Image encryption based on fractional chaotic pseudo-random number generator and DNA encryption method

Chunxiao Yang, Ina Taralova, Safwan El Assad, Jean-Jacques Loiseau

Noname manuscript No.
(will be inserted by the editor)

Image encryption based on fractional chaotic pseudo random number generator and DNA encryption method

Chunxiao YANG · Ina TARALOVA · Safwan EL ASSAD · Jean-Jacques LOISEAU

Received: date / Accepted: date

Abstract Nonlinear dynamic systems and chaotic systems have been quite exhaustively researched in the domain of cryptography. However, the possibility of using the fractional chaotic system in the cryptosystem design has been much less explored while it bears advantages such as enlarged keyspace, compared to classical nonlinear systems.

This paper, therefore, proposes a novel structure for the pseudo- random number generator based on fractional chaotic systems which consists of 3 different fractional chaotic systems, namely fractional Chen's system, Lu's system, and fractional generalized double-humped logistic map(FGDHL). Then, the outputs of this fractional chaotic pseudo-random number generator(FCPRNG) are used as a keystream for an image encryption scheme. The *confusion* layer of the scheme is conducted by a dynamic DNA encoding and decoding method combined with a 2D cat map for the permutation in the DNA-bases level. The *diffusion* layer is performed through the adoption of a 32 bits discrete logistic map. The performance and security analysis have been conducted for the above-designed cryptosystem, proving that the proposed cryptosystem is practical and secure in image encryption.

Chunxiao YANG
LS2N, UMR CNRS 6004, Ecole Centrale de Nantes, Nantes, France
E-mail: chunxiao.yang@ls2n.fr

Ina TARALOVA
LS2N, UMR CNRS 6004, Ecole Centrale de Nantes, Nantes, France
E-mail: ina.taralova@ls2n.fr

Safwan EL ASSAD
IETR, UMR CNRS 6164, Université de Nantes/Polytech Nantes, Nantes, France
E-mail: safwan.elassad@univ-nantes.fr

Jean-Jacques LOISEAU
LS2N, UMR CNRS 6004, Ecole Centrale de Nantes, Nantes, France
E-mail: jean-jacques.loiseau@ls2n.fr

Keywords fractional chaotic system · pseudo-random number generator · DNA encoding and decoding · image encryption

1 Introduction

Humans are now entering the Information Explosive Era with an unstoppable speed in an irreversible way due to the rapid advancement of information technology and digital communication techniques. Millions of gigabits of digital information are being transmitted every second via the internet and communication systems. In particular, under the still ongoing pandemic, much work has been done through remote working with lower security systems or computers[1]. In addition, after tasting the economic benefits gained from having their employees working remotely, many companies have now abandoned the traditional onsite working protocols to trim away their expenses on the fixed cost such as electricity and the rental fee for the workplace. All of these lead us to the point where people work more and more at home, generally in an informationally non-secure working environment, which heavily endangers the integrity of the transmitted information[2]. The confidentiality of the digital data people process through the internet or other communication systems, and the information stored locally in the personal or company computer is hence of great importance. Moreover, the standardization and generalization of sanitary passes, and many other applications aimed at preventing and controlling the epidemic during the post-epidemic era also calls for data security enhancement. Therefore, the urge to design advanced and secure cryptographic systems is more vital than ever before.

It goes without saying that a sufficiently sophisticated cryptosystem (consisting of key generation, encryption algorithm, and decryption algorithm), encrypting personal and(or) private information, is supposed to stay safe from hackers' attacks[3]. To achieve this goal, cryptosystem with *novel structures* and stronger *cipher* (especially the *encryption algorithm*) must be introduced and developed.

Concerning a secure cryptosystem's *structure*, a significant point is to be able to resist well-known attacks such as brute force attacks, chosen-plaintext attacks, and statistical attacks. The solution to resist brute force attacks is to design an encryption system with sufficiently large size of keyspace (each different key gives rise to a qualitatively dynamic different behavior). The keystreams generated by different keys should be independent and uniformly distributed[4]. Concerning this issue, new trends with less known but very promising non-linear functions such as fractional chaotic functions and the fractional chaotic pseudo-random number generator(FCPRNG) have started to be explored. The use of the latter assures that the pseudo-random and chaotic properties are preserved. Though other researchers also sought to design pseudo-random number generators(PRNG) using adaptive maps, such as Zaslasvisky and Chirikov adaptive map[47] and [48] with extra parameters to

enhance the cryptosystem security, to the best of our knowledge, none of them has discussed the encryption issues.

Historically speaking, fractional calculus as a branch of mathematical analysis has already been established and discussed purely in the scope of mathematics for centuries[5][6]. In the more recent decades, some researchers in the field of science and engineering have seized the opportunity to take advantage of the memory effect and the hereditary properties[7] possessed by the fractional calculus and systems to study their engineering applications. Other recent studies have also proved that the use of fractional dynamic systems in defining and modeling real-life systems is applicable and suitable in many disciplines such as physics[8], biology, economics[9], etc.

The obstacles that hinder the broader implementation of fractional systems, especially for the use of fractional chaotic systems for cryptosystem security enhancement, arise from the intrinsic complexity of fractional calculus and the aspects of its digital implementation. In addition, our recent research works [10][11] have revealed that applying different numerical calculation methods and adopting different fractional calculus characterizations may strongly impact the (short or long term) chaotic behavior of the resulting system. This renders the design of efficient FCPRNG a very challenging problem. As far as we know, no one else except us has ever accomplished the PCPRNG design, which can successfully pass the standard NIST randomness test suites.

Along with the above mentioned significant obstructions, one should also note the outstanding merits of the fractional chaotic system. From the cryptosystem design point of view, the inherent properties of the fractional chaotic systems can increase the complexity of the encryption scheme[12]. What's more, the security of the cryptosystem will be further enhanced with the employment of fractional chaotic systems. The fractional order may differ for each state component of the system, and it works as extra parameters, enlarging the size of the keyspace of the cipher. These remarkable features perfectly satisfy the increasingly solid needs for secure cipher algorithms and motivate the researchers to investigate the introduction of fractional chaotic systems in the cryptosystem design.

