An enhancement crypto-compression scheme for image Based on chaotic system

Manel Dridi  
National Engineering School of Monastir, University of Monastir, Tunisia.  
Laboratory of Electronic and Microelectronic, University of Monastir, Tunisia.

Belgacem Bouallegue  
Laboratory of Electronic and Microelectronic, University of Monastir, Tunisia.

Mohamed Ali Hajjaji  
Higher Institute of Applied Sciences and Technology of Kasserine, University of Kairouan, Tunisia.  
Laboratory of Electronic and Microelectronic, University of Monastir, Tunisia.

Abdellatif Mtibaa  
National Engineering School of Monastir, University of Monastir, Tunisia.  
Laboratory of Electronic and Microelectronic, University of Monastir, Tunisia.

Abstract
The largest growth rate of technology requires a new method of storing and transmitting data or information. These methods may guarantee an important ratio of security and storage. So it is necessary to compress and encrypt data (image, text, register) before its transmission or archiving. In this context, this work presents an enhanced scheme which combines between compression and encryption. Concerning the encryption process, it is based on AES and chaotic system (Henon map and Arnold cat map). However, the Advanced Encryption Standard (AES) is used to encrypt the DC coefficients and the Arnold cat map is used for shuffling. Finally, to show the efficiency of this proposed scheme in experimental results are presented (PSNR, UACI and NPCR).

Keywords: Encryption images, DCT transform, Chaos permutation, Arnold cat map, AES, AES_SBOX.

Introduction
In order to overcome the image/data encryption problems, a lot of cryptographic schemes for images have been proposed and/or developed. In the recent years cryptography based on chaotic system has increasingly attracted the researchers’ interest, so a lot of schemes based on a chaos image encryption have been developed. In what follows, we present in brief some interesting works. Actually, some researchers are interested in combining pixel scrambling and symmetric encryption.

Guanrong Chen et al [5], proposed in their paper a new symmetric image encryption scheme based on 3D chaotic cat maps. Fahad T.BinMuhaya in [1] proposed a secure image encryption based on chaotic map (Henon & Arnold) and AES. Where the Arnold’s cat map was used to shuffle an image and the Chaotic Henon map is used to generate a key. Xiang et al. proposed, in [2], a selective image encryption scheme: only a part of significant bits of each pixel were encrypted using the key-stream generated based on a map lattice. In the article of Chong Fu et al [3], they proposed a novel encryption technique of medical images based on chaotic systems: Arnold cat map and logistic map. Where the Arnold cat map was used to switch pixels and logistic map to substitute pixels. Furthermore Jolfaei and Mirghadri, [4] have proposed a novel encryption scheme based on the Baker map and modified AES. Basically, the Baker map was used to shuffle pixels.

In fact, the use of chaotic systems for image encryption provides resistance against statistical and correlations attacks, that is why, we will propose a scheme based on the chaotic system. Also, we will be interested in this work, in combining both compression and encryption processes, as some researches [15-16-17] have done. In [15] Lian proposed a new algorithm based on Discrete Cosine Transformation (DCT), in which the DC coefficient were encrypted and the sign bit of all AC coefficients were modified using a spatiotemporal chaos system. In [16] Ching et al. also used the DCT transformation with hash function (Secure Hash Algorithm-1: SHA-1) and the Tent map to compress and encrypt an image.

This paper is organized as follows. In the next section, we define the tool used. In the third section we elaborate the proposed scheme. Then, we present some of the treated examples and tests followed by some comparison with respect to other works. Finally, we conclude the paper.

Background Study
In our proposed images compression and encryption scheme, we use the JPEG as a compression technique and the AES with chaotic system as an encryption technique. Then in this manuscript, we will focus on the JPEG compression and on two different encryption approaches. The first one is the selective encryption while the second is the chaotic encryption.
Compressions which do not conserve the quality of the image, transmission. There are two types of compression: lossy compressions which do not conserve the quality of the image, and lossless compression which are totally reversible and doesn’t cause any visual problem. To evaluate a compression string we should consider the complexity of the application (the speed of the encoder and decoder), the compression ratio, and the quality of reconstructed image. The proposed context (the speed of the encoder and decoder), the compression ratio, and the quality of reconstructed image. The proposed context is based on DCT. Usually the steps required for compression are the DCT transform, thresholding, quantization and encoding.

