A Hybrid Modification Algorithm for Advanced Encryption Standard (AES)

Prof. Dr.Abdul Monem S. Rahma
Computer Science Department, University of Technology
Baghdad, Iraq
Email: monem.rahma@yahoo.com

Zainab J. Hanash Al- Abedi
Computer Science Department, University of Technology
Baghdad, Iraq
Email: zainab.master100@yahoo.com

Abstract—Security in transmission storage of digital images has its importance in today's image communications. Due to the increasing use of images in industrial process, it is essential to protect the confidential image data from unauthorized access. Advanced Encryption Standard (AES) is a well known block cipher that has several advantages in data encryption. In this paper, a modification to the (AES) to reflect a high level better image encryption and security has been presented. The modification is done by adjusting (the Shift Row Transformation, Generate random key, Generating S-Box randomly). The algorithm uses GF (2^8). It works in 15 rounds, which provides high speed along with the complexity, and applies encryption to the compressed image (JPEG), taking into consideration saving the image data from loss after decryption.

Keywords: cryptography, symmetric key; randomness, AES, blockcipher; polynomial, key generator; JPEG.

1 INTRODUCTION

Joint Photographic Experts Group (JPEG) is an advanced lossy compression method for color or grayscale still images (not videos). It does not manage bilateral level images (black and white) extremely well and it works best on continuous tone images.

The JPEG standard has proved successful and has become widely used for image compression, particularly in Web pages [1]. In this paper, this type of digital image is used.

2 THE CONCEPT OF BLOCK CIPHER

In a block cipher, a group of plain text elements greater than one charterer are encrypted together creating a group of cipher text. A block cipher is one in which a block of plain text is treated as a whole and used to produce cipher text blocks of equal length. Successful block cipher designs often integrate the concepts of confusion and diffusion [2].

2.1 Previous modifications of the AES algorithm

Many researchers have used the Advanced Encryption Standard (AES) algorithm as the basis for their work. Some of the published works which used AES are described as in the following:

1. Nada Hussein Mohammad, 2015, proposed "Encryption of the Audio File in Real Time", in which she used parallel programming techniques based on the implementation of the OpenMp tools available for C and C++. Parallel programming was applied to an improved version of AES, explained in details in that thesis, which provides a good time acceleration compared to the serial implementation of those improvements [3].

2. Kazys Kazlauskas, Jaunius Kazlauskas (2009) proposed a new approach to generate different S-Boxes in the AES algorithm. The secret key was generated in a random manner. The proposed algorithm was tested by generating different S-Box look-up tables by only changing one bit of the cipher key. The proposed algorithm offers a good solution for the problem of fixed structure for the S-Box table in the AES algorithm. This solution has led to an increase in the complexity. Therefore, it would raise the security level of this algorithm. The number of S-Boxes that were generated in this approach was very large, which was considered the main benefit of this algorithm [4].

3. Ali A. et al, in this researcher used a new AES cipher method which depended on shift register in addition to the chaotic map for encryption the image. The aim of this new method, was to reduce time and to increase encryption of image [5].

4. Omar A. Dawood, et al. (2015) proposed a cipher that uses the SPN structure and what is known as the Galois Field (GF) [2^8]. It is an iterated cipher that has a conservative design which is easily implemented in both hardware and software [6].

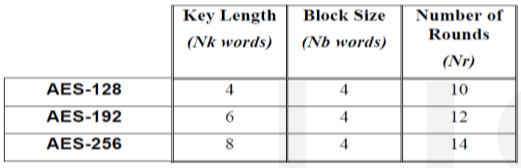
5. Atheer M. Abbas Al-Abbassi, (2015) provided a state of balance between time and complexity of encrypted documents by using multiple irreducible polynomials with order of 8, 4, and 2 (high, medium, and low complexity, respectively) depending on the importance of the document [7].

