Hardware Implementation of Dynamic S-BOX to Use in AES Cryptosystem

Sahar Darvish Motevali [a], Kooroush Manochehri [b,*]

[a] *Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[b] *Department of Computer Engineering and IT, Parand Branch, Islamic Azad University, Parand, Iran*

Received 6 December 2016; revised 20 April 2017; accepted 3 September 2017; available online 16 October 2017

Abstract

One of the major cipher symmetric algorithms is AES. Its main feature is to use S-BOX step, which is the only non-linear part of this standard possessing fixed structure. During the previous studies, it was shown that AES standard security was increased by changing the design concepts of S-BOX and production of dynamic S-BOX. In this paper, a change of AES standard security is studied by production of dynamic and key-dependent S-BOX. Also the LFSR random number generation hardware algorithm is applied in order to produce the dynamic S-BOX. In order to produce a dynamic and key-dependent S-BOX, the field bits of key are divided into separated bits at first and then a byte is selected by LFSR algorithm randomly. The number of selected bit is considered as the repeating number of LFSR algorithm and is applied in order to produce dynamic S-BOX. In the evaluation step, we compared the proposed model with fixed S-BOX model in the original AES algorithm. It was shown that the proposed implementation could increase the security as about 0.2%, 0.017% and 0.19%, 0.04 in the case of avalanche effect, output bits dependence criteria, compared with the strict avalanche criteria and in the case of linear criteria, respectively.

Keywords: *Advanced Encryption Standard, Changeable, Feedback Shift Register, Moving Boxes, Simultaneous Encryption.*

1. Introduction

Today, by increasing the communication models, the need for protecting the information transfer among users becomes more important than before. It is well known that the AES algorithm has been registered as the standard encryption model since 2001 [1]. So many researches have been devoted on increasing the security levels and changing the hardware structure of the algorithm**.**

In [2], Zhou-quan Du and colleagues proposed a model in which math functions are used to produce a robust S-box that could increase security. In their proposed model, the XOR operator combination was employed to design a new S-box. They evaluated the model by a Bemanie test. In the analysis outputs performed by the Bayesian test, it was found that all outputs were close to the SAC benchmark. As a result, this model was able to produce strong S-box.

In [3], Kalaiselvi and his colleagues tried to increase the security of the as algorithm by using the genetic algorithm and the neural network. They developed a stronger cryptomechanical algorithm in this research, the genetic algorithm was used to modify the s-box structure and the neural network was used to correct the structure.

* Corresponding author. Email: kmk1381@aut.ac.ir

In [4], Akram Belazi and colleagues used the Rossler chaos algorithm to produce s-box safe. In the evaluation step, three criterion including the output bits independence, strict avalanche, nonlinearity with and linear approximation probability were compared to the proposed model with previous design models. The results showed that the proposed model has better security than previous designs.

In [5], Julia Juremi and her team developed a new KEY-dependent AES using the S-BOX rotation. This encryption and decryption process is similar to the original AES, but the main AES consists of four steps. The value of rotation in this model is dependent on the key expansion, which has been able to minimize the amount of plantext. The evaluation of Avalanche Effect and Strict Avalanche Criterion on the proposed model and the AES model showed that the proposed model creates more confusion.

In [6] Iman Saberi and his team use the odd and even method in key expansion design in the AES-256 structure. In the method, a process is gone ahead based on the previous stage odd and even data. This issue has caused confusion, so that if the attacker knows some of the keys, he still cannot achieve the result. This indicates the high security of the proposed method. It was determined by the evaluation of the Avalanche Effect test.

