Indonesian Journal of Electrical Engineering and Computer Science
Vol. 19, No. 3, September 2020, pp. 1574 ~1581
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v19.i3.pp 1574 -1581

Enhancing advance encryption standard security based on dual dynamic XOR table and mixcolumns transformation

Adnan Ibrahem Salih[1], Ashwaq Alabaichi[2], Ammar Yaseen Tuama[3]
[1,3]Science College, Computer Science Department, University of Kirkuk, Iraq
[2]Engineering College, Biomedical Engineering Department, University of Kerbala, Iraq

Article Info

Article history:

Received Nov 19, 2019
Revised Feb 9, 2020
Accepted Mar 23, 2020

Keywords:

3D chaotic map
AES
Dynamic XOR
Mixcolumns
NIST

ABSTRACT

An efficient approach to secure information is critically needed at present. Cryptography remains the best approach to achieve security. On this basis, the National Institute of Standards and Technology (NIST) selected Rijndael, which is a symmetric block cipher, as the advanced encryption standard (AES). The MixColumns transformation of this cipher is the most important function within the linear unit and the major source of diffusion. Dynamic MixColumns transformation can be used to enhance the AES security. In this study, a method to enhance the AES security is developed on the basis of two methods. The first method is an extension of a previous study entitled " A novel Approach for Enhancing Security of Advance Encryption Standard using Private XOR Table and 3D chaotic regarding to Software quality Factor." In the current study, the fixed XOR operation in AES rounds is replaced with a dual dynamic XOR table by using a 3D chaotic map. The dual dynamic XOR tables are based on 4 bits; one of the xor tables is used for even rounds, and the other is used for odd rounds. The second method is dynamic MixColumns transformation, where the maximum distance separable (MDS) matrix of the MixColumns transformation, which is fixed static and public in every round, is replaced with a private dynamic MDS matrix utilizing 3D chaotic map. A 3D chaotic map is used to generate secret keys. These replacements enhance the AES security, particularly the resistance against attacks. Diehard and NIST tests, entropy, correlation coefficient, and histogram are used for security analysis of the proposed method. C++ is used to implement the proposed and original algorithms. MATLAB and LINX are used for the security analysis. Results show that the proposed method is better than the original AES.

Corresponding Author:

Ashwaq Alabaichi,
Engineering College,
Department of Biomedical Engineering,
University of Kerbala,
Hilla Road, Freha, Kerbala 56001, Iraq.
Email: ashwaq.alabaichi@gmail.com

1. INTRODUCTION

Safekeeping of information has been a trouble of concern over the past several decades. Nowadays, people exchange large amounts of data over unconfident channels, where any private data can be unprotected [1, 2]. Cryptography is probably the most important data protection technology. Numerous encryption algorithms have been developed to ensure data security. To establish an advanced encryption standard (AES), the National Institute of Standards and Technology (NIST) has selected Rijndael, a widely used symmetric key encryption standard, particularly when data confidentiality is a critical issue. This highly efficient cipher is particularly appropriate for encrypting relatively long plain data [1-3]. Three versions of AES based on key

length are conducted: AES-128, AES-192, and AES-256. The keys are represented in arrays with sizes of 4 × 4, 4 × 6, and 4 × 8, respectively. Among these arrays, a state is when 128-bit block data are arranged in a 4 × 4 array. Originally, four successive transforms are conducted on a state in 10, 12, or 14 rounds based on key length [4, 5].

The MixColumns transformation of the AES is one of the critical components and the responsible for diffusion. It plays an important role with respect to the wide trail strategy of the cipher. Each column is treated as a polynomial using Galois Field (GF) of 2^8. Modulo $x^4 + 1$ is multiplied by a fixed polynomial $c(x)=3x^3+x^2+x+2$. The inverse of this polynomial is $C\text{-}1(x) = 11x^3+13x^2+9x+14$. The MixColumns transformation can be performed by multiplying a coordinate vector of the four numbers within the Rijndael's Galois field with the circulated maximum distance separable (MDS) matrix as following.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

The math operation is conducted within the median Rijndael's Galois field, thus resulting in a complicated multiplication operation, whereas the addition operation is simple [5-7]. Figure 1 shows the AES MixColumns transformation.

