Wireless Pers Commun (2014) 79:811–829
DOI 10.1007/s11277-014-1888-7

# High Definition Image Encryption Algorithm Based on AES Modification

**Salim Muhsin Wadi · Nasharuddin Zainal**

Published online: 26 June 2014


**Abstract** In this article, a high-speed and highly restricted encryption algorithm is proposed to cipher high-definition (HD) images based on the modified advanced encryption standard (AES) algorithm. AES is a well-known block cipher algorithm and has several advantages, such as high-level security and implementation ability. However, AES has some drawbacks, including high computation costs, pattern appearance, and high hardware requirements. The aforementioned problems become more complex when the AES algorithm ciphers an image, especially HD images. Three modifications are proposed in this paper to improve AES algorithm performance through, decreasing the computation costs, decreasing the hardware requirements, and increasing the security level. First, modification was conducted using Mix-Column transformation in 5 rounds instead of 10 rounds in the original AES-128 to decrease the encryption time. Security is enhanced by improving the key schedule operation by adding MixColumn transformation to this operation as second modification. In addition, to decrease the hardware requirements, S-box and Inv. S-box in the original AES are replaced by one simple S-box used for encryption and decryption in the proposed method. The proposed AES version conducts one of the ciphering modes to solve the appearance pattern problem. Experimental results indicate that the proposed modifications to the AES algorithm made the algorithm more compatible with HD image encryption.

**Keywords** Image ciphering · AES algorithm · AES modification · Ciphering modes · HD image

S. M. Wadi (✉) · N. Zainal
Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia
e-mail: salimmw@eng.ukm.my

N. Zainal
e-mail: nash@eng.ukm.my

S. M. Wadi
Communications Technical Engineering Department, Foundation of Technical Education, Baghdad, Iraq

## 1 Introduction

Fast growth in communications techniques, for example satellite, mobile network, Internet, and ground communications, resulted in the urgent need to protect important individual, general, and universal equipments and their respective data against attackers, illegal copying and allocation [1]. Encryption algorithms are used as the best way to maintain the safety of transmitted data through converting the information into an incomprehensible form. Image encryption is conducted through the confusion (replacing the locations of the pixels) or/and diffusion (changing the values of the pixels) processes, where the perfect scene of the ciphered image is similar to the jamming scene in TV. Images are usually represented in two forms, either in the frequency domain or the spatial domain, where images are partially or fully encrypted in any domain [2]. Although most of the image-ciphering algorithms are in the spatial domain, the digital image could also be encrypted in the frequency domain [3]. Image-ciphering algorithms in the transform domain are based on fractional fourier transform [4,5], fast fourier transform [6], discrete cosine transform [7], and fresnel transform [8] and [9].

Encryption algorithms in the spatial domain effectively encrypt images in terms of security level and simplicity. Recently, chaos theory extensively used in image encryption operation produces a number series using initial conditions and parameters based on chaotic maps [3,10–15]. However, chaotic map-based encryption algorithms have some drawbacks, such as low security levels and high computation costs, because of the need to convert its resulting sequences to binary or integer numbers to make these sequences compatible with the ciphering requirements [16].

Decomposition technology is also used for ciphering purposes [17–22]. However, the decomposition-based encryption algorithms have the following problems: first, atonality exists in the security level because the number weights of bit planes are constant. Second, little or, in some cases, almost no key space is used in these approaches, which decreases the computational cost attacks.

The naive encryption algorithm is a public cryptography algorithm and is extensively applied in most enforcements, such as smart cards, cellular telephone, ATMs, and World Wide Web servers [23]. The data encryption standard (DES) was first introduced as a naive encryption algorithm [24]. After some years, the DES algorithm was broken. Therefore, advanced encryption standard (AES) was introduced as an alternative algorithm to DES [25]. However, image ciphering using the AES algorithm has some disadvantages, such as more computations required and artifacts appearing in the ciphered image, especially if the plain image has regions with high intensity[26–29].

Subramanyan et al. [30] and and F. Muhaya [31] are using the chaotic and Arnolds cat maps to create a key and shuffle the pixels in AES algorithm. These improvements strengthened the key, but the encryption time remained long. By contrast, Tran et al. [32] and Chen et al. [33] enhance the S-box of the AES algorithm. A new S-box, named the Gray S-box, has been proposed by Tran et al. [32]. The fixed S-box was replaced by the randomly selected inhomogeneous S-box proposed by Chen et al. [33]. The AES S-box is not the main factor in the consumption of encryption time. Therefore, modification of the AES S-box does not significantly decrease the time or pattern appearance.

In related works and published papers, many modifications, such as the modification of the S-box generation or MixColumn transformation, are suggested to enhance the performance of the AES algorithm.

As previously mentioned, if the plain image contains some identical color regions, then some patterns occur in the ciphered image in these areas. Chi-Wu et al. [29] proposed two approaches to overcome this problem. The first method modifies the electronic codebook

(ECB) mode applied on the AES algorithm. The suggested modification is based on the change or removal of these identical color regions. This task is achieved by adding three number sequences to the input image. These number sequences include the sequential numbers from the counter output, the non-sequential number from the accumulator output, and the random numbers generated from the cipher function itself. Another approach is based on compression of an image to decrease the occurrence of identical colors in those areas. However, these two methods add other computations to the original AES and then increase the encryption time, especially when applied to the image.

Kamali et al. [27] decreased the pattern appearance problem by adjusting the ShiftRow transform. Initially, the first byte in state [0 0] is determined whether it is even or odd. If the first byte is even, then the first and fourth row bytes are unchanged, whereas the second and third row bytes are shifted to the right three and two times, respectively. Meanwhile, if the first byte is odd, then the second and fourth row bytes are shifted to the left one and three times, respectively. However, the first and third rows are unchanged. The proposed method focused on only one problem of the AES algorithm and overlooks other problems, such as security level and computation costs.

Tran et al. [32] modified the AES algorithm security level by improving the algebraic expression in the S-box generation. Gray code conversion is applied to the AES S-box as a preprocessing step to make the algebraic expression more complicated. This enhancement strengthens the S-box and the AES algorithm against interpolation and algebraic attacks. However, a disadvantage of this improvement is increased computation cost.

In this paper, three modifications were suggested for the AES algorithm to make it suitable for high-definition (HD) image ciphering. Briefly, these enhancements are achieved by, firstly, conducting Mix-Column step 5 times instead of 10 times as in the initial AES-128, secondly, adding this operation to the key schedule, where these two modifications lead to the decrease in encryption time and reinforcement of the key schedule operation and the AES algorithm. In addition, we proposed a new S-box for encryption and decryption, instead of two in the original AES algorithm. The proposed algorithm was executed under one of the ciphering modes to decrease the pattern appearance problem.

The rest of this paper is organized as follows. Brief description of AES algorithm is introduced in Sect. 2. The details of the suggested enhancements of AES algorithm are given in Sect. 3. Experimental results with some comparisons are discussed in Sect. 4. Paper conclusions are presented in Sect. 5.

## 2 Initial Version of AES Algorithm

Before we discuss the proposed modifications, we must briefly show the operation of the AES algorithm.

