Multimedia Tools and Applications (2023) 82:14609–14631
https://doi.org/10.1007/s11042-022-13779-y

An extendable key space integer image-cipher using 4-bit piece-wise linear cat map

Gaetan Gildas Gnyamsi Nkuigwa[1] · Hermann Djeugoue Nzeuga[1] · J. S. Armand Eyebe Fouda[1,2] · Samrat L. Sabat[3] · Wolfram Koepf[2]

Received: 10 September 2021 / Revised: 30 June 2022 / Accepted: 5 September 2022 / Published online: 30 September 2022

Abstract

This paper presents a multiplierless image-cipher, with extendable 2048-bit key-space, based on a 4-dimensional (4D) quantized piece-wise linear cat map (PWLCM). The quantized PWLCM exhibits limit-cycles of 4-bit encoded integers with periods greater than 10^7. The synthesis of the PWLCM in a finite state space allows to eliminate the undesirable finite precision effect due to the hardware realization. The proposed image-cipher combines chaos, modular arithmetic, and lattice-based cryptography to encrypt a color image by performing pixel permutation and diffusion in a single operation. Further, an image-dependent confusion operation based on an 8-bit 2D-PWLCM is performed on the whole image to enhance security. In order to increase the key-space without key duplication, 16×16 sub-images are modified using sub-keys of different lattice length vectors generated from the external key. Both simulations and security analyses confirm that the proposed algorithm can resist common cipher attacks, in addition to its advantages such as simplicity, ease of implementation on low-end processors and extensibility of key-space that allows it to easily adapt even for future post-quantum computing attacks.

Keywords Cryptography · Modular arithmetic · Random integers · Security analysis

1 Introduction

The rapid development of digital image transmission over wireless communication media has increased the concern of data security, leading to the demand for image cipher. There exist several techniques for securing data, including steganography, watermarking, and data encryption [2, 25]. The steganography technique conceals the message data; hence it requires more communication bandwidth over a computer network, whereas the cryptography technique transforms the message data, needing approximately the same bandwidth as message data. Thus, encryption is the preferable and mature technology used in applications

✉ J. S. Armand Eyebe Fouda
fouda@mathematik.uni-kassel.de

Extended author information available on the last page of the article.

involving message transmission over a network [19]. In encryption, the message is scrambled using a secret key. The encryption's strength depends on the secret key's randomness strength. Different techniques such as linear congruential, additive congruentional, linear feedback shift register, multiple recursive generators and chaos based generators are used to generate the random sequences using the seed as the initial condition of the dynamical system that constitutes the secret key [18, 24, 37].

There exist different methods such as linear feedback shift register (LFSR), linear congruential generator (LCG), Multiple Recursive Generators (MRGs) for generating Pseudo-random Numbers (PRN). The disadvantages of these methods are the limited periodicity of the random number sequence, which can be mitigated using a suitable chaotic system.

Chaotic systems have intrinsic properties such as sensitivity to initial conditions, ergodicity, random-like dynamics behaviors, and unpredictability that are desired characteristics for designing secure ciphers. Hence, chaos-based encryption methods have emerged besides traditional ciphers such as Advanced Encryption Standard (AES), International Data Encryption Algorithm (IDEA), Data Encryption Standard (DES), and RSA to enhance data security. However, complex chaotic systems are required to make ciphers more secure. Broadly, there are two types of dynamical chaotic systems: continuous-time and discrete-time. The latter is suitable for data encryption as it is feasible to implement on digital hardware. Among the basic discrete-time systems, 2D logistic map, 2D standard map, 2D Henon map, and the 3D Baker map are used in cryptography [26, 34, 38, 39]. During the last decade, the performance of such systems has been improved to increase their complexity, leading to more randomness for secure data encryption. A combination of the piece-wise-linear chaotic map and linear Diophantine Equation (LDE) enhances the cipher's security and was used for image encryption [14]. Although the ciphered image was obtained after only one round, it has the drawback that the encryption process is independent of the plain-image characteristics. The authors mitigated the drawbacks by proposing another one-round encryption scheme in which large permutation and diffusion keys were generated by sorting the solutions of the LDE [13]. Multiple chaotic maps were used to derive the control parameters and initial values to increase the security level of the cipher [31]. It enhances the key-space; however, the short-length encryption key makes the cipher vulnerable against brute force attacks. Other ciphers based on the combination of chaotic systems were also proposed to increase the key-space and are still under investigation [22, 29, 30, 33, 36].

In all the above chaos-based ciphers, chaotic maps need to be quantized during the hardware implementation of the algorithm. Such a quantization reduces the randomness of the chaotic orbits; hence the security level of the cipher [41]. In order to overcome such a drawback, it is necessary to either evaluate the complexity of the chaotic system under limited precision conditions or to increase the computational precision. Most of the work reported in the literature implements chaotic systems with 32-bit floating-point arithmetic, which is hardware costly and requires high-end processors for execution [5, 15]. Hence, it is necessary to design quantized chaotic systems with a large period of limit-cycles to implement chaos-based ciphers on a low-end processor [12].

Arnold's cat map (ACM) preserves the mixing property even after quantization among the different chaotic maps. It has the advantage of (i) being easily defined both in the continuous phase space and the discrete phase space (quantized version) and (ii) a computationally simple 2D system that can be easily extended into a multi-dimensional system. It has been used in data encryption in the confusion step. Apart from the ACM, the combination of the Henon and Arnold cat maps was used in designing an image cipher [11]. Although the algorithm performed well, it has nevertheless been observed that the periodicity of restoring the original image is too short, leading to security issues. More recently, an encryption scheme based on continuous phase space of a generalized Arnold cat map was reported [20]. In [43], the 2D ACM was combined with an affine cipher to enhance the security level of the cipher. In all of these algorithms, the ACM is used in its continuous phase space version, and the precision of the generated obits is chosen as large as possible (32-bit, for example) to avoid short limit-cycles due to the quantization process. The quantized ACM (QACM) has been widely studied in the literature, and the relationship between its period and the number of encoding bits has been determined [6, 7, 27].

The period of the QACM is an important parameter that can induce security issues when using it in cryptography. It is known that the period of the QACM does not exceed $3m$, $m \in \mathbb{N}_{>1}$ being the modulo value. This limitation justifies the use of continuous phase space ACM ($m = 1$) compared to QACM in many cryptographic applications. As a solution, some researchers investigated the impact of the dimensionality and the control parameters on the period of the system. The investigation showed that the period does not significantly increase with the increase in dimension [16], where the conventional 2D Arnold's cat map was altered to a 3D map by introducing six control parameters. The obtained map allows for improvement, but the period distribution and its impact on the system dynamics are not evaluated. Due to the finite computer precision, chaotic sequences are transformed into periodic ones. Further, the output of the 3D map was perturbed to mitigate such degradation of chaotic sequences without investigating the impact of the perturbation on randomness.

The present paper proposes a multiplierless 2048-bit key secure cipher based on the quantized piece-wise linear cat map (PWLCM) obtained by perturbing the conventional QACM [11]. We aim to directly generate randomly distributed integers with the desired precision using the PWLCM. The proposed algorithm combines chaos, modular arithmetic, and lattice-based cryptography [3, 8, 35]. The latter allows to easily extend the external key length without duplication. Such a property is required as lattice-based ciphers are assumed to resist future attacks in the era of post-quantum computing [3, 17, 28, 35]. For the algorithm to be implemented even with low-end processors, we consider only 4-bit precision random numbers generated from a 4D PWLCM for performing the confusion and diffusion operations. We investigate the period of the generated random integers and their randomness to prove the high-security level of our cipher. In the proposed scheme, pixel positions and values are modified in a single operation within blocks of size 16 × 16 pixels with a 2D PWLCM before confusing the whole image. It helps to enhance the speed performance. During the confusion-diffusion operation, each 16 × 16 sub-image is modified using a different subkey corresponding to a combination of κ-length vectors (κ-dimensional lattice), $\kappa \in \mathbb{N}_{\geq 1}$ [28].

The key contributions of the current work are as follows:

1. A novel integer arithmetic multiplierless 2048-bit key image cipher combining chaos, modular arithmetic and lattice-based cryptography is proposed that uses a combination of 4-bit and 8-bit modular addition and subtraction operations only;
2. An extensible key management technique combining modular arithmetic and lattices is presented;

3. A 4-bit 2D and 4D PWLCM is proposed to generate large period pseudorandom sequences;
4. A performance analysis for different attacks is presented.

The rest of the paper is organized as follows: Section 2 presents a brief recalling of ACM and the definition of the PWLCM; Section 3 presents the proposed cipher; Section 4 presents the security analysis of the proposed encryption scheme, and conclusions are given in Section 5.