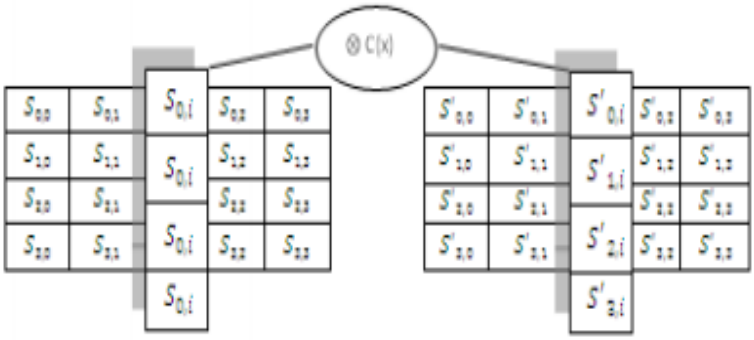

Figure 1. AES MixColumns transformation

The analysis of AES resistance to differential and linear cryptanalyses shows that random, unknown, and key-dependent permutations can enhance the resistance of block ciphers against differential and linear cryptanalyses and that block ciphers can be completely dynamic and unknown to such cryptanalyses. Therefore, dynamic MixColumns transformation can be applied to improve the AES security [7-9]. The fixed XOR table can be replaced with a dynamic XOR table during rounds in the AES factor for improving the AES security as well [1]. Chaos has competitive advantages, such as high sensitivity to initial values, ergodicity, mixing property, deterministic dynamics, and structure complexity. Consequently, chaos theory has attracted extensive attention from the field of cryptography [2, 9, 10]. On this basis, secret keys are in this study to enhance the AES security.

Many related studies proposed different technique for enhancing the security of AES. The predefined XOR operation is eplaced with a novel private XOR table during rounds in AES by using a 3D chaotic map as a private key [1]. The MixColumns transformation in the AES is replaced with dynamic MixColumns transformation on the basis of DNA processes and structure, which rely on keys [7]. MixColumns transformation in the AES is replaced by an alike dynamic MixColumns transformation. A dynamic MixColumns transformation includes dynamic MDS matrices, which are based on the default MDS matrix of AES and m-bit additional key. Here, m is a variable length that does not exceed the product of 31.97 and one less the number of encryption rounds [6]. S-boxes bank as a rotor mechanism and dynamic key MDS matrix is used in AES algorithm. However, in this manner, the AES becomes key dependent to enhance its resistance to frequency attacks [11]. The conservation of various good cryptographic assets of MDS matrices direct exponent transformation is reported. This report has important applications in constructing dynamic diffusion layers for block ciphers. The strength of ciphers against developing cryptanalyses can be enhanced by dynamic MDS diffusion layers [12]. Dynamic MDS matrices are generated

from direct exponent and scalar multiplication transformations on the basis of the original MDS matrices. Then, they calculated the outputs and inputs of these matrices when they are available. They reported that encryption and decryption by the dynamic MDS matrices is performed fastly by salvaging the original MDS matrices. An easy calculation for the direct exponent of MDS matrices based on a lookup table is presented [13]. The MixColumns Transformation in AES is replaced by used Bit Permutation since bit permutation is easy to implement and it does not have any complex mathematical compution [14]. In this study, the AES security is enhanced by two approaches. In the first approach, the fixed XOR is replaced with a dual dynamic XOR; one is used for odd rounds, whereas the other is used for even rounds. In the second approach, the fixed MDS matrix of the MixColumns transformation is replaced with a dynamic MDS matrix for all rounds. In both approaches, 3D Chebyshev is used to generate secret keys.

The remainder of the paper is organized as follows: In the second section, the properties of the chaotic map are explained. In the third section, the proposed algorithm is detailed. In the fourth section, the security of the proposed algorithm is analyzed and compared with that of the original AES. Finally, in the fifth section, future works and conclusions are presented.