Looking at the recent research work in the field of cryptography when employing fractional order chaotic systems, very few papers analyzed the pseudo-random features of the systems' outputs which is a fundamental feature for the generated keystreams. In [13], authors proposed an encryption scheme based on fractional-order Lorenz system and a simple algorithm based only on pixel confusion. In [14], authors explored and analyzed the characteristic of a new fractional-order complex system based on the Adomian decomposition method (ADM) with a novel image encryption algorithm proposed applying the discussed system and Galois field (GF). An "improper fractional chaotic system" was constructed and realized based on the DSP platform in [15], an image encryption algorithm was then structured and analyzed. In a recently published paper [16], a different encryption scheme based on a fractional hyperchaotic

system has been investigated where multiple grayscale images were fused into a color image and then scrambled and diffused.

All the above-mentioned papers deal with only one unique fractional chaotic system at a time. Most of the papers([13][14][15]) directly use the system's states as the keystream to permutate the image pixels as a scrambling procedure for the cryptosystem. Hence, the complexity of the generated keystream and the encryption scheme was relatively minor. Moreover, the chaotic and pseudo-random properties of the system's outputs over the parameters have not been tested. The increase of keyspace size, which some authors deemed (such as in [13] and [16]), was, therefore, not guaranteed. With respect to this problem, the implementation of the FCPRNG appears as a perfect solution. Indeed, as a pseudo-random number generator, its outputs must firstly be independent and secondly follow a uniform distribution; hence, a complex structure combining different fractional systems is usually adopted. In addition, different pairs of parameters (secret keys) must be tested for randomness as a requirement of the international standard for the design of pseudo-random number generators, which leads to the assertion of the homogeneous properties[17] over the keyspace for the cryptosystem.

As for the *encryption algorithm* of the cryptosystem, a new trend has emerged using DNA computing which has received unceasing attention [18][19] [20][21] [22]. Since the first experiment of DNA computing conducted by Adleman[23] in 1994, the idea of implementing DNA computing in the domain of cryptography reveals itself due to its numerous advantages such as massive parallelism, huge storage capacity, and energy consumption[24]. Under the belief that with the development and advance of biochemistry, the difficulties and limitations of DNA cryptography, which now exist at the experimental level, will be conquered in the foreseeable future. Many researchers have been triggered to explore the possibility of designing a good and practical encryption algorithm using DNA cryptography[24][25][26][27].

Regarding the application of image encryption, a dynamic DNA-based cipher with two chaotic maps, namely fractional Chen's system and Lorenz system, has been investigated in [28]. In [29], the authors analyzed an encryption algorithm combining a novel proposed 2D Hénon-sine map DNA ciphering methods and achieved better performance in terms of security. A new one-time pad encryption scheme has been designed based on the coupled map lattices (CML system) and DNA diffusion sequence in [30] with good security analysis results. In [31], authors proposed a color light field image encryption algorithm using DNA sequence and well-known chaotic systems(logistic map and Chen system), which is proved to be applicable, reliable, and secure enough. All these encryption schemes employed structures based on chaotic systems, which to some extent, render the security of the cryptosystem dependent on the performance of the chaotic systems. Besides, the papers brought in complex and cumbersome encryption operations, whose necessity and effectiveness are inconclusive, to overcome the shortcomings of DNA ciphering methods, such as poor diffusion properties.

To palliate these problems, in this paper, we introduce for the first time an original FCPRNG comprised of three different fractional chaotic systems to generate pseudo-random numbers. In addition, a novel image encryption algorithm based on DNA ciphering methods is also proposed and analyzed. The outputs of the FCPRNG work as the keystreams for the cryptosystem. The dynamic DNA encoding and decoding methods with the benchmark cat map are performed to achieve the permutation operation. The diffusion is conducted by a straightforward but efficient logistic diffusion process. The two essential components for this cryptosystem, the FCPRNG, and the cipher, are discussed separately in the following.

2 Preliminaries

In this part, we shall recall some preliminaries on fractional calculus, fractional systems and DNA computing.

2.1 Fractional calculus and fractional systems

2.1.1 fractional calculus

Fractional calculus discusses the integrals and derivatives of non-integer order. It is a generalization of integration and differentiation to non-integer order fundamental operator ${}_aD_t^\alpha$, where α is the non-integer order(fractional order), a and t are the bounds of the operation[32]. Along with the long existence of fractional calculus, various characterizations (also indicated as 'definitions' in many works) have been developed, such as Grünwald-Letnikov(GL) characterization, Riemann-Liouville(RL) characterization, Caputo characterization, and etc. These different characterizations can be equivalent when certain bounds and conditions are satisfied.

Hereafter, we list out two characterizations for fractional integrator and derivative, which are adopted for our fractional pseudo chaotic random number generator design. For a more comprehensive introduction on this topic, one can refer to [32] and other textbooks such as[33].

The fractional integral of fractional order α ($\alpha > 0$) under RL definition is described as follows,

$$ {}_aI_t^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} f(\tau) d\tau \tag{1} $$

The formula is a generalization of the standard integral, which is the particular case of RL integral when $\alpha = 1$. $\Gamma(.)$ in (1) represents the Euler Gamma function which is expressed as below,

$$ \Gamma(\alpha) = \int_0^\infty \frac{t^{\alpha-1}}{e^t} dt \tag{2} $$

The RL definition for the fractional derivative is the left inverse of ${}_aD_t^\alpha$ and is described using the formula below,

$$\begin{aligned} {}_aD_t^\alpha f(t) &= D^n {}_aI_t^{n-\alpha} f(t) \\ &= \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \end{aligned} \tag{3}$$

where $n = \lceil \alpha \rceil$ represents the smallest integer greater or equal to α. D^n denotes the standard integer-order derivative. a and t are the limits of operation ${}_aD_t^\alpha$. It is to be remarked that for a causal function $f(t)$, when $t < 0$, $f(t)$ is equal to 0, and we have $a = 0$. Therefore, a fractional derivative in the Caputo sense with $f(t)$ being causal can be defined as follows[34],

$$\begin{aligned} D^\alpha f(t) &= I^{n-\alpha} D^n f(t) \\ &= \frac{1}{\Gamma(n-\alpha)} \int_0^t (t-\tau)^{n-\alpha-1} f^{(n)}(\tau) d\tau \end{aligned} \tag{4}$$

where $n - 1 \leq \alpha \leq n$, $t \geq 0$.

