VOL. 12, NO. 3, FEBRUARY 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ADVANCED ENCRYPTION STANDARD ALGORITHM VERSUS EXTREME LEARNING MACHINE BASED WEIGHT: A COMPARATIVE STUDY

Hayfaa A. Atee[1, 2], Robiah Ahmad[2], Norliza Mohd Noor[2] and Abidulkarim K. Ilijan[2, 3]
[1]Foundation of Technical Education, Higher Education and Scientific Research, Baghdad, Iraq
[2]Department of Engineering, UTM Razak School of Engineering and Advanced Technology, UTM Kuala Lumpur, Jalan Sultan Yahya Petra, Kuala Lumpur, Malaysia
[3]College of Engineering, Almuthanna University, Samawa, Iraq
E-Mail: haifaaatee@gmail.com

ABSTRACT

Advanced Encryption Standard (AES) is being widely used ciphering/deciphering system has emerged as a standard benchmark. Due to rapid advancement in the hardware specifications, the architecture security of AES became a major concern. Furthermore, the newly developed machine learning dependent encryption architecture called Extreme Learning Machine Based Weight (ELMWi) appears more suitable for sundry cryptographic implementations. This article compares the performance of ELMWi with AES via statistical evaluation, where the parameters such as sensitivity, visual imperceptibility metrics, and key space are determined. Results reveal their similar performances. It is further argued that ELMWi outperforms the AES in perspective of architecture implementation.

Keywords: AES, machine learning, ELM, ELMWi, cryptography.

1. INTRODUCTION

With the rapid escalation of information communication technologies, easy transfer of digital multimedia data over the internet with absolute security, confidentiality, and authentication became a priority. Hackers or unauthorized users can intercept, interrupt, and change the data upon attack. Consequently, the free exchange of sensitive or private data among various users needs to be completely protected, which is achieved using cryptography. In the past, several cryptographic algorithms are developed for transferring the useful data easily over the internet. Despite many attempts to break the security level of AES by attackers, yet it acts as a robust encryption system against attacks. Thus, AES is emerged as a yardstick for measuring the efficiency of any encryption algorithm.

Many types of evaluations are performed for the standard AES involving software implementation, hardware modification, and in combination of both. The comparative including software implementation have been conducted in variant points [1]-[14], where AES based algorithms have mainly focused on different performance criteria. These are centered towards enhancing the security [1]-[6], [11], [13], speeding up [2], improving the complexity [8][10], enhancing the performance [12], evaluating the power analysis attack [14], and implementing the differential power analysis (DPA) [7] of the cryptosystem. Meanwhile, researchers used different tactics to enhance AES security in terms of S-box modifications and improvement of Shift-Row transformation [1]. Despite intensive research an accurate and efficient cryptosystem with supreme security for digital multimedia data transfer is far from being achieved. In this spirit, we modify the conventional AES algorithm to achieve high level data encryption security against statistical attack.

A comparative study in terms of security is made among various encryption schemes such as AES, Compression Friendly (CF), Chaotically Coupled Chaotic Map (CCCM) and Bernoulli Map (BM) [3]. It is revealed that each scheme has its own advantages in terms of correlation, entropy, diffusion, and key sensitivity. Several options for fault analysis resistance of field programmable gate array (FPGA) implementations of AES are explored [4]. Besides, a comparison is made between the throughput and area overheads associated with parity based error detection at the level of algorithm, round, and operation redundancy based countermeasures. Nonetheless, such approach does not provide enough security and limited to the detection of odd number of bit error only. A secure satellite image encryption scheme is proposed by combining the AES with chaotic map [5]. Meanwhile, the number of AES rounds are increased from 10 to 16 for enhancing the security level [6]. However, this became computationally time expensive.

