Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666

Volume 6 Issue 20 (March 2021) PP. 10-16
DOI: 10.35631/JISTM.620002

JOURNAL OF INFORMATION SYSTEM AND TECHNOLOGY MANAGEMENT (JISTM)
www.jistm.com

GLOBAL ACADEMIC EXCELLENCE

SECURITY ANALYSIS BETWEEN STATIC AND DYNAMIC S-BOXES IN BLOCK CIPHERS

Nur Hafiza Zakaria[1*], Azuan Ahmad[2], Azni Haslizan Ab Halim[3], Farida Hazwani Mohd Ridzuan[4]

[1] Faculty of Science and Technology, Universiti Sains Islam Malaysia
Email: mzhafiza@usim.edu.my
[2] Faculty of Science and Technology, Universiti Sains Islam Malaysia
Email: azuan@usim.edu.my
[3] Faculty of Science and Technology, Universiti Sains Islam Malaysia
Email: ahazni@usim.edu.my
[4] Faculty of Science and Technology, Universiti Sains Islam Malaysia
Email: farida@usim.edu.my
[*] Corresponding Author

Article Info:

Article history:
Received date: 06.05.2019
Revised date: 15.05.2019
Accepted date: 10.03.2021
Published date: 15.03.2021

To cite this document:

Zakaria, N. H., Ahmad, A., Ab Haim, A. H., & Ridzuan, F. H. M. (2021). Security Analysis Between Static and Dynamic S-Boxes in Block Ciphers. Journal of Information System and Technology Management, 6 (20), 10-16.

DOI: 10.35631/JISTM.620002.

Abstract:

The development of block ciphers has resulted in a number of cryptographic algorithms such as AES, aria, blowfish256, desl, and 3d-aes. AES is one of the best cryptographic algorithms that can be used to protect electronic data. However, the principal weakness in AES is the linearity in the s-box. The objective of this research is to investigate and evaluate the existing work related to the dynamic s-box. Other than that, the aim of this research is to design a dynamic s-box using affine transformation in order to increase the security of the encryption. The method to design is using java with the NetBeans software. The proposed block cipher will be tested using NIST statistical test suite to test the randomness of the algorithm. Besides, the strength of the s-box will be analyzed using the s-box evaluation tool (set). The cryptographic strength depends strongly on the choice of s-box. Therefore, this new proposed block cipher can be used by countries, organizations, stakeholders, or interested parties as one of the secure algorithms to increase the protection of the information and also will contribute as an alternative to other cryptographic algorithms in computer security research.

Keywords:

Box, AES, Affine Transformation, NIST, Cryptography

Introduction

Cryptography is defined as a method or technique of secure communication in the presence of third party. In modern age of computers, cryptography is a technique to scramble plaintext or original message into ciphertext by using cryptographic algorithms. Cryptography is an effective way of protecting sensitive information whether it is stored in media or transmitted. The main objective in cryptography is to attain data confidentiality, data integrity, authentication and non-repudiation.

Cryptography is generally divided into two categories which are known as symmetric key cryptography and asymmetric key cryptography. Symmetric key cryptography is a form of cryptosystem in which encryption and decryption are performed using the same key (Forouzan, 2008). Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm. It is widely used in symmetric ciphers for instance DES (Data Encryption Standard) and AES (Advanced Encryption Standard).

Asymmetric key encryption is a form of cryptosystem in which encryption and decryption are performed using the different keys. One is public key for encryption and the other one is private key for decryption. Asymmetric encryption can be used for confidentiality, authentication or both. The most widely used asymmetric key encryption are Rivest-Shamir-Adleman (RSA), Diffie-Hellman key exchange, ElGamal Cryptosystem and Elliptic Curve Cryptography (ECC) (Amandeep, 2017). The asymmetric algorithm as outlined in the Diffie-Hellman paper uses numbers raised to specific powers to produce decryption keys. RSA is the most widely used asymmetric algorithm is embedded in the Secure Socket Layer/Transport Layer Security (SSL/TLS) protocol which is used to provide communications security over a computer network.