### 2.1 AES Algorithm

#### *2.1.1 AES Algorithm Round*

The AES algorithm has three versions, depending key length (AES-128, AES-192, and AES-256). These keys are represented in arrays with sizes $4 \times 4$, $4 \times 6$, and $4 \times 8$ whilst, 128-bit block data are arranged in the $4 \times 4$ array named state [31]. In original AES, four consecutive transforms are carry out on a state in 10, 12, or 14 rounds based on key length, see Fig. 1. The AES transformations are explained as follows:

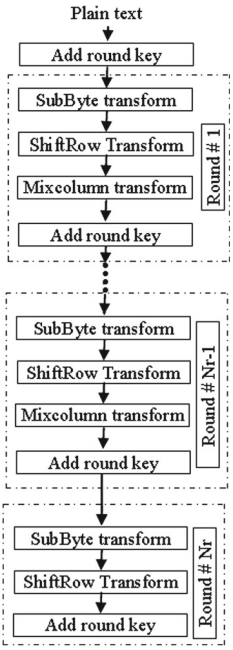


**Fig. 1** AES algorithm structure where Nr =10, 12, or 14 as key length (128, 192 or 256 bits), respectively

*1. Substitution Transformation* SubByte transformation is a nonlinear byte substitution operation which exchange bytes of the state independently using S-box table. The S-box is produced by taking the multiplicative inverse in the finite Glues Field $2^8$, as shown in (1), and then applying the affine transformation over it [34]:

$$b_i^{'} = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus c_i \tag{1}$$

where $0 \leq i \ll 8$, $b_i$ is the $i$th bit of the byte, and $c_i$ is the $i$th bit of byte c with value of 01100011. Equation (2) expresses the S-box affine transformation element in matrix form [35], as follows:

$$\begin{bmatrix} b_0^{'} \\ b_1^{'} \\ b_2^{'} \\ b_3^{'} \\ b_4^{'} \\ b_5^{'} \\ b_6^{'} \\ b_7^{'} \end{bmatrix} = \begin{bmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{2}$$

The S-box and SubByte transformations are shown in Fig. 2.

*2. ShiftRow Transformation* Shifting operation is applied to the state rows in this step, where the 1st row is not shifted; the 2nd, 3rd, and 4th rows are shifted to the right in one, two, and three steps, respectively. Figure 3 shows the ShiftRow transformation [26].

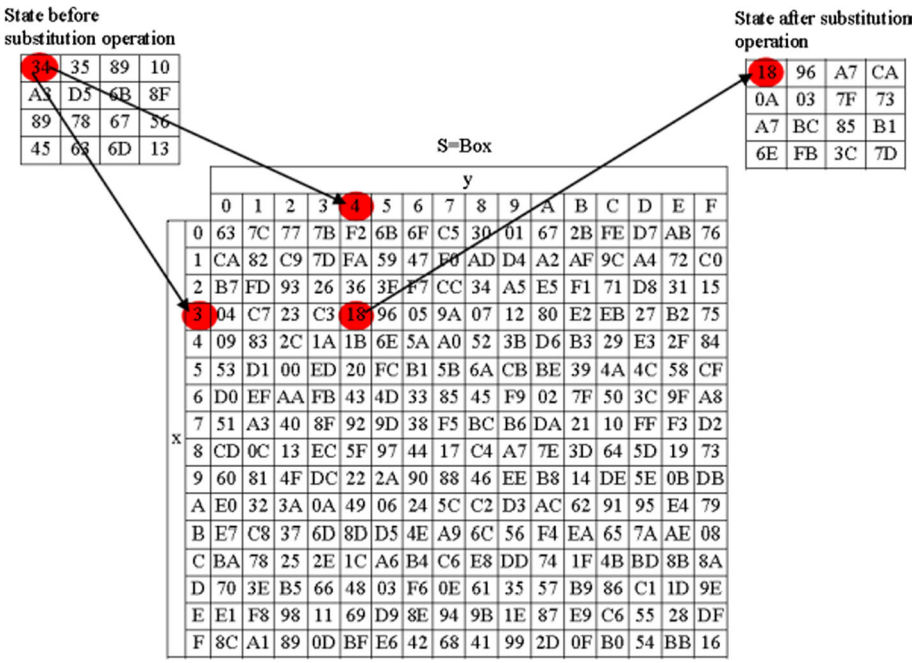


**Fig. 2** S-box matrix with explanation to substitution operation

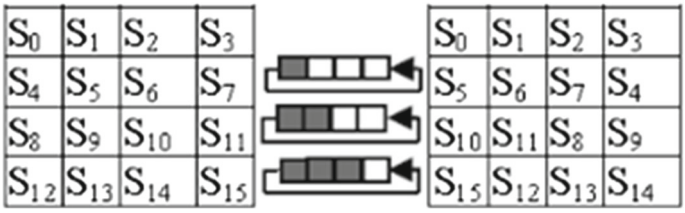


**Fig. 3** ShiftRow operation

*3. MixColumn Transformation* MixColumn transformation is applied on the state columns one by one. Byte value is changed based on values of all bytes in the column by multiplying the state in GF($2^8$), as in (4), using the polynomial shows in (3) [36]:

$$m(x) = x^8 + x^4 + x^3 + x + 1 \tag{3}$$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} \tag{4}$$

*4. Add Round Key Transformations* The last transform in the round is the AddRoundKey (ARK). ARK operation is done by carry out EX-OR logic operation between state array and subkey, see Fig. 4.

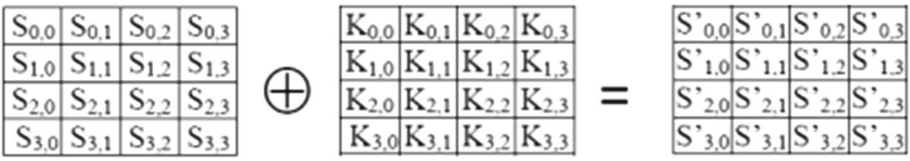


**Fig. 4** ShiftRow operation

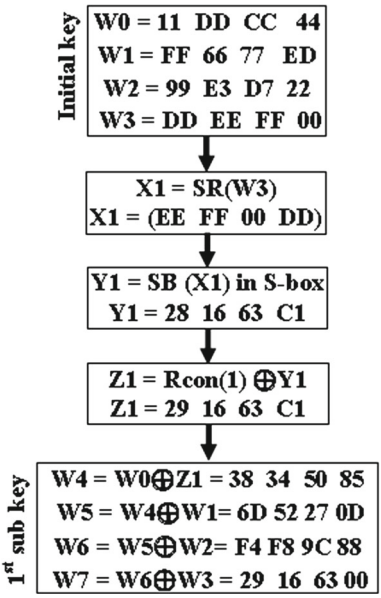


**Fig. 5** An example of key expansion operation

*2.1.2 AES Key Expansion*

This step uses a 16-byte initial key (4 words with 4 bytes or 32 bits for each word) and generate $Nb(Nr+1)$ words. Here, $Nb$ is the state columns number ($Nb = 4$), and $Nr$ is the rounds number ($Nr = 10$, 12, or 14). The number of words required in each round is equal to $Nb$. The key expansion operation include copy the key into the 1st set of four words and then composition subsets of four words, where each word depends on the values of the former set. The 1st word in each subkey specifically deals with SR + SB + Rcon of the former word before XORing it with the first word from the subsequent subkey [12]. Figure 5 shows an example of the key expansion operation for the first subkey.

