A new cryptosystem based on an enhanced Vigenere cipher incorporating large SBoxes

Abdelhakim Chemlal[1], Hassan Tabti[2], Hamid El Bourakkadi[1,], Abdellah Abid[1], Abdellatif Jarjar[1], Abdelhamid Benazzi[1]*

[1] MATSI Laboratory, ESTO, Mohammed First University, Oujda, Morocco
[2] LSIA Laboratory, FST, Sidi Mohamed Ben Abdellah University, Fez, Morocco

Abstract. The present paper explains the development of an innovative cryptographic system designed to encrypt color images at the pixel level. This technique relies on a significant enhancement of the conventional Vigenere method, incorporating the implementation of two large substitution tables generated from the widely used chaotic maps in cryptography. The system integrates new confusion/diffusion functions, governed by binary decision vectors of pseudorandom values. Simulations conducted on an arbitrary selection of images from a database demonstrate that our system can withstand all known attacks.

Keywords: S-box; Enhanced Vigenere function; confusion; diffusion

1 Introduction

The rapid advancement of the mathematical theory of chaos provides researchers with the opportunity to enhance some conventional encryption systems such as the classical Hill [1], [2], Vigenere technique [3], [4], [5], and Feistel [6], [7] to cope with known attacks. In light of the increased focus on security, numerous techniques for encrypting color images have invaded the digital domain, primarily using number theory and linear algebra. Others strive to adapt certain genetic operators in the encryption of medical images chaos [8], [9]. Regrettably, the lack of an encryption procedure and the absence of linkage between encrypted and plaintext blocks make most of these improvements vulnerable to differential attacks.

In the traditional Vigenère encryption method [10], the difficulty of determining the length of the cryptographic key and the use of a static substitution table can lead to the need to recalculate the index of coincidence. This process can then facilitate the reconstruction of the private key, making the algorithm vulnerable to statistical attacks. Ultimately, the inherent simplicity of this classic algorithm makes it susceptible to such attacks.

Our contribution in the present article is partitioned into two key parts. Firstly, we transform the plain image into a vector modelling. Next, we determine the initialization value needed to adjust the seed pixel, thereby initiating the encryption process. To initiate this process, we

*Corresponding author: hamid.elbourakkadi.d23@ump.ac.ma

implement a new transformation inspired by the Vigenere technique, incorporating robust confusion and diffusion functions.
The present manuscript is organized into several sections, namely, a section 2 to explain in detail different stages of our method, a section 3 for experimental results and the security analysis, and finally a section 4 for the conclusion.

2 Our method

Our approach involves a significant enhancement of the conventional Vigenere method, achieved across the utilization of broad replacement tables [11], [12] in conjunction with novel pseudorandom substitution functions. This methodology is founded on the following stages.

2.1 Chaotic sequences selection stage

2.1.1 A.J- map.

The author A.J. has proposed a novel map denoted as map (s_n) with chaotic behaviour [11], defined by a linear function expressed by equation (1).

$$\begin{cases} s_0 \in \left[\frac{1}{(1+\delta)} \quad \frac{\delta}{(1+\delta)}\right] \qquad \delta \in [1{,}47 \;;\; \varphi] \\ f(s_n)=s_{n+1}=\begin{cases} \delta^2 s_n & \text{if } 0 \le s_n \le \frac{1}{1+\delta} \\ \delta-\delta s_n & \text{if } \frac{1}{1+\alpha} \le s_n \le 1 \end{cases} \end{cases} \tag{1}$$

(δ): the control parameter, (s_0): the initial state, and φ=1.58 is the golden number.

2.1.2 Logistic map.

The 2nd sequence is produced using the logistic map [12], which is a straightforward polynomial-order recursive sequence given in system (2).

$$\begin{cases} l_0 \in]0{,}5; 1[\; et \; k \in [3{,}75; 4] \\ l_{n+1} = k.l_n(1-l_n) \end{cases} \tag{2}$$

(l_0): the initial state and (k): the control parameter.

