test results obtained in Table 3 show non-linearity value of 112 for all the 8 functions. The average degree of non-linearity of the Rijndael S-Box obtained is 112, which also agrees with the test results obtained by research results in 2012 [17].

The closer the measure of non-linearity is to the upper-bound of $N_f = 120$, for perfectly non-linear functions, the more favourable it is. Higher non-linearity values indicate a resistance to algebraic attacks due to the confusion effect created by the S-Box. This paper therefore aims at generating an S-Box that has a non-linearity value above the current S-Box‘s non-linearity of 112, closer to 120 and also passes the frequency, serial and correlation tests.

E. Formulation of an enhanced S-Box

In view of the results of the statistical tests in Section 3, it is observed that there is room for further enhancement of the security of Rijndael-AES algorithm. This is done by particularly improving the S-Box structure, so as to create a more non-linear function and lower degrees of correlation between adjacent pairs of bits or bit patterns.

Therefore, in this paper we suggest the implementation of a non-linear pseudo-random number generator to generate a more secure S-Box. The suggested S-Box generation technique is evaluated in terms of its effectiveness.

In this study, a qualitative evaluation of the expected generated S-Box output will be done so as to derive its improved encryption strength, its ability to create confusion and thus its security features.

Various methods of constructing an AES S-Box have been proposed by researchers previously, as shown in Table 4.

It can be observed that none of them proposes an S-Box which has more superior non-linearity than the current Rijndael S-Box. The optimal value of non-linearity for the (8×8) S-Box is 120.

The higher the non-linearity of the structure, the more superior its performance, the lower the predictability of its output and thus the higher the resistance to algebraic linear attacks. Therefore, this study proposes an S-Box construction technique that will produce an S-Box that has more superior non-linearity $(112 < N_f \le 120)$ than the current Rijndael-AES S-Box by using a non-linear pseudo-random number generator.

1) Non-linear Pseudo-Random Number Generator

A random number generator is a program or a routine that is used to produce a random number which is obtained from a set of pre-determined possible values, each of which have an equal probability of selection and are statistically independent of each other. This is useful in cryptographic applications for generation of random unique keys, simulation and modeling applications as well as selection of a random sample from a large set of data. This technique is chosen by mapping the S-Box design principles to the desirable properties of a random number generator as shown in Figure 5.

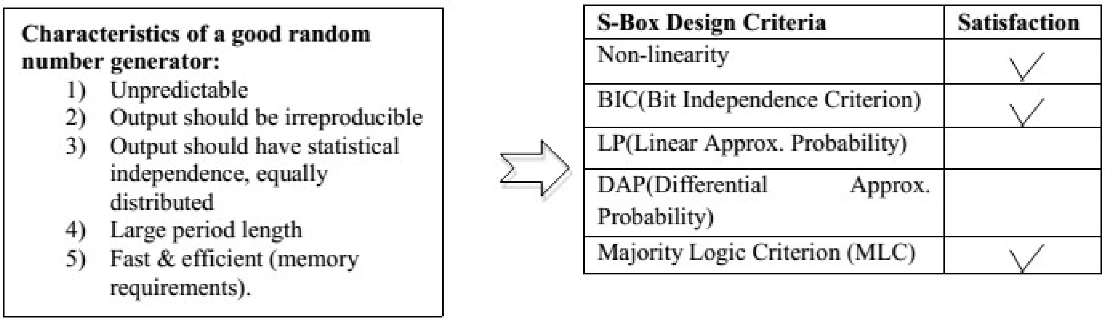
A pseudo-random number generator on the other hand, is an algorithm that applies a mathematical formulae or some

Table 3: Non-linearity test results.

Functions	F1	F2	F3	F4	F5	F6	F7	F8	Average
Non-linearity	112	112	112	112	112	112	112	112	112

Table 4: S-Box construction techniques.

S-Box	Non-linearity	Reference
Projective General Linear Group (PGL)	105.5	I. Hussain, Shah, et al., 2012b
Key-dependent S-Box		Krishnamurthy and Ramaswamy, 2008
Linear Fractional Transform	104.75	I. Hussain, Shah, Gondal, Khan, and Mahmood, 2012
Exchanges order of multiplicative inverse and affine transformation	104.75	X. Li et al., 2009
Binary Gray Code	112	Tran et al., 2008
Proposed S-Box	$112 < N_f \leq 120$	



Characteristics of a good random number generator:

1) Unpredictable
2) Output should be irreproducible
3) Output should have statistical independence, equally distributed
4) Large period length
5) Fast & efficient (memory requirements).

S-Box Design Criteria	Satisfaction
Non-linearity	✓
BIC(Bit Independence Criterion)	✓
LP(Linear Approx. Probability)	
DAP(Differential Approx. Probability)	
Majority Logic Criterion (MLC)	✓

Figure. 5: Mapping characteristics of a random number generator and the S-Box design criteria.

pre-calculated tables so as to produce strings or sequences of numbers that appear to be random in nature.