The Caputo definition is widely applied in engineering applications due to the fact that the fractional differential equations of the Caputo type are suitable in providing the applied problems with clearly interpretable initial conditions.

2.1.2 Fractional systems

The fractional system, as briefly explained in [33], is the dynamic system that can be modeled by differential equations with non-integer order derivatives. Hereafter, we illustrated the form of differential equation in the sense of Caputo characteristics as follows,

$$\begin{aligned} &D^\alpha x(t) = f(t, x(t)) \\ &x^{(k)}(0) = x_0^k, k = 0, 1, 2, ..., n-1. \end{aligned} \tag{5}$$

where $n = \lceil \alpha \rceil$ denotes the fractional order ceiling, and $x^{(k)}(0)$ represents the initial conditions for k-th order.

The fractional system can be expressed by a series of fractional differential equations. In our following work, the systems discussed have a same order between 0 to 1 for all their fractional derivatives. Hence, their system equations can be expressed in the following form,

$$\begin{aligned} &D_t^\alpha x_i(t) = f_i(x_1(t), x_2(t), ..., x_n(t), t) \\ &x_i(0) = c_i, i = 1, 2, ..., n. \end{aligned} \tag{6}$$

In equation (6), $c_i, i = 1, 2, ..., n$ denote the initial conditions of different component x_i, and α is the commensurate fractional order.

Table 1 DNA encoding and decoding rules

Rule	1	2	3	4	5	6	7	8
00	A	A	C	G	C	G	T	T
01	C	G	A	A	T	T	C	G
10	G	C	T	T	A	A	G	C
11	T	T	G	C	G	C	A	A

2.2 DNA encryption basis

2.2.1 DNA encoding and decoding

A DNA sequence in the biological sense is composed of four different nucleic acid bases, 'A' (Adenine),' T' (Thymine),' C' (Cytosine),' G' (Guanine). The composition of these nucleic acid bases follows the Watson–Crick principle, where 'A' and 'T' are complementary, and 'C' and 'G' likewise[35].

In DNA computing, other than the binary computation for traditional computers, the information is carried and expressed by these four acid bases 'A' (Adenine), 'T' (Thymine), 'C' (Cytosine), 'G' (Guanine). The transformation between the binary values and the DNA sequence involves the DNA encoding and DNA decoding process. Typically, to map a binary sequence by a DNA sequence, every two bits of binary sequence are grouped and encoded into one of the DNA nucleic acid-bases 'A', 'T', 'C', and 'G' through a specific DNA encoding rule. Its reverse procedure applies the DNA decoding rules and turns a DNA sequence back into a binary sequence. There are 24(4!) combinations for the mapping of 2-bit binary symbols to DNA bases. Among them, if we consider '00' and '11' as a pair of complement, '01' and '10' another, then only 8 combinations satisfy the above-mentioned Watson-Crick complementary principle, which are shown in the Table.1 working as 8 different encoding and decoding rules[36].

Intuitively, if the same rule has been chosen for both encoding and decoding processes, then the binary value remains unchanged after the process. Otherwise, the binary value is changed; hence, the original information is masked.

To further explain the application of this in image encryption, we take one 8-bit decimal pixel value '234' as an example to illustrate the DNA encoding process as shown in Fig.1. The decimal value of '234' is first converted to binary bits' 11101010'. Then adopting DNA encoding rule 4, the corresponding DNA sequence 'CTTT' is obtained. Obviously, with different encoding rules, the same value can be transformed into distinguished DNA sequences. (With rule 5, '11101010' turns to 'GAAA'). The same sets of rules are adopted for DNA decoding to turn the DNA bases back to binary values. Take the previously obtained DNA sequence 'CTTT' as an example. If the DNA decoding rule 8 is taken, we will get an 8 bits binary value of '10000000', whose corresponding decimal value is '128'. This decoding process is also illustrated in Fig.1.

For some early works of DNA encoding and decoding methods in cryptography, for instance, [37][38], the encoding and decoding rules are fixed for the whole encryption process. However, it has been argued that an encryption

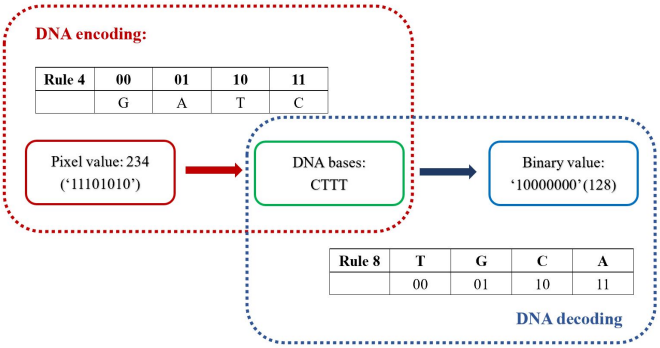

Fig. 1 Example of DNA encoding and decoding

scheme with a fixed rule can be easily detected and broken, thus is not competent enough for the design of cryptosystem[39][40]. So the dynamic DNA encoding and decoding method has been proposed. The principle of this method is to select different DNA encoding rules for the encryption of the plain text, which change dynamically during the whole encoding and decoding process. In this paper, we adopt the FCPRNG's outputs as the dynamic DNA encoding and decoding rules for the encryption of the image.