2 The 4D piece-wise linear cat map (PWLCM)

2.1 Brief recall on 2D PWLCM

Arnold's cat map is basically a 2D chaotic map of repeated folding and stretching in a limited area. It has been popularly used in multimedia chaotic encryption [5]. The 2D ACM is modeled as [23]:

$$\begin{cases} x(t+1) = x(t) + \alpha y(t) \\ y(t+1) = \beta x(t+1) + y(t) \end{cases} \bmod m, \tag{1}$$

which can be rewritten using matrix representation as

$$\mathbf{x}(t+1) = A\mathbf{x}(t) \bmod m \tag{2}$$

where

$$A = \begin{pmatrix} 1 & \alpha \\ \beta & \alpha \cdot \beta + 1 \end{pmatrix},$$

$(\alpha, \beta) \in \mathbb{N}^2_{\geq 1}$, and $\mathbf{x} = (x, y)^T$; $(\cdot)^T$ is the transpose of $(\cdot)$. The above map is a discrete time system and is continuous in the phase space for $(x, y) \in [0, 1)^2$ and $m = 1$. The QACM is obtained for $(x, y) \in [0, m)^2$ with $m \in \mathbb{N}_{>1}$. The QACM is periodic and its period depends on m and the parity of both α and β. It is shown that for $\alpha = \beta = 1$ and $m = 2^n$, the period Π_n behaves like [4, 10]

$$\Pi_n = 2 \cdot \Pi_{n-1},\ n > 2 \tag{3}$$

with $\Pi_1 = \Pi_2 = 3$ for the minimal period.

The period of the QACM can be computed using (3) and it is too short. As an example, for an 8-bit encoded phase space values, the period is $\Pi_8 = 192$. In order to increase the period of QACM, we proposed the PWLCM by introducing a nonlinear perturbation term to the conventional QACM, as shown below [9]:

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + \mathbf{x}_c(t) \bmod m, \tag{4}$$

where the perturbation $\mathbf{x}_c(t)$ is defined as

$$\mathbf{x}_c(t) = \begin{pmatrix} \sum_{i=1}^{M} \Big(a_i + y(t)\Big) \bmod c_i \\ \sum_{j=1}^{N} \Big(b_j + x(t+1)\Big) \bmod d_j \end{pmatrix} \tag{5}$$

with $(i, j) \in \mathbb{N}$. In (4), c_i and d_j are two natural numbers such that $0 \leq c_i < m + a_i$ and $0 \leq d_j < m + b_j$, $0 \leq a_i, b_j < m$ if $(c_i, d_j) = (0, 0)$; $0 \leq a_i < c_i$ if $c_i \neq 0$ and, $0 \leq b_j < d_j$ if $d_j \neq 0$. The parameters a_i, b_j, c_i and d_j are defined as perturbation parameters that can also be used as control parameters, while $\mathbf{a} = (a_i)$, $\mathbf{b} = (b_j)$, $\mathbf{c} = (c_i)$

and $\mathbf{d} = (d_j)$ are control vectors. We showed in [9] that the system in (5) can be put in the form

$$\mathbf{x}(t+1) = B\mathbf{x}(t) + C(t) \mod m \tag{6}$$

where

$$B = \begin{pmatrix} 1 & \alpha' \\ \beta' & \alpha'\beta' + 1 \end{pmatrix}, \tag{7}$$

with $\alpha' = M + \alpha$, $\beta' = N + \beta$ and $C(t) = \Big(C_1(t), C_2(t)\Big)^T$ such that

$$C(t) = \begin{pmatrix} \sum_{i=1}^{M} \Big(a_i - \varepsilon_i c_i \cdot u\Big(a_i + y(t) - c_i\Big)\Big) \\ \beta' C_1(t) + \sum_{j=1}^{N} \Big(b_j - \varepsilon_j d_j \cdot u\Big(b_j + x(t+1) - d_j\Big)\Big) \end{pmatrix}, \tag{8}$$

where $\varepsilon_i = \left\lfloor \frac{a_i + y(t)}{c_i} \right\rfloor$ and $\varepsilon_j = \left\lfloor \frac{b_j + x(t+1)}{d_j} \right\rfloor$. $u(t)$ is the Heaviside function defined as

$$u(t) = \begin{cases} 0, & \text{if } t < 0; \\ 1, & \text{otherwise.} \end{cases} \tag{9}$$

We verified that the PWLCM is a conservative system that exhibit large periods [9]. In order to use it both for image scrambling and diffusion, we suggest its 4D modelling.

2.2 The proposed 4D PWLCM

The above 2D PWLCM can be easily extended to a 4D PWLCM by coupling two 2D PWLCM $\mathbf{x} = (x, y)^T$ and $\mathbf{z} = (q, r)^T$ such that:

$$\begin{cases} x(t+1) = x(t) + \alpha y(t) + F_1(y, t) \\ y(t+1) = y(t) + \beta x(t+1) + F_2(x, t) \\ q(t+1) = q(t) + \gamma y(t+1) + F_3(y, t) \\ r(t+1) = r(t) + \zeta q(t+1) + F_4(q, t) \end{cases} \mod m, \tag{10}$$

where $(\alpha, \beta, \gamma, \zeta) \in \mathbb{N}^4$. $F_1(y, t)$, $F_2(y, t)$, $F_3(y, t)$ and $F_4(y, t)$ are the coupling nonlinear terms, defined as

$$\begin{cases} F_1(y, t) = \sum_{i=1}^{M} \Big(a_i + y(t)\Big) \mod c_i \\ F_2(x, t) = \sum_{j=1}^{N} \Big(b_j + x(t+1)\Big) \mod d_j \\ F_3(y, t) = \sum_{k=1}^{P} \Big(e_k + y(t+1)\Big) \mod g_k \\ F_4(q, t) = \sum_{l=1}^{W} \Big(f_l + q(t+1)\Big) \mod h_l \end{cases} \tag{11}$$

The 4D PWLCM defined in (10) is invertible and the corresponding inverse system is

$$\begin{cases} r(t) = z(t+1) - \zeta q(t+1) - F_4(q, t) \\ q(t) = q(t+1) - \gamma y(t+1) - F_3(y, t) \\ y(t) = y(t+1) - \beta x(t+1) - F_2(x, t) \\ x(t) = x(t+1) - \alpha y(t) - F_1(y, t) \end{cases} \mod m. \tag{12}$$

2.3 Stability analysis

In order to investigate the stability of the 4D PWLCM, we evaluated its Jacobian matrix. By using the same expansion as in (5)–(9), the system can be rewritten as in (6) with

$$B=\begin{pmatrix} 1 & \alpha' & 0 & 0\\ \beta' & 1+\alpha'\beta' & 0 & 0\\ \beta'\gamma' & \gamma'(1+\alpha'\beta') & 1 & 0\\ \beta'\gamma'\zeta' & \gamma'\zeta'(1+\alpha'\beta') & \zeta' & 1 \end{pmatrix}, \tag{13}$$

and $C(t)=\Big(C_1(t),C_2(t),C_3(t),C_4(t)\Big)^T$ such that

$$C(t)=\left(\begin{array}{c} \sum\limits_{i=1}^{M}\Big(a_i-\varepsilon_i c_i\cdot u\Big(a_i+y(t)-c_i\Big)\Big)\\ \beta' C_1(t)+\sum\limits_{j=1}^{N}\Big(b_j-\varepsilon_j d_j\cdot u\Big(b_j+x(t+1)-d_j\Big)\Big)\\ \gamma' C_2(t)+\sum\limits_{k=1}^{P}\Big(e_k-\varepsilon_k g_k\cdot u\Big(e_k+y(t+1)-g_k\Big)\Big)\\ \zeta' C_3(t)+\sum\limits_{l=1}^{W}\Big(f_l-\varepsilon_l h_l\cdot u\Big(f_l+q(t+1)-h_l\Big)\Big) \end{array}\right), \tag{14}$$

From the above equations, we deduce the Jacobian matrix as

$$J=\begin{pmatrix} 1 & J_{1,2}(t) & 0 & 0\\ J_{2,1}(t) & J_{2,2}(t) & 0 & 0\\ J_{3,1}(t) & J_{3,2}(t) & 1 & 0\\ J_{4,1}(t) & J_{4,2}(t) & J_{4,3}(t) & 1 \end{pmatrix}, \tag{15}$$

where

$$\begin{cases} J_{1,1}(t)=\alpha'-\sum_{i=1}^{M}(c_i\delta(y(t)+a_i-c_i))\\ J_{2,1}(t)=\beta'-\sum_{j=1}^{N}(d_j\delta(x(t+1)+b_j-d_j))\\ J_{2,2}(t)=1+J_{2,1}(t)J_{1,2}(t)\\ J_{3,1}(t)=J_{2,1}(t)(\gamma'-\sum_{k=1}^{P}(g_k\delta(y(t+1)+e_k-g_k)))\\ J_{3,2}(t)=J_{2,2}(t)(\gamma'-\sum_{k=1}^{P}(g_k\delta(y(t+1)+e_k-g_k)))\\ J_{4,1}(t)=J_{3,1}(t)J_{4,3}(t)\\ J_{4,2}(t)=J_{3,2}(t)J_{4,3}(t)\\ J_{4,3}(t)=\zeta'-\sum_{l=1}^{W}(h_l\delta(q(t+1)+f_l-h_l)) \end{cases},$$

and $\alpha'=\alpha+M$, $\beta'=\beta+N$, $\gamma'=\gamma+P$, $\zeta'=\zeta+W$. The 4D PWLCM thus defined is conservative as $\det(J)=1$, which implies that the sum of the four corresponding Lyapunov exponents is equal to 0. While computing the eigenvalues of J, we found $\Lambda_1=\Lambda_2=1$, the other eigenvalues are