Literature Review

Advanced Encryption Standard (AES)

AES is designed on the principle of combination of both substitution and permutation. The AES fixes the block length to 128 bits and supports key lengths of 128, 192 or 256 bits. It represented in matrix form called state matrix and operated on 4x4 column-major order matrix of bytes. Most AES calculations are done in a special finite field $GF(2^8)$. The element of a finite field can be represented in several different ways. For any prime power there is a single finite field, hence all representations of $GF(2^8)$ are isomorphic. Despite this equivalence, the representation has an impact on the implementation complexity. A byte b, consisting of bits $b_7\ b_6\ b_5\ b_4 b_3\ b_2 b_1 b_{0,}$ is considered as a polynomial with coefficient in {0,1}:

$$b_7\,x^7 + \ b_6\,x^6 + b_5\,x^5 + b_4\,x^4 + b_3\,x^3 + b_2\,x^2 + b_1 x + \ b_0$$

Example: the byte with hexadecimal value ‘57’ (binary 0101 0111) corresponds with polynomial

$$x^6 + \ x^4 + \ x^2 + x + \ 1.$$

The key size used for AES specifies the number of repetitions of transformation rounds. The number of cycles of repetition is as follows (Daeman, 2002):

i. 10 cycles of repetition for 128-bit keys.
ii. 12 cycles of repetition for 192-bit keys.
iii. 14 cycles of repetition for 256-bit keys.

Encryption process follows four (4) following steps which are

i. The ByteSub Transformation

The ByteSub Transformation is a non-linear byte substitution, operating on each of the State bytes independently. SubBytes is a bricklayer permutation consisting of an S-Box applied to the bytes of the state. Figure 1 below illustrates the effect of the SubBytes step on the state.

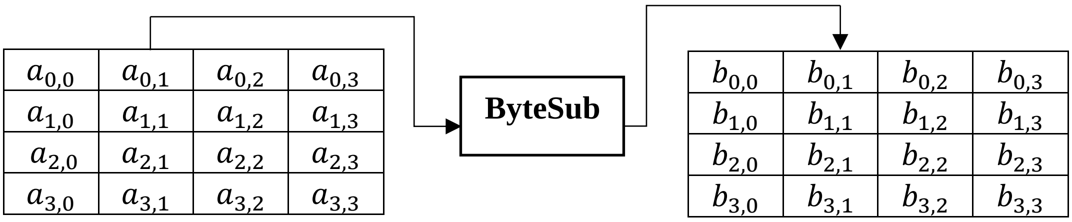

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ByteSub
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

Figure 1: SubBytes Transformation

The substitution table or known as S-Box is invertible and is constructed by the composition of two transformations:

a. First, taking the multiplicative inverse in $GF(2^8)$.
b. Then, applying an affine transformation defined by:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{bmatrix}$$

The affine transformation can also be described as a polynomial multiplication, followed by the XOR with a constant (Rijndael, 2002).

S-Box of AES is generated by equation: $Y = Ax \oplus c \; mod \; M$ (1)