3 Fractional pseudo random number generator design

For our cryptosystem, we use an original fractional chaotic pseudo-random number generator (FCPRNG) to generate a sequence of pseudo-random numbers which further works as the keystream for the encryption algorithm. It shoule be noticed that, the generator is called pseudo-random due to the fact that the numbers are generated by computer simulations rather than true random process.

The generator implemented in this work is mostly based on the FCPRNG proposed in our paper[41]. Unlike the generator proposed in [41], our generator employs two separate sets of non-uniform grids to solve the 3D fractional systems numerically. In the following, the fractional systems and the methods for their numerical solution approximation will be illustrated in detail. The FCPRNG structure and test results are given.

3.1 Fractional systems used for FCPRNG design

In this work, we adopt three fractional systems, 3D fractional chaotic Chen's systems (f_1), 3D fractional chaotic Lu's systems (f_2), and fractional generalized double-humped logistic system (FGDHL, f_g). The continuous time system equations of these three systems are given in equations (7), (8), and (9), respectively.

$$f_1(t, x(t)) = \begin{cases} D^{\beta_1} x_1(t) = a_c(x_2(t) - x_1(t)) \\ D^{\beta_1} x_2(t) = (c_c - a_c) x_1(t) - x_1(t) x_3(t) + c_c x_2(t) \\ D^{\beta_1} x_3(t) = x_1(t) x_2(t) - b_c x_3(t) \end{cases} \quad (7)$$

$$f_2(t, x(t)) = \begin{cases} D^{\beta_2} x_1(t) = a_l(x_2(t) - x_1(t)) \\ D^{\beta_2} x_2(t) = -x_1(t) x_3(t) + c_l x_2(t) \\ D^{\beta_2} x_3(t) = x_1(t) x_2(t) - b_l x_3(t) \end{cases} \quad (8)$$

$$f_g(t, x(t)) = D^{\alpha_g} x(t) = \rho((x(t) - c)^2 \left(c^2 - (x(t) - c)^2\right), t > 0 \quad (9)$$

In the above equations, β_1, β_2 and β_g are the commensurate fractional derivative orders of the systems (7), (8), and (9), respectively; (a_c, b_c, c_c), (a_l, b_l, c_l), and (c, ρ) denote the parameters of the systems. In order to be implemented into the FCPRNG design, the continuous time fractional systems need to be first approximated and discretized.

3.2 Numerical solutions calculation of the systems

3.2.1 Approximation for FGDHL

For the FGDHL system, a piecewise constant arguments discretization method as described in [42] is applied to solve the fractional differential equation. We omit the discretization process, which can be found in [42] and only give out the obtained formula as follows,

$$\mathrm{x}_{n+1} = \mathrm{x}_n + \frac{r^{\beta_g}}{\Gamma(1 + \beta_g)} \rho(\mathrm{x}_n - c)^2 \left(c^2 - (\mathrm{x}_n - c)^2\right) \quad (10)$$

where $\Gamma(.)$ is the gamma function given in 2, n is the discretized time. It is worth mentioning that r in equation (10) is a parameter introduced by the discretization procedure. With different r values, the solutions of the system for each iteration can be very different. For the sake of simplification and consistency, we set $r = 0.2$ for the following work.

3.2.2 Numerical solution of 3D systems with non-uniform grid

For the 3D fractional chaotic Chen's and Lu's system, the fractional Corrector predictor ABM method with non-uniform grid calculation method discussed in [12] is adopted to calculate the systems' states. Hereafter we only display and explain the formulas for the calculation of the states $\mathrm{X}_1(n) = \begin{bmatrix} x_{11}(n) \\ x_{12}(n) \\ x_{13}(n) \end{bmatrix}$ for

fractional Chen's system(8), to avoid repetitions.

$$\begin{aligned} \mathrm{X}_1(n+1) &= \mathrm{Ff}\left[h(n), \mathrm{X}_1(n)\right] \\ &= \mathrm{X}_1(0) + \frac{h(n)^{\beta_1}}{\Gamma(\beta_1+2)} f_1\left({\mathrm{X}_1}^{\mathrm{Pr}}(n+1)\right) \\ &+ \frac{h(n)^{\beta_1}}{\Gamma(\beta_1+2)} \sum_{j=0}^{n} a_{j,n+1}^{1} f_1\left(\mathrm{X}_1(j)\right) \\ & \qquad\qquad 0 < \beta_1 < 1 \end{aligned} \tag{11}$$

where $\Gamma(.)$ represents the gamma function, $h(n)$ stands for the non-uniform grid space(calculation step size) for n-th output, and the coefficient $a_{j,n+1}^{1}$ (indice '1' correspond to the discussed f_1 system)is given by

$$a_{j,n+1}^{1} = \begin{cases} n^{\beta_1+1} - (n-\beta_1)(n+1)^{\beta_1}, & \text{if } j = 0, \\ (n-j+2)^{\beta_1+1} + (n-j)^{\beta_1+1} & \\ \qquad -2(n-j+1)^{\beta_1+1}, & \text{if } 1 \le j \le n, \\ 1, & \text{if } j = n+1. \end{cases} \tag{12}$$

The $\mathrm{X}_1^{\mathrm{Pr}}(\mathrm{n}+1)$ in 11 denotes the prediction of $\mathrm{X}_1(\mathrm{n}+1)$ which is defined as,

$${\mathrm{X}_1}^{\mathrm{Pr}}(n+1) = \mathrm{X}_1(0) + \frac{1}{\Gamma(\beta_1)} \sum_{j=0}^{n} b_{j,n+1}^{1} f_1\left(\mathrm{X}_1(j)\right), 0 < \beta_1 < 1, \tag{13}$$

where $b_{j,k}^{1}$ is written as

$$b_{j,n+1}^{1} = \frac{h(n)^{\beta_1}}{\beta_1}\left((n+1-j)^{\beta_1} - (n-j)^{\beta_1}\right). \tag{14}$$

The non-uniform grid we propose and employ has in total 5 possible grid spaces varying from 0.001 to 0.005 with a gap of 0.001. That is to say, the step size $h(n)$ takes a value in the set $\mathbf{S} = \{0.001, 0.002, 0.003, 0.004, 0.005\}$.