In [7], Shivkumar and Umamaheswariy designed the S-BOX using the RC4 key. In this research, the key-dependent S-BOXs that were generated at each stage for each step of the S-BOX, as a feature is for making S-BOX 256, which is dependent on the input key. At each step, the key bytes of that step are XOR to use the amount of time for the S-BOX. Thus, if a byte of the input key is changed, 256 different values are obtained. Output bits independence criterion and Avalanche Effect tests showed that the proposed model has achieved the desired security. In [8], Zaibi and her team developed a dynamic S-BOX chaotic pattern by modifying the S-BOX structure, which increased the security criteria of blocked ciphers. This S-Box is based on a combination of two chaotic maps: one-dimensional and three-dimensional piecewise linear maps. The randomness of these maps was verified using the nist* test package. The Avalanche Effect and Differential Cryptanalysis analysis revealed binary sequences that generate dynamic S-BOX based on an anarchic dual-core that provides security.

In [9], Watanabe and his team designed S-BOX with nonlinear method. Differential and linear parameters have been investigated in this study. The AES has a nonlinear component called S-BOX. The ability of the S-BOX is depend on the Differential Cryptanalysis (DC) and the Linear Cryptanalysis (LC) to the maximum average differential probability (MADP) and the maximum average linear hull probability (MALHP). By adding a split (SP) process and a composite process (CP) in this model, the AES algorithm was changed nonlinearly. Finally, the analysis for MADP / MALHP was performed on 48 patterns and the results show that the M_S-BOX security level is efficient.

This article deals with studying the change of S-BOX1 structure in AES algorithm via LFSR hardware algorithm for increasing the security levels of system. In this regard, we can obtain novel ideas in the case of security improvement in addition to the results of the registered models analysis. However, in this article, Section A examined the overall structure of AES algorithm and S-BOX substitution tables. Section B dealt with the algorithms for the production of LFSR random numbers. In Section C, the proposed model of generating dynamic and key-dependent S-BOX was evaluated and finally in Section D, the proposed model was assessed and compared with the original one. According to the obtained results, it is revealed that the proposed model is an optimum case for increasing the safety of AES algorithm.

2. Reviewing the AES Algorithm

Advanced Encryption Standard has been developed based on the designing rule called Substitution-Permutation Network, either in hardware or software forms [6]. It is applied on a 4*4 matrix of bites in columns order, named State, as the maximum number of AES calculations are taken in a specific Finite Field [6]. The size of the key used in AES

*National Institute of Standards and Technology

cipher determines the number of repeats in transformation cycles, where it transforms the input, named Plaintext, to the output by the name of Cipher text. The numbers of repeating cycles are as 10, 12 and 14 for 128, 192 and 256 bit keys, respectively. The number of cycles is shown by Nr. Each round of the algorithm has 4 steps, except the final round possessing 3 steps. So mix column is not performed at the final round.

Sub-bytes: The non-linear substitution step, where each byte is replaced by another one based on a Lookup Table. S-Box [State (J, I)] =State (J, I) [10]

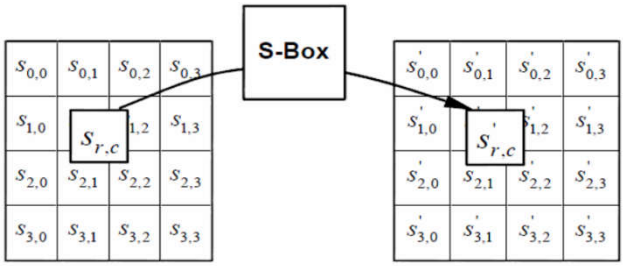

Fig. 1. Substitute byte transformation [10]

Shift Rows: The displacement step, where each row of state is shifted in some steps frequently.

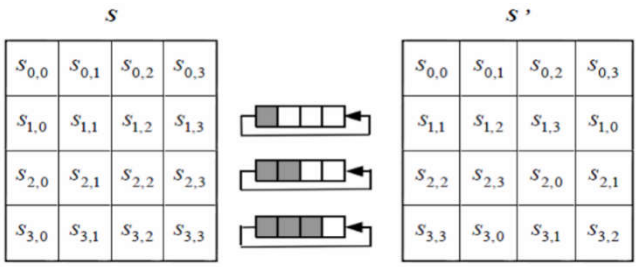

Fig. 2. Shift row transformation [10]

Mix columns: The process of mixing the columns performed on the state columns and combines four bites of each column.

