Al-Salam Journal for Engineering and Technology
Journal Homepage: http://journal.alsalam.edu.iq/index.php/ajest
e-IS S N: 2790-4822p-IS SN: 2958-0862

Generation Initial Key of the AES Algorithm based on Randomized and Chaotic System

Rusul Mansoor Al-Amri[1], Dalal N. Hamood[2], Alaa Kadhim Farhan[3]

[1]Computer Department/College of Science University AL-Nahrain, Baghdad, 10001, Iraq
[1]College of Nursing, University of Al-Ameed Karbala, PO No: 198 Iraq

[2]Computer Department/College of Science University AL-Nahrain, Baghdad, 10001, Iraq

[3]Department of Computer Sciences, University of Technology, Baghdad 10011, Iraq

*Corresponding Author: Rusul Mansoor Al-Amri

DOI: https://doi.org/10.55145/ajest.2023.01.01.007
Received June 2022; Accepted September 2022; Available online November 2022

ABSTRACT: Information security is a critical concern and a top priority for many communications networks. Cryptography is a suitable method of data security that can be used to maintain the secrecy and authenticity of information and can improve the privacy aspect of the data. The AES algorithm is one of the most important encryption algorithms and the best in protecting data. In this paper, a comparison was made between two different methods for the generation initial key of the key expansion of the AES-192 algorithm, in the first method, the initial key generated by the user (chose a random key by the user), and in the second method, the initial key generated by the chaotic system (which used the logistic function to generate the numbers chaotically). After testing the two methods, the second method that used a chaotic system is faster and more secure than the first method. Also, the encryption and decryption operations using the key generated by the chaotic system were faster when applying it to various sizes of text files and the plaintext is difficult to discover from the ciphertext. When comparing the proposed method to earlier experiments, it is both quicker and messier than earlier encryption techniques.

Keywords: Initial Key, Chaotic System, Randomize, NIST Tests.

1. INTRODUCTION

People may now execute a variety of tasks more quickly, correctly, and usefully thanks to communication and technology [1], [2]. One of the bad repercussions of technology, which also contains its positive advantages, is that data transmission is increasingly susceptible to trouble tapping. As a result, the security component of information sharing gives top priority. Information security is a critical concern and a top priority for many communications networks [3], [4], and [5]. Many significant and private pieces of information exist in the realm of information that the general public shouldn't be aware of. Cryptography is a suitable method of data security that can be used to maintain the secrecy and authenticity of information and can improve the privacy aspect of data [6], [7], [8]. Many other methods can be utilized to protect the information that is being conveyed. AES is one of the different algorithms that can be employed in cryptography (Advanced Encryption Standard) [9], [10], [11], [12]. The AES algorithm is one of the most important encryption algorithms and the best in protecting data. The AES-192-bit algorithm was used because it is less complex than the AES-256-bit algorithm and faster than the A algorithms [13], [1]. This chaotic system's discovery was regarded as a revolution that sparked a wide range of connected problems, new engineering characteristics, stability theory, and promises to differentiate signatures [15]. Also, there are important properties of the chaos system

exemplary qualities "sensitivity to beginning conditions," "strangeness," "fractal dimensions," and the Lyapunov exponent [16], [17]. This paper proposed two ways to generate the initial key of the AES-192 bit, the first method is to generate a key by the user (chosen randomly) and the second method is to generate the key by the chaotic system. Where the results indicated that the generation of the key by the chaotic system is the most rapid compared to the key generated by the user, and as if the chaotic key is more confidential because it is generated chaotically every time the key is different random numbers, as well as The messy key, proved its efficiency and speed by comparing with previous research. Related work explanation in section 2 and explanation of the proposed method in section 3, section 4 the results, section 5 conclusion, and section 6 deal with the source.

2. Related Work

Vatchara Saicheur and Krerk Piromsopa [18] the implementation of the (AES) algorithm for the Apple iPhone7 is suggested in this study. Their Researchers extend the default AES-128 algorithm to handle the 512-bit block size (AES-512). AES-512 performs better than AES-128 according to their implementation. Depending on the size of the key, the acceleration ranges from 1.20 to 1.58. A 128-bit key is the quickest size. They conclude that increasing the block size of the AES algorithm to 512 bits can increase throughput and accelerate it. Researchers assess the quality of this research by comparing it to AES-192 bit Key. This method will be symbolized in our research (AI AES OAMP) to be compared with a time (AES -192 bit) value mentioned in it.

Ria Andriani, Stevi Ema Wijayanti, and Ferry Wahyu Wibowo [19] To determine how much time and CPU power is required to encrypt and decrypt a file, this search will compare the AES algorithm from AES-128 bit, AES-192 bit, and AES-256 bit. The study's findings demonstrate that AES 128-bit processes information relatively more quickly than AES 192-bit and AES 256-bit. When compared to the other two AES algorithms, AES 192-bit uses the least amount of CPU power during the encryption and decryption of files. This method will be symbolized in our research (CO AES AFEDF) to be compared with a time (AES -192 bit) value mentioned in it.

Ibrahim Yasser, Mohamed A. Mohamed, Ahmed S. Samra, and Fahmi Khalifa [20], This work suggest brand-new 2D alteration models-based chaotic-based multimedia encryption techniques for highly secure data transfer. Perturbation For both bewilderment and propagation rounds, brand-new perturbation-based data encryption is they have a hybrid chaos structure where many maps are integrated for media encryption. The control parameters for the permutation (shuffle) and diffusion (substitution) structures are generated by blended chaotic maps. The proposed techniques are effective for secure multimedia transmission, and the encrypted media has resistance to attacks, according to extensive security and differential assessments in this search. The comparative performance findings showed that their techniques are more effective than their counterpart methods that are data-specific presented.