To acquire the step size for each n, we establish a multiplexing mechanism composed of two skew tent maps. One of them is used to assign different values to $h(n)$, and the other is adopted to increase the system's complexity by taking account of different step size allocation possibilities. For the sake of clarity, we first display the mathematical formula for step size $h(n)$ as following equation (15) to (20), then we discuss the interpretation of these equations briefly.

$$\begin{aligned} h(n) &= \mathrm{fh}[\mathrm{Xst}_1(n), \mathrm{Xst}_2(n)] \\ &= \sum_{i=1}^{5} 1_{\mathrm{A}_i}\left(\mathrm{Xst}_1(n)\right) \mathbf{H}\left(\left((2^{32}-1) \times \mathrm{Xst}_2\right) (\mathrm{mod}120) + 1, i\right) \end{aligned} \tag{15}$$

The $\mathrm{Xst}_1(\mathrm{n})$ and $\mathrm{Xst}_2(\mathrm{n})$ in the above equation are the n-th states of the skew tent maps given by (16) and (17), where p_1, p_2 are the control parameters,

$$\begin{aligned} \mathrm{Xst}_1(n) &= \mathrm{St}_1\left[\mathrm{Xst}_1(n-1)\right] \\ &= \begin{cases} \frac{\mathrm{Xst}_1(n-1)}{p_1}, 0 < \mathrm{Xst}_1(n-1) \le p_1 \\ \frac{1-\mathrm{Xst}_1(n-1)}{1-p_1}, p_1 < \mathrm{Xst}_1(n-1) < 1 \\ \mathrm{Xst}_1(n-1) - 0.05, \text{otherwise} \end{cases}. \end{aligned} \tag{16}$$

$$\begin{aligned} \mathrm{Xst}_2(n) &= \mathrm{St}_2[\mathrm{Xst}_2(n-1)] \\ &= \begin{cases} \frac{\mathrm{Xst}_2(n-1)}{p_2}, 0 < \mathrm{Xst}_2(n-1) \le p_2 \\ \frac{1-\mathrm{Xst}_2(n-1)}{1-p_2}, p_2 < \mathrm{Xst}_2(n-1) < 1 \\ \mathrm{Xst}_2(n-1) - 0.05, \text{otherwise} \end{cases} . \end{aligned} \tag{17}$$

In equation (15), $1_{\mathrm{A_i}}(.)$ is an indicator function expressed as

$$1_{\mathrm{A}_i}(\mathrm{Xst}_1(n)) = \begin{cases} 1, \mathrm{Xst}_1(n) \in \mathrm{A}_i \\ 0, \mathrm{Xst}_1(n) \notin \mathrm{A}_i \end{cases}, i = 1,2,3,4,5. \tag{18}$$

where A_i denoting the following interval

$$\mathrm{A}_i = \left(y | \frac{1}{5}(i-1) < y \le \frac{1}{5} i \right), i = 1,2,3,4,5. \tag{19}$$

$\mathbf{H}\left(((2^{32}-1) \times \mathrm{Xst}_2)(\mathrm{mod}120)+1, i\right)$ in equation (15) denotes the element on $\left(((2^{32}-1) \times \mathrm{Xst}_2)(\mathrm{mod}120)+1\right)$-th row and i-th column of the Matrix $\mathbf{H}$ who is of size 120×5 and is consists of all the possible combinations of the elements in set $\mathbf{S}$ and holds the form as following,

$$\mathbf{H} = \begin{pmatrix} 0.001 & 0.002 & 0.003 & 0.004 & 0.005 \\ 0.001 & 0.002 & 0.003 & 0.005 & 0.004 \\ 0.001 & 0.002 & 0.004 & 0.003 & 0.005 \\ 0.001 & 0.002 & 0.004 & 0.005 & 0.003 \\ & & \vdots & \vdots & \\ 0.005 & 0.004 & 0.003 & 0.002 & 0.001 \end{pmatrix} \tag{20}$$

A brief interpretation of the formula is given here. With an output of skew tent map in the range of $(0,1)$, we introduce 5 intervals $\mathrm{A}_i, (i = 1,2,\ldots 5)$ of the same size obtained by (19) for the assignment of $h(n)$. To match the intervals to the five possible step sizes in $\mathbf{S}$, we construct the matrix $\mathbf{H}$ (equation (20)). By performing the modulo operation, the states $\mathrm{Xst}_2(n)$ is processed to acquire a row indice r for the matrix. Then, the interval A_i are matched with the corresponding steps sizes value on the i-th column and r-th row of $\mathbf{H}$. Finally, the step size is assigned to $h(n)$ depending on which interval A_i among the five that $\mathrm{Xst}_1(n)$ lies in.

The calculation for the Lu system(7) states can be obtained by substituting all the control parameters and initial conditions for the systems and the non-uniform step size to the corresponding ones.

In our previous investigations concerning the nonuniform-grid calculation methods and their implementation for fractional chaotic systems ([10][11][41], etc.), we analyzed the systems dynamics of both 3D fractional Chen and Lu systems. Here, to recall the chaotic properties of the systems, some results of Chen and Lu systems with respect to their LEs values and bifurcation diagrams are given in Fig.2 and 3. It can be easily observed from Fig.2, for the fractional Chen system, only the first component of the system state x_1

