whereas, the bijective function h will be defined by a coupling of these two chaotic maps. The initialization value will be calculated from the clear image, for a possible change of the first pixel. This action greatly increases the avalanche effect and then starts the encryption process. In addition, a chaotic permutation will be applied on the output vector of the schema (see Figure 1) and this to increase the effect of confusion. This encryption technique is fully stated in Figure 2.

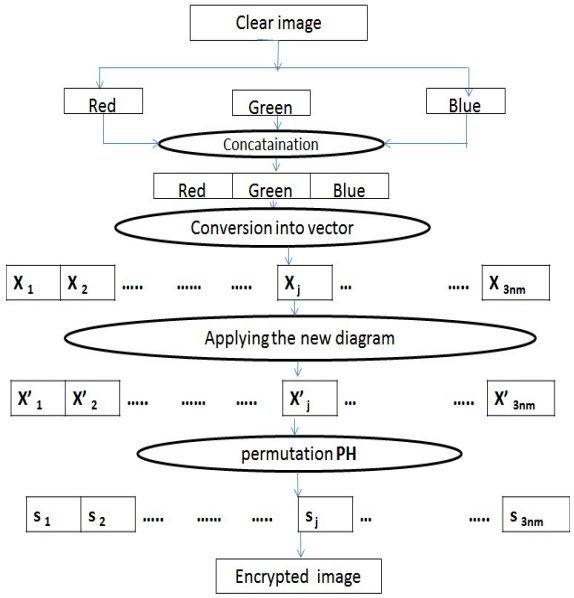

Figure 2. : Applying the new scheme to encrypt a colour image

We note that this technique is articulated along three major axes:

- Construction of the algorithm parameters
- Encryption of the clear image
- Decryption of the encrypted image

Finally, after an application of this algorithm on a database of colour images of different sizes and formats, a detailed analysis will be presented.

3. Construction of the Encryption Settings

This phase *is* the most important, it is used to define all the parameters necessary for the proper functioning of the new crypto system. It is based on the creation of two chaotic maps, namely, the logistic map and the map of Henon to a dimension. In this article, the initial values and control parameters of the two chaotic cards will be considered as a secret encryption key.

3.1 Construction of chaotic maps used

The logistics map is a recurring sequence defined by a particular polynomial of the second degree. This map is widely used in cryptography [10] of colour images. The choice of this map is due to the simplicity and is expression (see equation (1)):

$$\begin{cases} u_0 \in]0 \quad 1[\quad \mu \in [3,75 \quad 4] \\ u_{n+1} = \mu u_n (1 - u_n) \end{cases} \tag{1}$$

With u_0 an initial condition and μ a control parameter. It is known that this suite has a chaotic aspect for an initial value $u_0 \in]0.5 \quad 1[$ and a control parameter $\mu \in [3.75 \quad 4]$. In addition, this card has a high sensitivity to initial conditions due to the value of it is Lyapunov exponent $\lambda = Ln(2) \succ 0$. In the approach, as initial value $x_0 = 0.81$ et μ=3.71 as a control parameter. A first key space obtained is 128 bits.

3.2. Henon Map

The two-dimensional HENON chaotic map was discovered for the first time in 1978 (see equation (3)):

$$\begin{cases} x_0 \; et \; y_0 \quad a = 0.3 \quad et \quad b \in [1.07 \quad 1.4] \\ x_{n+1} = 1 + y_n - a x_n \\ y_{n+1} = b x_n \end{cases} \tag{3}$$

It is a very *simple* card to use in cryptography of colour images. It presents a chaotic aspect for $a=0.3$, $b \in [1.07 \quad 1.4]$ as a control parameter and $x_0 \in]0 \; 1[$, $y_0 \in]0 \; 1[$ as initial conditions. The two-dimensional HENON map expression can be converted to another much simpler one-dimensional chaotic map (see equation (4)):

$$\begin{cases} x_0 , x_1 \quad a = 0.3 \quad , \quad b \in [1.07 \quad 1.4] \\ x_{n+2} = 1 - a x_{n+1}^2 + b x_n \end{cases} \tag{4}$$