2. CHAOTIC MAP

In the past two decades, chaos theory has been extensively applied in scientific fields such as mathematics, physics, computer science, and engineering. Cryptography, which is a branch of mathematics and computer science, has attracted considerable attention. The intrinsic features of chaotic maps support the use of such maps in the design of such algorithms. These proprites comprise highly sensitive dependence on initial condition and control parameter, ergodicity, unpredictability, mixing, and random-like behavior. These properties are comparable with the confusion and diffusion assets of Shannon entropy. Therefore, chaotic dynamic is expected to provide a fast and easy way for building cryptography systems. Recently, researchers have studied high-dimensional chaotic systems to enlarge key space and address the issue of weak security; as a result of such intensive study, 2D chaotic maps have been generalized to 3D chaotic maps, such as 3D baker's map, 3D Cat map, 3D logistic map, and 3D Chebyshev, for designing a secure symmetric scheme that enhances cryptosystem security [1, 9, 10, 15-17, 18].

a) 3D CHEBYSHEV

The private keys which are necessary in the encryption are generated by the chebyshev polynomials. The Chebyshev polynomial Fn (x) of the first type, which is a polynomial in x of degree n, is a prototype of a chaotic map and is defined as follows:

$$F2(x) = 2 \times x^2 - 1 \quad (1)$$

$$F3(y) = 4y3 - 3y \quad (2)$$

$$F4(z) = 8z4 - 8z2 + 1 \quad (3)$$

Chebyshev polynominal map $Fp: [-1,1] \rightarrow [-1,1] of\ degree\ p, when\ p > 1$ [9, 10, 19, 20]

3. PROPOSED ALGORITHM

The proposed algorithm is executed as follows:

a) Three starting parameters of the 3D Chebyshev are initialized to obtain the sequences that will be used as secret keys.
b) Secret keys are generated from a 3D Chebyshev map by using (1)–(3).
c) The generated secret keys are converted from the above point to the decimal by using the following equations:
   $x_{i;j} = (x_{i;j} \times 10^{10} \bmod 16)$, (4)
   $y_{i;j} = (y_{i;j} \times 10^{10} \bmod 16)$, (5)
   $z_{i;j} = (z_{i;j} \times 10^{10} \bmod 4)$. (6)
d) The secret keys are stored in three arrays: two 16 × 16 arrays for x and y sequences and 4 × 4 for Z sequences.
e) Steps are taken to ensure that the arrays of the X and Y sequences follow the features reported in a previous paper [1].
f) The array of the X and Y sequences is used for odd and even rounds, respectively.
g) The numbers in the arrays of the Z sequences are ensured to be between 0–3 without any repeat value in each row. These values will represent the new positions of values in the MDS matrix of the

MixColumns transformation.
Figures 2 and 3 present the dynamic XOR tables for the even and odd rounds, respectively.

12	1	13	15	9	11	7	6	8	4	14	5	0	2	10	3
1	0	6	7	14	13	2	3	11	15	10	8	12	5	4	9
13	6	15	4	3	10	1	8	7	12	5	14	9	0	11	2
15	7	4	13	2	9	8	1	6	5	11	10	14	3	12	0
9	14	3	2	12	5	6	13	10	0	8	15	4	7	1	11
11	13	10	9	5	4	12	15	14	3	2	0	6	1	8	7
7	2	1	8	6	12	4	0	3	11	13	9	5	10	15	14
6	3	8	1	13	15	0	7	2	10	9	12	11	4	14	5
8	11	7	6	10	14	3	2	0	9	4	1	13	12	5	15
4	15	12	5	0	3	11	10	9	8	7	6	2	14	13	1
14	10	5	11	8	2	13	9	4	7	1	3	15	6	0	12
5	8	14	10	15	0	9	12	1	6	3	13	7	11	2	4
0	12	9	14	4	6	5	11	13	2	15	7	1	8	3	10
2	5	0	3	7	1	10	4	12	14	6	11	8	15	9	13
10	4	11	12	1	8	15	14	5	13	0	2	3	9	7	6
3	9	2	0	11	7	14	5	15	1	12	4	10	13	6	8