6.Hala Bahjat., and May A. Salih (2014) proposed "Speed Image Encryption Scheme using Dynamic Galois Field GF(P) Matrices", The study case shown in this paper works on GF(7) and for encryption key sizes varying from (3X3) to (12X12). The goal was to provide a highly secure encryption algorithm with a wide space for encryption speed [8].

2.2 The AES algorithm

This standard is used for the Rijndael algorithm, a symmetric block cipher that can deal with data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits. Rijndael was designed to manage additional block sizes and key lengths. However, they are not adopted in this standard. During the remainder of this description of the standard, the algorithm will be identified as “the AES algorithm.” The algorithm may be used with the three different key lengths referred to above, and therefore these various types will be referred to as “AES-128”, “AES-192”, and “AES-256” ([9] and [10]).

Table 1. Key-Block-Round Combinations



	Key Length (Nk words)	Block Size (Nb words)	Number of Rounds (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

2.3 Finite field arithmetic

All operations are implemented in 8-bit bytes in AES. In particular, the arithmetic operations of addition, multiplication, and division are executed over the finite field. The field is a set in which we can perform addition, subtraction, multiplication, and division. There is a method of defining a finite field including elements; such a field is indicated as GF (2^n). Consider the set of all polynomials of degree or less with binary coefficients. Thus, each polynomial has the form

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0 = \sum_{i=0}^{n-1} a_ix^i \quad \ldots 1$$

Where each a_i takes on the value 0 or 1.There are a total of 2^n various polynomials. For n=3, the 2^3 =8 polynomials in the set are [2]:

$$\begin{array}{llll} 0 & x & x^2 & x^2+x \\ 1 & x+1 & x^2+1 & x^2+x+1 \end{array} \quad \ldots 2$$

2.4 The AES structure and implementation

This algorithm is not a Feistel structure which is one noticeable feature of this structure. Recall that, in the classic Feistel structure, half of the data block is used to modify the other half of the data block, and then the halves are swapped. AES uses substitutions and permutation as a replacement for processing the entire data block as a single matrix during each round.

The key that is provided as input is extended into an array of forty-four 32-bit words, w[i]. Four distinct words (128 bits) serve as a round key for each round. Four various stages are used, three of substitution and one permutation:

1. Substitute bytes: byte-by-byte substitution of the block, using an S-Box to implement it.
2. ShiftRows: A simple permutation.
3. MixColumns: A substitution that makes use of arithmetic over GF (2^8).
4. AddRoundKey:

A simple bitwise XOR of the current block with a part of the expanded key produces a very simple structure. For both encryption and decryption, the cipher starts with an AddRoundKey phase, and then it continues with nine rounds that each contains all four phases, and it ends with a tenth round of three phases.

Only the AddRoundKey phase makes use of the key. Therefore, the cipher starts and ends with an AddRoundKey phase. Any other stage applied at the start or end is reversible without knowledge of the key and so would add no security.

The stage of AddRoundKey is in fact a form of Vernam cipher and by itself would not be remarkable. The other three phases together provide diffusion, confusion, and nonlinearity. However, by themselves they would not provide security because they do not use the key. We can view the cipher as alternating operations of XOR [11].

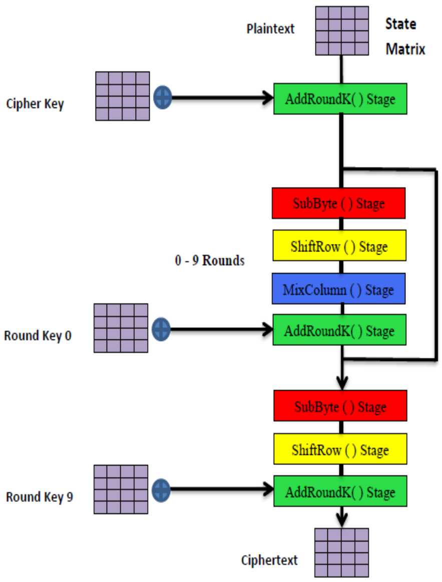


Figure 1. Block Diagram for the AES Structure.