The secret key is used to generate randomly the AES, where key dependent S-box is changed for each alteration of the secret key [2]. This approach is beneficial because by changing the secret key one can make a massive number of S-box tables. Besides, the AES S-box is generated using RC4 stream cipher to improve the security of AES [11]. The key expansion procedure is used to make the S-box key dependent and more dynamic. However, this increased the complexity due to the incorporation of rigor differential and linear cryptanalysis. Such key expansion alteration procedure with key dependent S-box has improved the security level of the proposed AES algorithm. A generalized multiple S-boxes scheme is introduced for improves security [13]. A method is suggested [15] based on the affine orbit and mapping of the power function used in AES S-box. In this scheme,

256 different S-boxes are produced to augment the complexity against different attacks.

A new algebraic technique is proposed and compared with AES, where an AES S-box is developed with improved complexity [7]. This modified AES S-box is found to be robust against differential cryptanalysis with high level of security. For the secured transmission of chaotic and AES encrypted images, a comparative evaluation is made among the fast Fourier transform (FFT) based orthogonal frequency division multiplexing (OFDM), the discrete cosine transform (DCT) based OFDM and the discrete wavelet transform (WDT) based OFDM techniques [9]. It is concluded that although each technique has some feature superior to others, but AES possessed the highest security.

Comparatives and modification of software implementation are performed to increase the speedup of AES. The efficiency of conventional AES and modified AES algorithm is tested by National Institute of Standards and Technology (NIST) [2]. Results showed that the proposed algorithm is around eight times faster than the conventional one. Furthermore, an improvement of the security and complexity of S-box is proposed [8][10]. Meanwhile, an improved AES S-box is developed to improve the complexity of algebraic expression with increasing number of terms from 9 to 255 [8]. The improved AES S-box revealed good properties of Boolean functions and remained robust against differential cryptanalysis with high security reliance. A new strategy called 'Gray augmentation' is introduced to enhance the security of S-box [10], which increased the algebraic complexity and provided further resistance against algebraic as well as interpolation attack. The proposed algorithm enhanced the security and algebraic complexity of AES by incorporating the binary Gray code encoding as the pre-processing step of S-box.

Performance enhancement in terms of time ciphering and pattern appearance is also suggested [12]. Two strategies are developed for such modifications. First, a new S-box is used instead of conventional AES S-box to increase the hardware requirements and to reduce the number of rounds to one. Experimental results showed that although the proposed modifications make the AES algorithm faster but one round is too few to maintain the high security level. Meanwhile, six different AES-S-box architectures are used to design and evaluate the power analysis [14]. The robustness of different AES S-box designs *field-programmable gate array* (FPGA) against chosen-plaintext attack (CPA) is evaluated by analyzing the *signal-to-noise ratio* (SNR). This approach is found to suffer from the slower speed. Many developments that occurred for hardware modification are systematically compared to determine their robustness against statistical attack [16]–[21]. Comparative evaluations are made in terms of complexity [15], fault detection [16], lower power and less delay [17], speed [18]–[20], energy efficiencies and high performance of area [21].

A new concurrent fault detection technique based on modification of the AES architecture for achieving a reliable AES implementation is introduced [16]. Results exhibited that the modified AES achieved high values for the fault coverage detection. A novel method based new S-box architecture is proposed to obtain least delay, low power, and area efficient S-box for AES processor [17]. It is acknowledged that the new design outperformed the conventional S-box in terms of power, delay time, and size. The speed is further enhanced following the existing strategies [18]–[20]. An integration architecture called multimode multiplier (MM) algorithm is designed for enhancing the performance of AES by saving the area cost [18]. The hardware performance of the conventional AES is compared with that of MM algorithm. It is shown that the performance of the proposed architecture is about two to six times higher than traditional AES algorithm.

A new AES encryptor architecture with efficient merging techniques is proposed to accelerate the AES algorithm [19]. This new architecture revealed remarkable improvements in terms of efficiency and speed. The speed of AES algorithm is enhanced in terms of the number of assembly instruction [20], where the advantages of using the Adder-Based Dynamic Architecture for Processing Tailored Operators (ADAPTO) is realized [22]. Results showed that the ADAPTO architecture is useful for accelerating the AES algorithm with most complex operation. Dedicated efforts are made towards software implementation and hardware modification [23]–[25] for strengthening the security level and performance of AES algorithm. The design and software implementation of a high-performance area-efficient AES cipher is performed on the many-core platform [23]. A comparison is made in terms of the throughput of the cipher and the number of processor utilized. The proposed design is found to outperform the conventional AES.