AES (Advanced Encryption Standard) algorithm
The AES[20] is a symmetric cryptographic algorithm, widely used according to its high security. The encryption processing is divided into two principle steps. The initial step is the key generation (pseudo-random generators, A5 / 1, W7, etc) and the second step is to apply the encryption function which is divided into three main parts: an initial transformation, a series of rounds and final processing.

What is more each round is divided into four different transformations(fig 1):

i) SubBytes: Replace each byte in the table state by its corresponding value in the S-Box
ii) ShiftRows: The lines are shifted cyclically with different offsets.
iii) MixColumns: multiply to a constant matrices with a state matrix (mixture of columns)
iv) Add round key: Apply XOR to the round key with the state.

In this work, we use a chaotic system, the Henon map, to modify the secret key after encrypting each bloc of 16 elements of the selected vector. Then in order to have a new key, we modify the initial value \((x_0, y_0)\) for every process.

The proposed encryption process uses a secret key of a length of \(16 \times n (n = [(LV) / 16])\) where LV is the length of selected vector (TC). Furthermore, the key is divided into blocks of 16. \(K = K_1, K_2, \ldots, K_n\).

The Henon map transforms a point \((x_n, y_n)\) in the plane to a new point \((x_{n+1}, y_{n+1})\) which is defined by:

\[
x_{i+1} = 1 - a \times x_i^2 + y_i \tag{1}
\]

\[
y_{i+1} = b \times x_i \tag{2}
\]

where: \(a=1.4; b=0.3; x_0=0.02; y_0=0.02;\)

The single secret key \(K_i\) is defined as follows. First, we generate the chaotic sequence \(x_i\) through eq (1) then we amplify a different value by a scaling factor (10), \(x_i\rightarrow (10 \times x_i)\). Next, we determine the maximum value \((v_{max}=\max(x_i))\). Finally, we determine \(x_{r1}=x_1-v_{max}\) and we round the value \(x_{r1}\).

In this work, we use a chaotic system, the Henon map, to modify the secret key after encrypting each bloc of 16 elements of the selected vector. Then in order to have a new key, we modify the initial value \((x_0, y_0)\) for every process.

The proposed encryption process uses a secret key of a length of \(16 \times n (n = [(LV) / 16])\) where LV is the length of selected vector (TC). Furthermore, the key is divided into blocks of 16. \(K = K_1, K_2, \ldots, K_n\).

The Henon map transforms a point \((x_n, y_n)\) in the plane to a new point \((x_{n+1}, y_{n+1})\) which is defined by:

\[
x_{i+1} = 1 - a \times x_i^2 + y_i \tag{1}
\]

\[
y_{i+1} = b \times x_i \tag{2}
\]

where: \(a=1.4; b=0.3; x_0=0.02; y_0=0.02;\)

The single secret key \(K_i\) is defined as follows. First, we generate the chaotic sequence \(x_i\) through eq (1) then we amplify a different value by a scaling factor (10), \(x_i\rightarrow (10 \times x_i)\). Next, we determine the maximum value \((v_{max}=\max(x_i))\). Finally, we determine \(x_{r1}=x_1-v_{max}\) and we round the value \(x_{r1}\). Therefore, \(K = x_{r1}, x_{r2}, \ldots, x_{r16}\).