2.5 Five Basic Tests:

Five statistical tests are used to determine whether the binary sequences possess some specific features. If a truly random sequence would appear, it confirms that the result of the test is not specific, but rather is probabilistic. If a sequence passes all five tests, it does not ensure that the sequence actually resulted from a random bit generator [12].

3 THE PROPOSED ALGORITHM FOR IMAGE ENCRYPTING:

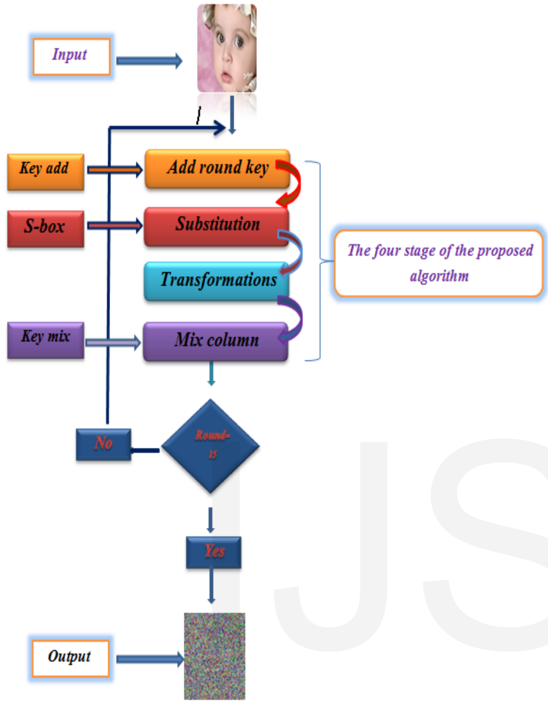


Figure 2. Block Diagram for the proposed algorithm

3.1 Description of the Proposed Algorithm:

The proposed algorithm is similar to the (AES) of the structural hand and stages and different from their in the number of rounds (since the proposed algorithm (15) round), key generator where the key is generated randomly within the range (0-255). The following stages show the changes that taken place in four stages of the proposed algorithm:

3.2 The proposed AddRoundkey stage.

In this proposed stage the size of element is fixed as well as the key matrix is constant in all (15) rounds. In this stage, the key matrix is provided with the same size of the block image, where the process(XOR) is between the elements of the block (4 * 4) and the key random generated (4 * 4), shown in the following algorithm (1):

```
Algorithm 1: The Proposed Add Round key
 Encryption/Decryption
 Transformation Function/128bits
Input: block image, Key add
Output: State1 matrix
Begin
Step1: The dimension of block matrix is 4*4 for GF ($2^8$).
Step2: Each cell in the matrix of block is added with the
cell key matrix based on GF table and the results are
stored in state1 matrix.
End.
```

3.3 The proposed Substitution stage.

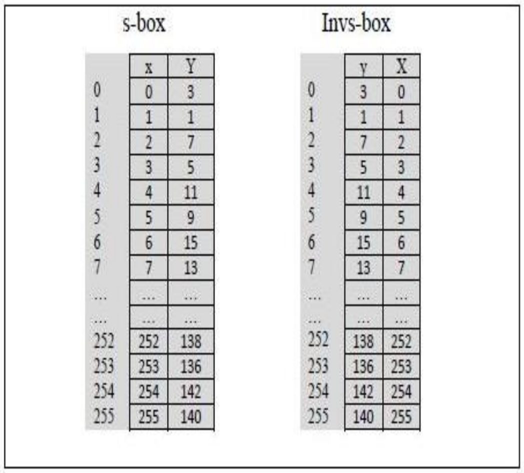
The proposed substitution stage is in the S-box. It follows the add Round key stage which is based on the GF(2^8) that has been chosen in the add Round key stage to construct the S-box based on irreducible polynomial ($x^8+x^6+x^5+x^4+1$) as shown in the following algorithm(2):

```
Algorithm2:The proposed substitution
Encryption/Decryption Transformation Function/128bits
Input: state1 matrix
Output: state2 matrix
Begin
Step1: It generats S-Boxes based on of GF ($2^8$).
Step2: The dimension of state1 matrix is 4*4 for
 GF ($2^8$).
Step3: pass any cell of the state1 matrix to the chosen
S-Box in step1 and store the result of the element
in the state2 matrix, and so on for the other cases.
End.
```