Figure 2. XOR table for even rounds

0	15	6	5	8	3	2	11	4	14	10	7	12	13	9	1
15	1	5	14	10	2	6	12	8	9	4	13	7	11	3	0
6	5	8	11	9	1	0	10	2	4	7	3	13	12	15	14
5	14	11	8	12	0	15	9	3	7	13	2	4	10	1	6
8	10	9	12	5	4	13	14	0	2	1	15	3	6	7	11
3	2	1	0	4	15	9	7	11	6	12	8	10	14	13	5
2	6	0	15	13	9	1	8	7	5	14	12	11	4	10	3
11	12	10	9	14	7	8	5	6	3	2	0	1	15	4	13
4	8	2	3	0	11	7	6	1	13	15	5	14	9	12	10
14	9	4	7	2	6	5	3	13	1	11	10	15	8	0	12
10	4	7	13	1	12	14	2	15	11	0	9	5	3	6	8
7	13	3	2	15	8	12	0	5	10	9	11	6	1	14	4
12	7	13	4	3	10	11	1	14	15	5	6	0	2	8	9
13	11	12	10	6	14	4	15	9	8	3	1	2	0	5	7
9	3	15	1	7	13	10	4	12	0	6	14	8	5	11	2
1	0	14	6	11	5	3	13	10	12	8	4	9	7	2	15

Figure 3. XOR table for odd rounds

As shown in (1) and (2) are conducted to generate dynamic XOR and MDS matrix, respectively. An example is provided below to show the dynamic MDS in an AES round.
Example 1:

Original MDS				Secret keys				Dynamic MDS			
2	3	1	1	3	2	0	1	1	1	2	3
1	2	3	1	2	3	1	0	3	1	2	1
1	1	2	3	1	0	3	2	1	1	3	2
3	1	1	2	1	0	2	3	1	3	1	2

As shown in the above figure, the first row of the original MDS is permuted on the basis of the values of the secret keys. The first row of the secret keys comprises 3, 2, 0, and 1. The fourth value of the original MDS will be the first value in the dynamic MDS. This action is repeated for the third value, then the first value, and finally the second. The same process is performed for the second, third, and fourth rows.

4. EXPERIMENTAL RESULTS

The proposed algorithm is implemented in C++ on 128 different sequences, where every sequence is 10^6 bits. Some criteria of the security analysis, such as the NIST test, are applied by using LINUX. Other criteria are used with MATLAB R2018a (Mathworks) on a computer with Windows 7 64-bit, Intel Core i5-4500U processor, 4 GB RAM, and 1600 MHz CPU clock speed. The following initial values of the 3D Chebyshev maps are used in all experiments:

For the 3D Chebyshev map, $x_0 = 0.234$, $y_0 = -0.398$, and $z_0 = -0.88$.

5. SECURITY ANALYSIS

In this section, the security of the proposed algorithm is analyzed using different criteria and compare with the origmal algorithm.

5.1. Nist Test

Randomness is a necessary property for a suitable encryption algorithm and is one of the important factors used to amount the confusion and diffusion assets of a novel cryptograpy system. The NIST test is

a statistical test suite used to test the randomness of any cryptographic algorithm. This test suite judges whether the outputs of the algorithms under assured test settings show assets that can be deemed to be arbitrarily produced outputs. The NIST test develops a statistical package that comprises of 15 tests. The proposed algorithm generates 128 different sequences with 128 different 16-byte keys, each with 10^6 bits. The P value is calculated in accordance with each corresponding sequence across all the 15 tests. The results are presented in Tables 1 and 2 for the proposed and the original algorithms respectively. The sequences pass all the 15 tests for the proposed algorithm while fail in FFT test for the original algorthim. Therefore, the proposed algorithm has good randomness and better than from the orginal algorithm. The symbol √ in the below tables indicates a passing result while the symbol x in Table 2 indicated a failing result [11, 15, 21-23].