2.2 Pseudorandom vectors construction stage

In our method, there are two sequences of perturbations (l) and (s) that are very sensitive to initial conditions and can be easily implemented in any cryptosystem, as presented above.

2.2.1 Sub keys construction.

Algorithm 1 below includes 5 pseudorandom vectors ($\boldsymbol{T1}$), ($\boldsymbol{T2}$), ($\boldsymbol{T3}$), and ($\boldsymbol{a}$) with coefficients inside the ring ($\boldsymbol{Z/256Z}$).

Algorithm 1. Pseudo-random vectors creation

$for\ i\ =\ 1\ to\ 3nm$
$\quad T1(\ i) \leftarrow\ (int(max(s(i); l(i)).10^{10})\ \ mod\ 251) + 4$
$\quad T2(\ i) \leftarrow\ (int(((s(i) + 3 * l(i))/4).10^{11})\ \ mod\ 252) + 3$
$\quad T3(\ i) \leftarrow\ \ (int(|s(i) - 2 * l(i)|.10^{12})\ \ mod\ 254) + 1$
$\quad a(\ i) \leftarrow\ \ (2 * int(|l(i) * s(i)|.10^{12}) + 1)\ \ mod\ 256$
$\quad b(\ i) \leftarrow\ \ (int(|l(i) + s(i)|.10^{12})\ \ mod\ 253) + 2\ :\ end\ for$

Furthermore, to control the ciphering operations, our system generates two binary vectors ($C1$) and ($C3$) as depicted in Algorithm 2.

Algorithm 2. ($\boldsymbol{C_1}$) and ($\boldsymbol{C_2}$) Control vectors creation

$for\ j \leftarrow 1\ to\ 3nm$	//Second binary vector construction
//First binary vector construction	$if\ \ s(j) \le l(\ j)\ then: C2(j)\ \leftarrow\ 1$
$if\ s(j)\ > l(j)\ then:\ C1(j)\ \leftarrow 1$	$else:\quad C2(j)\ \leftarrow\ 0$
$else:\ C1(j)\ \leftarrow 0\ :\ \ end\ if$	$end\ if:\ end\ for$

2.3 Substitution table design stage

Our set of rules necessitates the introduction of two fresh alternative tables (A) and (B), each of dimension ($256\ ;\ 256$) and containing pseudorandom values within the ring (Z/256Z).

2.3.1 (A) S-Box design.

The primary goal of this phase is to create a singular Vigenere substitution matrix, designated as (***A***), sized at (256; 256), adhering to the required commands outlined in Algorithm 3.

- The initial row in table (A) corresponds to the permutation ($Pt1$) of the initial 256 values from vector ($T1$), achieved through ordering them in descending arrangement.
- For lines of order greater than 1, the line values are taken from either a rank shift $T2(j)$ or $T3(j)$, contingent upon the fee of the manage vector $C1(j)$.

Algorithm 3. (A) Replacement S-Box design

$//\ First\ line$	for i $\leftarrow\ 1\ \ to\ \ 256$
$for\ i \leftarrow\ 1\ \ to\ \ 256$	$if\ C1(j) = 0\ then:$
$\quad A(1, i)\ \leftarrow\ Pt1(i)$	$\quad A(j, i)\ \leftarrow\ A(j - 1, mod(i + T2(j),256))$
$end\ for$	$else$: A$(j, i)\ \leftarrow\ A(j - 1, mod(i + T3(j),256))$
$//\ Next\ lines$	$end if$
$for\ j \leftarrow\ 2\ \ to\ \ 256$	$end\ for$
	$end\ for$

2.3.2 (B) S-Box design.

The method of building the brand novel replacement matrix (***B***) is outlined as follow:

- The 1st three lines consist of the rearrangements ($L1$), ($L2$), and ($L3$), which might be derived from a huge ascending type carried out on the preliminary 256 values of vectors ($T3$), ($T2$), and ($T1$), respectively.
- The i^{th} line (in which i is more than 3) is fashioned through combining either the line $(i - 3)$ and $(i - 2)$ or $(i - 3)$ and $(i - 1)$, based on the cost of the control vector ($C2$). The process is demonstrated through Algorithm 4 as depicted below.