## 3 Proposed Modifications

A significant challenge in the application of encryption algorithms for images is the encryption/decryption time because images, especially HD images, have large amount of information. Attacks are another trouble of all encryption algorithms. So, should be modifying the ciphering systems carefully such that they do not provide any hiatus to the attackers. One of famous types of attacker is the reduced-round attacker. The AES is a block encryption scheme has set of transformations executed several times (in iterations). Theoretically, the

reduced-round adversary can break the AES algorithm in $2^{120}$ iterations if reduced its rounds to 7 [34]. Therefore, decreasing the rounds number of the AES algorithm makes it weak against this type of attacker. ARK is one of the AES iteration transforms, where the EX-OR operation is conducted between subkey derived from the former subkey in the key schedule operation and state. This reason prevents us from reducing the number of AES rounds. The goal of our paper is to propose a high-speed and highly restricted ciphering algorithm for HD image encryption. Some modifications to the AES algorithm, such as decreasing the computation costs by decreasing the execution time of the MixColumn transformation and replacing the S-box with a new simple S-box, are proposed to make the algorithm compatible with our goal. In addition, increasing the security level through enhances the key schedule operation. The details of the suggested modifications are discussed in the following subsections.

### 3.1 MixColumn Reduction

As mentioned in Sect. 2.1, MixColumn step has a large amount of calculations contrast with other steps of AES (substitution, shifting, and ARK). So, we proposed decreasing the MixColumn execution times from 10 to 5, which will decrease computations in the reduction factor (RF) calculated using (5):

$$RF = 4 \times (MCC) \times (IP/16) \tag{5}$$

where $RF$ is the reducing factor, $MCC$ is the number of mathematical operations in the Mixcolumn steps, and $IP$ is the number of image pixels.

### 3.2 Key Schedule Modification

The key schedule operation has weaknesses because of the direct relationships between subkey bytes results from the key schedule operation, and attackers can be used in the AES round to break the key [34] and [35]. As an example, the adversary is able to determine $K_{13}^0$ (the 13th byte in the secret key) by using (6) (here, we note that the adversary should be a known block of plaintext and ciphertext) [36], as follows:

$$K_{13}^0 = ARK(PT_{13}; (SBI(SRI(ARK(CT_{13}; K_{13}^1))))) \tag{6}$$

Now, we can determine $K_9^1$ by using (7) as below:

$$K_{13}^1 = K_{13}^0 \oplus K_9^1 \tag{7}$$

We can then use similar equations to easily determine key bytes. MixColumn transformation creates a problem from the adversary because it deals with a full column (4 bytes in one time). Therefore, we proposed the application of MixColumn operation on the subkey before using it in the ARK operation to make the operation more complicated. This step produces the subkey from the schedule, which is not the same subkey used in ARK. Thus, $K_{13}^0$ in (6) is not the same as $K_{13}^0$ in (7). As a result, the adversary cannot use these equations to calculate the key directly.

### 3.3 New S-box

Initial S-box is replaced with new and simple one suggested based on (8) below:

$$x(m, n) = D_1 \; D_2 \tag{8}$$

| | | y | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| x | 0 | 0F | 0E | 0D | 0C | 0B | 0A | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| | 1 | 1F | 1E | 1D | 1C | 1B | 1A | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| | 2 | 2F | 2E | 2D | 2C | 2B | 2A | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| | 3 | 3F | 3E | 3D | 3C | 3B | 3A | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 |
| | 4 | 4F | 4E | 4D | 4C | 4B | 4A | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 |
| | 5 | 5F | 5E | 5D | 5C | 5B | 5A | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 |
| | 6 | 6F | 6E | 6D | 6C | 6B | 6A | 69 | 68 | 67 | 66 | 65 | 64 | 63 | 62 | 61 | 60 |
| | 7 | 7F | 7E | 7D | 7C | 7B | 7A | 79 | 78 | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 |
| | 8 | 8F | 8E | 8D | 8C | 8B | 8A | 89 | 88 | 87 | 86 | 85 | 84 | 83 | 82 | 81 | 80 |
| | 9 | 9F | 9E | 9D | 9C | 9B | 9A | 99 | 98 | 97 | 96 | 95 | 94 | 93 | 92 | 91 | 90 |
| | A | AF | AE | AD | AC | AB | AA | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| | B | BF | BE | BD | BC | BB | BA | B9 | B8 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| | C | CF | CE | CD | CC | CB | CA | C9 | C8 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |
| | D | DF | DE | DD | DC | DB | DA | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| | E | EF | EE | ED | EC | EB | EA | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| | F | FF | FE | FD | FC | FB | FA | F9 | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 | F0 |

**Fig. 6** S-box proposed

where $D_1 = m$ and $D_2 = F - n$. $x$ ($m$,$n$) is the value of element in new S-box with position determined by ($m$, $n$); $m$, $n$ and $F$ are hexadecimal numbers.

The new S-box array can be easily and simply constructed. Second advantage is through using one S-box for ciphering and deciphering operations (in SubByte and Inverse SubByte operations) instead of using two arrays as in original AES, to decrease hardware requirements. However, the proposed S-box has the disadvantage of low nonlinearity, which exacerbates the pattern appearance problem that decreases the use of ciphering modes. Figure 6 shows the proposed S-box.

## 4 Experimental Results

Ciphering modes are compared to determine which mode is more compatible with the proposed version of AES. The security of the new AES version is evaluated through visible scene, histogram analysis, entropy, and correlation of adjacent pixels. In addition to security analysis, the computation cost is compared between original and modified AES versions. The simulation time of encryption and decryption operations is also compared between two versions of the AES algorithm. Two types of HD images (red, green, and blue [RGB] and grayscale) were used as test images. MATLAB 2010b is used to simulate performance evaluation.

### 4.1 Security Analysis

#### *4.1.1 Visible Scene*

The perfect scene of the ciphered image is similar to the jamming scene in TV, which hides the smallest details. Two test HD images, one RGB and one grayscale, are used to evaluate the key schedule operation. The selected test images contained identical color regions to

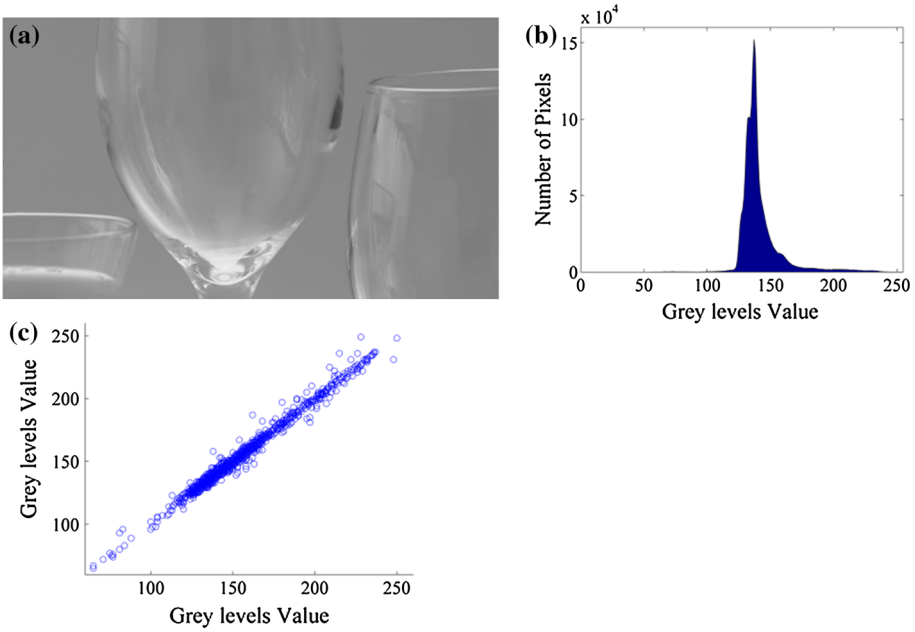


**Fig. 7** Test1 GS original image. **a** Visual scene, **b** histogram, **c** correlation coefficients

clarify the phenomenon of patterns, as shown in Figs. 7 and 12. Ciphered images under different ciphering modes are shown in Figs. 8, 9, 10, 11, 13, 14, 15 and 16 [37]. The AES version proposed in this paper satisfied the visual scene requirements when conducting the tests under cipher-block chaining (CBC) and cipher feedback (CFB) ciphering modes, as shown in Figs. 8, 9, 13 and 14.