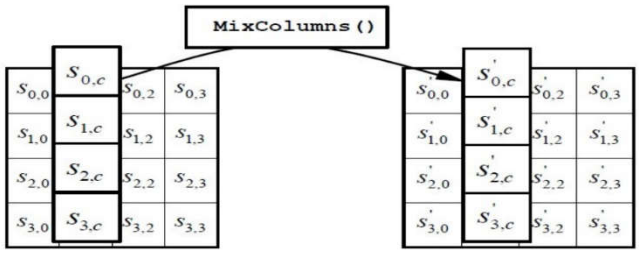

Fig. 3. Mix column transformation [10]

Add-round-key: Each bite of State is combined with cycle key by Xor, bit by bit.

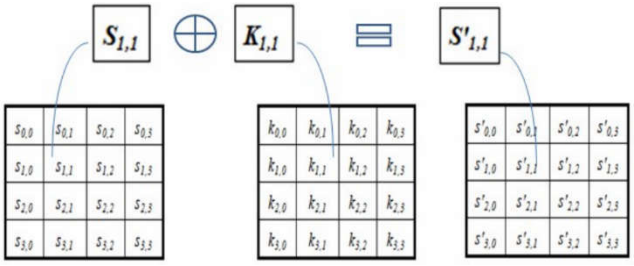

Fig. 4. Add-round-key [10]

As noted earlier, in this paper, the fixed structure of S-BOX was changed to dynamic structure for increasing the security and safety of system. In the main model of AES algorithm, Sub-Bytes take a non-linear substitution on the State bytes and use S-BOX for this purpose. Each byte of State is replaced by a value of S-BOX table, as its index is equal with State value. S-Box [State (j,i)] = State (j,i) [7]. We use the LFSR random number generator algorithm in our proposed model for producing the dynamic and key-dependent S-BOX.

3. Linear Feedback Shift Registers

The resulting sequences of shift registers are used either in cryptography or coding theory. Working basis of Stream encryption is based on shift registers, mostly applied in military encryption applications. Application of linear feedback shift registers in the hardware implementation is very convenient and simple. They also show proper statistical properties. The constituents of feedback shift register are included of shift register and feedback function. In one kind of LFSR generators, each bit is xored and placed in its bit location, instead of using the bits existed in tap sequences. Then the bit with the minimum value is considered as the output bit. This type of generator is called Galios configuration as shown below:

```
Function LFSR
the  input  is x array 8;
f is array 8;   i is integer;
for  i   from  1  to  255   by 1
f<------x; ** The value of x is copied to f
x<---------shift(x);**The value of a bit shifts to the left
```

x(6)<-------- f(7) xor f(1);The bits 2 and 1 are xor and transmitted to bit 6
x(5)<-------- f(6) xor f(1);The bits 2 and 1 are xor and transmitted to bit 5
x(4)<-------- f(5) xor f(1);The bits 2 and 1 are xor and transmitted to bit 4
output x
end;
End;

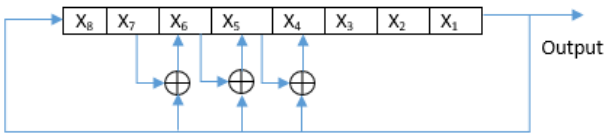

Fig. 5. Galios generator

The above generator is a single operator type, so it can be done in parallel. This type of configuration has greater speeds in hardware implementation, especially in special VLSI circuits. Thus, Galios configuration is a proper candidate, if we want to implement our model by hardware algorithm characterized by parallel operation. That's why we proposed the use of LFSR algorithm with Galios model for dynamic S-BOX unit [8].