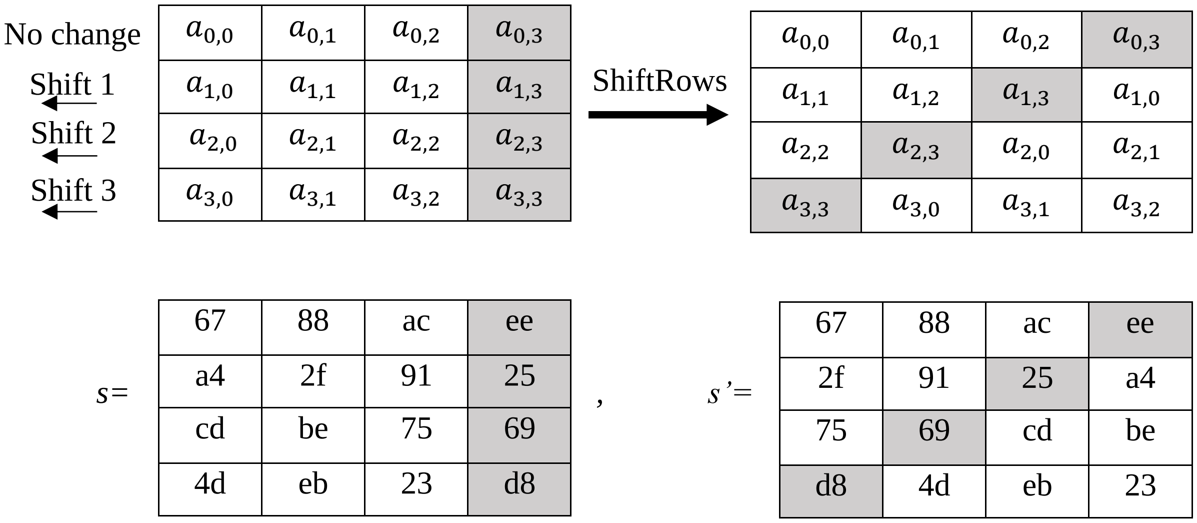
Where ‘A’ is represented as affine matrix, ‘x’ is a vector that is multiplicative inverse of element of state matrix, ‘c’ is affine constant 63 (01100011) and ‘M’ is irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. The S-Box generated by equation (1) is represented in the Table 1 below.

Table 1: AES S-Box

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

i. The ShiftRow Transformation

The ShiftRows step is a byte transposition that cyclically shifts the rows of the state over different offsets. In AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For example we get *s'* matrix from state matrix *s* after shift row in following transformation:



$$\begin{array}{r|c|c|c|c|}\hline \text{No change} & a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ \hline \text{Shift 1} \leftarrow & a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ \hline \text{Shift 2} \leftarrow & a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ \hline \text{Shift 3} \leftarrow & a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \\ \hline \end{array} \xrightarrow{\text{ShiftRows}} \begin{array}{|c|c|c|c|}\hline a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ \hline a_{1,1} & a_{1,2} & a_{1,3} & a_{1,0} \\ \hline a_{2,2} & a_{2,3} & a_{2,0} & a_{2,1} \\ \hline a_{3,3} & a_{3,0} & a_{3,1} & a_{3,2} \\ \hline \end{array}$$

$$s = \begin{array}{|c|c|c|c|}\hline 67 & 88 & ac & ee \\ \hline a4 & 2f & 91 & 25 \\ \hline cd & be & 75 & 69 \\ \hline 4d & eb & 23 & d8 \\ \hline \end{array} \quad , \qquad s' = \begin{array}{|c|c|c|c|}\hline 67 & 88 & ac & ee \\ \hline 2f & 91 & 25 & a4 \\ \hline 75 & 69 & cd & be \\ \hline d8 & 4d & eb & 23 \\ \hline \end{array}$$

Figure 2: ShiftRows Transformation

ii. The MixColumns Transformation

The MixColumns step is a bricklayer permutation operating on the column by column. In MixColumn, column vector is multiplied with a fixed matrix, where the bytes are treated as a polynomial rather than numbers.