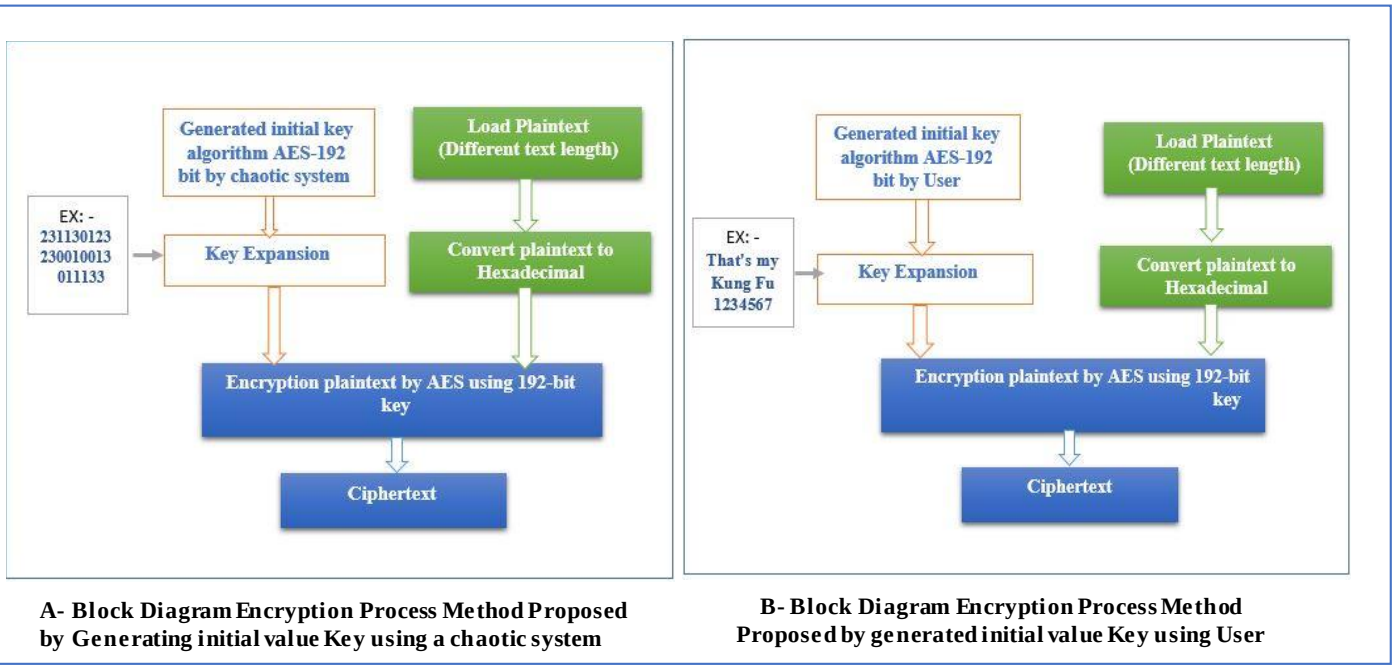
Mahdi Shariatzadeh, Mohammad Javad Rostami, and Mahdi Eftekhari [21] The Advanced Encryption Standard and the logistic chaotic map are the foundations of the new approach for image encryption proposed in this article, dubbed Dynamic AES (AES). The algorithm key is created and combined with the data needed for encryption at different stages using the logistic map. The computational capacity at Galois field 28 is utilized to encode images using a particular type of Advanced Encryption Standard (AES). Investigations and trials have produced data that demonstrate the proposed algorithm's superiority to other image encryption techniques against statistical and differential attacks. The suggested approach works as intended, as evidenced by the ideal NPCR value, nearly-ideal UACI and Entropy values, as well as the histogram and correlation studies of the nearby pixels

Chih-Hsueh Lin, Guo-Hsin Hu, Che-Yu Chan, and Jun-Juh Yan [22] in this study, synchronized dynamic keys based on chaos were designed, and the suggested synchronized random keys were used to create an enhanced chaos-based advanced encryption standard (AES) algorithm. Sliding mode control (SMC) technology was used, and a novel modified AES cryptosystem was presented with dynamic random keys based on chaos synchronization. The static key in the proposed design in this search is made dynamic and random by incorporating the synchronization technology of chaotic systems, and it is not necessary to store or broadcast it across open channels. Finally, a novel image encryption algorithm has been created using the proposed chaos-based AES (CAES) algorithm. To show the effectiveness and advancement of the proposed CAES cryptosystem, simulation experiments have been used to generate and assess the statistical analysis, histogram, information entropy, and correlation indices.

3. Method Research

This paper proposed two ways to generate the initial key of the AES-192 bit, the first method is to generate a key by the user (chosen randomly) and the second method is to generate the key by the chaotic system. Various texts of various lengths were encrypted by two methods and comparison results. The first method encrypts the texts by using the AES-192 bit algorithm with the key chosen randomly by the user (initial key), and the second method, encrypts the texts by using the AES-192 bit algorithm with the key generated by a chaotic system (by the chaotic logistic function). The method mechanism is described in the following block diagrams. Figure (1) illustrated the block diagram of the

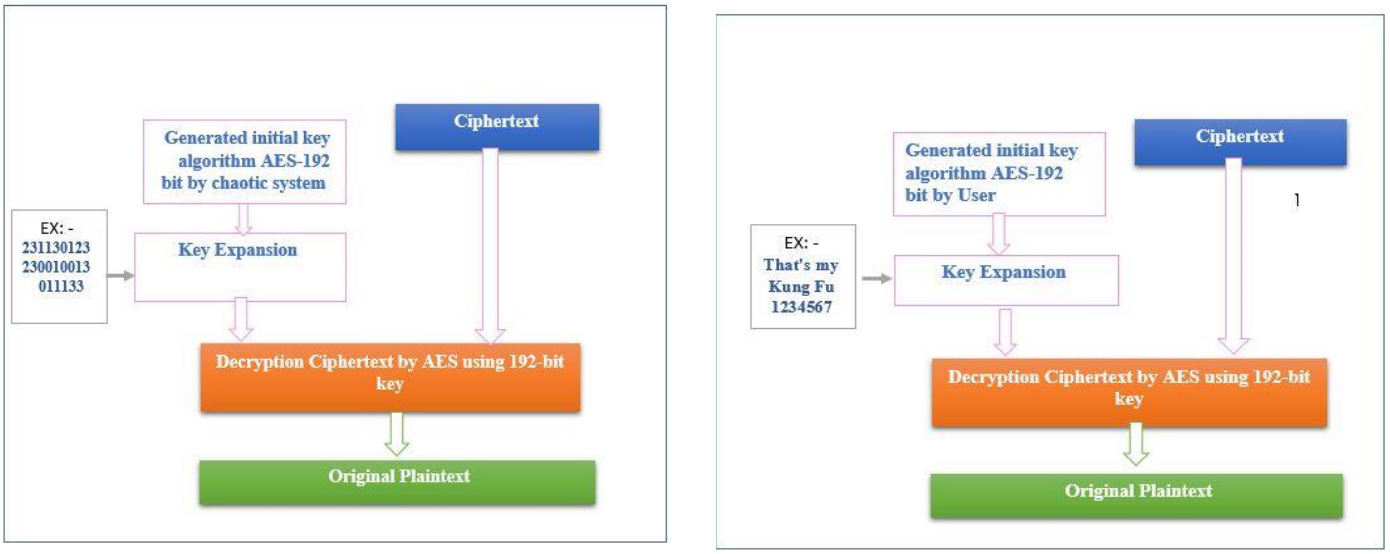
encryption process of the proposed methods and figure (2) illustrated the block diagram of the decryption process of the proposed methods.



A- Block Diagram Encryption Process Method Proposed by Generating initial value Key using a chaotic system

B- Block Diagram Encryption Process Method Proposed by generated initial value Key using User

FIGURE 1. - Block Diagram of Encryption Process of the proposed Methods



A- **Block Diagram Decryption Process Method Proposed by generated initial value Key using chaotic system**

B- **Block Diagram Decryption Process Method Proposed by generated initial value Key using User**

FIGURE 2. - Block Diagram of Decryption Process of the proposed Methods

3.1 Generate Initial Key by User

The AES algorithm was built to operate with a 192-bit key, which was generated throughout 12 cycles. After choosing the plaintext, the user generates a key:

1- Open the file containing the key.

2. The key must contain 24 symbols, for instance: That's my Kung Fu 1234567

3- Must convert the text key (That's my Kung Fu 1234567) to hexadecimal, as shown below.

[- 54 - 68 - 61 - 74 - 73 - 20 - 6D - 79 - 20 - 4B - 75 - 6E - 67 - 20 - 46 - 75 - 20 - 31 - 32 - 33 - 34 - 35 - 36 – 37]

4- Make a matrix for the key is size 4*6

$$\text{KEY 0} = \begin{matrix} \text{W1 W2 W3 W4 W5 W6} \\ \begin{bmatrix} \text{- 54 - 68 - 61 - 74 - 73 - 20 -} \\ \text{- 6D - 79 - 20 - 4B - 75 - 6E -} \\ \text{- 67 - 20 - 46 - 75 - 20 - 31 -} \\ \text{- 32 - 33 - 34 - 35 - 36 – 37 -} \end{bmatrix} \end{matrix}$$

The keys are calculated using the following Equation (1) for W0 from Key Expansion [23][24], (From Key 0 generated Key 1, from Key 1 generated Key 2, and so on until arriving at Key 12).