Based on HDL and IP core, a detail performance evaluation is made on two hardware models of Rijndael's AES encryption algorithm with key length of 128 bit [24]. Comparison between AES encryption and proposed model is conducted in terms of encryption time, size, speed, and memory consumption. Results displayed that the proposed model enhanced the throughput per slice measure. The software is combined with the architecture for improving the AES S-box [25], where a low power design technique is applied to reduce the power consumption. It is acknowledged that the improvements lead to the smallest, fastest, and least power consuming circuit for the serial, the outer round pipeline and inner-outer round pipeline implementation of the AES. In this view, we compare the performance of AES encryption algorithm of 128 key length and plaintext block of 128 bits with that of ELMWi algorithm [26]. The detailed statistical analysis is carried out and the parameters including sensitivity, visual imperceptibility metrics, and key space are determined.

This paper is organized as follows. Section 2 describes the AES algorithm and Section 3 highlights the features of ELMWi algorithm. Section 4 depicts the performance evaluation measures. The experimental results are discussed in Section 5. Section 6 concludes the paper.

2. AES ALGORITHM

In cryptography, the AES remains the most important algorithm [27] due to its high speed, strong encryption, and low memory costs. It is a block cipher algorithm with block length of 128 bits and the variable key length of 128, 192, or 256 bits, where the key length is indicated in the version of AES. The number of rounds is decided based on the key length (10, 12, or 14 rounds) [28]. Generally, N_b denotes the block length (32 bit words), N_k refers the key length, and N_r signifies the number of rounds. All AES versions work with a block size of 16 bytes (i.e. $N_b \times 32 = 4 \times 32 = 128$ bits). The internal functions of the AES work in a finite field in an irreducible polynomial modulo over GF (2^8), where the 8 bits can be expressed as a coefficient of polynomial [29] via:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^{7} b_i x^i \quad (1)$$

Commonly, AES works on array of 4 rows and 4 columns called State s. At the beginning the input bytes (in_0, …., in_{15}) are copied into the State s. After the initial implementation of Add-Round Key() transformation, the State s is transformed by applying a round function (10, 12, or 14 round, depending on the key length). All round function is consisted of four transformation steps (SubBytes, ShiftRows, MixColumns, and AddRoundKey) [30] except the last round which contains three steps without MixColumns step [29]. Each step performs a specific job as described below:

a) SubByte step: replaced the bytes of the State s according to the conventional table called S-box.
b) ShiftRows step: shifts the left rows of the State s by different offsets.
c) MixColumns step: mixed the data within each column of the State s.
d) AddRoundKey step: adds the round key to the State s.

Figure-1 illustrates the encryption and decryption processes of AES algorithm [31]. It is worth noting that AES is being a reversible algorithm the transformations that used in the encryption procedure can be inverted and applied in reverse order to produce AES decryption algorithm.

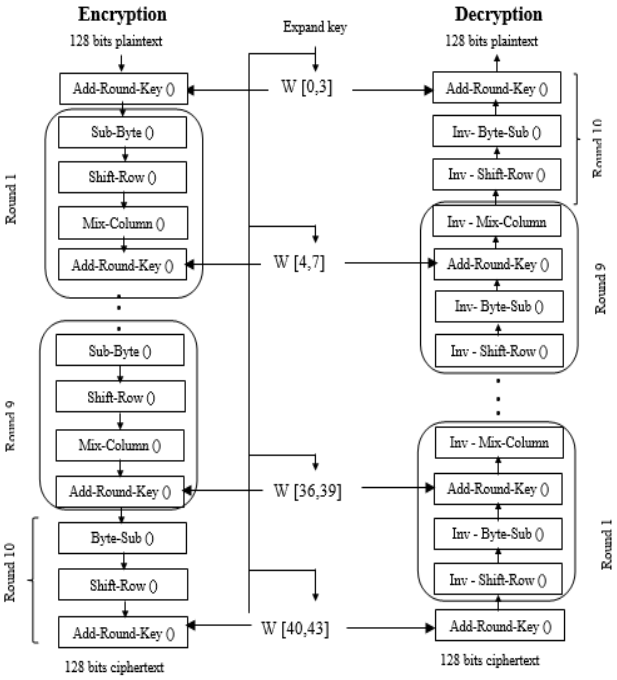

Figure-1. Encryption and decryption processes in AES algorithm.