3.4 The following tables show the construction process of s-boxes:

For the GF (2^8) with irreducible polynomial ($x^8+x^6+x^5+x^4+1$) the S- box and its inverse are shown in table (1).

Table 2. S-box and its inverse in GF (2^8)



	s-box			Invs-box	
	x	Y		y	X
0	0	3	0	3	0
1	1	1	1	1	1
2	2	7	2	7	2
3	3	5	3	5	3
4	4	11	4	11	4
5	5	9	5	9	5
6	6	15	6	15	6
7	7	13	7	13	7
...	...	...	...	...	...
...	...	...	...	...	...
252	252	138	252	138	252
253	253	136	253	136	253
254	254	142	254	142	254
255	255	140	255	140	255

3.5 The proposed Shift Row stage.

In the standard AES there is a fixed shift for each row, (0) for first row, (1) for second row, (2) for third row and (3) for fourth row. While in the proposed stage, the transport process in the third phase will be as follows:

There are three types of transport through the block:
1. Main diagonal (transport).
2. Upper (left to right).
3. Lower (top to bottom).

The process will be applied on the results of the previous stage on all blocks during the 15 round at this stage (the third stage of the algorithm), which is implemented in each round.

$$R[i] = \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} \\ \mathbf{5} & \mathbf{6} & \mathbf{7} & \mathbf{8} \\ \mathbf{9} & \mathbf{10} & \mathbf{11} & \mathbf{12} \\ \mathbf{13} & \mathbf{14} & \mathbf{15} & \mathbf{16} \end{pmatrix} \quad \text{>> Matrix before transport}$$

1. **Main diagonal (transport)- first transport**
Diffusion matrix: That resulting from previous operations (sub), where transport is to convert rows to columns. At this stage, it will be transferred to each row and a column, as shown in the example, as the row (1 2 3 4) turned into a column, and so on for the rest of rows, so the resulting matrix will be R [i] 1.

$$R[i]\ 1 = \begin{pmatrix} \mathbf{1} & \mathbf{5} & \mathbf{9} & \mathbf{13} \\ \mathbf{2} & \mathbf{6} & \mathbf{10} & \mathbf{14} \\ \mathbf{3} & \mathbf{7} & \mathbf{11} & \mathbf{15} \\ \mathbf{4} & \mathbf{8} & \mathbf{12} & \mathbf{16} \end{pmatrix} \quad \text{>> first transport}$$

Main diagonal (transport)

2. **Upper (left to right)**
At this stage, it will be moving in each line from left to right. That is to say, the top row of the matrix (1 2 3 4) will be (4 3 2 1) , (5 6 7 8) will be (8 7 6 5), (9 10 11 12) will be (12 11 10 9) and the bottom row of the matrix (13 14 15 16) will be (16 15 14 13), as shown in the example R[I]2.

$$R[i]\ 2 = \begin{pmatrix} \mathbf{4} & \mathbf{3} & \mathbf{2} & \mathbf{1} \\ \mathbf{8} & \mathbf{7} & \mathbf{6} & \mathbf{5} \\ \mathbf{12} & \mathbf{11} & \mathbf{10} & \mathbf{9} \\ \mathbf{16} & \mathbf{15} & \mathbf{14} & \mathbf{13} \end{pmatrix} \quad \text{>> second transport}$$

Upper (left to right)

3. **Lower (top to bottom)**

At this stage, the first row will turn down to the last row and the last row will turn to the top. That is to say, the top row of the matrix (1 2 3 4) will turn to the last row instead of (13 14 15 16). As well as for the last row (13 14 15 16) turns to the top where the same approach is intended and so on for each block [4 * 4] , shown below for R[i]3.