K[n]: W0 = K [n-1]: W0 XOR SubByte (K [n-1]: W3 >>8) XOR Rcon [i] ---------- (1)

Where,

SubByte = S-Box Lookup Table for Encryption Process for AES

W3>>8 = Shift for row W3 by an amount equal to 8 bits

Recon [i] = Rcon (It is a table with the special values that may be retrieved and replaced for each value.)

I= required key sequence value (example, if K=2 -→ Rcon [2])

Rcon [2] = 02 00 00 00

Either to W1, W2 …Wn Applied Equation (2) for Key Expansion

K[n]: W[i] = K [n-1]: W[i] XOR K[n]: W [i-1] ---------- (2)

Figure (3) illustrated 12 keys generated from an initial key (chosen randomly)

+ Key [0]: 54 - 68 - 61 - 74 - 73 - 20 - 6D - 79 - 20 - 4B - 75 - 6E - 67 - 20 - 46 - 75 -
+ Key [1] : 20 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - C3 - 6D - FB - 6C - B0 - 4D - 96 - 15 -
+ Key [2]: 90 - 6 - E3 - 7B - F7 - 26 - A5 - E - D7 - 17 - 97 - 3D - E3 - 22 - A1 - A -
+ Key [3]: 52 - 5F - 9C - 7D - E2 - 12 - A - 68 - 72 - 14 - E9 - 13 - 85 - 32 - 4C - 1D -
+ Key [4]: 52 - 25 - DB - 20 - B1 - 7 - 7A - 2A - 93 - 85 - 79 - B5 - 71 - 97 - 73 - DD -
+ Key [5]: 3 - 83 - 9A - CE - 86 - B1 - D6 - D3 - D4 - 94 - D - F3 - 65 - 93 - 77 - D9 -
+ Key [6]: 47 - 70 - 4C - F8 - 36 - E7 - 3F - 25 - 35 - 64 - A5 - EB - B3 - D5 - 73 - 38 -
+ Key [7]: 67 - 41 - 7E - CB - 2 - D2 - 9 - 12 - E2 - 71 - 85 - 8F - D4 - 96 - BA - AA -
+ Key [8]: E1 - F2 - 1F - 41 - 52 - 27 - 6C - 79 - 35 - 66 - 12 - B2 - 37 - B4 - 1B - A0 -
+ Key [9] : 4F - DE - 65 - 15 - 9B - 48 - DF - BF - 7A - BA - C0 - FE - 28 - 9D - AC - 87 -
+ Key [10]: 1D - FB - BE - 35 - 2A - 4F - A5 - 95 - 8B - D8 - 4F - F0 - 10 - 90 - 90 - 4F -
+ Key [11]: 6A - 2A - 50 - B1 - 42 - B7 - FC - 36 - 5F - 4C - 42 - 3 - 75 - 3 - E7 - 96 -

FIGURE 3. - Key Expansion from Initial Key Chose Randomly

3.2 Generate Initial Key by Chaotic System

A chaotic system was used to generate the initial key, selecting a different 24-digit number at random each time. The chaotic system was chosen for key generation to provide high protection, strength, and speed in generated key. There have been developed a series of chaotic values referred to as TOA (Time of Arrival) values when modern approaches were utilized in a chaotic sampling of data rather than random samples (chaos series). Non-linear chaotic systems have unstable structures, and occasionally display random behavior, and their output is influenced by both the

initial condition and the control parameter. To create 24 numbers, that represented the initial key of the AES-192 bit, used the chaotic logistics function, as shown in equation (3) [25] [26].

$$\left(X_{N+1}\right) = \lambda X_N(1 + X_N) \quad \text{(3)}$$

Where, N = 0, 1, 2 ……..... λ = control Parameter

When choosing the initial value of the chaotic logistic function obtained the initial key. Table 1 explains obtaining a different initial key when using different initial values. The initial value range is set between 0 and 1.

Table 1. - Various series with different initial value was produced by the chaotic system.

No	Initial generated value	Initial generated key (series)
1	0.35	131101301220123120013133
2	0.91	312322001223013021000123
3	0.25	123122301302300000122231
4	0.69	231130123230010013011133

The range of the generated integers was also established by specifying the lowest and maximum values, allowing the generated key to be restricted to values between 0 and 10, 1 and 5, or 2 and 40. In this paper, the range from 0 to 3 was established as shown in table 1 above. After choosing plaintext and generating an initial key by the chaotic system, one must create key0, that by performing the following steps:

1-The key must contain 24 digits, for instance: 123122301302300000122230

2- Must convert the key (123122301302300000122230) to hexadecimal, as shown below.

[- 31 - 32 - 33 - 31 - 32 - 32 - 33 - 30 - 31 - 33 - 30 - 32 - 33 - 30 - 30 - 30 - 30 - 30 - 31 - 32 - 32 - 32 - 33 – 30]

3- Make the matrix for the key size 4*6, as shown below.

Matrix for key is size 4*6

KEY 0 =

$$\begin{bmatrix} 31 & 32 & 33 & 31 & 32 & 32 \\ 33 & 30 & 31 & 33 & 30 & 32 \\ 33 & 30 & 30 & 30 & 30 & 30 \\ 31 & 32 & 32 & 32 & 33 & 30 \end{bmatrix}$$

The keys are calculated using the following Equation (1), explain in the previous section. Figure (4) illustrated 12 keys generated from an initial key (chaotic system).

```
+ Key[0]:  31 - 32 - 33 - 31 - 32 - 32 - 33 - 30 - 31 - 33 - 30 - 32 - 33 - 30 - 30 - 30 -
+ Key[1] :  30 - 30 - 31 - 32 - 32 - 32 - 33 - 30 - 13 - F1 - 37 - 12 - 21 - C3 - 4 - 22 -
+ Key[2] :  10 - F0 - 34 - 10 - 23 - C0 - 4 - 20 - 13 - F0 - 35 - 12 - 21 - C2 - 6 - 22 -
+ Key[3] :  34 - 9E - A4 - EF - 15 - 5D - A0 - CD - 5 - AD - 94 - DD - 26 - 6D - 90 - FD -
+ Key[4] :  35 - 9D - A5 - EF - 14 - 5F - A3 - CD - FF - 94 - 19 - 15 - EA - C9 - B9 - D8 -
+ Key[5] :  EF - 64 - 2D - 5 - C9 - 9 - BD - F8 - FC - 94 - 18 - 17 - E8 - CB - BB - DA -
+ Key[6] :  E8 - 7E - 4E - 8E - 2 - B7 - F7 - 56 - ED - D3 - DA - 53 - 24 - DA - 67 - AB -
+ Key[7] :  D8 - 4E - 7F - BC - 30 - 85 - C4 - 66 - 6F - 62 - 7D - 8A - 6D - D5 - 8A - DC -
+ Key[8] :  80 - 6 - 50 - 8F - A4 - DC - 37 - 24 - 7C - 92 - 48 - 98 - 4C - 17 - 8C - FE -
+ Key[9] :  BF - 6 - C6 - A3 - D2 - D3 - 4C - 7F - 52 - D5 - 1C - F0 - F6 - 9 - 2B - D4 -
+ Key[10] :  8A - 9B - 63 - 4C - C6 - 8C - EF - B2 - 9B - D9 - F1 - 17 - 49 - A - BD - 68 -
+ Key[11] :  63 - B6 - C6 - 25 - 2A - BC - 7B - 4D - 31 - 63 - DA - D5 - DC - B5 - 50 - 99 -
```

FIGURE 4. - Key Expansion from Initial Key of chaoticsystems

Figure (3) and figure (4) above generated 12 keys each one used in a specified round of the encryption (or decryption) process. STATE matrix converts to Binary code where each number in hexadecimal converts to Binary After converting all the numbers in each matrix.