It is a *recurrent*, quadratic and nonlinear sequence. The algorithm $x_0 = 0.01, x_1 = 0.02$, as initial *values* $a = 0.3$ *and* $b \in [1,07 \quad 1,4]$. So, the overall size of the encryption key is 384 bits. This value largely *protects* crypto system from brutal attacks, since any exhaustive attack requires at least $(2)^{384}$ Tests.

3.2.1. Construction of the 3nm random functions (g_i)

The encryption of a clear image of arbitrary size (n, m), a vector $(CL)$of size (1,256*3*nm) to coordinates in the ring G is calculated from the logistics map. This construction is provided by a modular function (see algorithm (1)).

Algorithm 1 Converting logistics map elements to G elements

$for\ i = 1\ to\ 256 * 3nm$

$CL(i) = \mathrm{mod}\left(E\left(u_i * 10^{10}\right), 256\right)$

$Next\ i$

The $E(x)$ denotes the integer part of the real x. Le vecteur (CL) *thus obtained is converted into a matrix* (CM)*of size (256, 3nm). Each column i of this matrix represents a random function* (g_i). Therefore the image of an element x_j of the ring G by the random function (g_i) is the element of the matrix (CM) located at the intersection of the line x_j and column i, that is to say that $g_i(x_j) = CM(x_j, i)$. *In this case, the matrix* (CM) *is a random matrix of dynamic size substitution closely related to that of the clear image, and of which each random function* (g_i) is not necessarily injective (see Example 1).

Example 1

(CM)	g_1	g_2	g_3	g_4	……	g_{210}	……		g_{3nm}
0	14	215				12			12
1	12	20				150			25
3	15	15				18			215
4	12	14				89			220
:	215	121				221			126
:	220	86				162			98
:	23	95				23			78
:		20				39			45
140	220					231			35
:	210					:			
:	12					23			
:		14				:			
255	145	222				96			25

so : $g_{210}(140) = CM(140, 210) = 231$

3.2.2. Building the 3nm encryption functions (f_i)

All encryption functions (f_i) used in the algorithm satisfy the property (H) described by A.JARJAR., Each function (f_i) algorithm is defined (see equation (4)):

$$\begin{aligned} f_i : G \times G &\rightarrow G \\ (x, y) &\mapsto a_i x \oplus y \end{aligned} \qquad (4)$$

All the terms of the suite (a_i) are invertible elements in the ring (G). The latter are odd numbers modulo 256. Building all cryptographic functions (f_i) requires the creation of a vector $(HN)$of size (1,3nm) to odd integer coordinates 256, from the one dimensional HENON chaotic map. This operation is performed by algorithm 2:

Algorithm 2 Vector creation (HN)

$for\ i = 1\ to\ 3nm$

$HN(i) = \mathrm{mod}\left(E\left(10^9 * v_i\right), 256\right)$

$HN(i) = \mathrm{mod}(2HN(i) + 1, 256)$

$Next\ i$

Therefore, expression of each cipher function (f_i), stated in algorithm (see equation (5)):

$$\begin{aligned} f_i : G \times G &\rightarrow G \\ (x, y) &\mapsto HN(i) * x \oplus y \end{aligned} \qquad (5)$$

3.2.3. Construction of the bijective function (h)

The bijective transformation (h) of the algorithm, is a mix of the two chaotic cards. It will act only on the last pixel of the clear image, to build the first pixel of the encrypted image. This action will start the encryption process (see (6) for the expression of this bijective function (h) approach).

$$x_1' = h(x_{3nm}) = HN(3nm) * x_{3nm} \oplus CL\left(\mathrm{mod}\left(E(n * HN(nm) + m), 3nm\right) + 1\right) \quad (6)$$

3.2.4. Construction of the permutation (PH)

By decreasing sort in the broad sense of the first 3nm values of the HENON map (v_i), we get a vector (PH). All the coordinates of this vector separate two by two. This vector is considered as a permutation in the ring $\left(Z/_{3nmZ}\right)$.