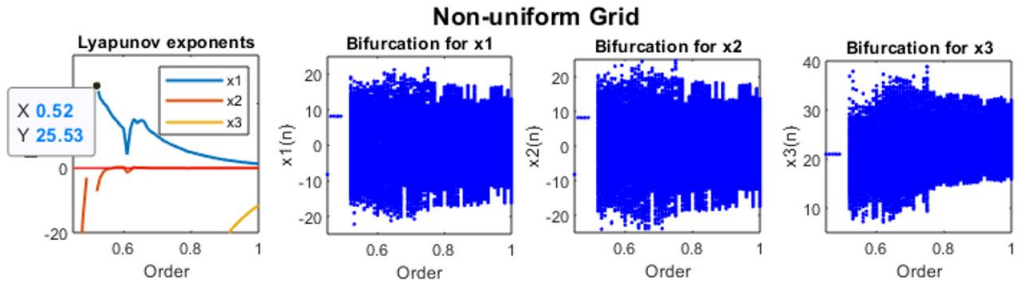

Fig. 2 Lyapunov Exponent and Bifurcation diagrams for Chen system of different fractional orders with $(a_c, b_c, c_c) = (35, 3, 28)$

has LEs greater than 0, and the chaotic behavior appears with the fractional order no smaller than 0.52. The bifurcation results in Fig.3 also reveal that while the fractional order is from 0.75 to 1, there is a promising mutual chaotic range for parameter c_l which lies in $[25, 30]$. Whereas for the parameter a_c of the Chen system, this range is between 35 to 40.

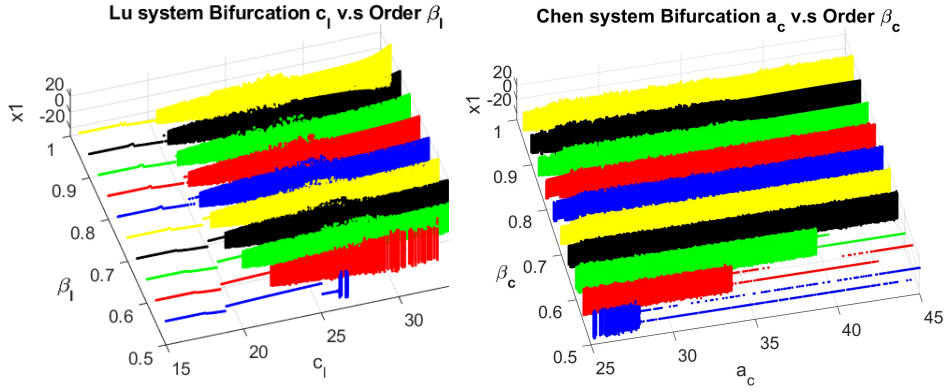

Fig. 3 Bifurcation diagram of Lu and Chen systems of different fractional orders

3.3 FCPRNG structure

The structure of the FCPRNG proposed is given in Fig.4. The systems (control) parameters, together with the initial conditions constitute the secret key. With the equations (15)-(20) formulated in the previous section, the grid spaces $h_c(n)$ and $h_l(n)$ are obtained through fh[$\mathrm{Xst_1(n)}$, $\mathrm{Xst_2(n)}$] and fh[$\mathrm{Xst_3(n)}$, $\mathrm{Xst_4(n)}$] given by equation (15). We then calculate the states of fractional Chen's and Lu's systems, $\mathrm{X_1(n)}$ and $\mathrm{X_2(n)}$, on the non-uniform grid $h_c(n)$ and $h_l(n)$ applying Ff[$\mathrm{h_c(n)}, \mathrm{X_1(n-1)}$] and Ff[$\mathrm{h_l(n)}, \mathrm{X_2(n-1)}$], respectively (equation (11)). Fg[$\mathrm{X_g(n-1)}$] in the figure is the function employed to

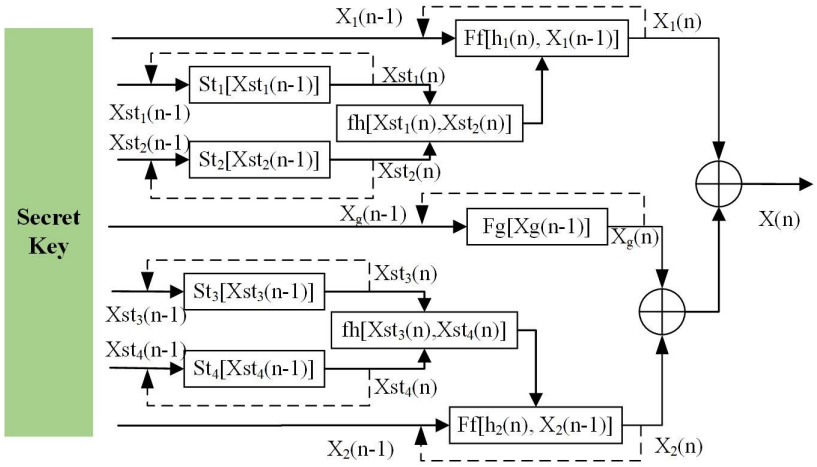

Fig. 4 Structure of the designed FCPRNG

calculate the states of FGDHL following the equation (9). The parameters and fractional derivative orders of Chen and Lu systems for the FCPRNG are as given: $\beta_1 \in [0.75, 1)$, $\beta_2 \in [0.75, 1)$, $a_c \in [35, 40]$, $b_c \in [1.5, 3.5]$, $C_c \in [23, 28]$, $a_l \in [30, 35]$, $b_l \in [3, 8]$, $c_l \in [20, 25]$.

The control parameters p_1, p_2, p_3, p_4 of the skew tent maps are all in the ranges of $(0, 1)$; and the initial conditions $\mathrm{Xst}_1(0)$, $\mathrm{Xst}_2(0)$(for the calculation of $h_c(n)$), and $\mathrm{Xst}_3(0)$, $\mathrm{Xst}_4(0)$ (for the calculation of $h_l(n)$) are also in the same range. In addition,the initial condition Xg(0) for FGDHL is equally adopted as a component of secret key, and it is in the range of $[0, 0.3]$.

It is proved that only the first component of the state vectors possess positive Lyapunov exponents(LEs) for both Lu and Chen 3D systems, which means that the first component among the three implies the chaotic dynamic of the whole system. (eg. $x_{11}(n)$ for Chen system states $\mathrm{X}_1(n) = [x_{11}(n), x_{12}(n), x_{13}(n)]$). Therefore, we only employ the first one for further use as the output of the FCPRNG.