4. Result and Discuss

In evaluating the proposed system, test a variety of keys created by the user and keys generated by chaotic systems that represented the initial key of AES. For testing the proposed algorithm, suppose ten plaintexts with different lengths for encryption and decryption using proposed keys created by the user and proposed keys generated by chaotic systems of AES - 192-bit Algorithm.

4.1 Generation Time of Keys created by the user

Testing 10 user-generated keys and calculating generation time for each key. Generation Time is measured by Picosecond, as shown in Table 2.

Table 2. - Generation Time of Keys created by User

NO	Keys	Keys generation time (Picosecond)
1	Thats my Kung Fu 1234567	00:00:04.1002682
2	Iraq my Country 19958999	00:00:02.6023839
3	University AL-Nahrain 22	00:00:03.9468202
4	Cryptography Text Part 1	00:00:03.4665153
5	Encryption AES -192 bits	00:00:02.5914488
6	Bagdad Center Iraq @8001	00:00:03.1512021
7	Computer Science College	00:00:02.4815038
8	Number Student Perfect =9	00:00:02.4266818
9	AES - 192 Algorithmgood	00:00:03.1413028
10	AESDecryption CipherText	00:00:03.8201124

From the above result in table 2, the outcome is shown Key generation time is measured in fractions of a second at a time. This process has a short time. Suppose different 10 text file sizes were also examined, and the AES -192 bit technique, which required the user to generate its key, was used to encrypt and decode the text files
Table 3 illustrated the encryption time and decryption time for ten plaintexts with different sizes.

Table 3. - Illustrate Length Text, Encryption, and Decryption Time for AES -192 bit with Key Generated by the user

No	Name text File (.txt)	Length Text file (Block)	Encryption Time (Picosecond)	Decryption Time (Picosecond)
1	TEXT 1 .txt	8B	00:00:00.0187602	00:00:00.0052835
2	TEXT 2 .txt	16B	00:00:00.0256724	00:00:00.0052297
3	TEXT 3 .txt	24B	00:00:00.0327285	00:00:00.0052315
4	TEXT 4 .txt	30B	00:00:00.0372706	00:00:00.0055239
5	TEXT 5 .txt	36B	00:00:00.0439545	00:00:00.0052105
6	TEXT 6 .txt	48B	00:00:00.0563520	00:00:00.0060620
7	TEXT 7 .txt	55B	00:00:00.0605771	00:00:00.0052143
8	TEXT 8 .txt	60B	00:00:00.0644492	00:00:00.0051866
9	TEXT 9 .txt	64B	00:00:00.0699278	00:00:00.0058820
10	TEXT 10 .txt	80B	00:00:00.0845763	00:00:00.0052944

The above result in table 3, shows that the encryption and decryption processes occur quickly in fractions of a second and that the generation of 24 elements consists of symbols, letters, and numbers from the key, making them more complex and achieving a high level of confidentiality for the encrypted data. This is one of the quality indicators of the performance of the algorithm, as the more complex the algorithm key, the more difficult the algorithm is to break by intruders. Table 4 illustrates the encryption process (12 rounds treated the key matrix and state matrix) example of the state matrix: - Iraq Great.

Table 4. - Ciphertext Generated (12 Round) by User initial key

KEYs	State	Encryption Rounds
54 - 68 - 61 - 74 - 73 - 20 - 6D - 79 - 20 - 4B - 75 - 6E - 67 - 20 - 46 - 75 -	- 49 - 72 - 61 - 71 - 20 - 47 - 72 - 65 - 61 - 74 - 20 - 20 - 20 - 20 - 20 - 20 - 20	Round 1 = E7 32 84 F9 67 0C 11 C6 EF 75 62 7E 5E 8B D1 D2
20 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - C3 - 6D - FB - 6C - B0 - 4D - 96 - 15 -	E7 - 32 - 84 - F9 - 67 - 0C - 11 - C6 - EF - 75 - 62 - 7E - 5E - 8B - D1 - D2 -	Round 2 = A5 87 4C FD 25 4C 7C 28 02 11 9E DE F9 AA CD DE
90 - 6 - E3 - 7B - F7 - 26 - A5 - E - D7 - 17 - 97 - 3D - E3 - 22 - A1 - A -	A5 - 87 - 4C - FD - 25 - 4C - 7C - 28 - 02 - 11 - 9E - DE - F9 - AA - CD - DE -	Round 3= 33 78 A7 53 0B 9B 60 38 82 87 F5E46F 75 683D
52 - 5F - 9C - 7D - E2 - 12 - A - 68 - 72 - 14 - E9 - 13 - 85 - 32 - 4C - 1D -	33 - 78 - A7 - 53 - 0B - 9B - 60 - 38 - 82 - 87 - F5 - E4 - 6F - 75 - 68 - 3D -	Round 4 = 32 78 AC 9F D8 13 7F 50 B2 27 B8 54 C2 07 64 47
52 - 25 - DB - 20 - B1 - 7 - 7A - 2A - 93 - 85 - 79 - B5 - 71 - 97 - 73 - DD -	32 - 78 - AC - 9F - D8 - 13 - 7F - 50 - B2 - 27 - B8 - 54 - C2 - 07 - 64 - 47	Round 5 = 0E 8D EA E4 4E F7 3F 86 E7 97 F5 8B E1 B4 AE 84
3 - 83 - 9A - CE - 86 - B1 - D6 - D3 - D4 - 94 - D - F3 -	0E - 8D - EA - E4 - 4E - F7 - 3F - 86 - E7 - 97 - F5 - 8B -	Round 6 = 0B D6 F5 7C 65 EF 41 85 B9 7E D9 09 2E 61 A8 F7