This means that it can be replicated if the algorithm and its initial conditions or input parameters are known, therefore satisfying the differential approximation probability which ensures uniform mapping between the input differential to a unique output differential. After certain duration of time, the sequence of the pseudo-random generator will be repeated. This repeat duration is referred to as period, which is an important measure of the quality of a generator. The period should exceed the required amount of numbers.

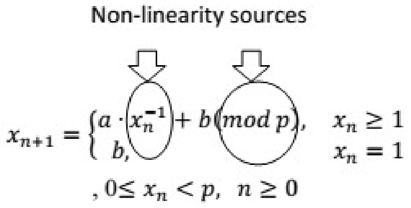
There are two categories of pseudo-random number generators: linear pseudo-random generators and non-linear pseudo-random number generators. Linear pseudo-random generators produce an output that is based on linear recurrences and the output also possesses a linear structure. On the other hand, non-linear pseudo-random number generators produce truly random outputs and they do not produce a lattice structure [18].

2) *S-Box Construction*

A more secure S-Box can be generated by using a non-linear pseudo-random generator to produce output that possesses good non-linearity properties closer to the optimal value of non-linearity of 120. The generated S-Box is expected to be resistant to statistical, linear and differential analysis attacks. There are several methods of generating non-linear output from a random number generator:

1. *Using a nonlinear transition function f*
2. *Using a linear transition function* to generate uniform random numbers and then transforming the resulting random numbers non-linearly by:
   (a) Applying a non-linear function g to produce the output
   (b) Combining two linear RNGs
   (c) By shuffling the resulting output values using another generator.

The second approach of first generating uniform random numbers and then converting them to non-linear output is the standard approach and will be applied in this paper. This method is applied by using the inverse congruential generator which has a characteristic feature of exhibiting strong non-linearity properties [19]. It also passes the serial test and it allows a large choice of parameters, meaning it has a large period. Therefore, the inverse congruential generator used to generate random numbers for the improved S-Box is shown using the Equation III-E.2.



$$x_{n+1} = \begin{cases} a \cdot x_n^{-1} + b \, (mod\, p), & x_n \geq 1 \\ b, & x_n = 1 \end{cases} \quad , 0 \leq x_n < p, \; n \geq 0$$

In which p is a prime number, a is the multiplier, b is the additive term and also a and b are positive integers. x is a unique integer and $0 \leq x \in < p$ and also $x \cdot x^{-1} \equiv 1\,(modp)$. The value x_0 is the seed or initial value input to the generator. The value of $x_{n+1} = b$ if $x_n = 0$. This generator can be denoted by ICG (p,a,b,x_0). The assumption that $p \geq 5$ is made to avoid triviality [20].

Non-linearity is achieved by using the multiplicative inverse modulo p operation. The values a and b are suitable constants and $a,b \in F_p$. Thus, pseudorandom numbers in $[0,1]$ are obtained by setting $u_n = \frac{x_n}{p}$. This means that x_n and p are co-prime numbers. If p is a prime number and if $x^2 - bx - a$ is a primitive polynomial over the finite F_p, then the length of the period is at its maximum value and is given by $\rho = p$ [21].

The random number generator operates by first initializing a random input referred to as a seed x_0, from which a long sequence of random numbers can be generated in a deterministic manner. The seed could be set from the system clock or alternatively generated manually.

A pseudo-random number generator (PRNG) is characterized by its very fast speed in producing random numbers [22]. It is also characterized by a uniform distribution, in that output is either 0 or 1 with equal probability, and also these output bits are independent of all other bits. The most common hardware implementation of PRNGs is in linear feedback shift registers (LFSR).

The sequence followed in formulating or generating of the S-Box values is as shown in the Figure 6. The process of generating values for the S-Box using a non-linear PRNG would begin by first setting the non-linearity value for the desired random string for an 8×8 matrix such that $112 < N_f \leq 120$.

Then a selection of the number of random strings to be checked is entered. Subsequently, the (least significant) LS bits from the string are extracted and the non-linearity calculated. This process is repeated up to the 8^{th} column. Only values which meet the selection criteria for non-linearity are displayed in the form of a table, expressed in binary numbers $(0,1)$.

These values, displayed in table form, can then be cascaded or arranged in the form of 8 columns, 256 rows as in Figure 3. The binary values of all the 256 rows, each eight binary bits in length are then converted to hexadecimal form and then used to populate the new proposed S-Box which is more non-linear.

In order to carry out decryption of ciphertext using the improved AES algorithm using the enhanced S-Box values, then the inverse S-Box needs to be generated. To find the inverse S-Box for the purpose of decryption, there are two alternatives:

1. *By going back in the program execution* for the incursive congruential PRNG. To do this, cache the most recent numbers and store some points at intervals so that the sequence can be recreated from there.