Permutation using Arnold cat map
In the permutation stage the pixel positions are rearranged so as to have a scrambled image while the values of pixels are unchanged. For this step, some authors usually employ the baker map, the Arnold cat map or the standard map [7, 13]. We have used the Arnold cat map because it is invertible. This map is a two-dimensional chaotic map which was first discovered in 1978. Its defined as follows:

\[
\begin{bmatrix}
x_{n+1}
\end{bmatrix} = \begin{bmatrix}
1 & c \\
-d & cd + 1
\end{bmatrix}
\begin{bmatrix}
x_n
\end{bmatrix} \mod 1
\tag{3}
\]

where \(c\) and \(d\) present the control parameters. To use this map for permutation, we need to modify the generalized map, while keeping some of its useful specificities such as the mixing property and the sensitivity to initial parameters and conditions. The cat map which allows the mixing of pixels is presented as follow:

\[
\begin{bmatrix}
x_{n+1}
\end{bmatrix} = \begin{bmatrix}
1 & c \\
-d & cd + 1
\end{bmatrix}
\begin{bmatrix}
x_n
\end{bmatrix} \mod N, \tag{4}
\]

where \(N\) is the number of pixels in one column(row), \((x_n, y_n)\) is an actual pixel values for the image at a specified position and \((x_{n+1}, y_{n+1})\) is the remplaced pixel value for the image. Inversely, it is easy to reproduce the image before shuffling by applying the inverse transform given by:

\[
\begin{bmatrix}
x_{n+1}
\end{bmatrix} = \begin{bmatrix}
d & -c \\
1 & -d
\end{bmatrix}
\begin{bmatrix}
x_n
\end{bmatrix} \mod N, \tag{5}
\]

Proposed scheme
The objective of this work is to propose a new and highly secure image crypto-compression scheme. The proposed scheme is divided in two levels of security: the first one is

Figure 1: AES encryption process

Key generation based on Henon map
The generation of the key is one of the important step in an encryption scheme. Indeed, if we use a not right key or a small key, the encryption system is easily broken even if the encryption algorithm is designed to be performant. Typically we choose to generate a key based on a chaotic system due to its characteristic of sensivity to initial conditions and control parameters and to ergodicity [14].
after the DCT transform to modify the DC coefficients and the second is after the Run Length Encoding (RLE) coding to make the cipher more robust against any attack and make the execution time shorter. Fig 3 illustrate different followed steps.

Encryption algorithm
Different steps to encrypt an image are represented as follows:

**Step 1:** Partition the image into 8x8 blocks.

**Step 2:** DCT transform: XDCT.

**Step 3:** Quantization: XDCTQ.

**Step 4:** Select the DC coefficients then keep them in a TC vector.

**Step 5:** Generate and encrypt the selected vector TC with AES scheme and the secret key (fig 2) produced by Henon map see section (II.3).

**Step 6:** Remit the encryption vector in XDCTQ.

**Step 7:** Apply zigzag and RLE coding. The RLE coding reduces the extends of data (if the same data value occurs in many consecutive data elements), where the data will be stored as a single data value followed by a count (value, counter). For example the value 22111153 become the pairs (2,2), (1,4), (5, 1), (3, 1).

**Step 8:** Use the Sbox generated in the encryption AES scheme after swapping the position of the elements matrices to modify the value of the resulting matrix from step 6. In this step, we use the Arnold cat for shuffling the position of the SBOX element.

**Step 9:** Shuffling the counter position in the RLE vector with the Arnold cat map: If we have a non-square matrix of MxN size, we apply the following steps. First, convert the matrix in vector then divided the vector into two vectors in the first one we store L*L elements which L=floor(√M × N) (floor (x) returns the nearest integers lower than or equal to x). Then in a second matrix we store the rest of elements. After that we reshape the first vector in order to have a square matrix with size L. Finally, the square matrix that we get is shuffled by using the Arnold cat map.

**Step 10:** Follow the compression steps: Huffman coding.

**Step 11:** Finally backup data compression.

Decryption algorithm
The decryption process and the encryption procedure are similar expect for some steps that are in a reversed order.

**Step 1:** IDCT transform.

**Step 2:** Dequantization: XDCDQ.

**Step 3, step 4 and step 5** are the same as those of the encryption process.

**Step 6:** Zigzag inverse and RLE inverse.

**Step 7:** Remove the effect of scrambled value of the resulting matrix from step 6 and the effect of shuffled position of the counter.