Table 1 Results of NIST test for proposed alagorithm

No	Test	Success sequence$Pvalue$	Proportion successful	Assessment
1	Frequency	0.949602	0.984375	√
2	Block frequency	0.026648	0.9921875	√
3	Accumulative sums (forward)	0.500934	0.984375	√
	Accumulative sums (reverse)	0.082177	0.984375	√
4	Run	0.517442	0.984375	√
5	FFT	0.534146	1	√
6	Non-overlapping template	0.116519	0.9921875	√
7	Overlapping template	0.100508	0.96875	√
8	Universal	0.404758243	0.990234375	√
9	Approximate entropy	0.452799	0.984375	√
10	Long run	0.023812	0.9765625	√
11	Rank	0.299251	0.9921875	√
12	Random excursions	0.554822125	0.993421053	√
13	Random excursions variants	0.466392	0.989473684	√
14	Serial 1	0.000316	1	√
	Serial 2	0.110952	0.984375	√
15	Linear complexity	0.756476	1	√

Table 2 Results of NIST test for original AES

No	Test	Success sequence$Pvalue$	Proportion successful	Assessment
1	Frequency	0.550148	0.9921875	√
2	Block frequency	0.481243	0.9765625	√
3	Accumulative sums (forward)	0.538846	0.9921875	√
	Accumulative sums (reverse)	0.538846	0.9921875	√
4	Run	0.432788352	0.96875	√
5	FFT	0.523526	0.9765625	√
6	Non-overlapping template	0.513129813	1	√
7	Overlapping template	0.148899	0.640625	X
8	Universal	0.500141138	0.991026182	√
9	Approximate entropy	0.557094391	0.9921875	√
10	Long run	0.461856563	1	√
11	Rank	0.568625938	1	√
12	Random excursions	0.315794968	0.9921875	√
13	Random excursions variants	0.317097578	0.99375	√
14	Serial 1	0.687015	0.9921875	√
	Serial 2	0.575824	0.9921875	√
15	Linear complexity	0.483133281	1	√

5.2. Diehard Test

The diehard test, which consists of 12 tests, is used to test the randomness of random number generators (RNGs) and to evaluate how well a pseudo-RNG produces values. The results are categorized in a way that clearly indicates the randomness of cipher texts produced by the tested block ciphers. The diehard test was originally developed by George Marsaglia and published in 1995. This test comprises several standard tests that operate by using the P-value method. The diehard test results are presented in Tables 3 and 4 respectively. From the resuts it can be seen the proposed algorithm passed all tests while the original algorithm fails in overlapping sums test [1, 24, 25].

Table 3. Results of the diehard test for proposed algorithm

Test	P-value	Assessment
Birthdays	0.531082	√
OPERM5	0.294862	√
Rank 31x31	0.342524	√
Rank 32x33	0.641602	√
Rank 6x8	0.425336	√
BITSTREAM	0.999992	√
OPSO	1	√
OQSO	0.884964	√
DNA	0.610058	√
Count-1s-star	0.643509	√
PARKING LOT	0.57112	√
Minimum Distance	0.999753	√
3D SPHERES TEST	0.884159	√
Squeeze	0.741109	√
OVERLAPPING SUMS	0.534791	√
Run	0.698266	√
Craps	0.333134	√

Table 4. Results of the diehard test for orginal AES algorithm

Test	P-value	Assessment
Birthdays	0.742062	√
OPERM5	0.237064	√
Rank 31x31	0.765169	√
Rank 32x33	0.621038	√
Rank 6x8	0.149836	√
BITSTREAM	0.533555	√
OPSO	0.561696	√
OQSO	0.465943	√
DNA	0.552726	√
Count-1s-star	0.542229	√
PARKING LOT	0.274812	√
Minimum Distance	0.509324	√
3D SPHERES TEST	0.670784	√
Squeeze	0.794714	√
OVERLAPPING SUMS	0.08783	x
Run	0.693404	√
Craps	0.267394	√