$$\Lambda_3(t)=1+\frac{1}{2}\Big(\alpha'-\sum\nolimits_{i=1}^{M}c_i\delta\Big(y(t)-\tau_y^i\Big)\Big)\Big(\beta'-\sum\nolimits_{j=1}^{N}d_j\delta\Big(x(t+1)-\tau_x^j\Big)\Big)\left(1+\sqrt{1+\frac{4}{\Big(\alpha'-\sum_{i=1}^{M}c_i\delta\Big(y(t)-\tau_y^i\Big)\Big)\Big(\beta'-\sum_{j=1}^{N}d_j\delta\Big(x(t+1)-\tau_x^j\Big)\Big)}}\right) \tag{16}$$

and

$$\Lambda_4(t) = 1 + \frac{1}{2}\left(\alpha' - \sum_{i=1}^{M} c_i\delta\left(y(t) - \tau_y^i\right)\right)\left(\beta' - \sum_{j=1}^{N} d_j\delta\left(x(t+1) - \tau_x^j\right)\right)$$
$$\left(1 - \sqrt{1 + \frac{4}{\left(\alpha' - \sum_{i=1}^{M} c_i\delta\left(y(t) - \tau_y^i\right)\right)\left(\beta' - \sum_{j=1}^{N} d_j\delta\left(x(t+1) - \tau_x^j\right)\right)}}\right) \tag{17}$$

The Lyapunov exponents corresponding to $\Lambda_{1,2}$ are equal to zero. The sum of the Lyapunov exponents being zero implies that $\Lambda_3 > 1$ (corresponding to a positive Lyapunov exponent) and $0 < \Lambda_4 < 1$ (corresponding to a positive Lyapunov exponent). The steady state of the system depends on the perturbing parameter and remains difficult to formally determine. Figure 1 presents the behavior of the largest Lyapunov exponent for arbitrary parameter setting and various initial conditions (x_0, y_0, q_0, r_0). We fixed $q_0 = r_0 = 1$ and set $z_0 = 2^n x_0 + y_0$, where $0 \leq x, y \leq 2^n - 1$, $a_r = \sum_{i=1}^{N} 2^{n(i-1)} a_N$, and $n = 4$ is the number of bits or precision. Figure 1(a) shows the Lyapunov exponent as a function of initial conditions z_0, where control vectors are set as $\mathbf{a} = \mathbf{e} = (1, 1, 0, 0)$, $\mathbf{b} = \mathbf{f} = (0, 2, 0, 0)$, $\mathbf{c} = \mathbf{g} = (0, 3, 1, 1)$, $\mathbf{d} = \mathbf{h} = (3, 5, 1, 1)$. Figure 1(b) depicts the Lyapunov exponent in terms of the control vector $\mathbf{a}$ represented as parameter a_r, with $N = 3$ where $x_0 = y_0 = 2$, $\mathbf{b} = (0, 2, 1)$, $\mathbf{c} = (15, 15, 15)$, $\mathbf{d} = (3, 5, 1)$, $\mathbf{e} = (1, 1, 0)$, $\mathbf{f} = (0, 2, 0)$, $\mathbf{g} = (0, 3, 1)$, and $\mathbf{h} = (3, 5, 1)$. These plots show that the largest Lyapunov exponent remains positive for the chosen initial conditions and control parameters.

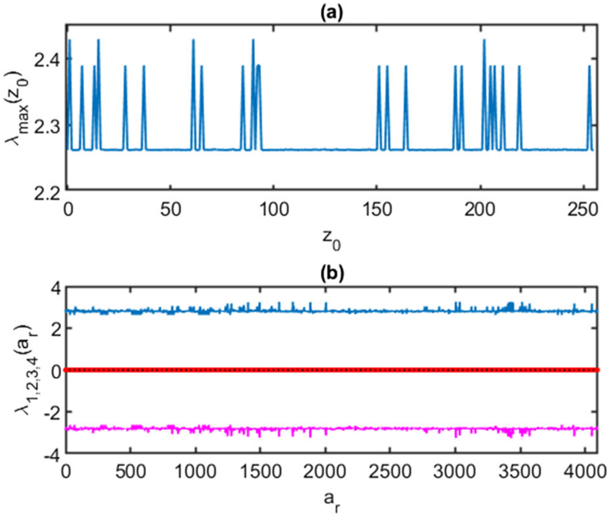

Fig. 1 Lyapunov exponent of the 4D PWLCM: (a) behavior of the Lyapunov exponent $\lambda(z_0)$ with respect to the initial condition $z_0 = 2^n x_0 + y_0$, for $n = 4$ and given control vectors; (b) behavior of the four Lyapunov exponents $\lambda_{1,2,3,4}(a_r)$ with respect to the control vector $\mathbf{a}$, for $n = 4$ and given initial condition and control vectors $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$, $\mathbf{e}$, $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{h}$

Table 1 Comparison of the 2D QACM and 2D PWLCM periods

n	(a_1, b_1)	(c_1, d_1)	Π_{QACM}	Π_{PWLCM}
2	(1,2)	(3,3)	3	90
3	(3,3)	(2,3)	6	780
4	(5,1)	(11,3)	12	12759390
5	(2,2)	(7,5)	24	56934108
6	(6,2)	(5,13)	48	1.7870e+12
7	(2,2)	(7,5)	96	1.2041e+16
8	(6,2)	(5,13)	192	6.918e+56

2.4 Period and randomness evaluation of the PWLCM

Similar to the QACM, the proposed 4D PWLCM is chaotic while used in a continuous phase space ($m = 1$). As we are interested in using it in a finite state phase space ($m = 4$ and $m = 8$), it is no longer chaotic but preserves the mixing properties of the corresponding chaotic systems. For it to be efficient for security applications, its period should be very large. In this subsection, we estimate the period Π_n of the proposed 4D PWLCM for different values of the precision n and some arbitrary values of the perturbation parameters. The periods of PWLCM are compared with the QACM and tabulated in Tables 1 and 2. From the Table 1, we can observe that for any value of n, the period of the proposed PWLCM is significantly higher than that of the corresponding QACM.

Table 2 shows the comparison of the 4D PWLCM and 4D QACM. It confirms that the periods of the proposed map are significantly higher than those of the QACM for any arbitrary parameter values, $\alpha = \beta = \gamma = \zeta = 1, M = N = 1$:

In order to testify the mixing property of the 4D PWLCM, we propose to shuffle a $2^{13} \times 2^{13}$ periodic image obtained by repeating sequences of 8-bit encoded unsigned integers. The image was shuffled with x and y coordinates of the 4D PWLCM. We set as initial conditions, $q_0 = 1$, $r_0 = 2$; $x_0, y_0 \in \left[0, 2^{13} - 1\right]^2$ and controls parameters $a_1 = 1, b_1 = 2, e_1 = 0, f_1 = 1, c_1 = 3, d_1 = 3, g_1 = 3$ and $h_1 = 3$; the corresponding period is 148740480. The NIST800-22 statistical test was performed after 30 iterations of image scrambling with the PWLCM and the QACM. The data set was divided into 100 sets of 1000000 bits and the results obtained are summarized in Table 3. The comparison of the two results confirms that the 4D PWLCM is suitable for the image scrambling as it passes all the tests.

Table 2 Comparison of the 4D QACM and 4D PWLCM periods

n	(a_1, b_1, e_1, f_1)	(c_1, d_1, g_1, h_1)	Π_{QACM}	Π_{PWLCM}
2	(1,2,1,0)	(3,3,3,3)	12	360
3	(6,3,1,1)	(11,1,5,11)	24	240240
4	(6,3,1,1)	(7,1,1,1)	48	340728960

Table 3 NIST 800-22 test results

Sub-Tests	4D QACM		4D PWLCM	
	P-value	Proportion	P-value	Proportion
Frequency	0.0	22/100	0.319084	99/100
Block frequency	0.0	0/100	0.574903	98/100
Cumulative sums (forward)	0.0	0/100	0.289667	100/100
Runs	0.0	0/100	0.657933	99/100
Longest run	0.0	0/100	0.017912	100/100
Rank	0.0	0/100	0.657933	100/100
FFT	0.0	0/100	0.319084	98/100
Non overlapping	0.0	0/100	0.971699	100/100
Overlapping	0.0	0/100	0.574903	99/100
Universal	0.0	0/100	0.066882	100/100
Approximate entropy	0.0	0/100	0.304126	100/100
Random excursions	0.0	0/61	0.957319	61/61
Random excursions variant	0.0	0/61	0.957319	61/61
Serial	0.0	0/100	0.955835	98/100
Linear complexity	0.0	0/100	0.955835	100/100

3 Proposed encryption algorithm

The proposed encryption algorithm has two stages, namely diffusion-confusion and confusion only. In the pixel diffusion-confusion stage, the pixel value of each sub-block is diffused and confused using the 4D PWLCM, whereas in the subsequent block confusion stage, each diffusion-confusion sub-block is split into sub-images that are confused using the 2D PWLCM.