65 - 93 - 77 - D9 - 47 - 70 - 4C - F8 - 36 - E7 - 3F - 25 - 35 - 64 - A5 - EB - B3 - D5 - 73 - 38 -	E1 - B4 - AE - 84 - 0B - D6 - F5 - 7C - 65 - EF - 41 - 85 - B9 - 7E - D9 - 09 - 2E - 61 - A8 - F7 -	Round 7 = 16 5B 2E 12 F8 0B B7 D0 58 2D 72 47 8C F3BA BB
67 - 41 - 7E - CB - 2 - D2 - 9 - 12 - E2 - 71 - 85 - 8F - D4 - 96 - BA - AA -	16 - 5B - 2E - 12 - F8 - 0B - B7 - D0 - 58 - 2D - 72 - 47 - 8C - F3 - BA - BB -	Round 8 = B8 04 A0 51 C9 C8 04 78 D6 0B 90 3F 2C F3 E1 8A
E1 - F2 - 1F - 41 - 52 - 27 - 6C - 79 - 35 - 66 - 12 - B2 - 37 - B4 - 1B - A0 -	B8 - 04 - A0 - 51 - *C9 - C8 - 04 - 78* - D6 - 0B - 90 - 3F - 2C - F3 - E1 - 8A -	Round 9 = AA 6D 85 3E A7 AB 35 68 A3 A6 7C E5 D5 D7 95 F2
4F - DE - 65 - 15 - 9B - 48 - DF - BF - 7A - BA - C0 - FE - 28 - 9D - AC - 87 -	AA - 6D - 85 - 3E - A7 - AB - 35 - 68 - A3 - A6 - 7C - E5 - D5 - D7 - 95 - F2 -	Round 10 = 61 29 75 85 2A 37 D5 DC D0 A3 9B 48 A1 98 01 AE
1D - FB - BE - 35 - 2A - 4F - A5 - 95 - 8B - D8 - 4F - F0 - 10 - 90 - 90 - 4F -	61 - 29 - 75 - 85 - 2A - 37 - D5 - DC - D0 - A3 - 9B - 48 - A1 - 98 - 01 - AE -	Round 11 = EA 76 D9 02 32 D0 10 27 3A A8 73 05 C6 F7 C5 23
6A - 2A - 50 - B1 - 42 - B7 - FC - 36 - 5F - 4C - 42 - 3 - 75 - 3 - E7 - 96 -	EA - 76 - D9 - 02 - 32 - D0 - 10 - 27 - 3A - A8 - 73 - 05 - C6 - F7 - C5 - 23 -	Round 12 = F7 58 3F 3F 3C 16 3A 62 50 24 9F 21 4B 96 FB 03

From the result in the above table, the ciphertext generated (12 Rounds) by the initial key-using user. Ciphertext **of Round 12 = F7 58 3F 3F 3C 16 3A 62 50 24 9F 21 4B 96 FB 03**

4.2 Generation Time of Keys generated by Chaotic System

Testing 10 chaotic function-generated keys and calculating generation time for each key. The generation time is measured by Picosecond, as shown in Table 5.

Table 5. - Generation Time of Keys created by a chaotic system

NO	Keys	Keys generation time (Picosecond)
1	2**3000013012201231201000 3**	00:00:00.0003683
2	2223**00130122012310000 3**	00:00:00.0002395
3	1223**10130121003112**30021	00:00:00.0002470
4	2313**13130122003112**30012	00:00:00.0002419
5	3122**1**3012**1**023**00331**232222	00:00:00.0002343
6	13**013112**21023**11131031200**	00:00:00.0002328
7	2223**1**0012**1**02**20**12**31**230123	00:00:00.0002442
8	2223**1**0012**1**02**20**12**31**230123	00:00:00.0002542
9	12313121312221002333 0131	00:00:00.0002358
10	13001313230031312222 3122	00:00:00.0002323

From the result in table 5, the process of creating random keys using integer numbers. Also observe that calculating the time for generating keys is quick, indicating that the algorithm's time for generating keys using the chaotic function is very speedy. This is one of the most significant excellent indicators of the algorithm's performance. Table 6 illustrated the encryption time and decryption time for ten plaintexts with different sizes.

Table 6.- Illustrate Length Text, Encryption, and Decryption Time for AES -192 bit with Key Generated by Chaotic function

No	Name text File (.txt)	Length Text file (Block)	Encryption Time text file (Picosecond)	Decryption Time text file (Picosecond)
1	TEXT 1 .txt	8B	00:00:00.0078260	00:00:00.0007443
2	TEXT 2 .txt	16B	00:00:00.0154350	00:00:00.0005584
3	TEXT 3 .txt	24B	00:00:00.0223596	00:00:00.0006928
4	TEXT 4 .txt	30B	00:00:00.0290594	00:00:00.0007015
5	TEXT 5 .txt	36B	00:00:00.0340131	00:00:00.0007222
6	TEXT 6 .txt	48B	00:00:00.0448685	00:00:00.0007923
7	TEXT 7 .txt	55B	00:00:00.0534529	00:00:00.0008328
8	TEXT 8 .txt	60B	00:00:00.0627826	00:00:00.0008918
9	TEXT 9 .txt	64B	00:00:00.0644645	00:00:00.0009349
10	TEXT 10 .txt	80B	00:00:00.0763015	00:00:00.0009936

The above result in table 6, shows that the encryption and decryption processes occur quickly in fractions of a second and that the generation of 24 integers numbers them more complex and achieves a high level of confidentiality for the encrypted data because difficult prediction generated random Numbers. Table 7 illustrates the encryption process (12 rounds treated the key matrix and state matrix) example of the state matrix: - Iraq Great.

Table 7.- Ciphertext Generated (12 Round) by chaotic initial key

KEYs	State	Encryption Rounds
31 - 32 - 33 - 31 - 32 - 32 - 33 - 30 - 31 - 33 - 30 - 32 - 33 - 30 - 30 - 30	- 49 - 72 - 61 - 71 - 20 - 47 - 72 - 65 - 61 - 74 - 20 - 20 - 20 - 20 - 20 - 20	Round 1 = EF 83 0C 8A 22 EC D1 FB DA CE B1 2D 35 08 CEA6
30 - 30 - 31 - 32 - 32 - 32 - 33 - 30 - 13 - F1 - 37 - 12 - 21 - C3 - 4 - 22 -	EF - 83 - 0C - 8A - 22 - EC - D1 - FB - DA - CE - B1 - 2D - 35 - 08 - CE - A6 -	Round 2 = 10 EC 82 73 CF B5 D9 8C **C2 DC 3A 60 24 DE 33 B5**
10 - F0 - 34 - 10 - 23 - C0 - 4 - 20 - 13 - F0 - 35 - 12 - 21 - C2 - 6 - 22 -	10 - EC - 82 - 73 - CF - B5 - D9 - 8C - C2 - DC - 3A - 60 - 24 - DE - 33 - B5 -	Round 3= 8A 2B B7 E3 AB A4 04 BF C4 BA 8E DC 4E 0D 93 E6
34 - 9E - A4 - EF - 15 - 5D - A0 - CD - 5 - AD - 94 - DD - 26 - 6D - 90 - FD -	8A - 2B - B7 - E3 - AB - A4 - 04 - BF - C4 - BA - 8E - DC - 4E - 0D - 93 - E6 -	Round 4 = 85 49 65 46 D4 EF DE C7 29 C6 E2 46 3A EF 54 DE
52 - 25 - DB - 20 - B1 - 7 - 7A - 2A - 93 - 85 - 79 - B5 -	85 - 49 - 65 - 46 - D4 - EF - DE - C7 - 29 - C6 - E2 - 46 -	Round 5 = 25 42 4D 10 F8 66 65 E6 69 B4 B8 BA DA E9 06 88