5.3. Correlation Coefficient

The correlation coefficient is calculated from the correlation between two random variables. It is considered one of the significant features of the security of block ciphers that handle the dependency of individual output bits on input bits. Moreover, it is a measure of how two variables affect each other, i.e., the degree of dependency of two variables on each other. In this study, it is used to measure the dependency between plaintexts and ciphertexts. The correlation values can determine the confusion effect of the block ciphers. Correlation coefficient is a number between (−1) and (1) and is the amount of the unit of linear relationship between two variables. If the correlation coefficient is (1), then the relationship is increasingly linear. If the correlation coefficient is (−1), then the relationship is decreasingly linear. In the case of independent variables, the correlation is 0. The test is run on 128 different sequences of 10^6 bits with 128 different 16-byte keys of both algoritms the proposed and the original algorithms. The average correlation coefficient is 0.008109 and 0.023021 repectively. Hence, the relation between plaintexts and ciphertexts is near zero. Figures 4 and 5 illustrate the correlation between plaintexts and ciphertexts for the proposed and original algorithms respectively. However the correlation between the plaintext and cipher text in the proposed algorithm is weaker than the original algorithm. This means the security of the proposed algorithm is beeter than the original [21, 26-28].

Figure 4. Correlation coefficient of the proposed algorithm

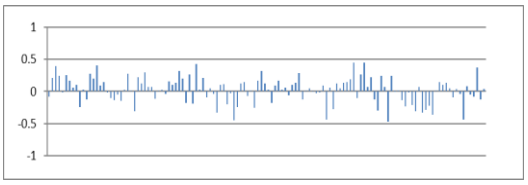

Figure 5. Correlation coefficient of the orginal AES algorithm

5.4. ENTROPY

The idea of entropy comes from information theory and ergodic theory. Shannon entropy is a metric associated with the information content of an input signal. Entropy analysis measures the complexity of encrypted data. The optimum entropy value is 8. Thus, a value that is close to 8 corresponds to the high complexity of encrypted data. The information entropy H(s) of a source s with 2^N symbols si is clear as follows:

$$H(s) = \sum_{i=0}^{2^{N-1}} P(s_i) log_2 s_i, \quad (4)$$

where P(si) means the likelihood of the symbol si being emitted from s. If s is a truly random source, that is, if P(si)= 2^{-N} for all i, then H(s)=$2^N 2^{-N} \log_2 2^N$ = N.

This test is run on the original AES and the proposed algorithm with 128 sequences of 10^6 bits with 128 different random keys. The results are presented in Figures 6 and 7. The average entropy of the original AES and the proposed algorithm are 7.9998299 and 7.9998319, respectively. Therefore, the entropy value of both algorithms is close to 8 [5, 15, 21].

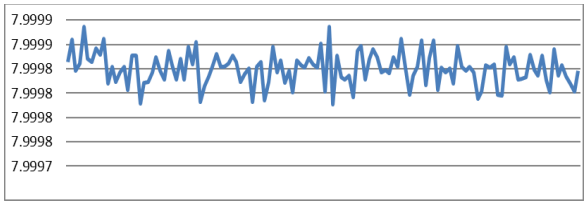

Figure 6. Entropy of the original AES algorithm

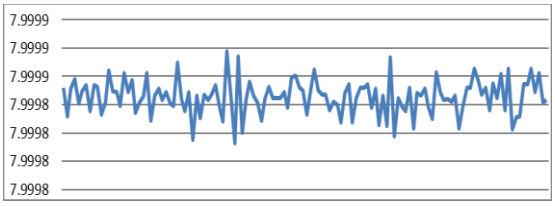

Figure 7. Entropy of the proposed algorithm

5.5. Histogram

Histograms present the number of occurrences (frequency) of every component, i.e., how many times each and every component appears in a sequence, in the form of a graph. The histogram of a text shows the frequency distribution of the characters in a text in graphical formula in a corresponding window. A good RNG should present a uniform histogram with any secret key to avoid detection by opponents, especially statistical attackers. This test is run on the proposed algorithm and the original algorithm on 10^4 with 8-bit number. The results are presented in Figures 8 and 9. The results show that both algorithms have a uniform histogram [21, 29, 30].