3. ELMWi ALGORITHM

The ELMWi algorithm is developed using the Real-time Recurrent Neural Network (RRNN). To overcome the weaknesses of RRNNs, ELMWi is derived from the gradient-based learning for symmetric cipher [32]. Furthermore, the cipher is trained on ELM, where the target plaintext vectors (Wi called weights) of the input-hidden layer are used as training data. Table 1 summarizes the pseudo-code for ciphering and deciphering algorithm. Figure-2 displays the chaining mode of the cryptography system.

Table-1. Pseudo code of ELM-Based RRNNs ciphering and deciphering using Wi as the target.

ELM-based RRNN ciphering	**ELM-Based RRNN Deciphering**
Input:	**Input:**
$M_i = \{M_1, M_2, \dots, M_n\}$,	$C_i = \{(V_1, E_1), (V_2, E_2), \dots (V_n, E_n)\}$,
Key={X,Y}, $RRNN_0$	Key={X,Y}, $RRNN_0$
Output: $C_i = \{(V_1, E_1), (V_2, E_2), \dots (V_n, E_n)\}$	**Output:** $M_i = \{M_1, M_2, \dots, M_n\}$,
Procedure:	**Procedure:**
$RRNN_1 = Train\_ELM(RRNN_0, Key, 1)$	$RRNN_N = Train\_ELM(RRNN_0, Key, 1)$
$M_0 = RRNN_N\_Out$	$M_0 = RRNN_N\_Out$
$Y_0 = M_0$	$Y_0 = M_0$
$i = 1$	$i = 1$
$j = 1$	$j = 1$
WHILE $(i \le n)$	**WHILE** $(i \le n)$
$X_i = Y_{i-1} \| M_i$	$Y_i = RRNN_j.F_2(V_i, W_i)$
$V_i, W_i = RRNN_j.F_1(X_i)$	$W_i = E_i + Y_i$
$Y_i = RRNN_j.F_2(V_i, W_i)$	$RRNN_{j+1} = Train(RRNN_j, M_i, 1)$
$E_i = W_i - Y_i$	
$RRNN_{j+1} = Train(RRNN_j, M_i, 1)$	Add_To_Plain (M_i)
Add_To_Cipher $((V_i, E_i))$	$M_i = \{M_1, M_2, \dots, M_n\}$,
$C_i = V_i \| E_i$	
$i = i + 1$	$i = i + 1$
$j = j + 1$	$j = j + 1$
END	**END**

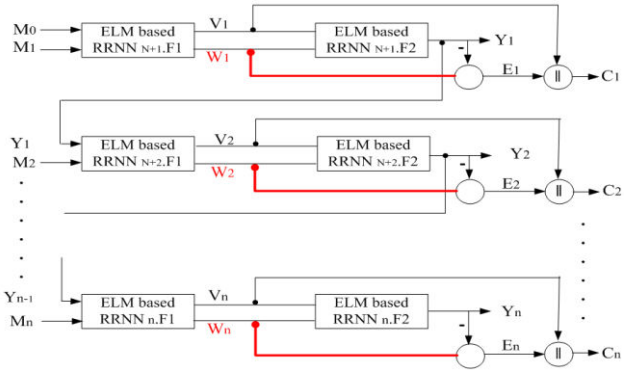

Figure-2. The chaining mode of the ciphering process in ELM-based RRNNs using Wi as the target.