2. *By computing the previous state*. This is done by reordering the operation so that

$$x_{n+1}^{-1} = a^{-1} \times (x-b)\, mod p \tag{6}$$

The new S-Box can then be subjected to the statistical tests conducted on the Rijndael S-box previously in this study so as to ascertain that it passes the frequency test, serial test and correlation tests while maintaining a more superior non-linearity. This will help to assure the continued enhanced security of Rijndael algorithm for use in safeguarding top secret information due to its resistance to statistical analysis attacks. In this paper, a qualitative analysis of the formulated S-Box is done in Section V.

IV. Qualitative Analysis

A qualitative evaluation of the satisfaction of the formulated S-Box to the same statistical tests carried out earlier in the study would provide a clear picture of the expected efficiency of the suggested technique of using the incursive congruential method. The tests under consideration are monobit/ frequency tests, serial test and correlation tests. These tests help to give a clear picture on whether there is a uniform distribution of bits in the S-Box and whether there is a minimal degree of correlation between adjacent pairs of bits.

The frequency/ monobit test evaluates the uniformity of sequences of successive random numbers generated for the S-Box table. Therefore, it tests the uniformity of the output for $s = 1$ (1 bit). Research has shown that a pseudo-random generator gives an output of random numbers that are uniformly distributed [23], thus the formulated S-Box passes the monobit test.

The serial test evaluates the uniformity of the output from the inversive congruential generator used to populate the S-Box table. It is also referred to as the s-dimensional test for $s \leq p-2$. One of the characteristics of the inversive congruential generator is that it passes the s-dimensional serial test, where $s \leq d$, in which case d is a specific value [19]. Therefore, this implies that the generator satisfies the serial test for $s = 1,2,3,4,\ldots,n-1$. Thus, the inversive congruential generator passes the serial test for 1 bit (similar to monobit test), 2 bits, 3 bits and 4 bits.

Research has shown that provided the values a and b are chosen appropriately so as to satisfy the condition for obtaining a maximum period length p, then it follows that the ICG (p,a,b,x_0) will exhibit excellent correlation properties [24]. Thus, the generated S-Box is expected to satisfy the majority logic criterion in terms of correlation analysis because the generated output will have excellent correlation properties.

V. Conclusion

This research involves an analysis of the Rijndael-AES S-Box so as to evaluate the weaknesses present in the S-Box. This research reveals weaknesses in terms of a significantly higher degree of correlation in comparison to the standard, with a deviation of up to 37.5% for 4-bit correlation. The research also reveals weaknesses in terms of a low degree of non-linearity of 112 in comparison to the optimal value of 120. This study therefore proposes a solution that produces a more non-linear output with a low degree of correlation which results in a more secure S-Box. The generation of a more secure S-Box can be done using a non-linear pseudo-random number generator (PRNG) that implements the incursive congruential method which generates highly non-linear output, with non-linearity closer to the optimal value of 120. The generated output is expected to pass the monobit, serial and correlation tests.

References

[1] R. Gupta, S. Gupta, and A. Singhal, “Importance and techniques of information hiding: A review,” *arXiv preprint arXiv:1404.3063*, 2014.

[2] M. Malhotra and A. Singh, “Study of various cryptographic algorithms,” *International Journal of Scientific Engineering and Research IJSER*, pp. 77–78, 2013.

[3] A. Kahate, *Cryptography and network security*. Tata McGraw-Hill Education, 2013.

[4] D. S. A. Elminaam, H. M. Abdual-Kader, and M. M. Hadhoud, “Evaluating the performance of symmetric

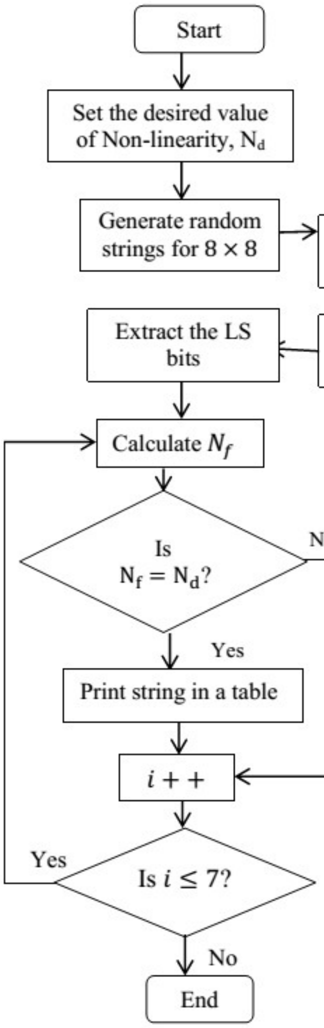


Figure. 6: The sequence of formulating S-Box values using a non-linear PRNG implementing inversive congruential method.

encryption algorithms.," *IJ Network Security*, vol. 10, no. 3, pp. 216–222, 2010.