4. The Proposed Model

Shivshankar Mishra et al., [8] have implemented the feedback shift register using the Galios generator via VHDL and showed that Galios generator is preferred to Fibonacci in hardware implementations. Thus, we used the Galios generator for generating the semi-random numbers and Ciphertext, while designing the dynamic S-BOX system. In following, we used the key-field dependent in generating S-BOX in order to increase the complexity of Ciphertext. In the initial step, the generated random number is used for selecting one byte of key-field. The random generated number determines the S-BOX value in the second step. The overall steps for generation of dynamic S-BOX are as shown below:

- Receiving the Key-field.
- Transforming the key-field to separated bytes: 16, 24 and 32 bytes for the 128, 192 and 256 bits field keys, respectively.
- Using the LFSR algorithm (in the first step for selecting one random byte among the separated bytes of key-field, as the number generated by LFSR algorithm, shown by Key (i) index) reflects the considered bytes values of the Key-field. If the value of Key (i) is equal to zero or 255, the KEY/2 fixed repeating number was applied for each one. In this way, repeats of 64, 96 and 128 are applied for 128, 192 and 256 bits, respectively, as the number of repeats in the second step is never zero or 255. Because in this case, the LFSR input and output values are the same.
- The number specified in the first step determines the number of LFSR algorithm repeats with Galios generator per each S-BOX unit. After obtaining the output from LFSR generator, the S-BOX value is determined in the second step.
- In the case of zero input of S-BOX, the Key-field is considered as the S-BOX value and the algorithm is run. On the contrary, for input values equal to key-field, the algorithm is not run and output value is zero.

5. Evaluation of the Proposed Model

Safe S-BOX is investigated by the criteria such as avalanche test, strict avalanche test, criteria for dependence of output bits and linear analysis. Thus, the proposed model would be evaluated in all cases to show the improvement of the security of the proposed model in relation to the AES algorithm's main model.

5.1. Avalanche Effect

A small change in the input text can cause an avalanche change in the output results. In other words, a function of f: {0,1} N{0,1}, N has the avalanche properties, if changing a bit of input values causes a change in half of output bits [9].

Table. 1. Avalanche test of the proposed model

INPUT	OUTPUT	Avalanche test
00000000	10000000	0.6
10000000	01011110	
11111111	11001010	0.5
01111111	01110110	

Table. 2. Avalanche test of the reference model

INPUT	OUTPUT	Avalanche test
00000000	10011101	0.5
10000000	01111001	
11111111	00010011	0.5
01111111	10111100	

5.2. The Dependence Criteria of the Output Bits

Where by changing each (i) bit in the input, output bits of (j) and (k) should change independently. The maximum correlation coefficients for each pair are considered as Bic pertaining to that pair. Then among all obtained Bic, the maximum value is set as the Bic function, here S-BOX. The Bic value of function lays in [1, 0] interval, considering zero and 1 at the best and the worst conditions. Results of evaluation of the dependency of the proposed model's output bit for 128 test samples showed 0.56 of dependency coefficient for 2.7 paired bits. Also, the results of evaluation of the dependency of the AES main model's algorithm output bit for 128 test samples showed 0.57 of dependency coefficient for 1.7 paired bits. Since the minimum and maximum allowed values were 0.43 and 0.93, respectively, the proposed model with the value of 0.56 shows the higher dependency to the main model, closer to the determined area [10].

$$Bic(f)=\max(Bic(ai,aj)) \quad 1\leq j,k\leq n,\ j\neq k \qquad (1)$$

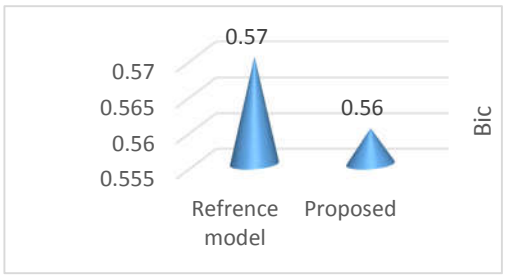

Fig. 6. Comparing the dependency of output bits of the proposed and original models

5.3. Strict Avalanche Criteria (SAC)

In this test, each of its output bits should be changed with a probability of a half, whenever a single input bit x is complemented. In general, the dependence matrix is used to test the SAC of an S-box. Here, the strict avalanche test was conducted on the proposed model and the original model for 100 samples for each bit and the results are shown in the table below [11].