**Step 8:** Follow the steps of decompression.

Experimental Analysis
The proposed approach for the crypto-compression image is implemented in Matlab using a personal computer with an Intel dual core 2.2 GHz processor, Windows7.

Experimental set up
Different tests have been performed on the proposed scheme in 256-level gray scale images of different size. Then the different result and interpretation show that the adopted technique offers a performance security also it makes it difficult to attack. We carry out many measurements for differential analysis such as: MAE, NPCR, UACI [2], PSNR, Universal Image Quality Index (UIQI), Pearson Correlation Coefficient and SNR.

Set two images P and C original and cipher images, which are the gray level of the pixels at the i\text{th} row and j\text{th} column of a W×L, respectively.

Experimental Results and Interpretation
Selective encryption with AES and result
Firstly, we present the result of the selective encryption image, so we notice in fig 4.b that the image is not all scrambled, then we can determine the contained image. The histograms of plain image, the ciphered image, produced by the selective encryption, and the decrypted image are

![Figure 2: First step of encryption.](image-url)
presented in fig (4), (c), (d ) and (f), respectively. The histogram of the resultant cipher-image is fairly distributed, while the histogram of the decrypted image is identical to that of the plain image.

Encryption Process and Results
Secondly, we present the result of the image encryption with the proposed scheme. The histograms of the plain image and the ciphered image, produced by the proposed scheme, and the decrypted image are presented in Fig.5.(c), (d), (g), (h), (k) and (l), respectively. The histogram of the cipher image is evenly fairly distributed. Moreover, the histogram of the shuffled image is evidently different from that of the plain image due to the significant substitution effect introduced in the scrambler process and show no statistical resemblance to the plain image.

![Image of histograms showing the comparison between plain, ciphered, and decrypted images.](image)

**Figure 4:** Histogram analysis for selective encryption

The MAE
It corresponds to the absolute error. The MAE between two images is defined according to the two images, $P(i, j)$ and $C(i, j)$, which correspond on the plain-image and the cipher image.

$$\text{MAE} = \frac{1}{W \times L} \sum_{i=1}^{W} \sum_{j=1}^{H} |P(i, j) - C(i, j)| \times 100\%$$

The PSNR
This parameter means the peak signal to noise ratio which can be used to evaluate an encryption scheme. It’s defined as in eq (7) in which $b$ is the number of the coding bit.

$$\text{PSNR} = 10\log_{10} \left( \frac{L \times W \times 255^2}{\sum_{i=0}^{L-1} \sum_{j=0}^{W-1} (P(i, j) - C(i, j))^2} \right)$$

NPCR & UACI
Where the symbol $T$ denotes the total number of pixels in the cipher image, $L$ and $W$ denote respectively the length and the width of the image and $|.|$ denotes the absolute value function.

Correlation analysis
In an ordinary image, each pixel is highly correlated with its adjacent pixels. Unlike the cipher image, the correlation of the adjacent pixels is sufficiently low [8]. However, the test of the correlations of the adjacent pixels in the encrypted image determines the superior confusion and diffusion properties [3, 8, 14]. The correlation of the adjacent pixels is calculated with the use of Eq.(11), (12), (13) and (14) to calculate the correlation coefficients in the horizontal, vertical and diagonal directions:

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

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

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

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

where $x$ and $y$ are grayscale values of two adjacent pixels in the image, $E(x)$ is the expectations of $x$, $D(x)$ is the estimation of the variance of $x$ and $\text{cov}(x, y)$ is the estimation of the covariance between $x$ and $y$.

In the image encryption community, we use the NPCR and UACI to evaluate the strength of an encrypted image scheme[6, 8]. The NPCR stand for the change rate number of pixels compared to a ciphered image while one pixel of a plain-image is changed. UACI represents that intensity rate with regards to $C1$ and $C2$, being two different cipher-images whose corresponding plaintext images differ by only one pixel.