Algorithm 4. (***B***) Substitution box construction

```
// First three lines                    //Other lines
for i ← 1 to 256                        for j ← 4 to 256
    B(1,i) ← L1(i)                        for i ← 1 to 256
    B(2,i) ← L2(i)                          if C2(j) then : B(j,i) ← B(j − 3, B(j − 2, i))
    B(3,i) ← L3(i)                          else: B(j,i) ← B(j − 3, B(j − 1, i)) : end if
end for                                 end for : end for
```

2.3.3 Improved confusion function.

This updated replacement function, which incorporates tables (***A***) and (***B***), is presented in Algorithm 5.

Algorithm 5. (***S***) Improved confusion function design

```
for
    S(X(i)) ← mod [ a(i) * A (T1(i), B(T2(i); X(i))) ; 256] ⊕ b(i)
    // Where, a(i) is invertible in Z/256Z
end for
```

2.3.4 Diffusion function.

The diffusion function involving table (***B***) is given by Algorithm 6.

Algorithm 6. (***D***) Diffusion fonction design

```
for
    D(X(i)) ← B(Z(i − 1); X(i))
end for
```

2.4 Encryption stage

The encryption stage is carried out through the following steps:

2.4.1 Plain image vectorization.

During this step, it is essential to load the original image whose dimensions are (n, m). Next, the channel vectors (R), (G), and (B) must be extracted, and they are combined to create a unified vector (X) of size (1,3nm). This operation is carried out under the guidance of the binary vector (C1). The Algorithm 7 presents the mathematical formulation of this step.

Algorithm 7. Plain image vectorization

```
for p ← 1 to nm                         else
  if C1(p) = 0 then                       X(3p − 2) ← R(p)⊕T3(p)
    X(3p − 2) ← R(p)⊕T1(p)                X(3p − 1) ← G(p)⊕T1(p)
    X(3p − 1) ← G(p)⊕T2(p)                X(3p) ← B(p)⊕T2(p)
    X(3p) ← B(p)⊕T3(p)                  end if: end for
```

2.4.2 Second encryption lap.

To address the challenge of differential attacks, an additional approach is employed through a second iteration that incorporates confusion/diffusion functions, using randomly generated vectors to establish the linkage between encrypted pixels and clear pixels. This approach

necessitates computing an initialization value associated with the resultant image following the initial iteration.

2.4.3 Initialization value calculation.

This process of confusion/diffusion begins with the calculation of the initialization value (I). This constant, intricately connected to the unique photograph, is supposed to adjust the cost of the initial pixel and commence the encryption procedure. Figure 2 illustrates the computation process of the initialization value. Algorithm 8 below provides an interpretation of this figure.

Algorithm 8. Initialization constant computation

$I = 0$
$for\ i\ =\ 2\ to\ 3nm$
  $if\ C2(i) = 1\ then$
    $I = I \oplus T3(i) \oplus X(i)$
  $else$
    $I = I \oplus T2(i) \oplus X(i)$
  $end\ if$
$end\ for$

2.4.4 Confusion/diffusion process.

To counter differential attacks, we can first perform diffusion using pseudo-random confusion vectors to set up a hyperlink among encrypted pixels and clear pixels the usage of a bijective function (S). The algorithm nine illustrates the confusion/diffusion technique.

Algorithm 9. Encryption process through the enhanced Vigenere cipher

// Encryption of the first pixel
$Z(1) = S(I \oplus X(1) \oplus T1(1))$
// Encryption of the next pixels
$for\ i\ =\ 2\ to\ 3nm$
  $y = D(X(i)) \oplus X(i)$
  $if\ C2(i) = 0\ then$
    $Z(i) = S(y \oplus T2(i))$
  $else$
    $Z(i) = S(y \oplus T3(i))$
  $end\ if$: $end\ for$

The graphical representation of this algorithm is illustrated in (Fig. 1).