### *4.1.2 Histogram Analysis*

The ciphered image histogram should be uniformly distributed to bar information detection by opponents, especially statistical attackers. In addition, the histogram of the encrypted image has low or no statistical identically to the histogram of the original image. Figures 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16 show the histogram of the original and ciphered images under different ciphering modes for two samples (one grayscale image and one RGB image). The results in Figs. 8, 9, 13 and 14 show that the histograms of the ciphered images by the modified AES version conducted using the CBC and CFB modes have uniform distribution, thus satisfying the good diffusion and confusion conditions [37].

### *4.1.3 Entropy*

The concept of entropy comes from information theory and ergodic theory. Shannon entropy is defined as a metric associated with information content of input signal. In image processes, entropy is known as the measurement to randomness, which can be exegesis as the average doubt of the information source. It is calculated from (9) [37], as follow:

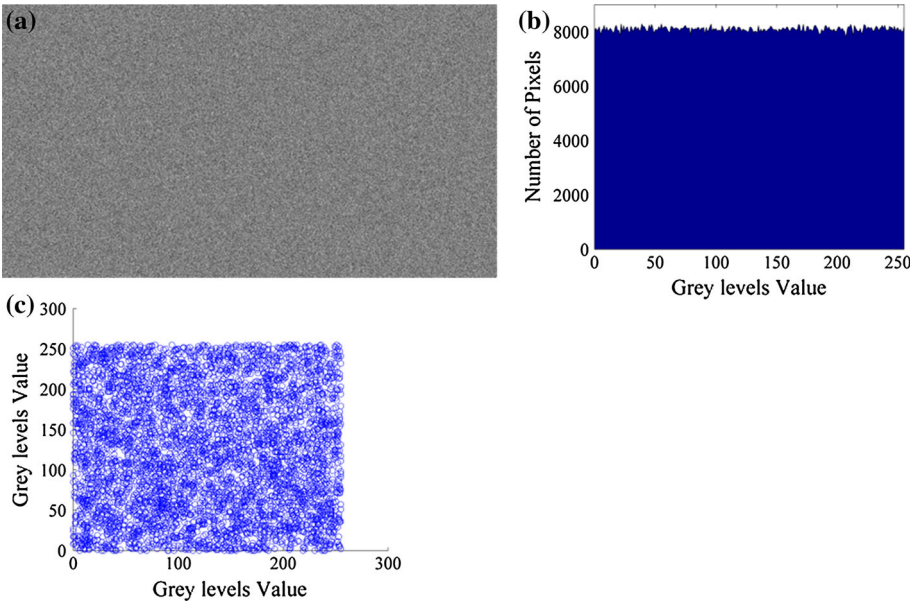


**Fig. 8** Test1 GS ciphered image CBC mode. **a** Visual scene, **b** histogram, **c** correlation coefficients

$$H(x) = \sum_{i=1}^{K} P(x_i) log_2 \frac{1}{P(x_i)} = -\sum_{i=1}^{K} P(x_i) log_2 P(x_i) \tag{9}$$

where the $P(x_i)$ is the probabilities of $x_i$.

Usually, grayscale image has 256 or $2^8$ grey levels. If the probabilities of the grey levels is equally, then by applying (9), entropy value must be equal to eight. The RGB image has three matrices (red, green, and blue), where each matrix has 256 gray levels similar to the grayscale image. The values of entropy for six samples (three grayscale images and three RGB images) are shown in Table 1. The results of Table 1 show that the entropy values for the modified AES algorithm executed using the CBC and CFB modes are near the ideal value of 8.

### *4.1.4 Correlation Coefficients Between Adjacent Pixels*

In natural images, ajusted pixels are highly correlated together. The best encryption scheme is a system that introduced an encrypted image with little correlation between the adjacent pixels. Equations (10)–(13) are calculate the correlation coefficients an image [11]:

$$E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{10}$$

$$D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2 \tag{11}$$

$$cov(x, y) = E[(x - E(x))(y - E(y))] \tag{12}$$

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{13}$$

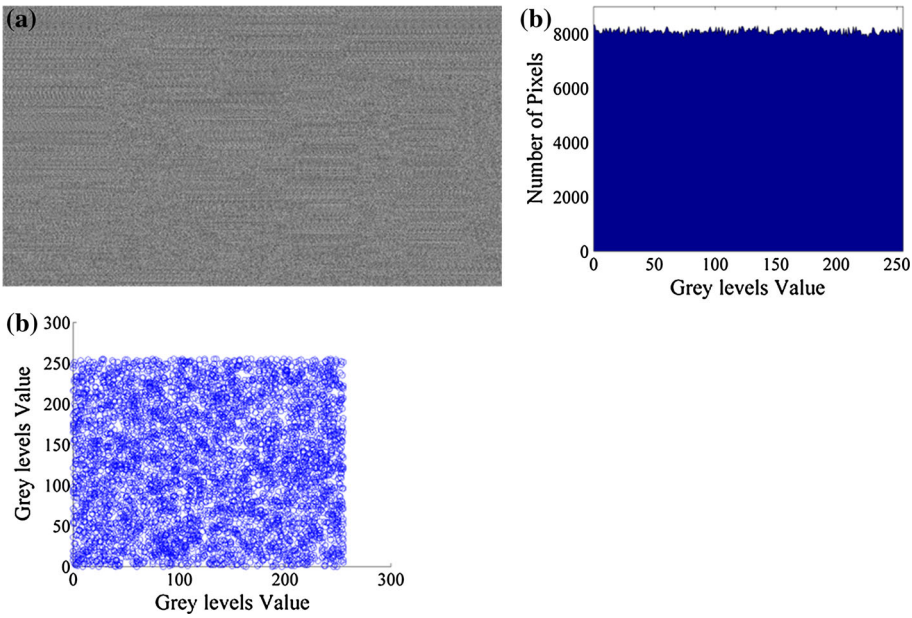


**Fig. 9** Test1 GS ciphered image CFB mode. **a** Visual scene, **b** histogram, **c** correlation coefficients

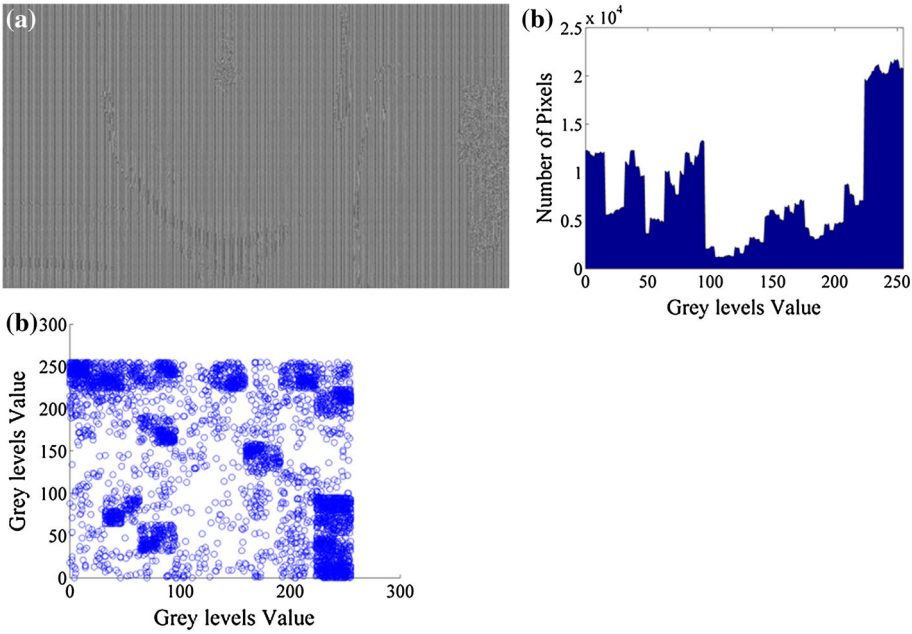


**Fig. 10** Test1 GS ciphered image ECB mode. **a** Visual scene, **b** histogram, **c** correlation coefficients

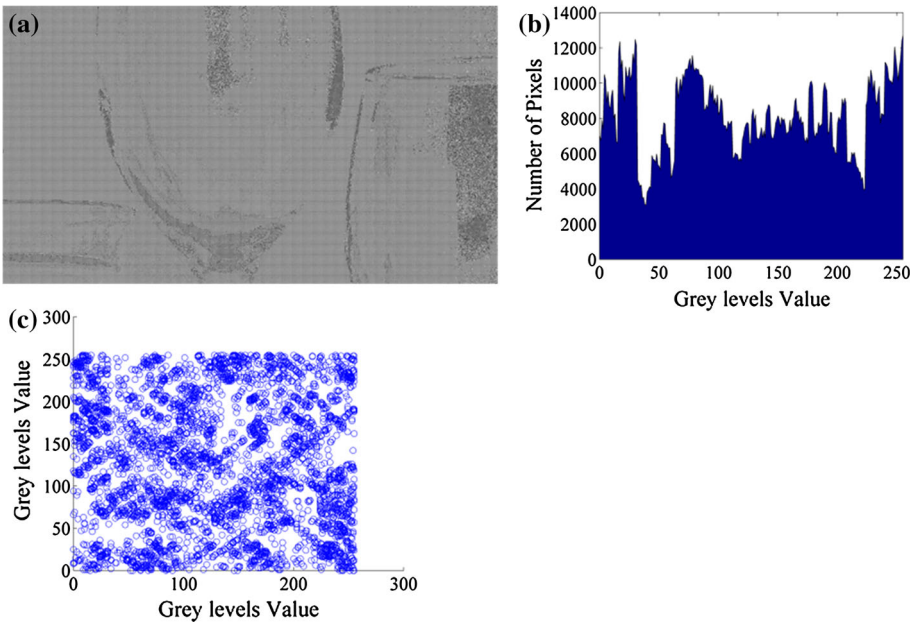


**Fig. 11** Test1 GS ciphered image OFB mode. **a** Visual scene, **b** histogram, **c** correlation coefficients

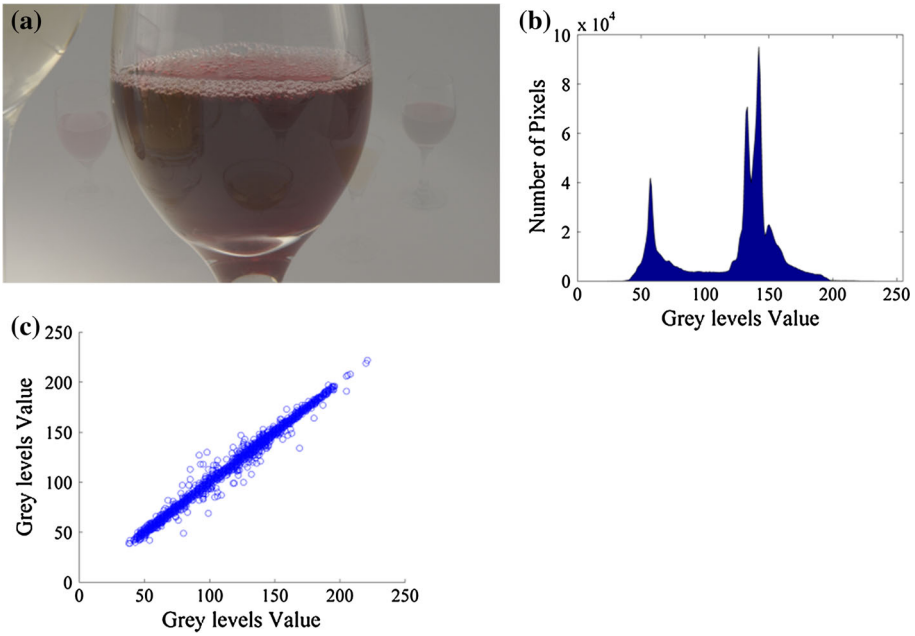


**Fig. 12** Test2 RGB original image. **a** Visual scene, **b** histogram, **c** correlation coefficients

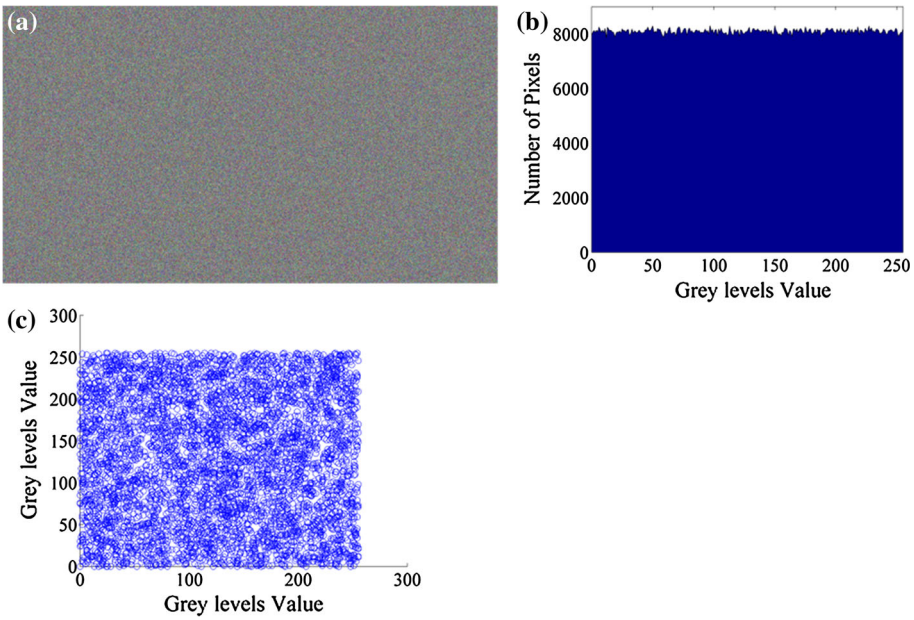


**Fig. 13** Test2 RGB ciphered image CBC mode. **a** Visual scene, **b** histogram, **c** correlation coefficients

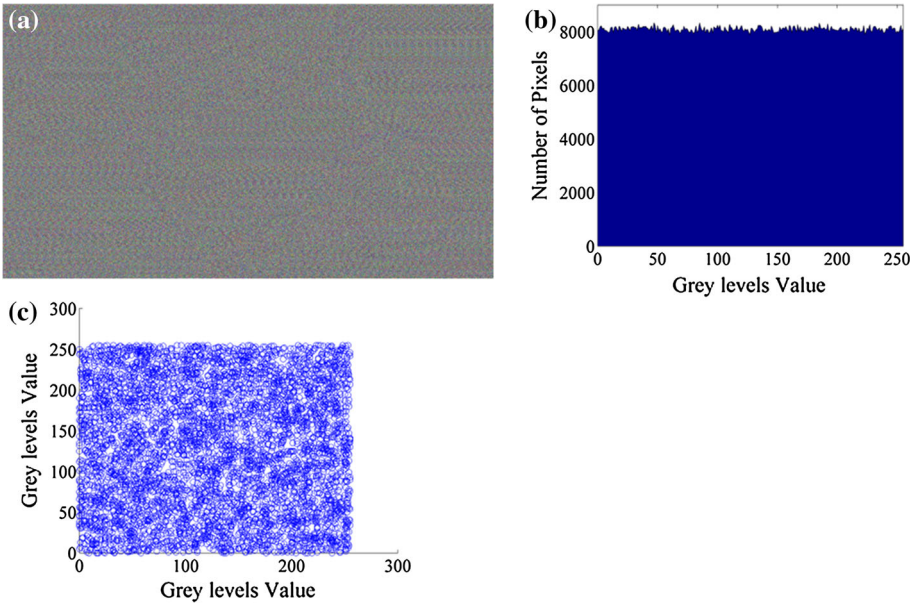


**Fig. 14** Test2 RGB ciphered image CFB mode. **a** Visual scene, **b** histogram, **c** correlation coefficients

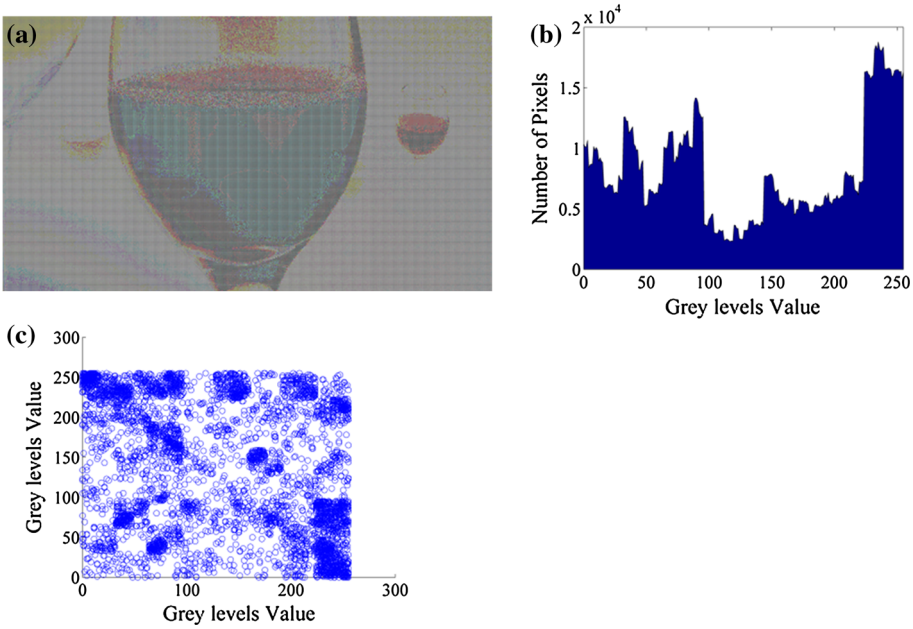


**Fig. 15** Test2 RGB ciphered image ECB mode. **a** Visual scene, **b** histogram, **c** correlation coefficients

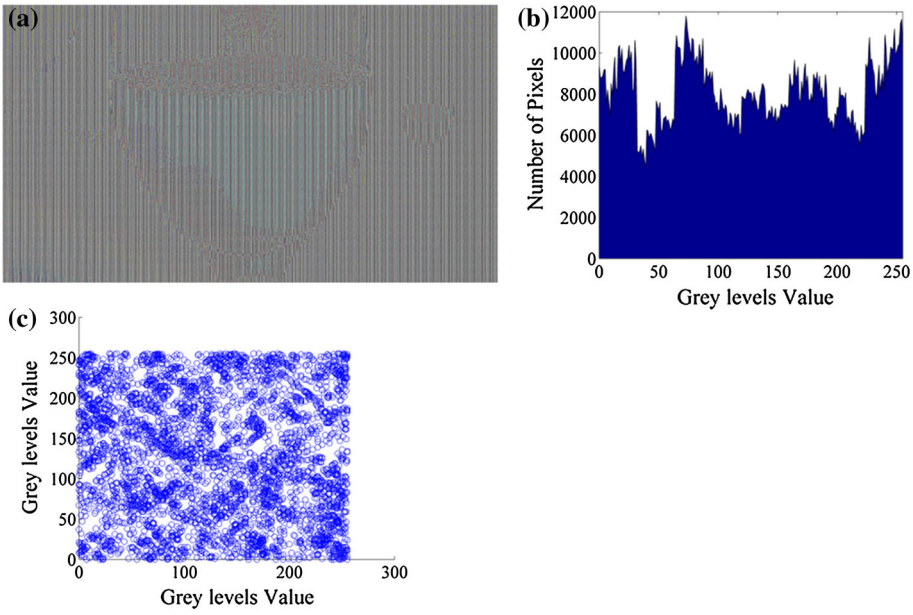