Table. 3. Dependence matrix of the proposed model

	1	2	3	4	5	6	7	8
1	0.49	0.52	0.54	0.54	0.55	0.55	0.52	0.51
2	0.49	0.52	0.43	0.58	0.52	0.45	0.54	0.46
3	0.5	0.47	0.42	0.52	0.51	0.48	0.59	0.47
4	0.46	0.48	0.47	0.52	0.56	0.5	0.56	0.5
5	0.52	0.49	0.48	0.53	0.5	0.47	0.6	0.53
6	0.46	0.51	0.5	0.51	0.47	0.56	0.56	0.51
7	0.49	0.48	0.52	0.54	0.53	0.53	0.48	0.46
8	0.51	0.49	0.6	0.53	0.51	0.53	0.49	0.49
Mean	**0.5094**							

Table. 4. Dependence matrix of the original model

	1	2	3	4	5	6	7	8
1	0.44	0.4	0.44	0.41	0.44	0.44	0.48	0.44
2	0.41	0.45	0.45	0.39	0.43	0.38	0.45	0.44
3	0.46	0.45	0.46	0.4	0.39	0.43	0.44	0.38
4	0.45	0.46	0.42	0.42	0.43	0.39	0.42	0.38
5	0.46	0.42	0.45	0.42	0.38	0.4	0.45	0.46
6	0.42	0.45	0.42	0.48	0.39	0.42	0.44	0.44
7	0.45	0.42	0.38	0.41	0.42	0.48	0.41	0.41
8	0.42	0.38	0.41	0.38	0.47	0.38	0.44	0.44
Mean	**0.4261**							

At the end, the mean of 100 samples for each bit showed a better result (0.5094) than the original model in the Strict Avalanche Criteria test.

5.4. The Linear Analysis Criteria

In this analysis, using the block cipher structure analysis, especially S-BOX and XOR, some bits of input texts and Cipher text become 0 or 1, named the linear equality. The purpose of this test is to create linear equalities between the input text and output cipher, and then counting all linear approximate probability of the S-BOX in a linear approximate table. The lower possibility leads to the higher complexity of the linear sentences [12]. For testing the criteria, the proposed model with 125 equalities and the initial model with 130 equalities were tested for the 256 texts. The results indicated that the proposed model showed better results.

$$Lp=Max_{\Gamma x\Gamma y\neq 0}((\pm\{x|x,\Gamma x=S(x),\Gamma y\})/2^{n})-1/2) \quad (2)$$

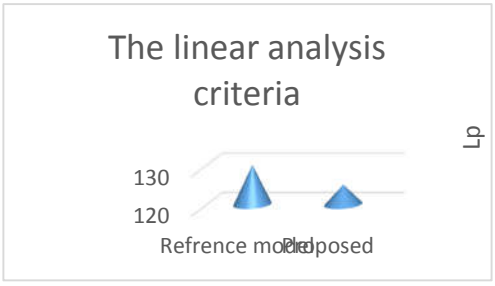

Fig. 7. Comparison of linear analysis of the proposed and reference models

5.5. Analysis of the Key-Field Sensitivity

The Key-field sensitivity is an important parameter in dynamic encryption, as changing one bit in the key-field leads to generation of a completely different S-BOX. To test this parameter, the proposed model was run by the two different Key-fields, considering one bit difference between the two fields.

The output results were shown as figures and graphs to represent the sensitivity difference of the key-filed in the proposed model.

128 bits Key-field No.1=01 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

128 bits Key-field No.2=01 01 02 03 04 05 06 07 08 09 1A 0B 0C 0D 0E 0F

The keys are introduced to the proposed algorithm and converted to the separate bytes. Based on the algorithm of generation of random numbers, one byte is selected for determining the frequency of LFSR algorithm. The S-BOX values have been produced and as shown, there is only one bit of difference between the two Key-fields. But as it could be seen in the Figure (3-5), results of calculations for 256 input are completely different values from 256 output which can be observed in Figures 3 and 4.