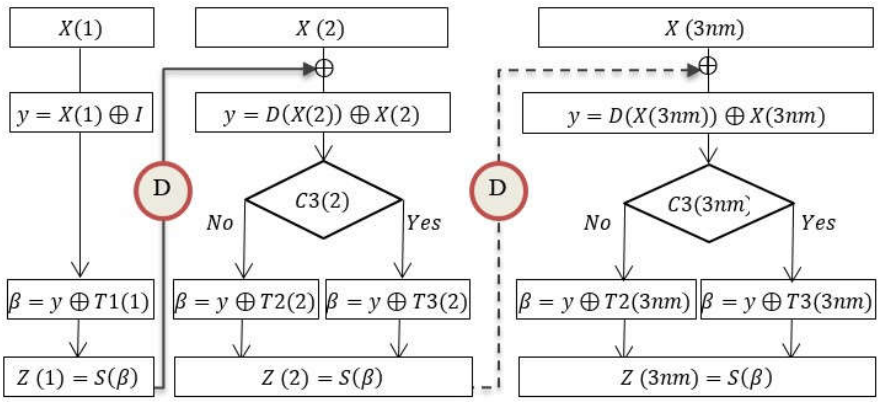


Fig. 1. Encryption process through the confusion/diffusion circuit

2.5 Decryption stage

Our algorithm is a symmetric ciphering system, meaning that the same key will be used in the decryption mechanism. Decryption initiates by reversing the final encryption degree, employing the inverse features of the encryption flowchart.

3 Simulations and discussion

The simulation results were carried out using Python on a Windows 10 running device, using hardware comprising a computer with 32 GB of RAM, an i7 processor, and 1 TB Hard drive. The image samples used for checking out have been sourced from [15]. The experimental parameters and keys had been derived from the chaotic maps distinct in advance.

3.1 Statistical attacks

Our new set of rules changed into examined on more than one randomly decided on reference pictures, and the following simulations were recorded:

3.1.1 Key-space analysis.

The studied crypto-system employs two chaotic maps derived from four actual attributes, every described by 32 bits, ensuing in a key size of a hundred and twenty bits. This layout guarantees robust resistance towards brute-force attacks.

3.1.2 Key-sensitivity analysis.

The high sensitivity of the initial situations of the 2 chaotic maps means that even a moderate alteration to the personal key will cause absolutely distinct encrypted snap shots, as depicted in (Fig. 2).

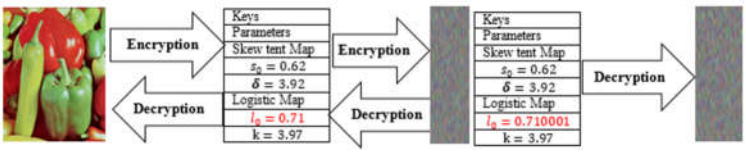
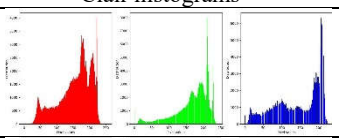
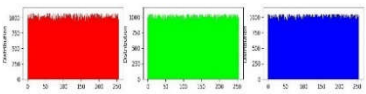
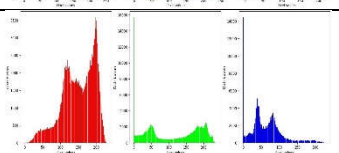
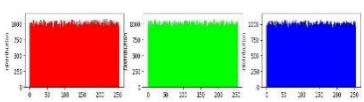


Fig. 2. Key sensitivity analysis

3.1.3 Histograms analysis.

Table 1 exhibits the RGB histograms of both the plain and cipher Peppers and Lena images utilizing our approach. The RGB histogram effects of the cipher snap shots generated through our algorithm reveal a steady distribution. These findings offer guarantee that our system is resilient in opposition to histogram-based assaults.