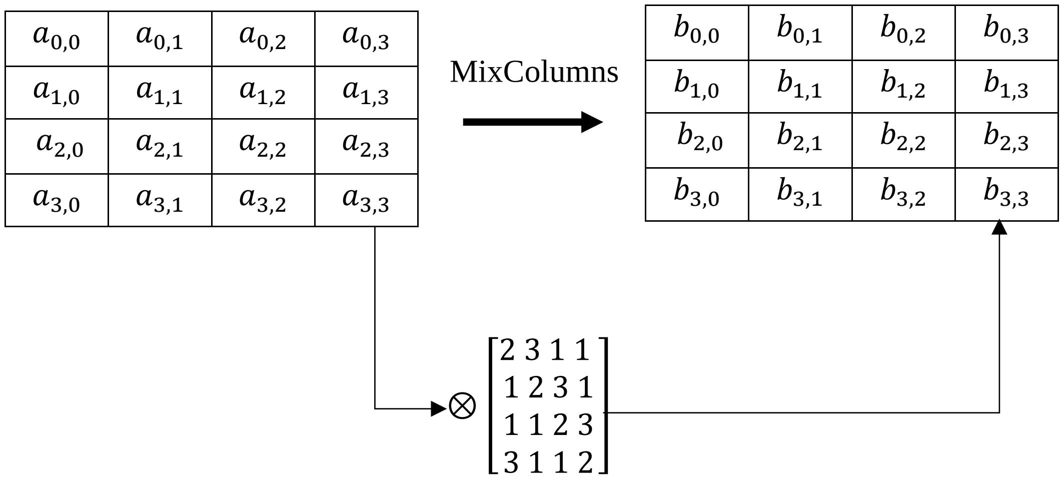

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
MixColumns
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
$\otimes \begin{bmatrix} 2\,3\,1\,1 \\ 1\,2\,3\,1 \\ 1\,1\,2\,3 \\ 3\,1\,1\,2 \end{bmatrix}$

Figure 3: MixColumns Transformation

iii. Add Round Key

The Add Round key operation is an XOR operation between the State and the Round Key. By doing XOR operation element by element of both State and Round Key, next state of matrix is obtained.

State' (a,b) = State(a,b) ⊕ Round Key(a,b)

$$\text{State} = \begin{bmatrix} 32\ 88\ 31\ e0 \\ 43\ 5a\ 31\ 37 \\ f6\ 30\ 98\ 07 \\ a8\ 8d\ a2\ 34 \end{bmatrix}, \text{Round Key} = \begin{bmatrix} 00\ a5\ a8\ a0 \\ e9\ 09\ bb\ 2a \\ c9\ d4\ b7\ ab \\ f2\ e8\ 60\ 08 \end{bmatrix}$$

The new state will be:

32(hex) ⊕ 00 (hex) = 0011 0010 ⊕ 0000 0000
= 0011 0010 (32)

88(hex) ⊕ a5 (hex) = 1000 1000 ⊕ 1010 0101
= 0010 1101 (2d)

$$\text{State'} = \begin{bmatrix} 32\ 2d\ 99\ 40 \\ aa\ 53\ 8a\ 1d \\ 3f\ e4\ 2f\ ac \\ 5a\ 65\ c2\ 3c \end{bmatrix}$$

S-Box Properties

The strength of block ciphers which work on substitution and permutation like AES is basically depends on the construction of S-Box. The S-Box maps an 8-bit input to an 8-bit output. Both the input and output are interpreted as polynomials over GF(2). First, the input is mapped to its multiplicative inverse in $GF(2^8) = GF(2)[x]/x^8 + x^4 + x^3 + x + 1$. The AES S-Box was specifically designed to be resistant to linear and differential cryptanalysis. This was done by minimizing the correlation between linear transformations of input/output bits, and at the same time minimizing the difference propagation probability. The AES S-Box can be edited which defeats the suspicion of a backdoor built into the cipher that exploits a static S-Box.

The properties of S-Box have been widely used as a base of new encryption technique for instance nonlinearity, differential uniformity and strict avalanche criterion (Zhang, 2013).

i. Bijection – requires a one-to-one and onto mapping from input vectors to output vectors if the S-Box is *n* by *n* bit.
ii. Strict avalanche criterion – 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, we will need a function that has a 50% dependency on each of its *n* input bits.
iii. Bit independence criterion or correlation immunity – requires that output bits act independently from each other's. In other words, there should not be any statistical pattern or statistical dependencies between output bits from the output vectors.
iv. Nonlinearity – requires that the S-Box is not a linear mapping from input to output. This would make the cryptosystem susceptible to attacks (Coppersmith, 1994). 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. The nonlinearity of an S-Box must be high to resist against linear cryptanalysis.
v. Balance – means that each Boolean vector responsible for the S-Box has the same number of 0's and 1's. The significance of the balance property is based on the higher the magnitude of function imbalance, a high probability linear approximation being obtained.
vi. Differential uniformity – the smaller is the differential uniformity, the better is the S-Box's resistance against differential cryptanalysis.