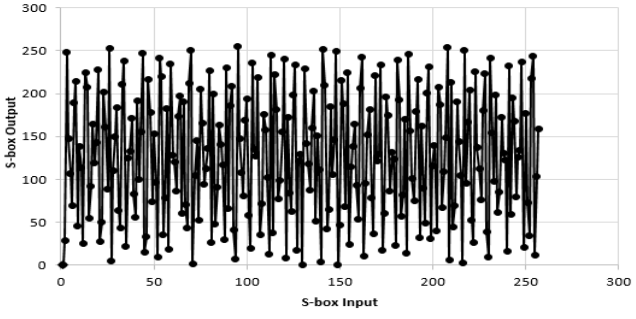

Fig. 8. Comapring the Key-sensitivity analysis for Key-1 proposed model

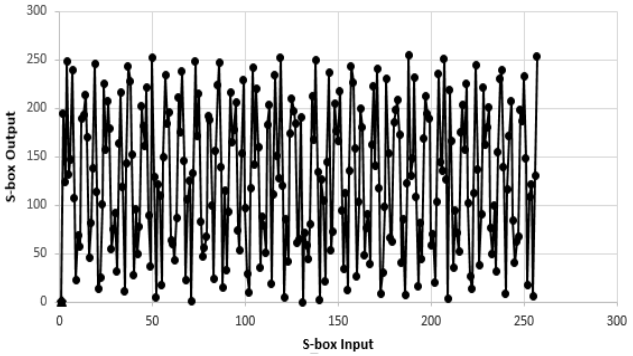

Fig. 9. Comapring the Key-sensitivity analysis for Key-2 proposed model

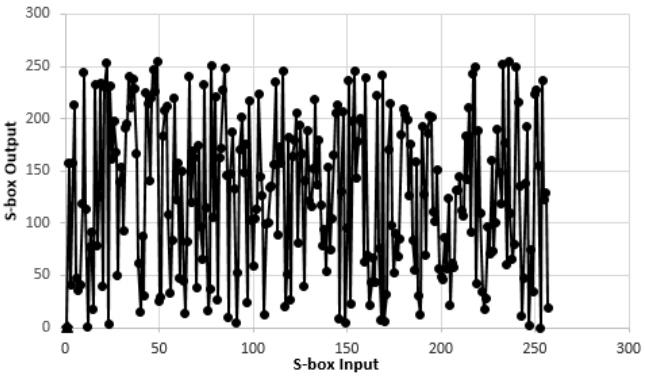

Fig. 10. Comapring the Key-sensitivity analysis for original model

6. Conclusion

According to the results of conducted analyses of the proposed model, all evaluations have been taken successfully and approved for increasing the security of hardware applications. In this regard, by generating the dynamic and Key-dependent S-BOX via LFSR random algorithm, known as hardware algorithms, we could improve the hardware safety and security of the system. The integrated LFSR algorithm sensitive to Clock Pulse can be used and the complexity of generated S-BOX can be studied and evaluated.

Table. 5.Overall comparison of the proposed model and reference model

	Avalanche	The dependence criteria of the output bits	Strict avalanche	The liner analysis criteria
Proposed Model	0.6	0.56	0.51	1.25
Reference Model	0.5	0.57	0.42	1.3

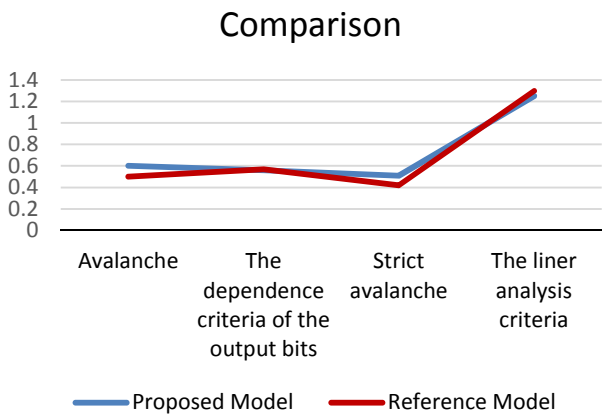

Fig. 11. Overall comparison of the proposed model and reference model

References

[1] Announcing the ADVANCED ENCRYPTION STANDARD (AES) Federal Information Processing Standards publication 197. United States National Institute of Standards and Technology (NIST). November 26, 2001. Retrieved October 2 (2012).