In the algorithm, ELMWi cipher is the first cipher that uses machine learning to capture information of the key for further using them in the ciphering process. The random nature of ELMWi has been exploited to achieve the desired random ciphering. Specifically, the training process in ELMWi is accomplished by assigning random weights of the input-hidden layer neurons, which prevents the convergence of error. Eventually, the error is added as one part of the encrypted data with the output of the first block in the cipher that represents the hidden layer and the activation function.

4. SECURITY PERFORMANCE ANALYSIS

To evaluate the security performance of the $ELMWi$ encryption algorithm and compare with AES algorithm, several criterions are used. Experiments are conducted on a specific message "abcdef1234567890" in terms of statistical analysis of confusion, diffusion, and correlation between plaintext and ciphertext, sensitivity of key and plaintext, visual imperceptibility metrics, and the key space. These measures are depicted hereunder.

4.1 Statistical analysis

To circumvent the known and chosen plaintext attack in block cipher algorithms, a good encryption algorithm must be considered with the desirable property. It must spread out the influence of a single plaintext bit over as much as possible so that the statistical structure of the plaintext are hidden [33]. Generally, block ciphers' security depends on their computing security which is determined by the confusion and diffusion criteria [34], [35]. Diffusion dissipates statistical structure of plaintext over bulk of ciphertext, whereas the confusion relates the ciphertext to the key as complex as possible.

To compare the robustness of the two encryption schemes, statistical analysis is performed. It demonstrated the superior confusion and diffusion properties with strongly resisting nature against the statistical attacks. In a cipher with good diffusion, the flipping of an input bit should change each output bit with a probability of one-half. This is performed by testing the number of bit change rate (NBCR), and Jaccard similarity (d_J) between plaintext and ciphered text besides the correlation. The confusion and diffusion criteria between the plaintext and the ciphertext are computed.

4.1.1 Number of bit change rate

In the image encryption field, the Number of Pixel Change Rate (NPCR) is a widely used in security analysis measure for deciding the level of differential attacks. In this article, it is used for plaintext evaluation because of images and texts are data, where bit employed instead of pixel. The percentage rate of changed bits yields:

$$NBCR = \frac{\sum_{i=1}^{N} D(i,j)}{N} \times 100 \quad (2)$$

where N is the number of bits, and $D(i,j)$ denotes the differences between the plaintext and the ciphertext. The higher percentage of changing bits indicates the extreme dissimilarity of the ciphertext from the plaintext.

4.1.2 Jaccard similarity index

Jaccard similarity (d_J) is being a common index (ranges between "0" and "1") for binary and non-binary variables is used to measure the similarity between two asymmetric information variables. The Jaccard index of the two sets is defined as the size of their intersection divided by the size of their union. A value "0" signify the complete dissimilarity of the two sets, a value "1" implies their true resemblances, and values between 0 and 1 represents a degree of their similarity. Jaccard similarity (d_J) is defined as:

$$d_J(\mathrm{x},\mathrm{y}) = \left(\frac{\sum_i \min(x_i,y_i)}{\sum_i \max(x_i,y_i)}\right) \quad (3)$$

where x_i, y_i denote data component in original plaintext and its corresponding value in its encrypted version, respectively.

4.1.3 Correlation

A robust encryption algorithm must be able to remove the correlation between the plaintext and the ciphertext. The less correlation between the plaintext bits and ciphertext bits indicates a preferable performance. The correlation (r) between the plaintext (x) and the ciphertext (y) of N bits is expressed as:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad (4)$$

With

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}\big(x_i - E(x)\big)\big(y_i - E(y)\big) \quad (5)$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (6)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \quad (7)$$

4.2 Sensitivity

The key sensitivity and plaintext sensitivity implies that a slight change in the key or plaintext data should result in great changes in the ciphertext [36]. This feature provides the cryptosystem with high security against statistical and differential attacks. The notions of key sensitivity and plaintext sensitivity are explained underneath.

4.2.1 Key sensitivity

The protection of the whole cryptosystem requires a key that should not only be large but also be of high sensitivity. The key sensitivity is determined by measuring the number of changes ciphertext bits after altering one bit in the original key. The calculation of the rate of change in the number of pixels between the two ciphertexts (encrypted plaintext) is performed after changing one bit in the original key. NBCR metric is used for calculating this evaluation performance.