**Fig. 16** Test2 RGB ciphered image OFB mode. **a** Visual scene, **b** histogram, **c** correlation coefficients

**Table 1** Entropy values of 6 tests

| Ciphering modes | S1 GSI | S2 GSI | S3 GSI | S4 RGBI | S5 RGBI | S6 RGBI |
|---|---|---|---|---|---|---|
| OI | 7.3856 | 7.3559 | 5.3519 | 7.5343 | 6.3389 | 6.4562 |
| ECB mode | 7.9921 | 7.9941 | 7.8333 | 7.9964 | 7.9981 | 7.9651 |
| OFB mode | 7.9976 | 7.9947 | 7.9598 | 7.9975 | 7.9961 | 7.9833 |
| CBC mode | **7.9999** | **7.9999** | **7.9999** | **7.9999** | **7.9999** | **7.9999** |
| CFB mode | **7.9999** | **7.9999** | **7.9999** | **7.9999** | **7.9999** | **7.9999** |

Bold values indicate the best values in table compared with other values in same table

**Table 2** Correlation coefficients of adjacent pixels of 6 tests

| Ciph. modes | S1 GSI | S2 GSI | S3 GSI | S4 RGBI | S5 RGBI | S6 RGBI |
|---|---|---|---|---|---|---|
| OI | 0.9971 | 0.9952 | 0.5902 | 0.9990 | 0.9627 | 0.9990 |
| ECB mode | 0.0362 | 0.0582 | −0.0372 | 0.0364 | 0.0373 | 0.0092 |
| OFB mode | −0.0302 | −0.0388 | −0.0727 | −0.0274 | −0.0543 | −0.1039 |
| CBC mode | **0.0208** | **0.0021** | **0.00012** | **−0.0283** | **0.0239** | **−0.0051** |
| CFB mode | 0.0219 | 0.0029 | −0.0164 | −0.0134 | −0.0104 | −0.0154 |

Bold values indicate the best values in table compared with other values in same table

where $x$ and $y$ are the values of the gray levels in images, and $N$ is the total number of samples.