71 - 97 - 73 - DD - 3 - 83 - 9A - CE - 86 - B1 - D6 - D3 - D4 - 94 - D - F3 - 65 - 93 - 77 - D9 -	3A - EF - 54 - DE - 25 - 42 - 4D - 10 - F8 - 66 - 65 - E6 - 69 - B4 - B8 - BA – DA - E9 - 06 - 88 -	
47 - 70 - 4C - F8 - 36 - E7 - 3F - 25 - 35 - 64 - A5 - EB - B3 - D5 - 73 - 38 -	6B - 2E - 5E - 8A - 57 - 51 - 39 - 71 - CD - 9C - 7C - 5C - 03 - 35 - 26 - FC -	Round 6 = 6B 2E 5E 8A 57 51 39 71 CD 9C 7C 5C 03 35 26 FC
67 - 41 - 7E - CB - 2 - D2 - 9 - 12 - E2 - 71 - 85 - 8F - D4 - 96 - BA - AA -	EE - 55 - 0A - 55 - 08 - 35 - 2D - 93 - 3A - FE - 46 - 0A - 87 - 9B - 4B - 2E -	Round 7 = EE 55 0A 55 08 35 2D 93 3A FE 46 0A 87 9B 4B 2E
E1 - F2 - 1F - 41 - 52 - 27 - 6C - 79 - 35 - 66 - 12 - B2 - 37 - B4 - 1B - A0 -	1A - 44 - 59 - F7 - C6 - 35 - D9 - EE - 09 - E4 - D4 - E3 - 59 - AA - 45 - 17 -	Round 8 = 1A 44 59 F7 C6 35 D9 EE 09 E4 D4E3 59 AA 45 17
4F - DE - 65 - 15 - 9B - 48 - DF - BF - 7A - BA - C0 - FE - 28 - 9D - AC - 87 -	F9 - 61 - 0A - E6 - BB - 5A - E1 - C7 - 69 - 06 - 12 - 97 - 7B - FB - DC - 6D -	Round 9 = F9 61 0A E6 BB 5AE1 C7 69 06 12 97 7B FB DC 6D
1D - FB - BE - 35 - 2A - 4F - A5 - 95 - 8B - D8 - 4F - F0 - 10 - 90 - 90 - 4F -	8F - B9 - F2 - 82 - 19 - 2A - 36 - 7C - 89 - 90 - A8 - 96 - F3 - AA - 9F - 73 -	Round 10 = 84 D3 55 5B AF 2C BA 1A 22 F5 CA 77 2F 39 74 9E
6A - 2A - 50 - B1 - 42 - B7 - FC - 36 - 5F - 4C - 42 - 3 - 75 - 3 - E7 - 96 -	84 - D3 - 55 - 5B - AF - 2C - BA - 1A - 22 - F5 - CA - 77 - 2F - 39 - 74 - 9E -	Round 11 = 84 D3 55 5B AF 2C BA 1A 22 F5 CA 77 2F 39 74 9E
		Round 12 = 3C 4C CD E5 C7 48 C1 CC B2 8E 49 B6 2E 58 C7 0B

From the table above this generated Ciphertext (State Round 12) by initial value key using a chaotic system. **Round 12 = cipherText = 3C 4C CD E5 C7 48 C1 CC B2 8E 49 B6 2E 58 C7 0B**

It has been noticed that the ciphertexts at the end are different for each key and this is due to the difference in the key and the way it is generated. Table 8 illustrated a comparison between the generation time of the keys created by the user and keys created by a chaotic function when the key size is equal to 24 symbols, characters, or numbers.

Table 8. - **Comparison between Key generation time by the user and Key generation time by a chaotic system**

No	Keys generation time by a user (Picosecond)	Keys generation time by Chaotic system (Picosecond)
1	00:00:04.1002682	00:00:00.0003683
2	00:00:02.6023839	00:00:00.0002395
3	00:00:03.9468202	00:00:00.0002470
4	00:00:03.4665153	00:00:00.0002419
5	00:00:02.5914488	00:00:00.0002343
6	00:00:03.1512021	00:00:00.0002328

7	00:00:02.4815038	00:00:00.0002442
8	00:00:02.4266818	00:00:00.0002542
9	00:00:03.1413028	00:00:00.0002358
10	00:00:03.8201124	00:00:00.0002323

From the above result in table 8, it can be shown that the chaotic system generates the key more quickly than the user does. Figure 5 explains the comparison of the generation time.

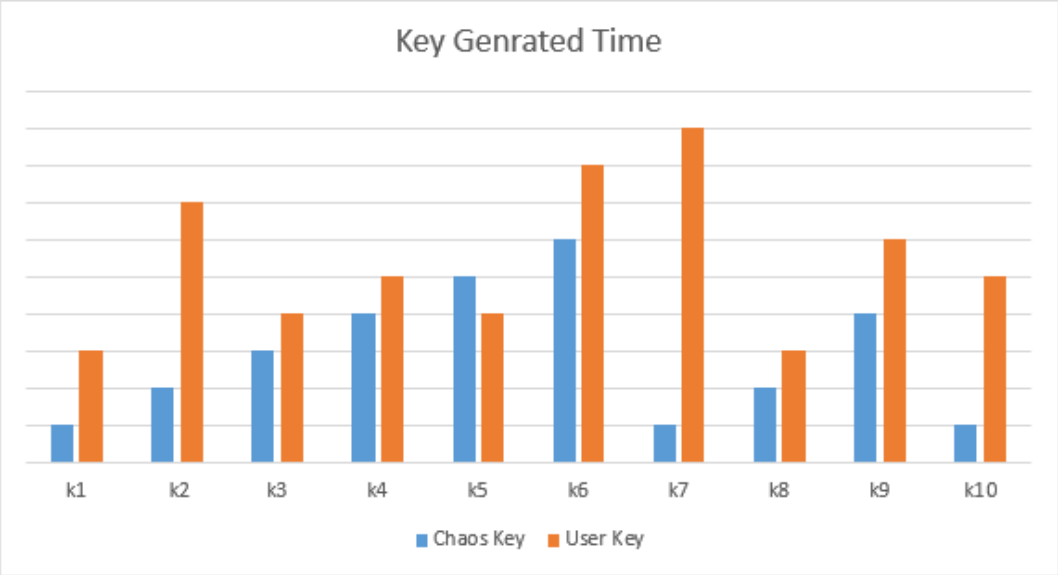


FIGURE 5. - **Compares Generation Time between Chaos Key and User Key**

And this means the calculated time that the encryption and decryption process will be different as shown in table 9.

Table 9. - compares the Encryption and Decryption time between the Key generated by the user and the Key generated by the chaotic system

NO	Length Text file size (Block)	Keys generation by the user		Keys generation by chaotic System	
		Encryption Time (Picosecond)	Decryption Time (Picosecond)	Encryption Time (Picosecond)	Decryption Time (Picosecond)
1	Text1 (8B)	00:00:00.0187602	00:00:00.0052835	00:00:00.0078260	00:00:00.0007443
2	Text2 (16B)	00:00:00.0256724	00:00:00.0052297	00:00:00.0154350	00:00:00.0005584
3	Text3 (24B)	00:00:00.0327285	00:00:00.0052315	00:00:00.0223596	00:00:00.0006928
4	Text4 (30B)	00:00:00.0372706	00:00:00.0055239	00:00:00.0290594	00:00:00.0007015
5	Text5 (36B)	00:00:00.0439545	00:00:00.0052105	00:00:00.0340131	00:00:00.0007222
6	Text6 (48B)	00:00:00.0563520	00:00:00.0060620	00:00:00.0448685	00:00:00.0007923
7	Text7 (55B)	00:00:00.0605771	00:00:00.0052143	00:00:00.0534529	00:00:00.0008328
8	Text8 (60B)	00:00:00.0644492	00:00:00.0051866	00:00:00.0627826	00:00:00.0008918
9	Text9 (64B)	00:00:00.0699278	00:00:00.0058820	00:00:00.0644645	00:00:00.0009349

10	Text10 (80B)	00:00:00.0845763	00:00:00.0052944	00:00:00.0763015	00:00:00.0009936

It can be seen from the comparison process in the table above that the AES -192 bit method, whose key is generated by the chaotic system, performs encryption and decryption more quickly than the algorithm whose key is generated by the user. A chaotic key is preferable since it is faster, which strengthens the algorithm utilized because speed is crucial for encryption techniques when they are transmitted via networks. Additionally, the chaotic process of generating the key generates random numbers, and it is assumed that the recipient has a copy of the program to be able to decrypt it. However, in the case of the key generated by the user, the entered key must be sent to the recipient to be able to decrypt, indicating the strength of protection it is challenging for the intruder. The chaotic system maintains great confidentiality for data transport and information. Figure 6 shows compare the Encryption and Decryption Time of the AES -192 Bit Algorithm with keys Generated by Chaos and keys generated by User Key.