4.2.2 Plaintext sensitivity

Plaintext sensitivity is determined using NBCR and $d_J$metric for two cipherstexts after changing one bit in the plaintext.

4.3 Visual imperceptibility metrics

For the performance evaluation, experiments are conducted on the test message. Measures such as the Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Structural Similarity (SSIM) index, and d_J are determined, which are regarded as the measures for the modification of the original text from the encrypted one. In the present cryptography context, each one is interpreted differently. The value of the metrices such as PSNR, SSIM and d_J indicate the resmeblance between the original and the encrypted text. Thus, by minimizing these measures a better performance can be attained. Conversely, MSE signifies how much change occurs between the original text and its encrypted version. Therefore, a better performance can be achieved by maximizing these measures. The following relations are used to calculate the aforementioned metrices:

$$PSNR = 10 * log\left[\frac{255^2}{MSE}\right] \quad (8)$$

$$\text{MSE} = \frac{1}{\text{N}\times\text{M}}\sum_{\text{i}=0}^{\text{N}-1}\sum_{\text{j}=0}^{\text{M}-1}[\text{Origianl(i, j)} - \text{Encrypted(i, j)}]^2 \quad (9)$$

$$SSIM = \frac{(2\mu_x\mu_y+C_1)(\ \sigma_{xy}^2+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (10)$$

where μ_x, μ_y, σ_x, σ_y, and $\sigma_{xy}$are the local means, standard deviation, and cross-covariance for plaintext x and ciphertext y, respectively with $C_1$and C_2as constant.

4.4 Key space

The more the possible keys for an cryptosystem, the higher is the resistances against the brute force attack [37]. The key space (2^n) that is determined by the number of bits (n) used in the key is a considered as the measure of robustness to brute force attack.

5. EXPERIMENTAL RESULTS

Experiments are conducted on text comprising of 16 bytes because of the AES is the block cipher algorithm. Table 2 compares the performance of AES algorithm with that of $ELMWi$ algorithm.

Table-2. Comparative performance of AES algorithm to the $ELMWi$ algorithm.

Category	Performance Criteria	Metric	AES	ELMWi
Statistical analysis	Confusion and diffusion (Between Plaintext and ciphertext)	NBCR	100%	99.01%
		d_J	0	0.0099
	Similarity	Correlation	-0.0902	-0.1462
Sensitivity	Key sensitivity One bit changes in the key (Between two ciphertexts with one-bit changes in the key)	*NBCR*	100%	98.82%
	Plaintext sensitivity and dissimilarity One bit changes in plaintext (Between two ciphertexts with one bit changes in Plaintext)	*NBCR*	100	98.82%
		d_J	0	0.0118
Visual imperceptibility metrics	Visual imperceptibility (Between the ciphertext and the Plaintext)	*PSNR*	8.5372	10.3433
		MSE	9106.70	6192.30
		SSIM	0.0147	0.00048865
		d_J	0	0.0099
Key space	Number of possible permutations of keys	2^n	2^{128}, 2^{192}, 2^{256}	No constraints on the key length

The results (Table-2) clearly show that ELMWi and AES have competitive performance. Yet, AES has outperformed the ELMWi within a small range of values of some evaluation measures. However, ELMWi is more flexible for hardware implementation with assured tolerance against attack. This enhanced robustness of ELMWi algorithm against attack is attributed to its fixed architecture, where enough alteration in the training data is possible during attack. This in turn generates totally different algorithmic behavior of encryption and remains resistant against attack. Besides, ELMWi algorithm has potential for computational optimization due to the presence of one hidden layer. On top, the key cannot be obtained using brute force attack due to unrestricted key length. Thus, the attacker cannot break the encryption at all without obtaining the topology of the proposed algorithm.