The proposed scheme generates the initial values and the control parameter values of 4D PWLCM from the external key, whereas the control parameters of the 2D PWLCM is derived from the external key along with the diffusion-confusion image. Thus, it is image-dependent. The detailed procedure for generating these parameters is described in Section 3.1.

The algorithmic steps of the proposed cipher are mentioned below:

The block diagram of the proposed image cipher is shown in Fig. 2. The minimum number of rounds for the algorithm to be secure is $R = 2$. Indeed, once the image-dependent step is applied in the first round, we need to go for a second round for the image-dependent shuffling to take effect in the diffusion process, thus increasing the algorithm's sensitivity to the plain-image.

3.1 External key management

The external key is defined by using N_K ASCII characters, $\{C_k\}$, $0 \leq k \leq N_K - 1$, to derive κ-length ($\kappa = 2\lfloor \frac{N_K}{8} \rfloor$) control vectors **a**, **b**, **c**, **d**, **e**, **f**, **g**, **h** whose coordinates are 4-bit encoded unsigned integers. As ASCII characters are 8-bit encoded, each character is divided into two blocks of 4 bits that are used as coordinates of each control vector. Therefore, $\theta = \frac{\kappa}{2}$ ASCII characters are required to determine the coordinates of each control vector. In

1. Consider an $N_L \times N_C$ image and reshape it into a 1D image.
2. Split the 1D image into blocks of length 1 × 256; for 16 × 16 sub-block.
3. Set the external key and deduce control vectors **a**, **b**, **c**, **d**, **e**, **f**, **g**, **h** as κ-length vectors of 4-bit encoded integers as shown in Section 3.1.
4. Set the control vector coordinates corresponding to the sub-image rank as detailed in Section 3.3. Define the corresponding 4D PWLCM as in (22).
5. Decompose the actual pixel into two 4-bit encoded integers $0 \leq q, r \leq 15$ as detailed in Section 3.2.
6. Apply the 4D PWLCM to each pixel by using as input its position referenced as $0 \leq x, y \leq 15$ and its gray-level value represented by $0 \leq q, r \leq 15$.
7. Repeat steps 5-6 until the whole sub-image is modified and consider the next sub-image.
8. Repeat steps 5-7 until the whole image is modified.
9. Use the image obtained from step 8 to define an image-dependent key by defining the 2D PWLCM control parameters as detailed in Section 3.4.
10. Confuse the block positions by using the image dependent 2D PWLCM to obtain a one round modified image.
11. Consider the previous image and go to step 4 for another round.

Algorithm 1

the case of 256-bit key for example, $\theta = 4$ and characters C_0 to C_3 are used to determine coordinates of the control vector **a**, C_4 to C_7 are used for **b**, C_8 to C_{11} for **c**, C_{12} to C_{15} for **d**, C_{16} to C_{19} for **e**, C_{20} to C_{23} for **f**, C_{24} to C_{27} for **g** and C_{28} to C_{31} are used for **h**. In the case of a 2048-bit key, $\theta = 64$ ASCII characters are required to determine each control vector. Therefore, **a** for example is defined as:

$$\begin{cases} \mathbf{a}(2\xi - 1) = \left\lfloor \frac{C_{\xi-1}}{16} \right\rfloor \\ \mathbf{a}(2\xi) = C_{\xi-1} \mod 16 \end{cases} \tag{18}$$

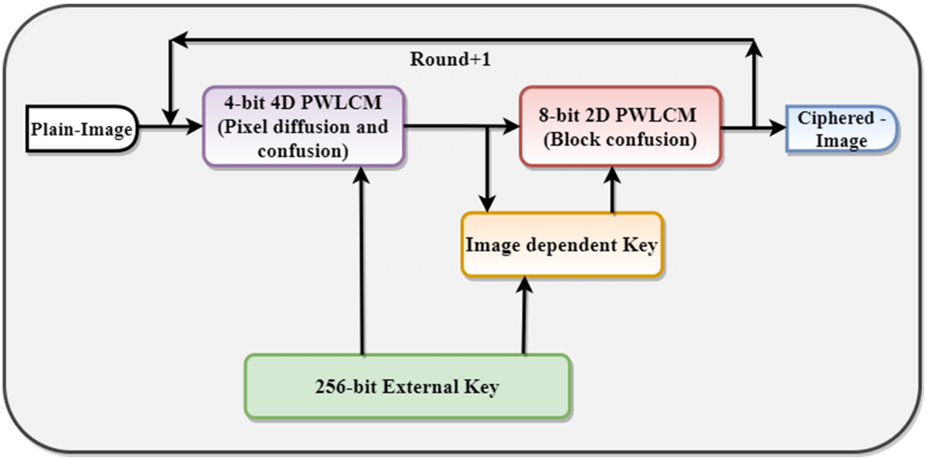
Round+1
Plain-Image
4-bit 4D PWLCM (Pixel diffusion and confusion)
8-bit 2D PWLCM (Block confusion)
Ciphered - Image
Image dependent Key
256-bit External Key

Fig. 2 Synoptic of the proposed cipher

where $1 \leq \xi \leq \theta$. The other vectors **b**, **e** and **f** are defined using the same principle, from the corresponding ASCII symbols. Similarly, **c** is defined as:

$$\begin{cases} \mathbf{c}(2\xi-1) = \mathbf{a}(2\xi-1) + \left\lfloor \frac{C_{\xi+\kappa-1}}{16} \right\rfloor \\ \mathbf{c}(2\xi) = \mathbf{a}(2\xi) + \left(C_{\xi+\kappa-1} \mod 16\right) \end{cases} \tag{19}$$

where $1 \leq \xi \leq \theta$. The coordinates of the other control vectors **d**, **g** and **h** can be defined using the same approach. Thus, control vectors can be considered as belonging to an κ-D lattice.

3.2 Pixel decomposition

In step 5, individual pixel, i of sub-image S_j, $j \in \mathbb{N}$ are decomposed into two 4-bit encoded integers q_i and r_i. For an 8-bit encoded pixel P_i, q_i and r_i are obtained as

$$q_i = \left\lfloor \frac{P_i}{16} \right\rfloor.$$

and

$$r_i = P_i \mod 16$$

3.3 Pixel confusion-diffusion process

In step 6, the confusion and diffusion operations are combined in a single operation (confusion-diffusion). Indeed, the coordinates x_i and y_i, as well as the intensity coordinates q_i and r_i of the pixel P_i are used as initial conditions of the 4D PWLCM to output new coordinates $x_{i'}$, $y_{i'}$, $q_{i'}$ and $r_{i'}$ using (20). As S_j is a vector, there is a relationship between x_i, y_i and i such that

$$x_i = i \mod 16,$$

and

$$y_i = \left\lfloor \frac{i}{16} \right\rfloor.$$

For each sub-image S_j, only a single coordinate $\mathbf{a}(k)$, $\mathbf{b}(k)$, $\mathbf{c}(k)$, $\mathbf{d}(k)$, $\mathbf{e}(k)$, $\mathbf{f}(k)$, $\mathbf{g}(k)$, and $\mathbf{h}(k)$, $k > 0$, is used as control parameter to 4D PWLCM as given below:

$$\begin{cases} x(t+1) = x(t) + \alpha y(t) + (\mathbf{a}(k) + y(t)) \mod \mathbf{c}(k) \\ y(t+1) = y(t) + \beta x(t+1) + (\mathbf{b}(k) + x(t+1)) \mod \mathbf{d}(k) \\ q(t+1) = q(t) + \gamma y(t+1) + (\mathbf{e}(k) + y(t+1)) \mod \mathbf{g}(k) \\ r(t+1) = r(t) + \zeta q(t+1) + (\mathbf{f}(k) + q(t+1)) \mod \mathbf{h}(k) \end{cases} \mod 16, \tag{20}$$

where $k = 1 + j \bmod \kappa$. The new position of the diffused pixel i' is obtained after three iterations of the PWLCM as

$$i' = 16 \cdot y_{i'} + x_{i'} \tag{21}$$

The corresponding intensity value is obtained as

$$P_{i'} = 16 \cdot q_{i'} + r_{i'}. \tag{22}$$

3.4 Image-dependent confusion

In order to enhance the security level of the cipher and prevent chosen-plaintext attacks, an additional image-dependent confusion step is performed using the 2D PWLCM, described as

$$\begin{cases} x(t+1) = \left(x(t) + \alpha y(t) + \sum\limits_{k=1}^{16} \left(\mathbf{a_1}(k) + y(t)\right) \bmod \mathbf{c_1}(k)\right) \bmod m_1 \\ y(t+1) = \left(y(t) + \beta x(t+1) + \sum\limits_{k=1}^{16} \left(\mathbf{b_1}(k) + x(t+1)\right) \bmod \mathbf{d_1}(k)\right) \bmod m_2 \end{cases} \tag{23}$$

where $\mathbf{a_1}$, $\mathbf{b_1}$, $\mathbf{c_1}$ and $\mathbf{d_1}$ are 16-length control vectors whose coordinates are 4-bit encoded values derived from the image of step 8 and the external key as:

$$\begin{cases} \mathbf{a_1}(1:\kappa) = \Gamma \quad \bmod \mathbf{c} \\ \mathbf{a_1}(\kappa+1:2\kappa) = \Gamma \quad \bmod \mathbf{d} \\ \mathbf{b_1}(1:\kappa) = \Gamma \quad \bmod \mathbf{g} \\ \mathbf{b_1}(\kappa+1:2\kappa) = \Gamma \quad \bmod \mathbf{h} \\ \mathbf{c_1}(1:\kappa) = \mathbf{a_1}(1:\kappa) + \Gamma \quad \bmod \mathbf{a} \\ \mathbf{c_1}(\kappa+1:2\kappa) = \mathbf{a_1}(\kappa+1:2\kappa) + \Gamma \quad \bmod \mathbf{b} \\ \mathbf{d_1}(1:\kappa) = \mathbf{b_1}(1:\kappa) + \Gamma \quad \bmod \mathbf{e} \\ \mathbf{d_1}(\kappa+1:2\kappa) = \mathbf{b_1}(\kappa+1:2\kappa) + \Gamma \quad \bmod \mathbf{f} \end{cases} \tag{24}$$

where

$$\Gamma = \sum_{i=1}^{N_L} \sum_{j=1}^{N_C} I_c(i, j). \tag{25}$$

I_c is the intermediate ciphered image obtained in step 8. m_1 and m_2 are defined such that

$$\begin{cases} m_1 = \frac{N_L}{T_1} \\ m_2 = \frac{N_C}{T_2} \end{cases} \tag{26}$$

with $(T_1, T_2) \in \mathbb{N}^2_{\geq 1}$. $T_1 \times T_2$ is the size of sub-images to be shuffled, $m_1 \times m_2$ is the number of sub-images and $N_L \times N_C$ is the size of the image.

4 Results and security analysis

The performance of the proposed encryption algorithm is analyzed by encrypting standard images like “Lena”, “Baboon”, “Airplane”, “Peppers”, of size 512 × 512 and 256 gray levels. Figure 3 respectively represents the plain-text “Lena” image, its encrypted image, and the decrypted image with the same key. All simulations are performed using MATLAB 2018b on a CPU with an Intel(R) Core (TM) i5-8250u CPU @ 1.60 GHz and 8 GB RAM with the Windows 10 operating system. In the current simulation, a 256-bit external encryption key is set as $K1 = \mathit{azertyuiopqsdfgjazertyuiopqsdfg0}$. We also set $\alpha = \beta = \gamma = \zeta = 1$ to make the algorithm multiplierless.

4.1 Evaluation metrics

This subsection presents the definition of evaluation metrics used to measure the proposed cipher’s strength.

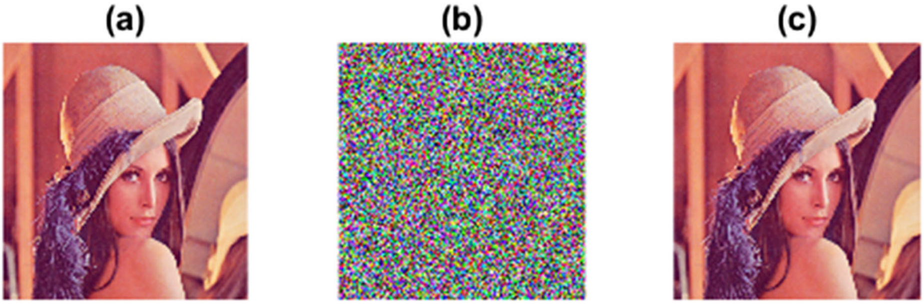

Fig. 3 Example of ciphered image of Lena using the proposed encryption algorithm: (a) plain-text image; (b) ciphered image and (c) decrypted image

4.1.1 Entropy measure

The Shannon entropy of the ciphered image is the primary indicator to confirm that the cipher is secure against permutation of pixels. It measures the disorder or randomness of pixels. For an 8-bit encoded image, it is determined as

$$H = -\sum_{i=0}^{255} p(v_i)\log_2(p(v_i)), \tag{27}$$

where $0 \leq v_i \leq 255$ are pixel values and $p(v_i)$ the probability of v_i. It is to be noted that for an 8-bit encoded image, the maximum value of the entropy is $H = 8$.

4.1.2 Correlation coefficient

The correlation coefficient is used to measure the similarity between two images A and B, and is defined as

$$\rho_{A,B} = \frac{E\Big((A-\overline{A})(B-\overline{B})\Big)}{\sigma_A \cdot \sigma_B}, \tag{28}$$

where $E(\cdot)$ is the expectation value; A and B are images between which the correlation coefficient needs to be evaluated. $\overline{A}$ and $\overline{B}$ represent the mean value of images A and B respectively. Similarly, σ_A and σ_B represent the standard deviation of image A and B respectively. In general, for any plain-text image there exists high correlation between adjacent pixels, whereas in the ciphered image it should be close to 0.

4.1.3 *NPCR* and *UACI* measures

The number of changing pixel rate ($NPCR$) and unified averaged changed intensity ($UACI$) of an image are the commonly used parameters to measure the change in encrypted pixels by modifying the value of a single-pixel in the original image. These metrics are commonly used to evaluate the strength of ciphers for differential attacks. For a 256-gray level image, the NPCR and UACI between two images are defined as

$$NPCR_{A,B} = \frac{\sum_{i=1}^{N_L}\sum_{j=1}^{N_C} D(i,j)}{N_L \times N_C} \times 100 \tag{29}$$

where

$$D(i, j) = \begin{cases} 1, & \text{if } A(i, j) \neq B(i, j); \\ 0, & \text{otherwise.} \end{cases} \tag{30}$$

and

$$UACI_{A,B} = \frac{100}{255} \frac{\sum_{i=1}^{N_L} \sum_{j=1}^{N_C} |A(i, j) - B(i, j)|}{N_L \times N_C} \tag{31}$$

where $\mathscr{F} = 255$ is the largest supported value of 256 gray level images. A high NPCR ($NPCR > 99.5810$) and UACI ($33.3445 \leq UACI \leq 33.5826$) [21] imply a high resistance of the cipher to differential attacks.

4.2 Key-space analysis

The key-space analysis is performed by evaluating the key-space and key sensitivity.

4.2.1 Key-space

Key-space is an ensemble of all possible combinations of keys that are used for encryption. A 256-bit key is known to be sufficiently large to prevent brute force attacks. One of the proposed cipher's main advantages is its key-space's extensibility. To simulate a post-quantum computing attack, we extended the key to 2048 bits. Indeed, most chaos-based ciphers exhibit a large key-space with keys that cannot easily be proven to be different. However, the sensitivity of the key is verified by changing its least significant bit (LSB). However, by our approach, the key-space can be extended as desired without sacrificing the independence of the keys. This approach consists of affecting to each sub-image j a sub-key ($\mathbf{a}(k)$, $\mathbf{b}(k)$, $\mathbf{c}(k)$, $\mathbf{d}(k)$, $\mathbf{e}(k)$, $\mathbf{f}(k)$, $\mathbf{g}(k)$, $\mathbf{h}(k)$) corresponding to a map QACM$_k$, with $k = 1 + j \bmod \kappa$. Each sub-image j is encrypted with a different map QACM$_k$, such that any permutation in the set $\{\text{QACM}_k\}_{1 \leq k \leq \kappa}$ affects the behaviour of the cryptogram, thus giving the possibility to extend the key-space.

4.2.2 Sensitivity of the key

Key sensitivity measures the sensitivity of the encryption algorithm to a small change in the key value. A high sensitivity of the key is required to prevent adaptive chosen-plaintext attacks and linear cryptanalysis. To evaluate the sensitivity of our cipher to the external key, we have encrypted the same image with two slightly different keys $K1$ as mentioned above and $K2 =$ *azertyuiopqsdfghazertyuiopqsdfg1*. The key $K2$ has only one-bit change from key $K1$. The plain-text image is encrypted with keys $K1$ and $K2$; UACI, NPCR, and correlation coefficients between the two encrypted images are tabulated in Table 4. The NPCR and UACI values are more than the reference values, and correlation coefficients close to zero suggest that the algorithm is highly sensitive to the key. In addition, the entropy of the image encrypted with key $K1$ and decrypted with key $K2$ is computed and tabulated in the same table. The entropy close to eight demonstrates that the decryption is unsuccessful; hence, the proposed algorithm is extremely sensitive to the key. We repeated the same experiment with a 2048-bit key. We set the 2048-bit as $K'1 = K1K1K1K1K1K1K1K1$ and a one-bit different key $K'2$ as $K'2 = K1K1K1K1K1K1K1K2$. The sensitivity of the key was evaluated under the same conditions as for the 256-bit key and the results are tabulated in Table 4. It demonstrates that the sensitivity of the key analysis is the same as in the case of the 256-bit key. Therefore, we can conclude that the proposed cipher presents an extensible key space than can be adapted depending on the desired security level.