$D$ is a bipolaire array defined in eq(8) NPCR and the UACI are mathematically defined by the following equations (eq 9 and eq 10).

$$D(i, j) = \begin{cases} 0, & \text{if } C_1(i, j) = C_2(i, j) \\ 1, & \text{if } C_1(i, j) \neq C_2(i, j) \end{cases}$$

$$\text{NPCR} = \frac{\text{TP}}{T \times W} \times 100\%$$

$$\text{UACI} = \frac{\text{CI}}{T \times W} \times 100\%$$
NPCR = \sum_{i,j} D(i,j) \times \frac{W \times L}{\sum_{i,j}} \times 100\% \quad (9)

UACI = \frac{1}{W \times L} \sum_{i,j} \left| C_1(i,j) - C_2(i,j) \right| T \times 100\% \quad (10)

**Sensitivity key**

For testing the key’s sensitivity of the proposed image encryption procedure, we have carried out the following steps:

An original image I (Fig.5.g) is encrypted using the secret key k_1 and the resultant image is the encrypted image C_1 (Fig.6.a). The same original image is encrypted by another secret key k_2 and the resultant image is the encrypted image C_2 (Fig.6.b). Finally, the difference image between the two ciphered images (a) and (b) is presented in Fig.6(c), which is obtained by eq (15).

I_{diff} = |C_1(i,j) - C_2(i,j)| \quad (15)

Again, the encrypted image (Fig.6.b) is decrypted by a key different from k_2 and the resultant image is referred to as the image D (Fig. 6.d).

**Figure 5:** Histogram analysis for proposed scheme

<table>
<thead>
<tr>
<th>a. Original image</th>
<th>b. Selective encrypted-compression image with modify value</th>
</tr>
</thead>
<tbody>
<tr>
<td>c. Histogram of original image</td>
<td>d. Histogram of encryption image</td>
</tr>
<tr>
<td>e. Decrypted image</td>
<td>f. Original image</td>
</tr>
</tbody>
</table>

| g. Histogram of decrypted image | h. Original image |
| i. Selective crypto-compressed image with modify value | j. Decrypted image |
| k. Histogram of Selective crypto-compression image with modify value | l. Histogram of Decrypted image |

**Figure 6:** Key sensitive test.

**Interpretation**

- The observation in Fig.6 shows that just a small change in the secret keys results in an incorrect decrypted image. Then we note that a wrong generation of initial parameters for chaotic function makes decryption impossible. This is due to the chaotic properties and the efficiency of the proposed scheme.
- The MAE experiment results for ciphered image are
shown in table 1. It is shown that there is a fluctuation between the MAE of the selective encrypted image and the encrypted image. The MAE of the encrypted image is 30 percent higher than the MAE of the selective encrypted image.

<table>
<thead>
<tr>
<th>Encryption image: crane (dB)</th>
<th>Selective encryption image (dB)</th>
<th>Decryption: crane (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR 8.86061 dB</td>
<td>9.45323 dB</td>
<td>48.75336</td>
</tr>
<tr>
<td>UIQI 0.00550</td>
<td>0.12496</td>
<td>0.95362</td>
</tr>
<tr>
<td>SNR -18.60483 dB</td>
<td>-18.01221 dB</td>
<td>21.28792</td>
</tr>
<tr>
<td>MAE 95.28487</td>
<td>65.04745</td>
<td>0.54724</td>
</tr>
</tbody>
</table>

The NPCR and the UACI test results are 89% and 22%, respectively. The NPCR and the UACI test results are presented in tab 2. As seen in table 2, we can conclude that the proposed scheme presents an adequate value.

<table>
<thead>
<tr>
<th>Image</th>
<th>UACI%</th>
<th>NPCR%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>22.7661</td>
<td>89</td>
</tr>
<tr>
<td>Crane</td>
<td>22.5647</td>
<td>91.11</td>
</tr>
<tr>
<td>Cameramen</td>
<td>23.2132</td>
<td>88.14</td>
</tr>
<tr>
<td>mandril_gray</td>
<td>23.2349</td>
<td>89.29</td>
</tr>
<tr>
<td>goldhill</td>
<td>19.1250</td>
<td>86.8404</td>
</tr>
</tbody>
</table>

We observe in fig (7) that the pixel values are very close, which is the same over a large number of neighboring pixels. However in fig.7.a. a sudden variation of the values of nearly pixels is shown [10].