First, 5,000 pairs of adjacent pixels were selected randomly from original and ciphered images in the horizontal direction only (for the RGB image, the pairs were taken from the green matrix only). Figures 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16 show the correlation coefficients of adjacent pixels of two test images in different ciphering modes. These figures showed that the correlation between adjacent pixels in encrypted images is decreased compared with the strong correlation in the plain image. The measured correlation coefficients of the plain and ciphered images for six tests (three grayscale images and three RGB images) are listed in Table 2. We verified that the values of the correlation coefficients are low, especially when we applied the modified algorithm using the CBC and CFB ciphering modes, compared with ECB and output feedback. This low correlation coefficient strengthens the algorithm against various attacks. The results discussed in this subsection clearly showed that the modified AES version performed well and satisfied most of the encryption requirements, especially when applied under the CBC ciphering mode.

### 4.2 Comparisons of Computations and Execution Time

The main purpose of the modifications to the AES algorithm is to decrease the encryption time and then make it suitable for image encryption, especially in HD images. In this subsection, the comparison between original and modified AES versions will be conducted based on the required computations and execution time. Table 3 shows the reduction achieved in the addition and multiplication operations only because these operations exist only in the MixColumn transformation.

The percentage reduction in computation using the modified version is approximately (29 %). The execution time required for the original and modified versions of the AES algorithm for different ciphering modes are shown in Table 4. We observed that the best execution

**Table 3** Computations cost comparison between initial and modified AES versions

| Transformation | Computation operation | | | | |
|---|---|---|---|---|---|
| | Multiplication | Addition | X-OR | Shifting | Substitution |
| *Initial AES algorithm* | | | | | |
| S-box gen. | 2,048 | 2,048 | – | – | – |
| Key expansion | **10** | – | 216 | 10 | 40 |
| One block ciphering | 576 | 144 | 176 | 60 | 160 |
| HD GS image ciph. | 74,651,658 | 18,666,496 | 22,809,600 | 7,776,000 | 20,736,056 |
| HD RGB image ciph. | 223,950,858 | 55,995,392 | 68,428,800 | 23,328,000 | 62,208,168 |
| *Modified AES algorithm* | | | | | |
| S-box gen. | – | **256** | – | – | – |
| Key expansion | 650 | 176 | 216 | 10 | 40 |
| One block ciphering | **320** | **80** | 176 | 60 | 160 |
| HD GS image ciph. | **41,472,650** | **10,368,432** | 22,809,600 | 7,776,000 | 20,736,056 |
| HD RGB image ciph. | **124,416,650** | **31,104,432** | 68,428,800 | 23,328,000 | 62,208,168 |

Bold values indicate the best values in table compared with other values in same table

**Table 4** Execution time comparison between initial and modified AES (in seconds) under different ciphering mode

| Ciph. modes | Initial AES version | | Modified AES version | |
|---|---|---|---|---|
| | GS image | RGB image | GS image | RGB image |
| ECB mode | 1,186 | 3,439 | 704 | 2,031 |
| CBC mode | **1,219** | **3,534** | **704** | **2,029** |
| CFB mode | 1,256 | 3,660 | 712 | 2,050 |
| OFB mode | 1,287 | 3,732 | 709 | 2,043 |

Bold values indicate the best values in table compared with other values in same table

time was achieved using the CBC mode for the original and modified versions. Simulation is achieved using MATLAB version 2010b on a laptop with the following specifications: HP Pavilion g4, Intel Core i5-2340M @ 2.40 GHz (4 CPUs) processor, 2.4 GHz, 8 GB random access memory, Windows 7 Home Premium 64-bit (6.1, Build 7601). As shown in Table 4, the execution time of the modified AES is decreased by approximately (35 %) of the original AES.

## 5 Conclusions

Some attempts have been made to modify the AES algorithm. However, all of these modifications did not focus on the execution time. Therefore, the original and previously modified versions of the AES algorithm take more time with image encryption, especially HD images. The largest number of computations in the AES algorithm was decreased in the MC transformation. Therefore, the first modification was based on the reduction of the execution times of the MC transformation from 10 to 5 (executed only in the first, third, fifth, seventh, and ninth rounds). Therefore, the computation costs and the encryption/decryption time will be

decreased, which makes the modified AES more compatible with image ciphering, especially HD images.

Another problem of the AES algorithm, which was not processed effectively in previous works, was weakness in key schedule operation against reduced-round attackers. This impairment resulted from the direct relationship between subkey bytes and cipher text in one round, as discussed in Sect. 3.2. To resolve this problem, we proposed to conduct the MC transformation in the key schedule because it deals with one column completely and then breaks the relationship between subkey bytes and cipher text, thus restricting the reduced-round attackers.

In addition, to decrease the hardware requirements and computation costs, a new and simple S-box was proposed as an enhancement of the original S-box. The proposed S-box can be used itself for the encryption and decryption operations. By contrast, the original AES needs two S-box matrices, one for the encryption operation (S-box) and the other for the decryption operation (Inv. S-box). A new version of the AES algorithm is applied using the CBC ciphering mode to fix the pattern appearance problem.

The results of some factors, such as entropy, correlations between adjacent pixels, and histogram distribution, proved that the AES with the proposed modifications is strong against most attacks. Table 3 shows that the computational complexity of AES is decreased by approximately (29 %) in the proposed version, which makes it more favorable than the original AES for HD image ciphering.

**Acknowledgments** The authors would like to thank the editor-in-chief and anonymous reviewers for helpful comments and suggestions that improved the quality and readability of the paper. The authors would also like to thank Universiti Kebangsaan Malaysia for supporting this work under UKM-GUP-2011-060 grant funds. Also the corresponding author would like to thanks the Foundation of Technical Education-Baghdad for supporting him by 7-17-20066 Grant scholarship.