Table 4 Detailed statistical properties of images encrypted with two slightly different keys $K1$ and $K2$ in the case of 256-bit, and $K'1$ and $K'2$ in the case of 2048-bit

		256-bit				2048-bit			
Image	Color	NPCR	UACI	Entropy	Correlation	NPCR	UACI	Entropy	Correlation
Lena	Red	99.6078	33.4238	7.9994	0.0017	99.6239	33.4431	7.9994	0.0004
	Green	99.6098	33.4896	7.9994	– 0.0015	99.5922	33.4277	7.9991	0.0000
	Blue	99.6136	33.5038	7.9993	– 0.0010	99.6296	33.5368	7.9993	– 0.0043
Baboon	Red	99.6048	33.4987	7.9994	0.0012	99.6002	33.4614	7.9994	0.0004
	Green	99.5956	33.4434	7.9992	0.0006	99.6178	33.5194	7.9992	– 0.0010
	Blue	99.6243	33.4855	7.9992	– 0.0015	99.6094	33.4075	7.9993	0.0017
Airplane	Red	99.6063	33.4631	7.9993	0.0007	99.5899	33.4199	7.9992	0.0004
	Green	99.6208	33.4720	7.9993	– 0.0014	99.6155	33.4583	7.9992	0.0008
	Blue	99.5884	33.5359	7.9993	– 0.0043	99.6162	33.4986	7.9994	0.0005
Peppers	Red	99.6117	33.5103	7.9993	– 0.0017	99.6014	33.5019	7.9993	– 0.0027
	Green	99.6426	33.3958	7.9992	0.0028	99.5899	33.5445	7.9993	– 0.0033
	Blue	99.6315	33.3403	7.9988	0.0039	99.6273	33.4066	7.9993	0.0006

$NPCR$, $UACI$ and entropy are to be compared with reference values $NPCR > 99.5810$ and $33.3445 \leq UACI \leq 33.5826$ and 8 respectively. For this experiment, we used $R = 3$ rounds

In Fig. 4, an example of ciphering/deciphering realized with two ($R = 2$) rounds is presented. The plain-image is encrypted with $K'1$. After that, the ciphered image is successfully decrypted with $K'1$. However, the decryption fails with $K'2$, which attests to the high sensitivity of the cipher to the encryption key.

Security of the proposed cipher, although we set $T_1 = T_2 = 2$. Figure 5 shows the behavior of the entropy values of the RED component of the image of Lena encrypted with $K1$ and those of the corresponding attempt for decryption using $K2$. The entropy is evaluated by varying T_1 and T_2 and plotted in terms of $\mu_1 = \log_2(T_1)$ and $\mu_2 = \log_2(T_2)$ in Fig. 5.

From this figure, we observed that the sensitivity of the key depends on the choice of (T_1, T_2). We verified that the system is secure to one-bit change in the external key as $(\mu_1, \mu_2) < (4, 4)$. The upper limit $\mu_m = 4$ of μ_1 and μ_2 corresponds to the binary logarithm of the block length of the confusion-diffusion step. We observe that the entropy decreases with an increase in T_1 or T_2, leading to a decrease in the key sensitivity. Further, in some cases, entropy values are close to those of the plain-image entropy ($H_R = 7.2531$). It attests to the vulnerability of the proposed scheme for these values of μ_1 and μ_2. In the proposed method, such a sensitivity decrease is compensated by increasing the number of rounds of the algorithm.

4.3 Statistical analysis

The histogram, the correlation of adjacent pixels (ρ_h : correlation coefficient between horizontal adjacent pixels; ρ_v : correlation coefficient between vertical adjacent pixels; ρ_d : correlation coefficient between diagonal adjacent pixels) and the information entropy of the ciphered image are evaluated for several 256 gray-scale images. Figure 6 shows the results

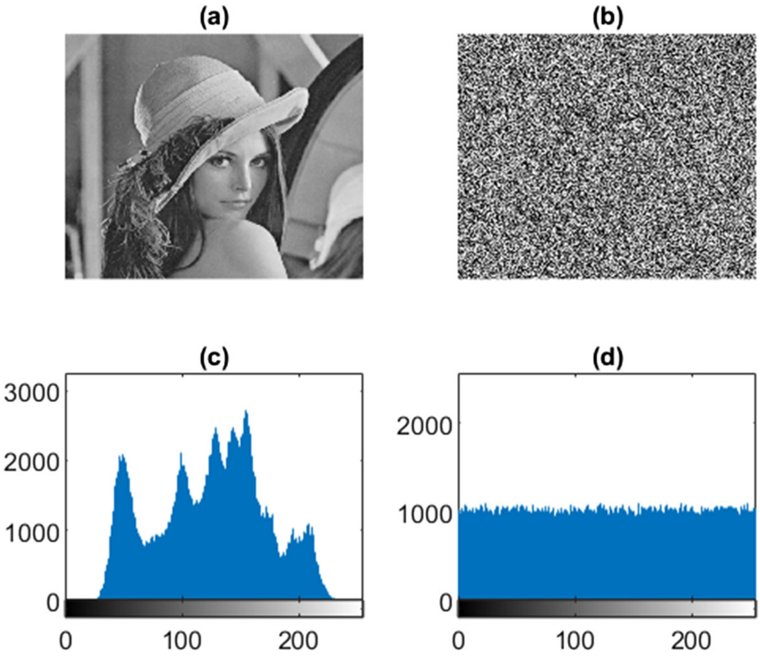

Fig. 4 Sensitivity of the key to one-bit change: (a) successfully decrypted (original) image with $K'1$, (b) unsuccessfully decrypted image with $K'2$, (c) histogram of the original image and (d) histogram of the unsuccessfully decrypted image

obtained for the color image of Lena (Fig. 4(a)) by varying number of rounds R between 1 and 10. Figure 6 suggests that the entropy is independent of the number of rounds R for $R > 1$, thereby attests that the minimum number of rounds required for the cipher to be secure is $R = 2$. Similarly, it also shows a satisfactory analysis result for the correlation of horizontally (ρ_h), vertically (ρ_v), and diagonally (ρ_d) adjacent pixels. Indeed, it can be concluded that the correlation coefficients of the three image components, i.e., RED, GREEN, and BLUE, are close to zero, independently of the number of rounds R. Thus, the proposed cipher satisfies the zero-correlation property necessary to resist statistical attacks.

Figure 7 shows the histograms of two round ciphered images of Lena for the RED, GREEN, and BLUE components. It appears that the histogram of each encrypted component is fairly uniform and significantly different from that of the corresponding plain-image component. It demonstrates that deducing the secret key from the ciphertext during the known/chosen plaintext attacks is hard.

4.4 Differential attack

For the cipher to resist differential attacks, it must be sensitive to a small change (single-pixel change) in the plain-image. We evaluated the robustness of our cipher against the differential attacks by comparing the $NPCR$ and $UACI$ values of a two-round ciphered image of Lena to their reference values. We diagonally varied the position (x, y) of changed pixels as (k, k), with $1 \leq k \leq 512$ by the step size of 2, and computed the corresponding $NPCR$ and $UACI$ for the RED, GREEN and BLUE components of the color image of Lena, and plotted in Fig. 8.

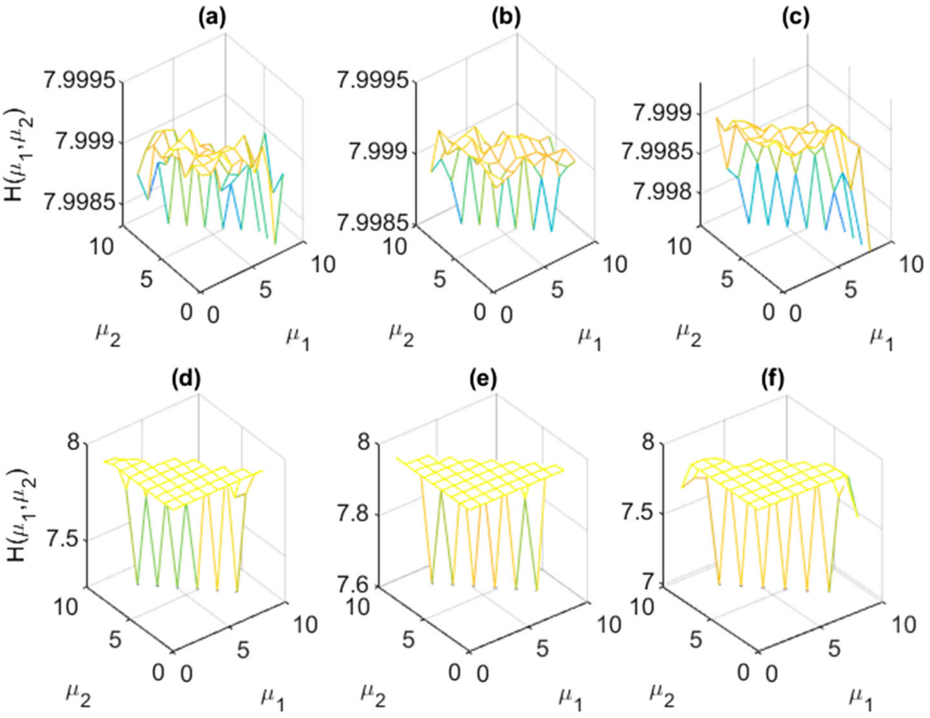

Fig. 5 Behaviour in term of μ_1 and μ_2 of entropy values of the ciphered and decrypted components of the color image of Lena($R = 2$): (a)-(c) cases of the respective RED, GREEN and BLUE components encrypted with $K1$; and (d)-(f) cases of attempt for decryption with $K2$ for RED, GREEN and BLUE components, respectively. Satisfactory entropy values after an attempt for decryption are observed for $(\mu_1, \mu_2) < (4, 4)$

A cipher is secure as $NPCR > 99.5810$ and $33.3445 \leq UACI \leq 33.5826$ ($\nu = 0.01$ significance level) for gray images of size 512 × 512 [21]. The result in Fig. 8 shows that our cipher is sensitive to one-pixel change for $R > 1$. From this figure, it appears that the proposed cipher can resist differential attacks as the $NPCR$ and $UACI$ remain in the good range independently of the coordinate of the pixel change. We also observed that the values of the $NPCR$ and $UACI$ depend on the input image. Indeed, for two different components, the $NPCR$ values obtained from the same pixel change coordinate are different; the same observation is made for the $UACI$. Table 5 shows comparative values of the proposed cipher's $NPCR$ and $UACI$ and other recent cipher ($R = 3$). This comparison also confirms the effectiveness of our algorithm.