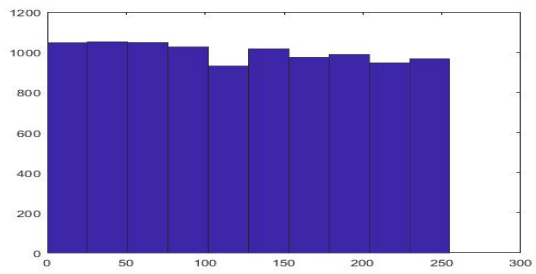

Figure 8. Histogram of the original AES algorithm

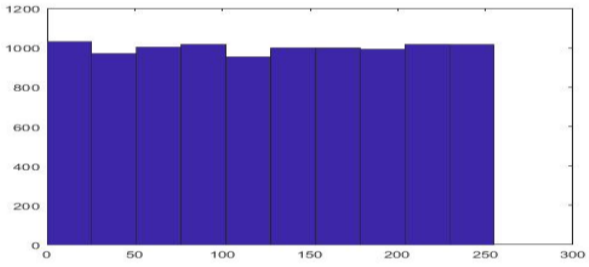

Figure 9. Histogram of the original AES algorithm

6. CONCLUSION AND FUTURE WORKS

According to numerous studies, the dynamic component of block cipher algorithms is more secure than their fixed component. Thus, this study aims to increase the AES security by proposing dual dynamic XOR table and MixColumns transformation based on 3D Chebyshev instead of fixed XOR table and MixColumns transformation. The experiments show that the proposed algorithm can provide high-level security. In future works, this proposed algorithm will be developed for use in bioinformatics.

REFERENCES

[1] A.I. Salih, A. Alabaichi, and A. S. Abbas, "A novel approach for enhancing security of advance encryption standard using private Xor table and 3d chaotic regarding to software quality factor," *ICIC Express Letters Part B: Applications*, vol. 10, no. 9, pp. 823– 832, 2019.

[2] U. R. Atique, *et al*., "A new image encryption scheme based on dynamic s-boxes and chaotic maps," *3D Research*, vol. 7 no. 1, 2016.

[3] M. Nishtha and R. Bansodeb, "AES based text encryption using 12 rounds with dynamic key selection," *7th International Conference on Communication, Computing and Virtualization*, pp. 1036-1043, 2016.

[4] S. Mona, *et al*., "Design of DNA-based advanced encryption standard (AES)," *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS'15)*, pp.390-397, 2015.

[5] M. W. Salim, N. Zainal, "High definition image encryption algorithm based on AES modification," *Wireless Pers Commun*, vol. 79, no. 2, pp. 811–829, 2014.