6. CONCLUSIONS

We evaluate and compare the security performance of AES and ELMWi encryption algorithm using various statistical measures. Performance evaluation is made in terms of statistical analysis, sensitivity, visual imperceptibility metrics, and key space. Experimental results revealed somewhat similar security performance of both AES and ELMWi in all aspects of attack. Moreover, ELMWi is found to outperform the AES algorithm and provided an additional level of security. This superior nature of ELMWi encryption algorithm is ascribed to the users' allowance of changing the algorithmic core of the cipher by simply altering the data weights of the neural network. Thus, it enabled the hardware programmed ciphers to be more secure with less cost. It is worth to implement the ELMWi encryption system in hardware

architecture such as field-programmable gate array (FPGA) for evaluating its computational and fault tolerance attributes as compared to traditional AES based system.

REFERENCES

[1] S. H. Kamali, R. Shakerian, M. Hedayati, and M. Rahmani. 2010. A New Modified Version of Advanced Encryption Standard Based Algorithm for Image Encryption. in International Conference on Electronics and Information Engineering (ICEIE). 1: 141–145.

[2] K. Kazlauskas, G. Vaicekauskas and R. Smaliukas. 2015. An Algorithm for Key-Dependent S-Box Generation in Block Cipher System. Informatica. 26(1): 51-65.

[3] J. Ahmad, S. O. Hwang and A. Ali. 2015. An Experimental Comparison of Chaotic and Non-chaotic Image Encryption Schemes. Wirel. Pers. Commun. 84(2): 901-918.

[4] A. A. Kamal and A. M. Youssef. 2009. An FPGA Implementation of AES with Fault Analysis Countermeasures. in International conference on Microelectronics (ICM). (2): 217-220.

[5] B. Muhaya and F. T. 2011. Chaotic and AES cryptosystem for satellite imagery. Telecommun. Syst. 52(2): 573-581.

[6] [6] P. Kumar and S. B. Rana. 2016. Development of modified AES algorithm for data security. Opt. - Int. J. Light Electron Opt. 127(4): 2341-2345.

[7] M. Petrvalsky, M. Drutarovsky and M. Varchola. 2013. Differential Power Analysis of Advanced Encryption Standard on accelerated 8051 processor. In: 23rd International Conference. pp. 334-339.

[8] J. Liu, B. Wei, X. Cheng and X. Wang. 2005. An AES S-Box to Increase Complexity and Cryptographic Analysis. 19th Int. Conf. Adv. Inf. Netw. Appl. 1(AINA Papers): 724-728.

[9] M. Hilmey Shaheen, S. M. Elhalafwy and M. A. Zein Eldin. 2009. Efficient transmission of chaotic and AES encrypted images with OFDM over an AWGN channel. Proc. - 2009 Int. Conf. Comput. Eng. Syst. ICCES'09. pp. 353-358.

[10] T. M. Triet, B. D. Khanh, and D. A. Duc. 2008. Gray S-Box for Advanced Encryption Standard. in 2008 International Conference on Computational Intelligence and Security. pp. 253-258.

[11] S. Shivkumar and G. Umamaheswari. 2011. Performance Comparison of Advanced Encryption Standard (AES) and AES Key Dependent S-Box - Simulation Using MATLAB. in 2011 International Conference on Process Automation, Control and Computing. pp. 1-6.

[12] S. M. Wadi and N. Zainal. 2013. Rapid Encryption Method based on AES Algorithm for Grey Scale HD Image Encryption. in The 4th International Conference on Electrical Engineering and Informatics (ICEEI). 11(Iceei): 51-56.

[13] M. Khan and N. A. Azam. 2015. S-Boxes Based on Affine Mapping and Orbit of Power Function. 3D Res. 6(2): 12.

[14] [14] K. H. B. K. H. Boey, P. Hodgers, Y. L. Y. Lu, M. O'Neill, and R. Woods. 2010. Security of AES Sbox designs to power analysis. in Electronics Circuits and Systems ICECS 2010 17th IEEE International Conference on. pp. 1232-1235.

[15] L. Cui and Y. Cao. 2007. A New S-Box Structure Named Affine-Power-Affine. Inf. J. Innov. Comput. Inf. Control. 3(3): 751-759.