4.5 Speed performance analysis

The execution speed of the proposed cipher is evaluated using Matlab 18b in the CPU as specified above. Table 6 compares the execution speed of the proposed algorithm with other chaos-based ciphers. We used the gray-scale images of cameraman (256 × 256) and Lena (512×512) for this experiment. The average execution time, for $R = 2$, is about 0.4478 s for the 512 × 512 gray-scale images. Although this speed is comparable to that of Refs. [1] and [42], it remains low as compared to the one obtained in [13]. Such a low execution speed is justified by the multiple data conversion involved in the algorithm, which is not necessary

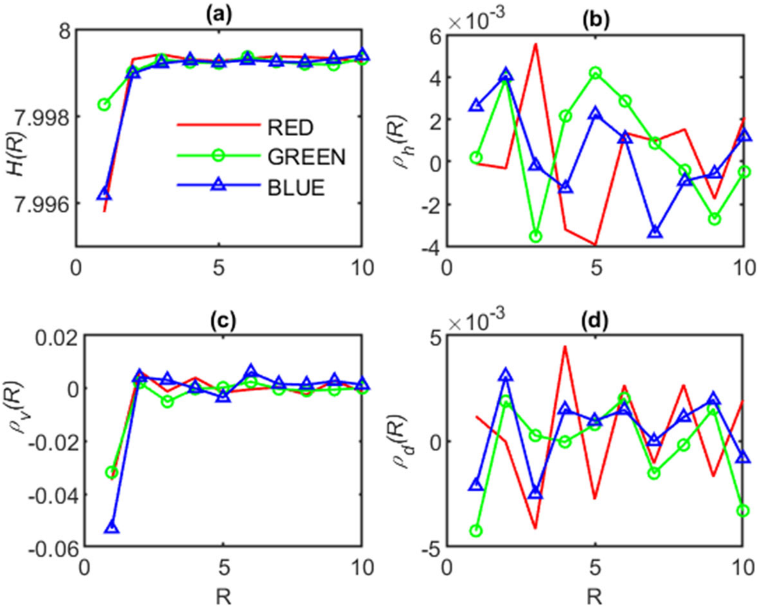

Fig. 6 Entropy values H and correlation coefficients ρ in terms of the number of rounds R. (a) Entropy values (H), (b)-(d) correlation coefficients of respectively horizontal (ρ_h), vertical (ρ_v) and diagonal (ρ_d) adjacent pixels for the RED, GREEN and BLUE components

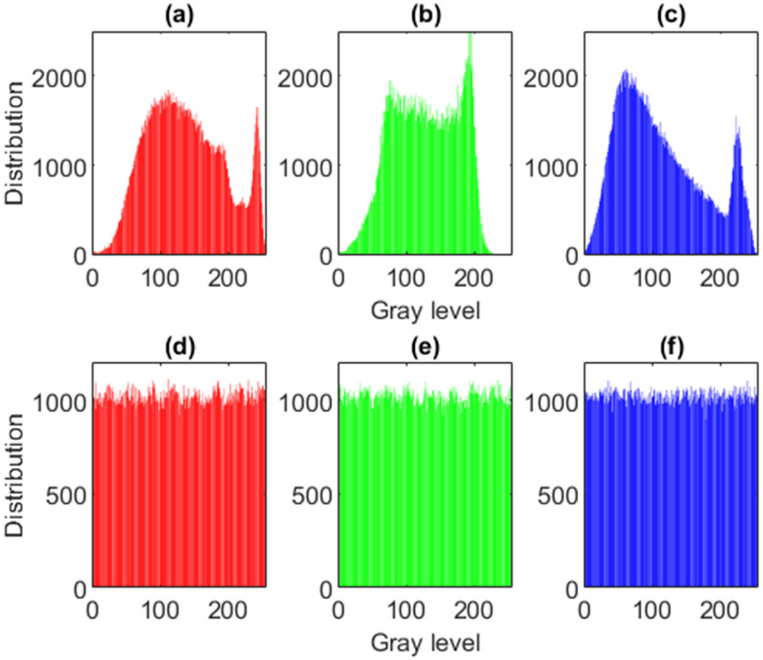

Fig. 7 Histograms of the image of Lena: first line shows from left to right histograms of the (a) RED, (b) GREEN and (c) BLUE components of the original image, respectively; second line (d)-(f) shows histograms of the corresponding components of the ciphered image

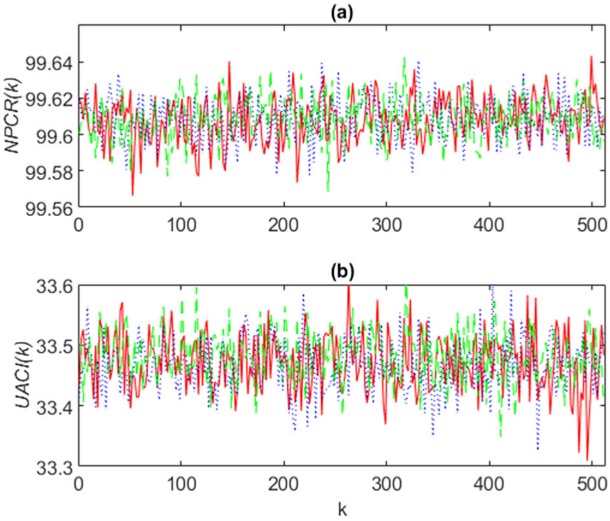

Fig. 8 Dependence of NPCR and UACI on position $((k, k))$ of the changed pixel: case of the color image of Lena. The change applies to each image component and the NPCR of the RED, GREEN and BLUE components are computed separately for each of the components

for hardware implementation. For example, the decomposition of the pixel gray-level into two 4-bit encoded integers $q(t)$ and $r(t)$ implies a division and a modulo operation. In the hardware implementation, these operations correspond to the four most significant bits (MSB) as the quotient $q(t)$ and the four least significant bits (LSB) as the remainder $r(t)$, thereby is time-saving. Similarly, the inverse of this decomposition is time costly in Matlab and corresponds to the concatenation of the two 4-bit outputs $q(t + 1)$ and $r(t + 1)$ in the hardware implementation. Thus, the proposed algorithm is more suitable for low-cost hardware implementation than software implementation. The overall comparison shows that the cipher in Ref. [13] is running faster in the Matlab environment than in ours; however, it requires floating-point arithmetic that is hardware costly than the 4-bit integer arithmetic used in the proposed scheme. Our algorithm offers the advantage of combining only 4-bit and 8-bit integer arithmetic operations, precisely addition and subtraction. These basic operations make simpler its hardware implementation even with low-end microprocessors

Table 5 Comparison of $NPCR$ and $UACI$ values for color images

		$NPCR$ (%)			$UACI$ (%)		
Cipher	Image	RED	GREEN	BLUE	RED	GREEN	BLUE
Proposed	Baboon	99.6429	99.6437	99.6464	33.5764	33.5881	33.6311
	Peppers	99.6479	99.6410	99.6403	33.6121	33.5629	33.5863
Ref. [32]	Baboon	99.6246	99.5914	99.5972	33.4702	33.4641	33.4625
	Peppers	99.6315	99.6017	99.6380	33.5577	32.7183	33.5702
Ref. [40]	Baboon	99.6056	99.6164	99.6256	33.4478	33.4495	33.4401
	Peppers	99.6098	99.6218	99.5948	33.5012	33.4415	33.4536

Table 6 Comparison of encryption time of different algorithms

Image size	Execution time (s)			
	Proposed cipher	Ref. [13]	Ref. [1]	Ref. [42]
256 × 256	0.1335	0.0202	0.1789	0.1950
512 × 512	0.4478	0.0708	0.6639	0.6500

The average execution time (in second) of the proposed cipher are obtained with $R = 2$ and $K1$ as encryption key

without losing its security properties. Moreover, the algorithm can easily be parallelized according to its architecture shown in Fig. 2.