[5] A. R. Tonde and A. P. Dhande, "Review paper on fpga based implementation of advanced encryption standard (aes) algorithm," 2014.

[6] *Operating Systems*. Cognizant Academy, 2004.

[7] A. F. Behrouz and M. Debdeep, "Cryptography and network security," *McGramHill, International Edition*, 2008.

[8] R. Ciprian, D. Grecu, F. Medeleanu, N. Jula, and D. RĂDUCANU, "Comparative analisys for the new generation of encryption algorithms involved in nessie and cryptrec projects," in *The 38-th International Symposium of the Military Equipment & Technologies Research Agency, Bucureşti*, 2008.

[9] J. Fuller and W. Millan, "Linear redundancy in s-boxes," in *Fast Software Encryption*, pp. 74–86, Springer, 2003.

[10] J. Daemen, V. Rijmen, and P. S. Barreto, "Rijndael: beyond the aes," in *Mikulsk kryptobesdka 2002–3rd Czech and Slovak cryptography workshop*, 2002.

[11] C. S. Kinga Mrton, Alin Suciu and O. Cre, "Generation and testing of random numbers for cryptographic applications," *international conference Romanian Cryptology Days vol RCD-2011*, no. Proceedings of the Romanian Academy, Series A, 2011.

[12] S. Faith, *Statistical Analysis of Block Ciphers and Hash Algorithms*. Thesis, Middle East Technical University, 2011.

[13] L. Li, *Testing Several Types Of Random Number Generator*. Thesis, 2011.

[14] J. Soto, "Statistical testing of random number generators," in *Proceedings of the 22nd National Information Systems Security Conference*, vol. 10, p. 12, NIST Gaithersburg, MD, 1999.

[15] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," tech. rep., DTIC Document, 2001.

[16] I. Hussain, T. Shah, H. Mahmood, and M. A. Gondal, "A projective general linear group based algorithm for the construction of substitution box for block ciphers," *Neural Computing and Applications*, pp. 1–9, 2012.

[17] I. Hussain, T. Shah, M. A. Gondal, W. A. Khan, and H. Mahmood, "A group theoretic approach to construct cryptographically strong substitution boxes," *Neural Computing and Applications*, pp. 1–8, 2012.

[18] J. E. Gentle, W. K. Hrdle, and Y. Mori, *Handbook of computational statistics: concepts and methods*. Springer, 2012.

[19] A. Gille-Genest, "Implementation of the pseudo-random numbers generators and the low discrepancy sequences," p. 16, 2012.

[20] W. Meidl and A. Topuzolu, *On the Inversive Pseudorandom Number Generator*, book section 5, pp. 103–125. Physica-Verlag HD, 2010.

[21] P. Hellekalek, "Inversive pseudorandom number generators: concepts, results and links," 1995.

[22] C. Gregory Gordon Rose, Dan Diego, C. (US); Alexander Gantman, PoWay, and C. U. (US); Lu Xiao, San Diego, "Cryptographically secure pseudo-random number generator," 2011.

[23] J. Bubicz, Jaroslaw Stoklosa, "Compound inversive congruential generator design algorithm," *parameters*, vol. 2, no. 4, p. 6, 2006.

[24] P. Hellekalek, "Inversive pseudorandom number generators: concepts, results and links," in *Proceedings of the 27th conference on Winter simulation*, pp. 255–262, IEEE Computer Society.

Author Biographies

Juliet N. Gaithuru was born in Thika, Kenya on the 15th of November 1985. She attained a BSc. in Computer Information Systems from Kenya Methodist University in 2011. She also holds a Master of Computer Science in Information Security from Universiti Teknologi Malaysia, 2013, where she carried out research on the S-Box in the Advanced Encryption Standard (AES) Algorithm. She is currently pursuing a Ph.D degree in computer science specializing in the field of information security at Universiti Teknologi Malaysia. Her research interests are in the field of symmetric and asymmetric cryptography with particular interest in post-quantum cryptography.

Majid M. Bakhtiari is a Senior Lecturer in the Faculty of Computing, Univesity Technology Malaysia (UTM). He holds a BSc in electronics, MSc in information security and PhD in computer science (cryptography). His research interests center around the area of cryptography and cryptanalysis. He has over 30 years experience in the field of cryptography and cryptanalysis. He has designed, educated and installed three generations of data crypto-systems in the Ministry of Foreign Affairs of Isl. Rep. of Iran from 1985 to 2006. He has experience in algorithm breaking in the field of voice encryption and data encryption and is an expert in designing security systems for large organizations . His current research work concentrates on cryptography, cryptanalysis, security in cloud computing, steganography and watermarking.