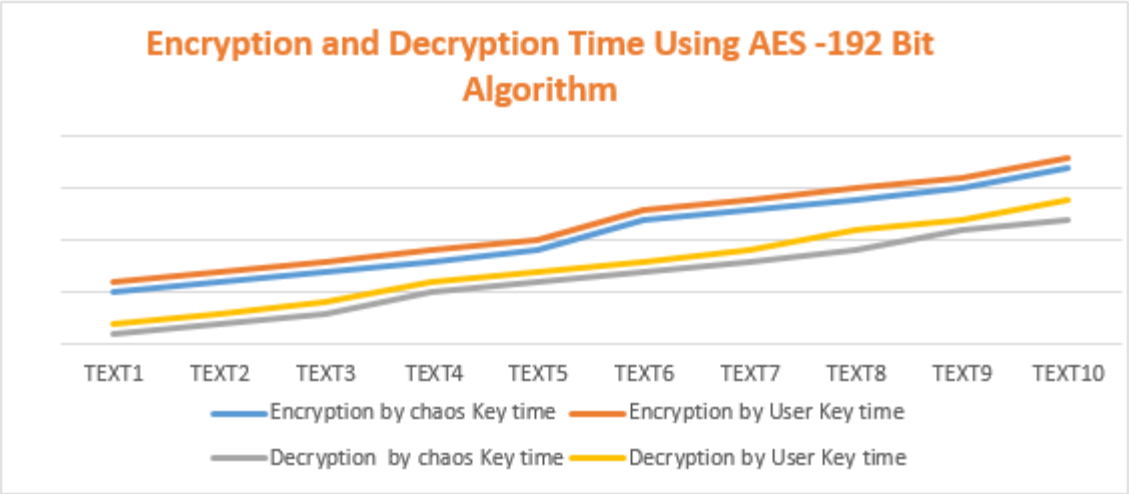


FIGURE 6. - Compare Encryption and Decryption Time Using AES -192 Bit Algorithm Generated by Chaos Key and User Key

Table 10 illustrated a comparison between Ciphertext that is encrypted by an Initial user key and ciphertext that is encrypted by an initial chaotic key by NIST tests.

Table 10. - NIST tests for ciphertext

Test Name	Ciphertext (Initial User Key)	Ciphertext (Initial Chaotic Key)
Frequency	0.8989	0.9889
Block Frequency	0.697	0.7870
Cumulative Sums	0.7896	0.8796
Runs	0.9001	0.9901
Longest Run	0.8204	0.9104
Rank	0.7196	0.8096
FFT	0.8195	0.9090
NonOverlaping Template	0.8896	0.9790
Overlapping Template	0.8536	0.9436
Universal	0.833	0.9230
Approximate Entropy	0.9062	0.9962
Serial	0.824	0.9140
Linear Complexity	0.8359	0.9259
Random Exclusions	0.8527	0.9427
Random Exclusions Varients	0.8989	0.9454

From the above result in table 10, the ciphertext encrypted with keys generated from the initial chaotic system very stronger and safer than the ciphertext encrypted with keys generated from the initial user key, as shown in figure (7).

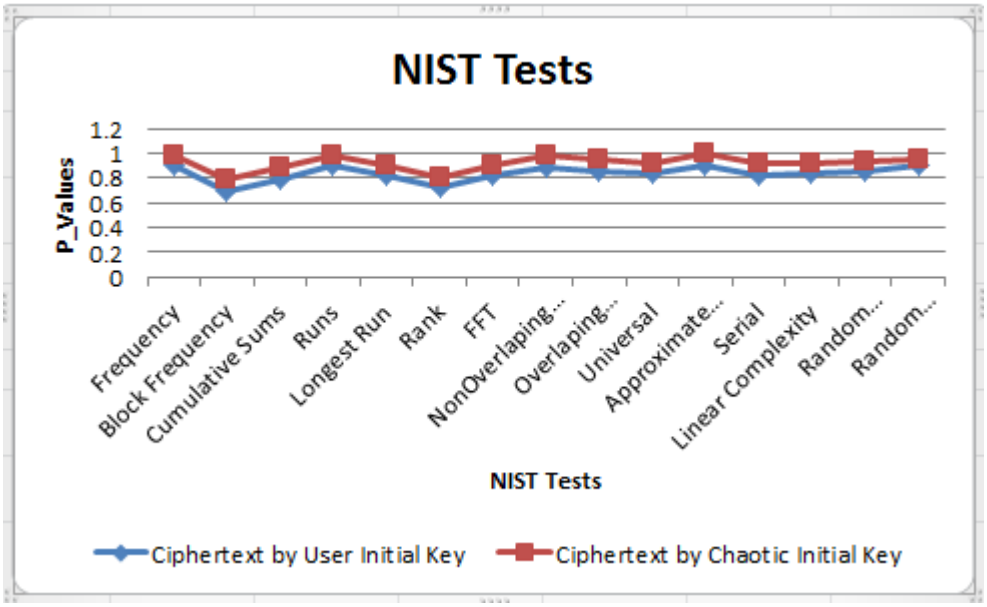


FIGURE 7. - NIST Tests for the ciphertext

Table (11) explain the comparison between the proposed method and the previous methods, there are three measurements for comparison: File size, encryption time, and decryption time.

Table 11. - Comparison between Proposed Method and previous Study

Name Method	Length Text file size (Block)	Time unit	Encryption Time (AES-192 bit)	Decryption Time (AES-192 bit)
The proposed method (Chaos Key)	80B	(Picosecond)	00:00:00.0763015	00:00:00.0052944
The proposed method (User Key)	80B	(Picosecond)	00:00:00.0845763	00:00:00.0052944
CO AES AFEDF	64 KB	(in a second)	1.22	----------
AI AES OAMP	546 kb	(in a second)	1.848	1.499

The proposed work and the earlier studies are juxtaposed in the table above. The file size, encryption, and decoding times were all taken into consideration throughout the comparison. It was discovered that the proposed work was faster than the earlier work, particularly in the approach that used the key produced by the chaotic system. The chaos used to generate the numbers increases the confidentiality of the encryption in addition to high speed. The projected effort produced great outcomes in terms of timing. This was the verdict in contrast.

5. CONCLUSION

The data is encrypted to protect it from illegal access, which is important given the importance of protecting sensitive data and information. As the key generated by the chaotic system was quicker than the key generated by the user and also the results of the suggested approach, which depended on the chaotic system, were superior to those of the previous methods. In the proposed method the time of encryption and decryption to a few fractions of a second. In addition, the encryption quality test and high security were by NIST Tests for the ciphertext, where the results were

excellent. Also, the key generated by the chaotic system produced a very strong and very safe ciphertext. The recommended paper showed that the speed of encryption is significantly influenced by how the encryption algorithm key is formed. The key generated by the chaotic system is used for live broadcasts and other applications because of its excellent data protection, difficulty in forecasting chaotic formation, difficulty in discovering plaintext from the ciphertext, and cost-effectiveness in terms of time.