[2] Du, Z.; Zhang, J.; Zhang, Q.; Liu, Z.; Zhu, B., "The Research of S-Box Based on Reconstruction Design Method" International Conference on Computational Intelligence and Security (CIS), pp. 571- 577 (2016).

[3] Kalaiselvi, K.; Kumar, A., "Enhanced AES cryptosystem by using genetic algorithm and neural network in S-box", IEEE International Conference on Current Trends in Advanced Computing, pp. 1- 6 (2016).

[4] Belazi, A; Rhouma, R; Belghith, S., "A novel approach to construct S-BOX based on Rossler system", Wireless Communications and Mobile Computing Conference (IWCMC), International, pp. 611– 615 (2015).

[5] Juremi, J; Ramlan, M.; Sulaiman, S., "A proposal for improving AES S-BOX with rotation and key-dependent Cyber Security", Cyber Warfare and Digital Forensic (CyberSec), International Conference, pp. 38– 42 (2012).

[6] Saberi, I.; Shojaie, B.; Salleh, M., "Enhanced Key Expansion for AES-256 by using Even-Odd method Research and Innovation in Information Systems (ICRIIS)", International Conference on , pp. 1– 5 (2011).

[7] Shivkumar. S; Umamaheswari, G., "Performance Comparison of Advanced Encryption Standard (AES) and AES Key Dependent S-BOX - Simulation Using MATLAB Process Automation, Control and Computing (PACC)", International Conference, pp. 1– 6 (2011).

[8] Zai˙bi, G.; Peyrard, F.; Kachouri, A.; Fournier-Prunaret, D.; Samet, M., "A new design of dynamic S-BOX based on two chaotic maps"; Computer Systems and Applications (AICCSA), IEEE/ACS International Conference on, pp. 1– 6 (2010).

[9] Watanabe, A.; Haruki, H.; Shimotomai, S.; Saito, T.; Nagase, T.; Yoshioka, Y.; Hasegawa, Y., "A New Mutable Nonlinear Transformation Algorithm for S-BOX",Advanced Information Networking and Applications Workshops, Vol. 1, pp. 246- 251 (2007).

[10] Stallings, W., Cryptography and Network Security Principles and Practices, Fourth Edition, (2005).

[11] Hussain, I.; Shah, T.; Gondal, M. A., "A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm", Nonlinear Dynamic, Vol. 70(3), pp. 1791- 1794 (2012).

[12] Mishra, S.; Racksha Tripathi, R.; Kr. Tripathi, D., "Implementation of configurable linear feedback shift register in VHDL", International Conference on Emerging Trends in Electrical Electronics and Sustainable Energy Systems (ICETEESES) pp. 342– 346 (2016).

[13] Mandal, A. K.; Tiwari, A., "Comparative study of Avalanche effect in DES using binary codes Computing and Communication Systems", (NCCCS) National Conference, pp. 1– 4 (2012).

[14] Webster, A. F.; Tavares, S. E., On the design of S-boxes, Adv. in Cryptology-Crypto'85, LNCS, Springer-Verlag, Berlin/Heidelberg/New York, Vol. 218, pp. 523-534 (1986).

[15] Deng, Y.; Shi, H.; Gong, J.; Xie, T., "Research on the avalanche property of the Camellia S-box", Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference pp. 4268– 4271 (2011).

[16] Jakimoski, G.; Kocarev, L., Chaos and Cryptography: block encryption ciphers, IEEE Transaction on Circuits and Systems I, Vol. 48(2), pp. 163- 169 (2001).