[6] M. Ghulam, *et al*., "Fortification of AES with dynamic mix-column transformation," *IACR Cryptology ePrint Archive*, pp.4-8, 2011.
[7] A. H. S. Al Wattar, *el al*., "A new DNA based approach of generating key dependent mixcolumns transformation," *International Journal of Computer Networks & Communications (IJCNC*), vol. 7, no.2, pp.93-102, 2015.
[8] A. Alabaichi, and A. I. Salih, "Enhance security of advance encryption standard algorithm based on key-dependent S-box", *2015 Fifth International Conference on Digital Information Processing and Communications* (ICDIPC), pp.44-53, 2015.
[9] A. Alabaichi, "Color image encryption using 3D chaotic map with AES key dependent S-Box", *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 11, pp. 105-115, 2016.
[10] A. Alabaichi, "True color image encryption based on DNA sequence, 3D chaotic map, and key-dependent DNA S-Box of AES," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 2, pp.304-321, 2018.
[11] A. Fatma, and D. Elkamchouchi, "Strongest AES with S-Boxes bank and dynamic key MDS matrix (SDK-AES)," *International Journal of Computer and Communication Engineering*, vol. 2, No. 4, pp. 530-534, 2013.
[12] Luong, Tran Thi, and Nguyen Ngoc C., "The preservation of good cryptographic properties of Mds matrix under direct exponent transformation,", *Journal of Computer Science and Cybernetics*, vol. 31, no. 4, pp.291–303, 2015.
[13] T. L. Tran, N. N. Cuong, and H. D. Tho, "On the calculation of input and output for dynamic MDS matrices in diffusion layer of SPN block ciphers," *KICS-IEEE International Conference on Information and Communications with Samsung LTE & 5G Special Workshop*, pp.281-287, 2017.
[14] H.V. Gamido1, A.M. Sison, R. P. Medina, "Modified AES for text and image encryption," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 3, pp. 942-948, 2018.
[15] A. Kanso and M. Ghebleh, "A novel image encryption algorithm based on a 3D chaotic map," *Commun Nonlinear Sci Numer Simulat*, vol. 17, no. 7, pp. 2943–2959, 2012.
[16] Z. Yikui, S. Lin, and Q. Zhang, "Improving image encryption using multi-chaotic map," *Workshop on Power Electronics and Intelligent Transportation System*, pp.143-148, 2008.
[17] B. Hossain, *et al*., "A new approach of image encryption using 3D chaotic map to enhance security of multimedia component," *3rd International Conference on Informatics, Electronics & Vision (ICIEV)*, pp. 1-6, 2014.
[18] A. Anees and I. Hussain, "A novel method to identify initial values of chaotic maps in cybersecurity," *symmetry*, vol. 11, no. 2, pp. 1-21, 2019.
[19] P. Khade, M. Narnaware, "3D chaotic functions for image encryption," *IJCSI International Journal of Computer Science*, vol. 9, no. 3, pp.323-328, 2012.
[20] L. Kocarev and Z. Tasev, "In Public-key encryption based on Chebyshev maps," *Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS '03)*, 2003.
[21] Ç. Ünal, *et al*., "A novel hybrid encryption algorithm based on chaos and S-AES," *Nonlinear Dynamics*, vol. 92, no. 2, pp.1745–1759, 2018.
[22] A. Alabaichi, *et al*., "Randomness analysis on Blowfish block cipher using ECB and CBC modes," *Journal of Applied Sciences*, vol. 13, no. 6, pp.768-789, 2013.
[23] A. Alabaichi, R. Mahmod, F. Ahmad, "Randomness analysis of 128 bits blowfish block cipher on ECB and CBC modes," *International Journal of Digital Content Technology and its Applications*, vol. 7, no. 15, pp.77-89, 2013.
[24] B. James, "Randomness of D sequences via DieHard testing", *arXiv:1312.3618*, pp.1-8, 2008.
[25] A. Mohammed, "Testing randomness in ciphertext of block-ciphers using DieHard tests," *International Journal of Computer Science and Network Security*, vol. 10, no. 4, pp.53- 57, 2010.
[26] A. Suriyani, *et al*., "An immune system-inspired byte permutation function to improve confusion performance of round transformation in symmetric encryption scheme," *Computer Science and its Applications*, vol. 203, pp. 339-351, 2012.
[27] A. Alabaichi, F. Ahmad, R. Mahmod, "Security analysis of blowfish algorithm," *Second International Conference on Informatics & Applications (ICIA)*, pp.12-18, 2013.
[28] A.S. Saljoughi and H. Mirvaziri, "A new method for image encryption by 3D chaotic map," *Pattern Anal. Appl*., vol. 22, no. 1, pp. 243–257, 2019.
[29] H. A. Abdullah and H. N. Abdullah, "FPGA implementation of color image encryption using a new chaotic map," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 1, pp. 129–137, 2019.
[30] A. Alabaichi, M. A. A. K. Al-Dabbas, and A. Salih, "Image steganography using least significant bit and secret map techniques," *Int. J. Electr. Comput. Eng*., vol. 10, no. 1, pp. 935–946, 2020.