Table 1. Histograms for Peppers and Lena plain and cipher images

Images	Clair histograms	Cipher histograms
Lena		
Peppers		

3.1.4 Entropy analysis.

The entropy of an image can be calculated using equation (5).

$$S(MC) = \frac{-1}{3nm} \sum_{i=1}^{3nm} p(i).log_2(p(i)) \qquad (5)$$

$p(i)$: The probability of level (i) within an image.
Table 2 affords evaluation of the entropy degrees among our gadget and different similar algorithms, as documented in references [13] [14]. These findings underscore the advanced performance of our approach compared to the algorithms discussed, therefore asserting the robustness of our device towards statistical attacks.

Table 2. Comparison of entropy results for image "Lena"

Method	Proposed	[13]	[14]
Entropy	7.9974	7.9973	7.9900

3.1.5 Correlation analysis.

Equation (6) offers the correlation for an image of size (n, m).

$$corr = \frac{cov(X,Y)}{\sqrt{var(X)}.\sqrt{var(Y)}} \qquad (6)$$

Table 3 provides a comprehensive assessment of the correlation between our method and different similar approaches. As illustrated in Table four, our technique demonstrates advanced performance compared to the algorithms referenced in references [13] [14]. These consequences affirm the robustness of our cryptographic system in opposition to statistical assaults.

Table 3. Comparison of correlation results for image "Lena"

Algorithm		Proposed	[13]	[14]
Correlation	0,0027	0.0096	-0.0058	-0.0058
	-0,0037	-0,0071	0,0043	0,0043
	-0,0003	-0,0079	-0,0004	-0,0004

3.2 Analysis of differential attacks

To compare the robustness of our architecture towards differential attacks, metrics which include the unified common change intensity (UACI), the wide variety of pixel exchange prices (NPCR), and the avalanche effect are utilized.

3.2.1 UACI analysis.

This metric can be given by equation (7) below.

$$UACI = \left(\frac{1}{3nm} \sum_{i,j=1}^{3nm} \frac{|Img_1(i,j) - Img_2(i,j)|}{255} \right) * 100 \qquad (7)$$

With, $Img_1(i,j)$ is the cipher modified image pixel of rank(i,j), while $Img_2(i,j)$ is the cipher image pixel of rank(i,j).

3.2.2 NPCR analysis.

This metric can be given by equation (8) below.

$$\begin{cases} NPCR = \left(\frac{1}{3nm} \sum_{i,j=1}^{3nm} Df(i,j) \right) * 100 \\ Df(i,j) = \begin{cases} 1 & if \quad Img_1(i,j) \neq Img_2(i,j) \\ 0 & if \quad Img_1(i,j) = Img_2(i,j) \end{cases} \end{cases} \tag{8}$$

Table 4 showcases the UACI and NPCR values computed from the images evaluated with the aid of our set of rules. The differential metrics computed through our algorithm align with global standards, as determined in previous references [13] [14]. This validation underscores the superior overall performance of our technique, making sure sturdy protection towards capability differential assaults.

Table 4. UACI and NPCR of Lena and Peppers comparison results

Algorithm	Lena		Peppers	
	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
Proposed	99.68	33.49	99.67	33,45
[13]	99.64	32.66	-	-
[14]	99.67	33.33	-	-

4 CONCLUSION

The images examined by our algorithm all exhibited differential and statistical constants consistent with established international norms and requirements. Furthermore, the use of two S-Boxes in the implementation of dynamic confusion and diffusion functions has endowed our system with robustness, enabling it to withstand any known attack. Consequently, the minimal encryption time encourages the use of our encryption system for video sequence encryption.