## References

1. Jong-Wook, H., Choon-Sik, P., Dae-Hyun, R., & Eun-Soo, K. (1999). Optical image encryption based on XOR operations. *Optical Engineering*, *38*(1), 47–54.
2. Daesung, M., Yongwha, C., Sung, P., Kiyoung, M., & Kyo, C. (2006). An efficient selective encryption of fingerprint images for embedded processors. *ETRI Journal*, *28*(4), 444–452.
3. Shahram, B., & Mohammad, E. (2013). Chaotic image encryption system using phase-magnitude transformation and pixel substitution. *Telecommunication Systems*, *52*(2), 525–537.
4. Nanrun, Z., Yixian, W., & Lihua, G. (2011). Novel optical image encryption scheme based on fractional Mellin transform. *Optics Communications*, *284*(13), 3234–3242.
5. Fan, G., Linfei, C., & Daomu, Z. (2008). A half-blind color image hiding and encryption method in fractional Fourier domains. *Optics Communications*, *281*(17), 4254–4260.
6. Shahram, B. (2000). Speech encryption based on fast Fourier transform permutation. In *7th IEEE international conference on electronics, circuits and systems*. Jounieh, Lebanon.
7. Zhengjun, L., Lie, X., Ting, L., Hang, C., Pengfei, L., Chuang, L., et al. (2011). Color image encryption by using Arnold transform and color-blend operation in discrete cosine transform domains. *Optics Communications*, *284*(1), 123–128.
8. Hsuan, C., Hone-Ene, H., & Cheng-Ling, L. (2011). Position multiplexing multiple-image encryption using cascaded phase-only masks in Fresnel transform domain. *Optics Communications*, *284*(18), 4146–4151.
9. Hwang, H. (2011). An optical image cryptosystem based on Hartley transform in the Fresnel transform domain. *Optics Communications*, *284*(13), 3243–3247.
10. Akhshani, A., Behnia, S., Akhava, A., Abu Hassan, H., & Hassan, Z. (2010). A novel scheme for image encryption based on 2D piecewise chaotic maps. *Optics Communications*, *283*(17), 3259–3266.
11. Fu, C., Lin, B., Miao, Y., Xiao, L., & Jun-jie, C. (2011). A novel chaos-based bit-level permutation scheme for digital image encryption. *Optics Communications*, *284*(23), 5415–5423.

12. Liu, H., & Wang, X. (2011). Color image encryption using spatial bit-level permutation and high-dimension chaotic system. *Optics Communications*, *284*(16–17), 3895–3903.
13. Akhavan, A., Samsudin, A., & Akhshani, A. (2011). A symmetric image encryption scheme based on combination of nonlinear chaotic maps. *Journal of the Franklin Institute*, *348*(8), 1797–1813.
14. Ruisong, Y. (2011). A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism. *Optics Communications*, *284*(22), 5290–5298.
15. Zhi-liang, Z., Wei, Z., Kwok-wo, W., & Hai, Y. (2011). A chaos-based symmetric image encryption scheme using a bit-level permutation. *Information Sciences*, *181*(6), 1171–1186.
16. Alexander, P., & Massimiliano, Z. (2012). Chaotic map cryptography and security. In *Encryption: methods, software and security* (pp. 301–332). Nova Science Publishers.
17. Jiancheng, Z., Rabab, W., & Dongxu, Q. (2004). A new digital image scrambling method based on Fibonacci numbers. In *International conference on circuits and systems*, Vancouver, Canada.
18. Jiancheng, Z., Rabab, W., & Dongxu, Q. (2004). The generalized Fibonacci transformations and application to image scrambling. In *IEEE international conference on acoustics, speech, and signal processing*, Montreal, Canada.
19. Linfei, C., Daomu, Z., & Fan, G. (2013). Image encryption based on singular value decomposition and Arnold transform in fractional domain. *Optics Communications*, *291*, 98–103.
20. Qiudong, S., Wenying, Y., Jiangwei, H., & Wenxin, M. (2012). Image encryption based on bit-plane decomposition and random scrambling. In *2nd International conference on consumer electronics, communications and networks*, Hubei, China.
21. Yicong, Z., Karen, P., & Sos, A. (2009). Image encryption algorithms based on generalized P-Gray code bit plane decomposition. In *Conference record of the forty-third Asilomar conference on signals, systems and computers*, CA, USA.
22. Zheng, W., Cheng, Z., & Cui, Y. (2008). Image data encryption and hiding based on wavelet packet transform and bit planes decomposition. In *4th International conference on wireless communications, networking and mobile computing*, Dalian, China.
23. Nandi, S., Kar, B., & Chaudhuri, P. (1994). Theory and applications of cellular automata in cryptography. *IEEE Transactions on Computer*, *43*(12), 1346–1357.
24. FIPS PUB 46-3: Data encryption standard (DES), 1999.
25. Daemen, J., & Rijmen, V. (2000). The block cipher Rijndael. In: *Smart card research and applications. Lecture notes in computer science* (pp. 277–284). Berlin: Springer.
26. Kamali, S., Shakerian, R., Hedayati, M., & Rahmani, M. (2010). A new modified version of advanced encryption standard based algorithm for image encryption. In *International conference on electronics and information engineering*, Kyoto, Japan.
27. Grangetto, M., Magli, E., & Olmo, G. (2006). Multimedia selective encryption by means of randomized arithmetic coding. *IEEE Transactions on Multimedia*, *8*(5), 905–917.
28. Wadi, S., & Zainal, N. (2013). A low cost implementation of modified advanced encryption standard algorithm using 8085A microprocessor. *Journal of Engineering Science and Technology*, *8*(4), 406–415.
29. Huang C. W., Tu Y. H., Yeh H. C., Liu S. H., & Chang C. J. (2011). Image observation on the modified ECB operations in Advanced Encryption Standard. In *International conference in information society (i-Society)*, London, UK.
30. Subramanyan, B., Chhabria, V. M., & Sankarbabu, T. G. (2011). Image encryption based on AES key expansion. In *IEEE computer society meeting*, Kolkata, India.
31. Fahad, M. (2013). Chaotic and AES cryptosystem for satellite imagery. *Telecommunication Systems*, *52*(2), 573–581.
32. Tran M. T., Bui D. K., & Duong A. D. (2008). Gray S-box for advanced encryption standard. In *IEEE computer society meeting*, Suzhou, China.
33. Yicheng, C., Xuecheng, Z., Zhenglin, L., Xiaofei, C., & Yu, H. (2008). Dynamic inhomogeneous S-Boxes design for efficient AES masking mechanisms. *The Journal of China Universities of Posts and Telecommunications*, *15*(2), 72–76.
34. Bouillaguet, C., Derbez, P., Dunkelman, O., Fouque, P., Keller, N., & Rijmen, V. (2012). Low-data complexity attacks on AES. *IEEE Transaction on Information Theory*, *58*(11), 7002–7017.
35. Dunkelman, O., & Keller, N. (2010). The effects of the omission of last rounds mixcolumns on AES. *Information Processing Letters*, *110*(8–9), 304308.
36. Announcing the Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, 26 of November, 2001.
37. Gonzalez, R., Woods, E., & Eddins, L. (2008). *Digital image processing using Matlab*. NJ: Gatesmark LLC.

**Salim Muhsin Wadi** was born in Najaf, Iraq, on April 26, 1980. He received the B.Sc. degree in Communication Techniques Engineering from Technical College, Najaf, Iraq, in 2002. He received the M.Sc. degree in Communication Engineering from University of Technology, Baghdad, Iraq, in 2005. He is currently a Ph.D. student in Electrical Electronic & System Eng., Faculty of Engineering and Built Environment, National University of Malaysia UKM. His main research interests are Image processing, Encryption and Steganography, and Image Enhancement.

**Nasharuddin Zainal** was born in Kuala Lumpur, Malaysia, on September 12, 1974. He received the B.E. degree from Tokyo Institute of Technology in 1998, M.E. degree from The National University of Malaysia in 2003 and the Ph.D. degree from Tokyo Institute of Technology in 2010. He is also member of IEEE, corporate member of The Institution of Engineers Malaysia and certified professional Engineer of Board of Engineers Malaysia. His researches are on image and video processing, pattern recognition and robotics.