$$R\ [i]\ 3= \begin{pmatrix} \mathbf{13} & \mathbf{14} & \mathbf{15} & \mathbf{16} \\ \mathbf{9} & \mathbf{10} & \mathbf{11} & \mathbf{12} \\ \mathbf{5} & \mathbf{6} & \mathbf{7} & \mathbf{8} \\ \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} \end{pmatrix} \quad \text{>> Third transport}$$

Lower (top to bottom)

The results are considered as shown in the following algorithm (3):

```
Algorithm 3: The Proposed shift Rows Encryption
/Decryption Transformation Function/128bits

Input: State2 matrix
Output:State3 matrix
Begin
Step1: Apply the first operation on the blocks
(Main diagonal(transport))
Step2: Apply the second operation on the blocks
(Upper (left to right))
Step3: Apply the third operation on the blocks
(Lower (top to bottom))
End.
```

3.6 The proposed Mix column stage:

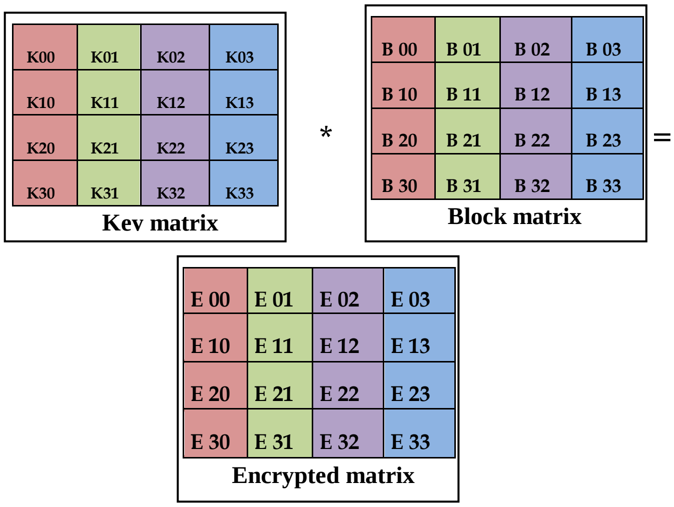
A process of fourth phase of each round involves the multiplication between the keys matrix and the resulting matrix of the previous phase (shift), where each is (4 * 4). The process is a multi-(row * column).

The results are shown in the following Algorithm (4):

```
Algorithm 4: The Proposed Mix column
Encryption/Decryption
Transformation Function/128bits

Input:
State3 matrix (block image); Key mix
Output:
Encrypted matrix (block image).
Begin
Step1: Generating a matrix at (4*4).
The elements of this matrix are random
 numbers. Based on GF ($2^8$) and it is a singular
, it  has inverse.
Step2: The block state in the matrix with size
either (4*4).
Step3: Compute sum of products of elements of
the row of key matrix with the Column
elements of the block matrix.
Step4: Store the result in the encrypted matrix.
End.
```

For GF(2^8), the size of block matrix and key is (4*4) and the element of the encrypted state matrix is represented by the products sum of block column elements by row key matrix, so that the first element in encrypted state matrix is represented by sum of product of the elements of the first row of key matrix with the elements of column of block matrix , the individual additions and multiplications are executed in GF(2^8) with irreducible polynomial ($m(x)=x^8+x^6+x^5+x^4+1$) as shown in figure (3).