3.2.5. Calculation of the initialisation value (IV)

After extracting the three colour channels (RGB) from the sharp image, the three matrices obtained are concatenated to form a matrix of size (n, 3m), this matrix is converted into a vector $X(x_1, x_2, \ldots\ldots, x_{3nm})$. The initialization value (IV) is the sum by exclusive or of all the coordinates (x_i) of the vector (X). This initialization value is intended only to change the intensity of the first pixel (x_1) of the vector (X). This operation is stated in algorithm 3.

Algorithm 3 Calculating the initialization value

```
for i = 1 to 3nm
    IV = IV ⊕ x_i
Next i
```

4. Encryption of the clear image

After having determined all the parameters necessary to ensure a perfect functioning of the algorithm, and by application of the new A.JARJAR schema, the formulas of ciphers applied in the algorithm stated in the equation 7:

$$\begin{aligned}
&x_1' = h(x_{3nm}) = HN(3nm) * x_{3nm} \oplus CL\left(\left(\mathrm{mod}\left(E(n * HN(nm) + m), 3nm\right) + 1\right)\right)\\
&x_1 = IV \oplus x_1\\
&x_2' = f_1\left(x_1, g_1(x_2)\right) = HN(1) * x_1 \oplus CM(x_2, 1)\\
&\vdots\\
&\vdots\\
&x_{k+1}' = f_k\left(x_k, g_k(x_{k+1})\right) = HN(k) * x_k \oplus CM(x_{k+1}, k)\\
&\vdots\\
&x_{3nm}' = f_{3nm-1}\left(x_{3nm-1}, g_{3nm-1}\left(x_{3nm}\right)\right) = HN(3nm-1) * x_{3nm-1} \oplus CM\left(x_{3nm}, 3nm-1\right)
\end{aligned} \quad (7)$$

The permutation (PH) will be applied on the output vector $CLH(x_1', x_2', \cdots\cdots, x_{3nm}')$, and this, to increase the effect of confusion and give birth to a new vector $MI(s_1, s_2, \cdots\cdots, s_{3nm})$. This operation is illustrated in algorithm 4:

Algorithm 4 Application of the permutation (PH)

```
for i = 1 to 3nm
    s_i = MI(i) = CLH(PH(i))
Next i
```

After conversion, the vector (MI) into matrix of size (n, 3m), then dissociation into three matrices of size (n, m), we finally obtain the encrypted image by application of the new A.JARJAR scheme.

5. Decryption of the encrypted image

The **decryption** process of the encrypted image, requires the calculation of the decryption functions (t_i), reciprocal bijection $(h)^{-1}$ of (h) and reciprocal permutation (HP) of (PH). This step is fully described in Figure 3.