After converting X_1(n), X_2(n) and X_g(n) into 32 bits binary values, the final output of the FCPRNG X(n) is obtained by performing or-exclusive operations (XOR) between the outputs of these three fractional systems. It is worth mentioning that in order to improve uniformity for the output sequence distribution, we inject the state values of the two fractional 3D systems into the interval of $[-10, 10]$ by a folding mechanism, and the states of FGDHL is truncated with a window of $[-0.15, 0.7]$ as it has been demonstrated in our previous paper [41].

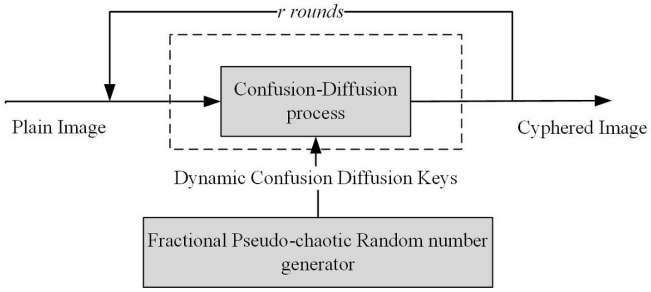

Fig. 5 General block diagram

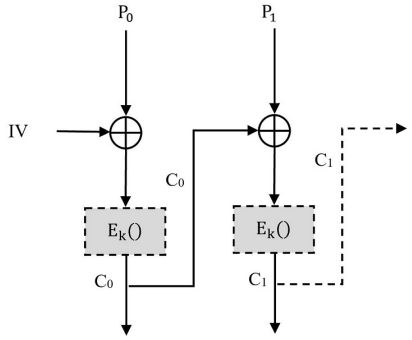

Fig. 6 CBC mode encryption structure

4 Proposed encryption scheme

4.1 General concept of the encryption scheme

For our proposed encryption scheme, we employ a CBC mode block cipher of size $b_s = 1024$ (32*32 pixels) based on the encryption scheme discussed in [43] and a dynamic DNA encoding and decoding method. The general block diagram of the proposed encryption scheme is given in Fig.5. r rounds of this confusion and diffusion process are performed to the whole image in order to obtain the secured ciphered image. During each round, the images are encrypted block by block through the CBC mode by the encryption algorithm $\mathrm{E_k}(.)$. The structure of the encryption scheme is illustrated in Fig.6. In the figure, P_0 stands for the first block of size 1024 bits from the plain image. IV is the Initial Vector pre-generated, C_0 is the first encrypted block. The cyphered block is then working as the initial vector to encrypt the next block etc.

Within each block, the permutation is performed by dynamic DNA encoding and decoding method with 2D cat map, which will be explained in the following sections. A logistic map is then used to complete the encryption by further diffusing the resulting cyphered block.

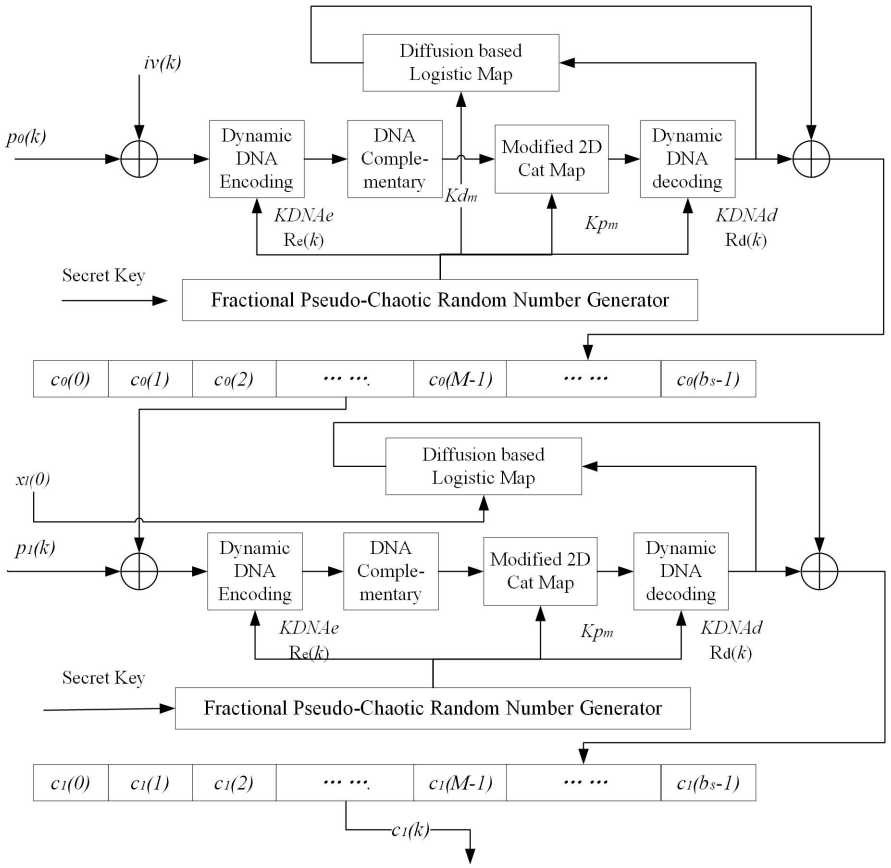

Fig. 7 Encryption structure of the cryptosystem

4.2 Encryption scheme of the proposed cryptosystem

The proposed encryption scheme is illustrated in Fig.7. For the first block of 1024 pixels, each pixel $p_0(k), (k = 1, ..., b_s)$ are first XOR-ed with the byte$(iv(k))$ of the initial vector(IV) of the same size given randomly by the function **randi** in MATLAB. Then, a value in the range of 1 to 8 is obtained by converting 3 bits of the generated FCPRNG outputs *KDNAe* to a decimal value. Adding one to this value gives us the dynamic DNA encoding rules $R_e(k)$ to encode the 8 bits of the XOR-ed pixels. After that, the DNA complementary rules are employed to switch the encoded DNA bases to their complementary ones('A'-'T', 'C'-'G'). After encoding all the pixels in the block, we will get a block of DNA bases that has 64×64 (One DNA base consists of 2 binary bits, the total number of 1024×8 bits divided by 2, equal to $4096 = 64 \times 64$ DNA bases). To relocate the acquired DNA bases to a new position, the modified 2D cat map discussed in [44] is performed on the DNA base level. Up to this point, the confusion of the first block has been accomplished. For the diffusion