References

1. H. Rrghout, M. Kattass, Y. Qobbi, N. Benazzi, A. Jarjar, and A. Benazzi, 'Robust Image Encryption Algorithm Using a New Variant of Hill Cipher', in *Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security*, New York, NY, USA: ACM, May 2023, pp. 1–6. doi: 10.1145/3607720.3607750.
2. M. A. Lone and S. Qureshi, 'RGB image encryption based on symmetric keys using Arnold transform, 3D chaotic map and affine hill cipher', *Optik (Stuttg)*, vol. 260, p. 168880, (Jun. 2022), doi: 10.1016/j.ijleo.2022.168880.
3. M. Kattass, H. Rrghout, M. Jarjar, A. Jarjar, F. Gmira, and A. Benazzi, 'Chaotic Image Encryption Using an Improved Vigenère Cipher and a Crossover Operator', 2024, pp. 181–191. doi: 10.1007/978-3-031-53717-2_17.
4. H. El Bourakkadi, A. Chemlal, H. Tabti, M. Kattass, A. Jarjar, and A. Benazzi, 'Improved vigenere using affine functions surrounded by two genetic crossovers for image encryption', *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 3, p. 1787, (Jun. 2024), doi: 10.11591/ijeecs.v34.i3.pp1787-1799.
5. H. El Bourakkadi, A. Chemlal, H. Tabti, M. Kattass, A. Jarjar, and A. Benazzi, 'Improved Vigenere approach incorporating pseudorandom affine functions for

encrypting color images', *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 3, p. 2684, (Jun. 2024), doi: 10.11591/ijece.v14i3.pp2684-2694.

6. H. Tabti, H. El Bourakkadi, A. Chemlal, A. Jarjar, S. Najah, and K. Zenkouar, 'Novel cryptosystem integrating the Vigenere cipher and one Feistel round for color image encryption', *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 5, p. 5701, (Oct. 2024), doi: 10.11591/ijece.v14i5.pp5701-5714.
7. A. Chemlal, H. El Bourakkadi, H. Tabti, M. Jarjar, A. Jarjar, and A. Benazzi, 'Image encryption through two rounds of Feistel operating at the bit level', in *2024 International Conference on Circuit, Systems and Communication (ICCSC)*, IEEE, Jun. 2024, pp. 1–6. doi: 10.1109/ICCSC62074.2024.10616710.
8. J. G. Sekar, E. Periyathambi, and A. Chokkalingam, 'Hybrid chaos-based image encryption algorithm using Chebyshev chaotic map with deoxyribonucleic acid sequence and its performance evaluation', *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, p. 6952, (Dec. 2023), doi: 10.11591/ijece.v13i6.pp6952-6963.
9. H. Tabti, H. EL Bourakkadi, A. Chemlal, A. Jarjar, K. Zenkouar, and S. Najah, 'Genetic Crossover at the RNA Level for Secure Medical Image Encryption', *International Journal of Safety and Security Engineering*, vol. 14, no. 1, pp. 201–216, (Feb. 2024).
10. S. Li, Y. Zhao, B. Qu, and J. Wang, 'Image scrambling based on chaotic sequences and Veginère cipher', *Multimed Tools Appl*, vol. 66, no. 3, pp. 573–588, (Oct. 2013), doi: 10.1007/s11042-012-1281-z.
11. A. JarJar, 'New chaotic map development and its application in encrypted color image', *Journal of Multimedia Information System*, vol. 8, no. 2, pp. 131–142, (Jun. 2021), doi: 10.33851/JMIS.2021.8.2.131.
12. S. R. Victor Juvvanapudi, P. Rajesh Kumar, and K. V. V. Satyanarayana Reddy, 'Hybrid chaotic map with L-shaped fractal Tromino for image encryption and decryption', *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, p. 389, (Feb. 2024), doi: 10.11591/ijece.v14i1.pp389-397.
13. S. Taha Allawi and D. Riadh Alshibani, 'Color Image Encryption Using LFSR, DNA, and 3D Chaotic Maps', *International journal of electrical and computer engineering systems*, vol. 13, no. 10, pp. 885–893, (Dec. 2022), doi: 10.32985/ijeces.13.10.4.
14. S. T. Allawi and N. A. A. Mustafa, 'Image encryption based on combined between linear feedback shift registers and 3D chaotic maps', *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 3, p. 1669, (Jun. 2023), doi: 10.11591/ijeecs.v30.i3.pp1669-1677.