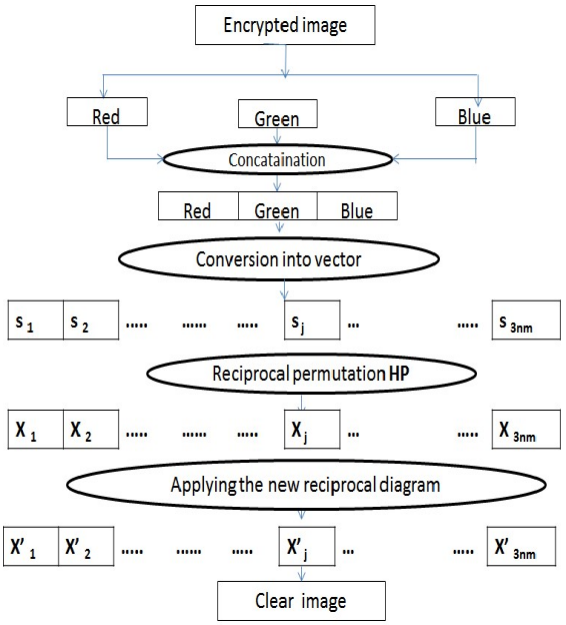

Figure 3. Applying new scheme to decrypt a colour image

5.1. Construction of the inverse permutation (HP)

Reciprocal permutation (HP) is a vector of size (1,3nm), whose coordinates are the indices of those of (PH). This operation is defined in algorithm5:

```
For i = 1 to 3nm
    HP(PH(i)) = i
Next i
```

5.2. Vector calculation (NH)

To determine the decryption functions (t_i) associated with encryption functions (f_i), we will need the values of the inverses of each component of the vector (HN). These inverses will be stored in a vector (NH). This step is stated in algorithm 6:

```
for i = 1 to 3nm
    for j = 1 to 255 step 2
        S = 1
        S = S * j
        r = mod(S, 256)
        if r = 0 then
            NH(i) = j
            exit for
Next j, i
```

5.3. Construction of the reciprocal bijective function $(h)^{-1}$

In the algorithm, reciprocal bijection is applied to the first pixel of the encrypted image, to obtain the last pixel of the net image. This operation is illustrated in equation (8):

$$\begin{aligned} x_1' &= h(x_{3nm}) \\ &= HN(3nm) * x_{3nm} \oplus CL\left(\left(\mathrm{mod}\left(E(n * HN(nm) + m), 3nm\right) + 1\right)\right) \\ x_{3nm} &= h^{-1}(x_1') \\ &\text{So} \\ x_{3nm} &= NH(3nm) * \left(x_1' \oplus CL\left(\left(\mathrm{mod}\left(E(n * HN(nm) + m), 3nm\right) + 1\right)\right)\right) \end{aligned} \tag{8}$$

5.3.1 Calculation of decryption functions (t_i)

The calculation of the decryption functions which is described by A.JARJAR is shown in the algorithm 7:

$$\forall \quad (x, y, z) \in G^3$$
$$with$$
$$z = f_i(x, y) = HN(i) * x \oplus y$$
$$HN(i) * x = z \oplus y$$
$$so$$
$$x = NH(i) * (z \oplus y)$$
$$so$$
$$t_i(x, y) = NH(i) * (x \oplus y)$$

In the algorithm, expression of decryption functions (t_i) or everything $i \in [1 \quad 3nm]$ is given by equation (9)

$$\begin{aligned} x'_{i+1} &= f_i(x_i, g_i(x_{i+1})) \\ &so \\ x'_{i+1} &= HN(i) * x_i \oplus C \\ &so \\ x_i &= t_i(x'_{i+1}, g_i(x_{i+1})) \end{aligned} \quad (9)$$

At first, the encrypted image is converted into a vector $CLH(s_1, s_2, \cdots\cdots, s_{3nm})$. on which the permutation is applied (HP). This action is given by the algorithm9. It causes a vector output $MI(x'_1, x'_2, \ldots\ldots, x'_{3nm})$. The coordinates of this vector are given by the algorithm8:

$$\begin{aligned} &for\ i = 1\ to\ 3nm \\ &\quad x'_i = MI(i) = CLH(HP(i)) \\ &Next\ i \end{aligned}$$