Conclusion

AES has been designed to have a very strong resistance against the classical attacks such as linear cryptanalysis and differential cryptanalysis. However, since AES is very algebraic, new algebraic attacks was appeared (Ferguson, 2001). Therefore, the need for design dynamic S-Box which is key-dependent will help to increase the security of encryption. Most of researchers aware that S-Box properties are very crucial in order to make the S-Box strong enough and secure. However, most of them analyse their S-Box properties based on their perceptions without any proper guideline. In addition, cryptanalysis attempt to break the S-Box properties with all kinds of methods.

References

Abdurashid, M., Herman, I. and Moesfa, S. M. (2009). Practical Bijective S-Box Design. Proceesings of the 5thbAsian Mathematical Conference.

Adams, C. and Tavares, S. (1990). The structured design of cryptographically good s-boxes. Journal of Cryptology, 3(1):27–41.

Amandeep, S., Praveen, A. and Mehar, C. (2017). Analysis of Development of Dynamic S-Box Generation. Computer Science and Information Technology.

Coppersmith, D. (1994). The data encryption standard and its strength against attacks. IBM Journal of Research & Development, 38(3):243.

Cui, J., Huang, L., Zhong, H., Chang, C. and Yang, W. (2011). An Improved AES S-Box and Its Performance Analysis. International Journal of Innovative Computing, Information and Control.

Daemen, J. and Rijmen, V. (2002). The Design of Rijndael: AES The Advanced Encryption Standard. Springer-Verlag.

Drashti, O. V. and Purvi, H. T. (2015). Study of Avalanche Effect in AES. National Conference on Recent Advances in Engineering for Sustainability.

Ferguson, N., Schroeppel, R. and Whiting, D. (2001). A Simple Algebraic Representation of Rijndael. Selected Areas in Cryptography.

Forouzan, B. (2008). Traditional Symmetric-Key Cipher. Introduction to Cryptography and Network Security. 1st ed New York: McGraw Hill.

Ghada, Z., Abdennaceur, K., Fabrice, P. and Daniele, F-P. (2009). On Dynamic Chaotic S-Box. Institute of Electrical and Electronics Engineers (IEEE) Access.

Jie, C., Liusheng, H., Hong, Z., Chinchen, C. and Wei, Y. (2011). An Improved AES S-Box and Its Performance Analysis. International Journal of Innovative Computing.

Kazys, K, Gytis, V and Robertas, S. (2014). An Algorithm for Key-Dependent S-Box Generation in Block Cipher System. Informatica.

Krishnamurthy, G. N. and Ramaswamy, V. (2008). Making AES Stronger: AES with Key Dependent S-Box. International Journal of Computer Science and Network Security (IJCSNS).

Manjula, G. and Mohan, H. S. (2016). Constructing key dependent dynamic S-Box for AES Block Cipher System. International Conference on Applied and Theoretical Computing and Communication Technology (ICATCCT).

Muhammad, F. K., Addel, A. and Khalid, S. (2019). A Novel Cryptographic Substitution Box Design Using Gaussian Distribution. Institute of Electrical and Electronics Engineers (IEEE) Access.

Musheer, A., Hitesh, C., Avish, G. and Prateek, Singla. (2013). A Chaos Based Method for Efficient Cryptographic S-Box Design.

Sliman, A., Abdellatif, H., Abderrahim, T. and Salah, E. K. (2013). Implementation of Stronger AES by Using Dynamic S-Box Dependent of Master Key. Journal of Theoretical and Applied Information Technology.

Tianyong, A., Jinli, R., Kui, D. and Xuecheng, Z. (2017). Construction of High Quality Key-dependent S-Box. IAENG International Journal of Computer Science.

Zakaria, N. H. (2016). A Block Cipher based on Genetic Algorithm. Universiti Putra Malaysia.