<table>
<thead>
<tr>
<th>Encryption</th>
<th>Decryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selective</td>
<td></td>
</tr>
<tr>
<td>OI (vs crp)</td>
<td>Encryption</td>
</tr>
<tr>
<td>mandril_gray</td>
<td>-0.00423</td>
</tr>
<tr>
<td>goldhill</td>
<td>-0.00092</td>
</tr>
</tbody>
</table>

Figure 7: Profile of origin and encrypt image along the row number 164: The top graph shows the pixel intensity and the graph shows the corresponding normalized correlation over a 128 pixel displacement. In table 4, we see that the correlation coefficients of the proposed scheme are all smaller than the selective encryption scheme.

Table 3: Correlation coefficients of two adjacent pixels for encrypted image

<table>
<thead>
<tr>
<th>Image</th>
<th>Correlation Vertical Correlation</th>
<th>Horizontal Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crane (256x256)</td>
<td>0.00319 -0.05130</td>
<td>0.02846</td>
</tr>
<tr>
<td>Lena (128x128)</td>
<td>0.00989 0.09811</td>
<td>0.07090</td>
</tr>
<tr>
<td>Cameramen (256x256)</td>
<td>0.00634 -0.03679</td>
<td>0.07339</td>
</tr>
</tbody>
</table>
Comparison between other works
Different proposed algorithm integrate encryption process after DCT[16, 9, 21], then the RLE coding become without influence in compression scheme(see section III.1 step 7). In our work we integrate the encryption process after RLE coding in order to assure a best rate of compression with a good result for encryption. In order to show the efficiency and the performance of the proposed scheme, the horizontal correlation of the encryption image of the proposed scheme, also the value of the PSNR of the encrypted image has been compared with others works. To refer to tab 5 and tab 6, the proposed image encryption method has a good result compared to other proposed algorithms.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Correlation</th>
<th>horizontal Correlation of encrypted image</th>
<th>PSNR of encrypted image (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coefficient</td>
<td>Correlation</td>
<td>horizontal Correlation of encrypted image</td>
<td>PSNR of encrypted image (dB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vertical</td>
<td>-0.08606</td>
<td>0.23109</td>
<td>5.001</td>
</tr>
<tr>
<td>horizontal</td>
<td>-0.05329</td>
<td>-0.13211</td>
<td>5.97219</td>
</tr>
</tbody>
</table>

Table 5: Correlation

<table>
<thead>
<tr>
<th>Proposed (cameramen)</th>
<th>Horizontal Correlation of Encrypted Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>[12] (cameramen)</td>
<td>0.0122</td>
</tr>
<tr>
<td>[8]</td>
<td>0.0088</td>
</tr>
</tbody>
</table>

Table 6: PSNR

<table>
<thead>
<tr>
<th>Proposed (cameramen)</th>
<th>PSNR of encrypted image (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[12] (cameramen)</td>
<td>5.34</td>
</tr>
<tr>
<td>[19]</td>
<td>5.001</td>
</tr>
</tbody>
</table>

Conclusion
The proposed algorithm uses both permutation and substitution processes which are the most used (respected) methods in the encryption schemes based on the chaotic system. In this work, the cipher image is produced by encrypting the DC coefficients with the AES scheme, scrambling the values using the S-box, already mixed by Arnold map, and shuffling the counter with the Arnold cat map chaotic function. Especially, we use chaotic maps which are easy to be implemented. Therefore, the encryption time is substantially reduced and the result is performed. Referring to the Histogram analysis, the PSNR value, the encryption quality, the correlation analysis, differential analysis and the key sensitivity analysis shows that the proposed scheme is performed against some attack such as the statistical and differential attacks. The different algorithms presented in the manuscript are tested and evaluated.

References
[10] K. S. Thyagarajan, Still Image and Video Compression with MATLAB.
[13] Xin Ma, Chong Fu, Wei-min Lei and Shuo Li, " A