K00	K01	K02	K03
K10	K11	K12	K13
K20	K21	K22	K23
K30	K31	K32	K33

B 00	B 01	B 02	B 03
B 10	B 11	B 12	B 13
B 20	B 21	B 22	B 23
B 30	B 31	B 32	B 33

E 00	E 01	E 02	E 03
E 10	E 11	E 12	E 13
E 20	E 21	E 22	E 23
E 30	E 31	E 32	E 33

Figure 3. Encryption process in modified Mix Column stage in GF (2^8)

Where:

E00= (K00*B00+K01*B10+K02*B20+K03*B30) mod m(x)

E01= (K00*B01+K01* B11+K02*B21+ K03*B31) mod m(x)

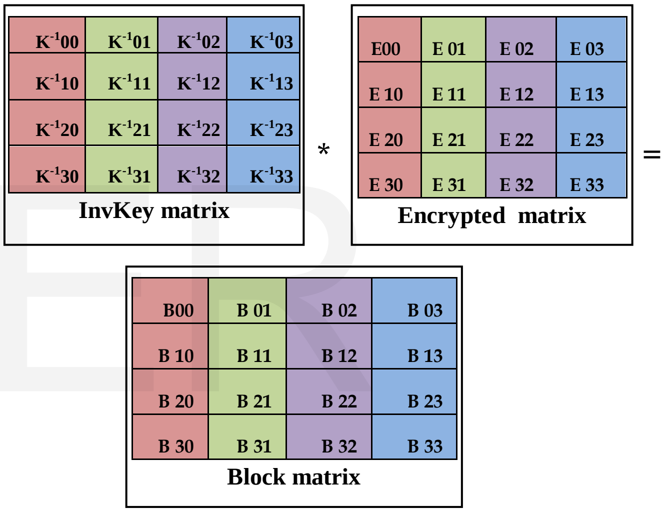
E02= (K00*B02+K01* B12+K02*B22+ K03*B32) mod m(x)

E03= (K00*B03+K01* B13+K02*B23+ K03*B33) mod m(x)

As concerning the decryption process, the first element of the product (block) matrix is the sum of products of elements of the first row of the inverse of key matrix with the first column of encrypted matrix, the individual additions and multiplications are executed in GF(2^8) with irreducible polynomial

{$m(x)=x^8+ x^6+x^5+ x^4+1$} as shown in figure (4).

Where:



$K^{-1}00$	$K^{-1}01$	$K^{-1}02$	$K^{-1}03$
$K^{-1}10$	$K^{-1}11$	$K^{-1}12$	$K^{-1}13$
$K^{-1}20$	$K^{-1}21$	$K^{-1}22$	$K^{-1}23$
$K^{-1}30$	$K^{-1}31$	$K^{-1}32$	$K^{-1}33$

E00	E 01	E 02	E 03
E 10	E 11	E 12	E 13
E 20	E 21	E 22	E 23
E 30	E 31	E 32	E 33

B00	B 01	B 02	B 03
B 10	B 11	B 12	B 13
B 20	B 21	B 22	B 23
B 30	B 31	B 32	B 33

Figure 4. Decryption process in modified Mix Column stage in GF (2^8)

B00= ($K^{-1}00$*E00+ $K^{-1}01$*E10+ $K^{-1}02$*E20+ $K^{-1}03$*E30) mod m(x)

B01= ($K^{-1}00$*E01+ $K^{-1}01$*E11+ $K^{-1}02$*E21+ $K^{-1}03$*E31) mod m(x)

B02= ($K^{-1}00$*E02+ $K^{-1}01$*E12+ $K^{-1}02$*E22+ $K^{-1}03$*E32) mod m(x)

B03= ($K^{-1}00$*E03+ $K^{-1}01$*E13+ $K^{-1}02$*E23+ $K^{-1}03$*E33) mod m(x)

3.7 The process of generation keys:

the Generation of the keys will be done by key- process which is a process of generating random keys through all the stages of the encryption by a function taking into consideration the size of the keys for GF(8) that have been described previously. The key provider is characterized by the ability to generate keys for all rounds of algorithm, where its certain keys are generated for the Add Round stage and have been stored in the (key matrix1). The

Mix column stage matrix is generated and characterized as a singular (it has inverse).

The encryption process in the proposed system begins by using a matrix with size based on the GF (8) then the block is passed from one stage to another inside the four stages of algorithm. After the first round is finished for processing the first block, the result is passed for second time for the next round and so on. Concerning the decryption process, the same steps are followed in the encryption process, but with inverse of keys of stages.