ACKNOWLEDGEMENT

The authors would like to thank Al-Nahrain University, the University of Technology, the University of Al – Ameed, and the Iraqi Ministry of Education for their support to conduct the work published in this paper.

CONFLICTS OF INTEREST

The authors declare no conflict of interest

REFERENCES

[1] Weijia Xue, Congli Wang, and Jinhua Wang, "Research on Cryptography as a Service Technique Based on Commercial Cryptography", IEEE, 25 July 2022, DOI:10.1109/ICETCI55101.2022.9832226.

[2] Geethanjali G, C Ashwin, Bharath V P, Avinash A, and Anurag Hiremath, "Enhanced Data Encryption in IOT using ECC Cryptography and LSB Steganography", IEEE,27 September 2021, DOI: 10.1109/ICDI3C53598.2021.00043.

[3] Fangfang Dang, Huichao Liang, Shuai Li, Dingding Li, and Han Liu, "Design and Implementation of Computer Network Information Security Protection Based on Secure Big Data", IEEE, 01 February 2021, DOI: 10.1109/IICSPI51290.2020.9332352.

[4] Wenjie Jia and Tao Jiang, "Information-defined networks: A communication network approach for network studies", IEEE, 26 July 2021, DOI: 10.23919/JCC.2021.07.016.

[5] Weijia Xue, Congli Wang, Jinhua Wang, "Research on Cryptography as a Service Technique Based on Commercial Cryptography", IEEE, 2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI), 25 July 2022, DOI: 10.1109/ICETCI55101.2022.9832226.

[6] Anjula Gupta and Navpreet Kaur Walia, "Cryptography Algorithms: A Review", International Journal of Engineering Development and Research, 2014 IJEDR | Volume 2, Issue 2 | ISSN: 2321-9939.

[7] Omar G. Abood and Shawkat K. Guirguis, "A Survey on Cryptography Algorithms", International Journal of Scientific and Research Publications, Volume 8, Issue 7, July 2018 495, ISSN 2250-3153, DOI: 10.29322/IJSRP.8.7.2018.p7978.

[8] Shaymaa Ammar Fadhil, Alaa Kadhim Farhan, and A. Hussien Radie, " Visual Cryptography Techniques: Short Survey ", IEEE, Published in 2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA), 25 February 2022, DOI: 10.1109/IICETA51758.2021.9717352.

[9] Keshav Kumar, K.R. Ramkumar, and Amanpreet Kaur, "A Design Implementation and Comparative Analysis of Advanced Encryption Standard (AES) Algorithm on FPGA", 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 15 September 2020, DOI: 10.1109/ICRITO48877.2020.9198033.

[10] Qingsheng Hu, Xiangning Fan, and Qiaowei Zhang, "An Effective Differential Power Attack Method for Advanced Encryption Standard", Published in 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 02 January 2020,10.1109/CyberC.2019.00019.

[11] Khairul Muttaqin1 and Jefril Rahmadoni, "ANALYSIS AND DESIGN OF FILE SECURITY SYSTEM AES (ADVANCED ENCRYPTION STANDARD) CRYPTOGRAPHY BASED ", Journal of Applied Engineering and Technological Science Vol 1(2) 2020: 113-123.

[12] BRANDON LANGENBERG, HAI PHAM, AND RAINER STEINWANDT4, "Reducing the Cost of Implementing the Advanced Encryption Standard as a Quantum Circuit ", IEEE, publication January 16, 2020.

[13] Noemie Floissac and Yann L'Hyver, "From AES-128 to AES-192 and AES-256, How to Adapt Differential Fault Analysis Attacks on Key Expansion", Published in 2011 Workshop on Fault Diagnosis and Tolerance in Cryptography, IEEE, 14 November 2011, DOI: 10.1109/FDTC.2011.15.

[14] Liang Han , Ning Wu , Fen Ge , Fang Zhou , Jin Wen and Peiyao Qing , "Differential Fault Attack for the Iterative Operation of AES-192 Key Expansion", Published in: 2020 IEEE 20th International Conference on Communication Technology (ICCT), 24 December 2020 , DOI: 10.1109/ICCT50939.2020.9295779 .

[15] Ricard V. Soléa and JordiBascompteb, "Measuring chaos from spatial information," Journal of Theoretical Biology Volume 175, Issue 2, 21 July 1995, Pages 139-147.

[16] Ziaur Rahman, Xun Yi, Ibrahim Khalil, and Mousumi Sumi, "Chaos and Logistic Map based Key Generation Technique for AES-driven IoT Security". Electronics 2022, 1, 0.doi.org/10.3390/electronics1010000.

[17] Ray Huffaker, Marco Bittelli, and Rodolfo Rosa, "Nonlinear Time Series Analysis with R," Oxford Scholarship Online: February 2018, DOI:10.1093/oso/9780198782933.001.0001.

[18] Vatchara Saicheur and Krerk Piromsopa, "An implementation of AES-128 and AES-512 on Apple mobile processor", IEEE, 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 07 November 2017, and DOI: 10.1109/ECTICon.2017.8096255.

[19] Ria Andriani, Stevi Ema Wijayanti and Ferry Wahyu Wibowo,"Comparision of AES 128, 192 And 256 Bit Algorithm for Encryption and Description File", IEEE .2018 3rd International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia, DOI: 10.1109/ICITISEE.2018.8720983.

[20] Ibrahim Yasser, Mohamed A. Mohamed, Ahmed S. Samra, and Fahmi Khalifa, "A Chaotic-Based Encryption/Decryption Framework for Secure Multimedia Communications", Entropy 2020, 22(11), 1253, doi.org/10.3390/e22111253.

[21] Mahdi Shariatzadeh, Mohammad Javad Rostami, and Mahdi Eftekhari," Proposing a novel Dynamic AES for image encryption using a chaotic map key management approach", Optik - International Journal for Light and Electron Optics 246 (2021) 167779, Volume 246, November 2021, https://doi.org/10.1016/j.ijleo.2021.167779Get.

[22] Chih-Hsueh Lin, Guo-Hsin Hu, Che-Yu Chan, and Jun-Juh Yan, "Chaos-Based Synchronized Dynamic Keys and Their Application to Image Encryption with an Improved AES Algorithm", MDPI, Applied Sciences, Appl. Sci. 2021, 11, 1329. https://doi.org/10.3390/app11031329

[23] Md. Anwar Hussain and Popi Bora, "A Highly Secure Digital Image Steganography Technique Using Chaotic Logistic Map and Support Image", Published in 2018 IEEE International Conference on Information Communication and Signal Processing (ICICSP),29 November 2018, DOI: 10.1109/ICICSP.2018.8549790.

[24] Abdulrahm Moh'd , Yaser JararwehYaser and JararwehLoai Tawalbeh , "AES-512: 512-Bit Advanced Encryption Standard algorithm design and evaluation",IEEE,Conference: 7th International Conference on Information Assurance and Security, IAS 2011, Melacca, Malaysia, December 5-8, 2011 ,DOI: 10.1109/ISIAS.2011.6122835.

[25] Sachin Dhawan and Rashmi Gupta," Analysis of various data security techniques of steganography: A survey," Information Security Journal: A Global Perspective, 12 Aug 2020, Volume 30, 2021 - Issue 2, doi.org/10.1080/19393555.2020.1801911.

[26] Mohammed Mahdi Hashim and Mohd Shafry Mohd Rahim," Image Steganography Based On Odd/Even Pixels Distribution Scheme and Two Parameters Random Function". Journal of Theoretical and Applied Information Technology, 30th November 2017. Vol.95. No 22.