[16] H. Mestiri, F. Kahri, B. Bouallegue and M. Machhout. 2015. A high-speed AES design resistant to fault injection attacks. Microprocess. Microsyst. 41(C): 47-55.

[17] F. S. Hossain and M. L. Ali. 2015. A Novel Byte-Substitution Architecture for the AES Cryptosystem. PLoS One. 10(10): 1-17.

[18] C. H. Wang, C. L. Chuang, and C. W. Wu. 2010. An efficient multimode multiplier supporting AES and fundamental operations of public-key cryptosystems. IEEE Trans. Very Large Scale Integr. Syst. 18(4): 553-563.

[19] I. Hammad and K. El-Sankary. 2010. High-Speed AES Encryptor with Efficient Merging Techniques. Embed. Syst. Lett. IEEE. 2(3): 67-71.

[20] G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, S. Pontarelli, M. Re, and A. Salsano. 2011. Implementation of the AES Algorithm Using a Reconfigurable Functional Unit. in Signals, Circuits

and Systems (ISSCS), 2011 10th International Symposium on. IEEE, 2011. pp. 1-4.

[21] H. Naveen and M. Ramesh. 2014. Parallel AES Encryption Engines for Many-Core Processor Arrays. Int. J. Innov. Res. Comput. Commun. Eng. 2(1): 242-247.

[22] G. C. Cardarilli, L. Di Nunzio, and M. Re. 2008. A full-adder based reconfigurable architecture for fine grain applications: ADAPTO. In: International Conference on Electronics, Circuits, and Systems. pp. 1304-1307.

[23] B. Liu and B. M. Baas. 2011. A high-performance area-efficient AES cipher on a many-core platform. In: 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR). pp. 2058-2062.

[24] J. Yenuguvanilanka and O. Elkeelany. 2008. Performance evaluation of hardware models of advanced encryption standard (AES) algorithm. In: IEEE Southeast Conference 2008. pp. 222-225.

[25] D. Bayhan and S. B. Ors. 2014. Analyzing and comparing the Montgomery multiplication algorithms for their power consumption. J. Intell Manuf. 25(2): 263-271.

[26] H. A. Atee, R. Ahmad, N. M. Noor, A. M. S. Rahma, and M. S. Sallam. 2016. A novel Extreme Learning Machine-based cryptography system. Secur. Commun. Networks.

[27] P. Hellekalek and S. Wegenkittl. 2003. Empirical evidence concerning AES. ACM Trans. Model. Comput. Simul. 13(4): 322-333.

[28] A. Biryukov and I. Nikolić. 2010. Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others. in Advances in Cryptology - Eurocrypt 2010, H. Gilbert, Ed. Springer. pp. 322-344.

[29] O. Rolf. 2005. Contemporary Cryptography. Boston - London: Artech House Inc.

[30] N. Sklavos and O. Koufopavlou. 2002. Architectures and VLSI implementations of the AES-Proposal Rijndael. IEEE Trans. Comput. 51(12): 1454-1459.

[31] W. Stallings. 2011. Cryptography and Network Security Principles and Practice, 5th ed. Pearson.

[32] M. Arvandi, S. Wu, and A. Sadeghian. 2008. On the Use of Recurrent Neural Network to Design Symmetric Ciphers. IEEE Comput. Intelligence Mag., no. May, pp. 42-53.

[33] Bruce Schneier, Applied cryptography, 2nd ed. New York: John Wiley and Sons press, 1996.

[34] C. E. Shannon. 1948. Communication Theory of Secrecy Systems. Best Syst. Tech. J. 28(4): 656-715.

[35] G. Chen, Y. Mao, and C. K. Chui. 2004. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons and Fractals. 21(3): 749-761.

[36] S. Lian, J. Sun, and Z. Wang. 2005. Security Analysis of A Chaos-based Image Encryption Algorithm. Phisica A, Elsevier Sci. pp. 1-16.

[37] M. J. Rostami, S. Saryazdi, H. Nezamabadi-pour and A. Shahba. 2015. Chaos-Based Image Encryption Using Sum Operation Modulo 4 and 256. IETE J. Res. 2063(July): 179-188.