3.8 Decryption for the proposed algorithm

It may be noted that to the decryption process the same steps, the decryption for the proposed algorithm works through implementing the inverse of all four operations which are described previously by using the inverse of key.

4 THE RESULT OF MODIFIED ALGORITHM:

Table 3. Comparison of Time, between the Proposed Algorithm and the AES

Algorithms	No. of image	No. of bit	Key Size	Time Encryption Time(M.S.MS)	Time Decryption Time(M.S.MS)
Original-AES	text	240128 bit	128bit	1.27.796	-
The Proposal algorithm	1. image (a)	240128 bit	128bit	0.0.753	0.1.227
	2. image (b)	240128 bit	128bit	0.0.697	0.1.209
	3. image (c)	240128 bit	128bit	0.0.698	0.1.209
	4. image (d)	240128 bit	128bit	0.0.739	0.1.234

In this table, the elapsed time is examined to encrypt a given block between the proposed algorithm and the AES algorithm. The time elapsed in the proposed block encryption algorithm is less than is spent in the algorithm (AES). And it should be noted for the time in the proposed algorithm which is attributed to the uneven time despite the uniform size of the image. Using a random key for each image, as shown in Table 3.

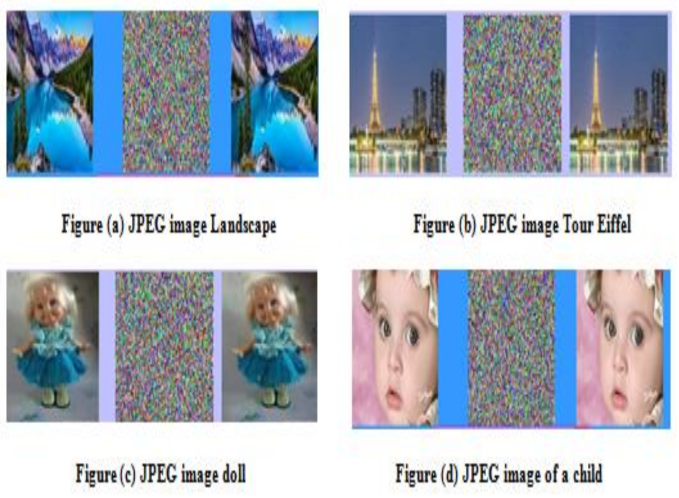


Figure (a) JPEG image Landscape
Figure (b) JPEG image Tour Eiffel
Figure (c) JPEG image doll
Figure (d) JPEG image of a child

Figure 5. Illustrates the image encryption.

5 THE RANDOMNESS TESTS:

Table 4. Randomness Test to the Image (JPEG) - 15 rounds

NO. of Image	Frequency test	ITH FREEDOI DEGREE & MUST BE	Run test		VITH FREEDOM DEGREE & MUST BE	
1. Image (a)	1.450	1	T0	12.905	13	<=22.078
		<= 3.84	T1	27.161	17	<= 27.303
2. Image (b)	3.546	1	T0	16.302	16	<= 26.012
		<= 3.84	T1	12.220	16	<= 26.012
3. Image (c)	0.054	1	T0	11.913	17	<= 27.303
		<= 3.84	T1	19.846	15	<= 24.712
4. Image (d)	0.228	1	T0	19.097	14	<= 23.401
		<= 3.84	T1	12.919	15	<= 27.303

Table 5. Randomness Test to the Image (JPEG) - 15 rounds

NO. of Image	Poker test	WITH FREEDOM DEGREE & MUST BE	Serial test	WITH FREEDOM DEGREE & MUST BE
1. Image (a)	1.622	<= 11.1 5	2.413	3 <= 7.81
2. Image (b)	1.734	<= 11.1 5	6.487	3 <= 7.81
3. Image (c)	1.409	<= 11.1 5	2.142	3 <= 7.81
4. Image (d)	1.431	<= 11.1 5	4.125	3 <= 7.81