On the vector $MI(x'_1, x'_2, \ldots\ldots, x'_{3nm})$ obtained at the output, the inverse scheme described by A.JARJAR is applied. The decoding formulas applied to this vector are obtained from the last pixel. In the algorithm these formulas are given by equation 10:

$$\begin{aligned} x_{3nm} &= h^{-1}(x'_1) = NH(3nm) * x_{3nm} \oplus CL(\mathrm{mod}(E(n * HN(nm) + m), 3nm) + 1)) \\ x_{3nm-1} &= t_{3nm-1}(x'_{3nm}, g_{3nm-1}(x_{3nm})) = NH(x'_{3nm}) * (x_{3nm} \oplus CM(x_{3nm}, 3nm - 1)) \\ &\vdots \\ x_k &= t_k(x'_{k+1}, g_{k-1}(x_{k+1})) = NH(k+1) * (x_{3nm} \oplus CM(x_{3nm}, 3nm - 1)) \\ &\vdots \\ x_1 &= t_1(x'_2, g_1((x_2))) = NH(2) * (x_2 \oplus CM(x_2, 1)) \end{aligned} \quad (10)$$

To change the value of the first pixel, it is necessary to recalculate the initial value, and thus to find the true value of the pixel (x_1). The study done on a database of images gave the following results :

5.4. Examples and simulations

The approach has been tested on a database of colour images of different sizes and formats. All the encrypted images obtained have a flat histogram. This ensures that the crypto system is safe from statistical attacks.

5.4.1. Analysis and security of the crypto system

A good crypto system must deal with all known attacks.

5.4.2. Key space

Encryption key of the algorithm is 384 bits in size. This size is more than enough to protect the crypto system from exhaustive attacks

5.4.3. Entropy Analysis

According to SHANON, the entropy of information is the amount of information encompassed or released by a source of random information. In particular, the more redundant the source, the less information it contains. Therefore, the entropy is maximum for a source whose all symbols are equiprobable. The expression of entropy is given by SHANNON in equation (11). For an image MC of size (n, m), we put t = nm:

$$Entropy \quad H(MC) = \frac{1}{t}\sum_{i=1}^{t} -p(i)\log_2\left(p(i)\right) \tag{11}$$

The famous colour images in "Lena", "Babon", "Peppers" and "Men" image cryptography are part of the database tested by the algorithm. Table 1 shows the computation of their entropies and those of the encrypted images.

Table 1. Computation of entropies and encrypted images

Net image	Size	Entropy of the net image	Entropy of the encrypted image
Lena	512x512	7,7502	7,9995
Babon	512x512	7,7624	7,9992
Peppers	512x512	7,6658	7,9994
Man	1024x1024	7,5337	7,9993

The values of the entropy of the images encrypted by the algorithm are close to 8, this is the maximum value for an 8-bit colour image. It confirms the uniformity of the histograms. This proves that this approach is safe against any entropy attack.

5.5. Correlation analysis

Correlation is a technique that makes it possible to compare two images to estimate the displacements of the pixels of an image relative to another reference image. Adjacent pixels of a standard image of a clear image have a strong correlation. A good crypto image system must remove such a correlation in order to avoid any statistical attack. The expression of the correlation is defined by equation (10):

$$correlation \quad r = \frac{cov(x,y)}{\sqrt{V(x)}\sqrt{V(y)}} \tag{10}$$

Table 2. Correlation values of the Images

Image	Original image			Encrypted image		
Correlation	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena	0.9774	0.9881	0.9696	0.0024	-0.0004	-0.0001
Babon	0.9047	0.8520	0.8238	-0.0009	-0.0002	0.0003
Peppers	0.9786	0.9820	0.9694	-0.0002	0.0005	0.0006
Man	0.9774	0.9813	0.9668	-0.0001	-0.0003	-0.0001

Table 2 shows that the correlation values of the encrypted images are close to zero. This ensures great security against any attack by correlation.