5 Conclusion

This paper proposed an extendable integer image cipher based on the combined 4-bit and 8-bit PWLCM. The cipher does not require any multiplication operation. The PWLCM is obtained by perturbing the convectional QACM and presents an extended period. The key space extensibility is achieved using different lattice length (κ) of the PWLCM control vectors. The extended key space helps to avoid key duplication. We evaluated the sensitivity of the key under 256-bit and 2048-bit key conditions and verified the high sensitivity of the key for both cases. Such flexibility for the extensibility of the key-space makes the proposed cipher a good candidate to resist brute-force attacks even under the post-quantum computing situation. The main advantages of the proposed cipher are the low number of bits involved in its hardware implementation and the extensibility of its key-space. Moreover, only unsigned integer addition and subtraction operations are used, contrary to other chaos-based ciphers that involve floating-point arithmetics, including time-consuming operations like multiplications and divisions. The statistical, differential, and key-space analysis demonstrate the robustness of the proposed cipher against known attacks. We further expect to reduce the block size while maintaining a high-security level of the cipher, simplifying its implementation with low-end processors under the limited memory space constraints.

Acknowledgements This work was partially supported by the Erasmus+ program under Ref KA107 Call 2019.

We thank the reviewers for their valuable comments.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability Data sharing not applicable to this article as no datasheets were generated or analyzed during the current study.

Declarations

Conflict of Interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Ahmed A, El-Latif A, Li L, Niu X (2014) A new image encryption scheme based on cyclic elliptic curve and chaotic system. Multimed Tools Appl 70:1559–1584
2. Atawneh S, Almomani A, Bazar H, Sumari P, Gupta BB (2017) Secure and imperceptible digital image steganographic algorithm based on diamond encoding in DWT domain. Multimed Tools Appl 76:18,451–18,472
3. Bajard JC, Eynard J, Merkiche N (2018) Montgomery reduction within the context of residue number system arithmetic. J Cryptogr Eng 8:189–200
4. Chen F, Wong KW, Liao X, Xiang T (2012) Period distribution of generalized discrete Arnold cat map for $n = p^e$. IEEE Trans Inf Theory 58:445–452
5. Chui CK, Chen G, Mao Y (2004) A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons Fractals 21:749–761
6. Crutchfield JP (1998) Spatio-temporal complexity in non linear image processing. IEEE Trans Circ Syst 35:770–780
7. Cui M, Tong X (2008) Image encryption with compound chaotic sequence cipher shifting dynamically. Image Vis Comput 26:843–850
8. Didier LS, Dosso FY, Véron P (2020) Efficient modular operations using the adapted modular number system. J Cryptogr Eng 10:111–133
9. Djeugoue H, Gnyamsi GG, Eyebe Fouda JSA, Koepf W (2022) On the implementation of large period piece-wise linear Arnold cat map. Multimed Tools Appl. https://doi.org/10.1007/s11042-022-13175-6
10. Dyson FF, Falk H (1992) Period of a discrete cat mapping. Am Math Mon 99:603–614
11. Elshamy AM, Hussein AI, Hamed HFA, Abdelghany MA, Kelash HM (2019) Color image encryption technique based on chaos. Procedia Comput Sci 163:49–53
12. Eyebe Fouda JSA, Koepf W (2022) An 8-bit precision cipher for fast image encryption. Multimed Tools Appl. https://doi.org/10.1007/s11042-022-12368-3
13. Eyebe Fouda JSA, Effa JY, Ali M (2014) A fast chaotic block cipher for image encryption. Commun Nonlinear Sci Numer Simulat 19:578–588
14. Eyebe Fouda JSA, Effa JY, Ali M (2014) Highly secured chaotic block cipher for fast image encryption. Appl Soft Comput 25:435–444
15. Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurcat Chaos 8:1259–1284
16. Gu G, Linga J (2014) A fast image encryption method by using chaotic 3d cat maps. Optik 125:4700–4705
17. Herbert V, Biswas B, Fontaine C (2019) Design and implementation of low-depth pairing-based homomorphic encryption scheme. J Cryptogr Eng 9:185–201
18. Hua Z, Zhou Y, Huang H (2019) Cosine-transform-based chaotic system for image encryption. Inf Sci 480:403–419
19. Ilan Y (2019) Generating randomness: making the most out of disordering a false order into a real one. J Transl Med 17(49):1479–5876
20. Kaixin J, Ye G, Dong Y, Huang X, He J (2020) Image Encryption Scheme based on Generalized Arnold Map and RSA Algorithm. Secur Commun Netw 2020:9721,675
21. Kang S, Liang Y, Wang Y, VI M (2019) Color image encryption method based on 2D-variational mode decomposition. Multimed Tools Appl 78:17,719–17,738
22. Kaur G, Agarwal R, Patidar V (2020) Chaos based multiple order optical transform for 2d image encryption. Eng Sci Technol Int J 23:998–1014
23. Keating JP, Mezzadri F (2000) Pseudo-symmetries of Anosov map and spectral statistics. Nonlinearity 13:747–775

24. Kopparthi VR, Kali A, Sabat SL, Anumandla KK, Rangababu P, Eyebe Fouda JSA (2022) Hardware architecture of a digital piecewise linear chaotic map with perturbation for pseudorandom number generation. Int J Electron Commun (AEÜ) 147:154,138
25. Li D, Deng L, Gupta BB, Wang H, Choi C (2019) A novel cnn based security guaranteed image watermarking generation scenario for smart city applications. Inf Sci 479:432–447
26. Lian S, Sun J, Wang Z (2005) A block cipher based on a suitable use of the chaotic standard map. Chaos, Solitons Fractals 26:117–129
27. Liao X, Xiang T, Wong KW (2007) Selective image encryption using a spatio temporal chaotic system. Chaos 17:0231,151–0231,512
28. Malina L, Popelova L, Dzurenda P, Hajny J, Martinasek Z (2018) On feasibility of post-quantum cryptography on small devices. IFAC PapersOnLine 51-6:462–467
29. Mondal B, Behera PK, Gangopadhyay S (2021) A secure image encryption scheme based on a novel 2D sine–cosine cross-chaotic (SC3) map. J Real-Time Image Proc 18:1–18
30. Panwar K, Purwar R, Jain A (2018) Cryptanalysis and improvement of an image encryption scheme using combination of one-dimensional chaotic maps. J Electron Imaging 27:053,037
31. Pareek NK, Patidar V, Sud KK (2006) Image encryption using chaotic logistic map. Image Vis Comput 24:926–934
32. Patro KAK, Acharya B (2019) An efficient colour image encryption scheme based on 1-D chaotic maps. J Inf Secur Appl 46:23–41
33. Ping P, Fan J, Mao Y, Xu F, Gao J (2018) A chaos based image encryption sheme using digit-level permutation and block diffusion. IEEE Access 6:67,581–67,593
34. Ping P, Xu F, Mao Y, Wang Z (2018) Designing permutation-substitution image encryption networks with Henon map. Neurocomputing 283:53–63
35. Schoinianakis D (2020) Residue arithmetic systems in cryptography: a survey on modern security applications. J Cryptogr Eng 10:249–267
36. Wang M, Wang X, Zhang Y, Zhou S, Zhao T, Yao N (2019) A novel chaotic system and its application in a color image cryptosystem. Opt Lasers Eng 121:479–494
37. Wikramaratna RS (2008) The additive congruential random number generator—a special case of a multiple recursive generator. J Comput Appl Math 216:371–387
38. Wu Y (2012) Image encryption using the two-dimensional logistic chaotic map. J Electron Imag 21:013,014
39. Ye G, Huang X (2016) A secure image encryption algorithm based on chaoticmaps and SHA-3. Secur Comm Netw 9:2015–2023
40. Zhang Y, He Y, Li P, Wang X (2020) A new color image encryption scheme based on 2DNLCML system and genetic operations. Opt Lasers Eng 128:106,040
41. Zhao Y, Gao C, Liu J, Dong S (2019) A self-perturbed pseudo-random sequence generator based on hyperchaos. Chaos, Soliton Fractals: X 4:100,023
42. Zhu ZL, Zhang W, Wong KW, Yu H (2011) A chaos-based symmetric image encryption scheme using a bit-level permutation. Inf Sci 181:1171–1186
43. Zhua H, Zhao C, Zhanga X, Yanga L (2014) An image encryption scheme using generalized Arnold map and affine cipher. Optik 125:6672–6677

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Gaetan Gildas Gnyamsi Nkuigwa[1] · **Hermann Djeugoue Nzeuga**[1] · **J. S. Armand Eyebe Fouda**[1,2] · **Samrat L. Sabat**[3] · **Wolfram Koepf**[2]

Gaetan Gildas Gnyamsi Nkuigwa
gaetangildas@yahoo.fr

Hermann Djeugoue Nzeuga
mahernzeuga@yahoo.fr

Samrat L. Sabat
slssp@uohyd.ac.in

Wolfram Koepf
koepf@mathematik.uni-kassel.de

[1] Départment de Physique, Université de Yaoundé I, Yaoundé, Cameroun

[2] Institute of Mathematics, University of Kassel, Kassel, Germany

[3] Centre for Advanced Studies in Electronics Science and Technology, University of Hyderabad, Hyderabad, India