Table 6. Randomness Test to the Image (JPEG) - 15 rounds

Auto Correlation test for ten bits - in 15 rounds				
	Image (a)	Image (b)	Image (c)	Image (d)
Shift 1	0.292	0.001	0.284	0.116
Shift 2	0.034	2.560	0.022	1.420
Shift 3	0.418	0.334	0.972	0.759
Shift 4	0.052	0.481	0.499	0.023
Shift 5	0.310	0.001	0.023	0.001
Shift 6	0.313	0.001	0.024	0.001
Shift 7	0.403	0.624	0.222	1.545
Shift 8	0.132	1.727	0.558	0.005
Shift 9	1.174	0.387	0.697	4.375
Shift 10	3.522	0.470	2.705	0.252
WITH FREEDOM DEGREE " 1 " MUST BE <= 3.84				

6 Security complexity analysis

The algorithm has been designed to provide high security, with the required complexity and speed, depending on the encryption blocks. From table (7), the number of each block is calculated, and then the possibilities are multiplied in the (15) round.

Table 7. Illustration of Security Complexity Analysis

The operation		In15 round
1.	The Round Key Addition function	(4*4*256)*15
2.	Sub Byte Transformation function	(4*4*256)*15
3.	Transformations function	(3*4*4*256)*15
4.	Mix Column Transformations	((4+4)*256)*16) *15

7 Conclusion

The implementation of a set of block- encrypting functions is carried out. (The Round Key Addition function; Sub Byte; Transformations function and Mix Column Transformations), on the compressed image (JPEG), with using the mathematical operations based on Gf *(2^8)* and generating the (random key) to encrypting of blocks, led to a reduction in the encryption and decryption time, with raising the complexity of this algorithm.

8 ACKNOWLEDGMENT

The authors would like to extend special thanks to Dr. Najm Al-Rubaye for his great contribution in editing and proof reading this paper from a linguistic point of view.

References

[1] Salomon, D.,"Data Compression", Fourth Edition ,Springer- Verlag London Limited 2007.

[2] Stallings W., "Cryptography and Network Security: Principles and Practice", Fifth Edition, Prentice Hall, 2011.

[3] Hussein N., “Encryption of the Audio File in Real Time”, Compute Science Department / University of Technology, 2015.

[4] Kazlauskas, K. and Kazlauskas, J., "Key-Dependent S-Box Generation in AES Block Cipher System", INFORMATICA, Vol.20, No. 1, 2009.

[5] Abdulgader, A., Ismail, M., Zainal, N. and Idbaa, T., "Enhancement of AES Algorithm Based on Chaotic Maps and Shift Operation for Image Encryption", Journal of Theoretical and Applied Information Technology, ISSN: 1992-8645, 2015.

[6] Omar A. Dawood, Abdul Monem S. Rahma and Abdul Mohsen J. Abdul Hossen, “The New Block Cipher Design (Tigris Cipher)”, I. J. Computer Network and Information Security, 2015.

[7] Metaab, A., "Automating security and time balancing in institution daily work on data and time balancing transfer", 2015.

[8] Bahjat H., & Salih M., " Speed Image Encryption Scheme using Dynamic Galois Field GF(P) Matrices ", International Journal of Computer Applications (0975 – 8887) Volume 89 – No.7, March 2014.

[9] J. Daemen and V. Rijmen, AES Proposal: Rijndael, AES Algorithm Submssion, September 3, 1999.

[10] J. Daemen and V. Rijmen, The block cipher Rijndael, Smart Card research and Applications, LNCS 1820, Springer-Verlag, pp. 288-296.

[11] NIST," Advanced Encryption Standard (AES)", Federal Information Processing Standards Publication (FIPS PUB) 197, Nov 2001.

[12] Andrew, R. & Juan, S. & Miles, S. ,"Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications ",NIST Special Publication 800, 2001.