5..6. Differential analysis

In general, an attacker can make a slight change to the clear image (for example, change only one pixel, see a single bit), and then observe the change in the result. In this way, he may be able to find a meaningful relationship between the simple image and the encrypted image. If a minor modification of the simple image can cause a significant change in the encrypted image, with respect to diffusion and confusion, then this differential attack would become very inefficient and virtually useless. To test the influence of the change of a pixel on the entire image encrypted by the proposed algorithm, two common measurements were used: the rate of change of the number of pixels (NPCR) and the unified average changing intensity (UACI). Let us note two encrypted images, whose corresponding clear images (C_1) and (C_2) have only one pixel difference, by and, respectively. The expressions of these two statistical constants are stated in equation (11), for a size image (n, m):

$$NPCR = \left(\frac{1}{nm} \sum_{i,j=1}^{n\,m} D(i,j) \right) * 100$$

$$With \quad D(i,j) = \begin{cases} 1 & if \quad C_1(i,j) \neq C_2(i,j) \\ 0 & if \quad C_1(i,j) = C_2(i,j) \end{cases} \tag{11}$$

$$UACI = \left(\frac{1}{nm} \sum_{i,j=1}^{n\,m} Abs\left(C_1(i,j) - C_2(i,j)\right) \right) * 100$$

Table 3. Calculated values of the PSNR, NPCR and the UACI of the crypto system

Image	Man	Lena	Haouse	Aigle	Tigre
Size	1024x1024	512x512	256x256	153x222	320x240
PSNR	8.0475	8.6510	8.9475	6.6526	8.1252
NPCR	99.94	99.67	99.63	99.67	99.67
UACI	33.36	33.35	33.33	33.25	33.32

6. Conclusion

In this article, we have detailed a new crypto system of colour images of any size (n, m), this algorithm is a direct application of the scheme improved by A.JARJAR. for this fact, we have witnessed the creation of 3nm random functions of encryption satisfying the property (H), described by A.JARJAR. All of these functions were created from the one-dimensional HENON map. To the construction of 3nm other random functions from the logistics map. To start the encryption process, we applied a bijection h on the last pixel of the net image, and this to obtain the first pixel of the encrypted image. This transformation is built by mixing the two chaotic cards. Finally, after the new schema is released, a chaotic permutation is applied to the output vector. This transformation greatly increases the effect of confusion. The application of the new schema is very simple, very efficient, and jointly builds the effect of confusion and diffusion, and this by applying a single trick. We notice that this scheme can be improved, to give more satisfactory results.

7. References

[1] B. Schneier, (1994). Fast software encryption Cambridge Security Workshop Proceedings, Springer-Verlag, pp. 191-204, December.

[2] C. E. Shannon, (1949). Communication theory of secrecy systems » Bell syst Tech Journal. pp 656-715.1949.

[3] NIST, (1977). Data Encryption Standard, Federal Information Processing Standards National Bureau of Standards, US Department of Commerce.

[4] C. Adams, (1997). The CAST-128 Encryption Algorithm, RFC 2144.

[5] E. Petrisor, (2003). Entry and exit sets in the dynamics of area preserving Hanon map, Chaos Solutions and Fractals. pp 651-658, October.

[6] Hraoui S., Gmira F., Jarar A.O., Satori K., Saaidi A, (2013). Benchmarking AES and chaos based logistic map for image encryption. Computer Systems and Applications, ACS International Conference.

[7] Xiao Feng, Xiaolin Tian and Shaowe iXia (2011). An Improved Image Scrambling Algorithm Based On Magic Cube Rotation and Chaotic Sequences", IEEE press,4th International Congress on Image and Signal Processing, pp.1021-1024.

[8] ScienceDirect, (2011). Encryption scheme using a bit-level permutation, Information Sciences, pp 1171–1186.

[9] G. Alvarez., S. J. Li, (2006). Some basic cryptographic requirements for chaosbased cryptosystem", Int. Journal Bifurcat. Chaos, pp 2129–2151.

[10] A.JARJAR, (2018). Improvement of Feistel method and the new encryption scheme. ScienceDirect OPTIK Volume 157, March, pp 1319-1324.

[11] L. Guo-hui, Z. Shi-ping, X. De-ming, L. Jian-wen, (2003). An Intermittent Linear Feedback Method for Controlling Henon like Attractor, Journal of Applied Sciences. pp 288-290, December.