Algorithm 1 Encryption steps

Generate the IV values to encrypt the first block B_1 for $r = 1$
for $r = 1 : rg$ **do**
  for $j = 1 : Bn$ **do**
    if $r = 1$ and $j = 1$ **then**
      $iv(k) = IV(k)$
    else if $r \neq 1, j = 1$ **then**
      $iv(k) = C_{B_N}(k)$
    else
      $iv(k) = C_{j-1}(k)$
    end if
    Calculate $y_j(k) = P_j(k)$ XOR $iv(k)$
    Get $R_e(k)$ by converting 3 bits in $KDNAe$ to decimal value
    Encode $y_j(k)$ by the rule $R_e(k)$ given in Table.1 to $y_{j,\mathrm{DNA}}$
    Apply the DNA complementary rules to change $y_{j,\mathrm{DNA}}$ to $y'_{j,\mathrm{DNA}}$
    Reshape $y'_{j,\mathrm{DNA}}$ to a matrix of DNA bases My_{DNA} of size $\sqrt{4b_s} * \sqrt{4b_s}$
    for $i = 1 : \sqrt{4b_s}$ **do**
      for $l = 1 : \sqrt{4b_s}$ **do**
        Calculate (i_{new}, l_{new}) using Equation 21
        $My'_{\mathrm{DNA}}(i_{new}, l_{new}) = My_{\mathrm{DNA}}(i, l)$
      end for
    end for
    Reshape My'_{DNA} to a DNA bases string $y'_{j,\mathrm{DNAnew}}$
    Get $R_d(k)$ by converting 3 bits in $KDNAd$ to decimal value
    Decode $y'_{j,\mathrm{DNAnew}}$ to $y_{j,new}$ applying $R_d(k)$
    if $j = 1$ and $r = 1$ **then**
      Get $x_l(0)$ from KMp
    else if $r \neq$ and j=1 **then**
      $x_l(0)$ equals the 32 bits decimal value converted from the binary string consisted of $[c_{B_n}(b_s - 3), c_{B_n}(b_s - 2), c_{B_n}(b_s - 1), c_{B_n}(b_s)]$
    else
      $x_l(0))$ equals the 32 bits decimal value converted from the binary string consisted of $[c_j(b_s - 3), c_j(b_s - 2), c_j(b_s - 1), c_j(b_s)]$
    end if
    Calculate $s(0) = f_l(x_l(0)$ using Equation 23
    Convert $y_{j,new}$ to string $y'_{j,new}$ consisted of 32-bits decimal values
    for $t = 1 : b_s/4$ **do**
      if $t = 1$ **then**
        $s(1) = f_l(s(0))$ using Equation 23
      else
        Calculate $s(t) = f_l(x(t-1))$ using Equation 23
      end if
      $x(t) = y'_{j,new}(t)$ XOR $s(t)$
    end for
    Get $C_j(k)$ from converting the string of x to 8 bits values
  end for
end for

layer, the permuted DNA bases are decoded to binary bits by the dynamic DNA decoding method, where the decoding rules $\mathrm{R_d(k)}$ are again acquired through FCPRNG outputs *KDNAd* the same way as introduced above for the dynamic DNA encoding. Then, a discrete logistic map of 32 bits is employed to construct the final cyphered output. During this process, every 4 pixels (32

bits) of decoded DNA bases are XOR-ed with the output of the discrete logistic map, and the input of the map is the 32 bits decimal value converted by the 4 pixels of previously decoded DNA bases.

The encryption process of the second block to the last block, block number B_N, is almost the same(B_N denotes the total number of blocks in the image). However, rather than use the initial vector IV, each pixel $p_l(k), (l = 1, ..., B_N - 1)$ is XOR-ed with the pixel at the same position of the previous cyphered block($c_{l-1}(k)$) to achieve the CBC mode. In addition, the first input of the diffusion based discrete logistic map is acquired by processing the last 4 pixels of the previous cyphered block, namely $c_{l-1}(b_s-3)$, $c_{l-1}(b_s-2)$ and $c_{l-1}(b_s-1)$, $c_{l-1}(b_s)$(in total 1024 pixels in each block). The four pixels are converted to 8 bits binary values $c^b_{l-1}(b_s - i)$, $(i = 3, 2, 1, 0)$ and form a 32 bits string which is then converted to the decimal value $x_l(0)$ as the input of the discrete logistic map.

The 2D map adopted here for the permutation is derived from the Arnold's cat map and has been discussed in [45]. The equations for the map is defined as follows,

$$\begin{bmatrix} i_{new} \\ j_{new} \end{bmatrix} = mod\left(A_0 \times \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} rl + rc \\ rc \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix} \right) \tag{21}$$

The matrix A_0 in equation (21) is defined as

$$A_0 = \begin{bmatrix} 1 & u \\ v & 1 + u \times v \end{bmatrix} \tag{22}$$

The determinant of matrix A_0 is equal to 1, which indicates that each point (i, j) of the square matrix is transferred to a unique point at position (i_{new}, j_{new}). The parameter u, v, rl, rc are the outputs of FCPRNG and form the dynamic key (Kp_m). The use of rc, rl is under the intention of overcoming the fixed point problem, possessed by the Arnald Cat map[4].

It is to be noticed that these dynamic keys u, v, rl, rc are fed by the proposed FCPRNG, and the values for them change every block.

For the digital implementation, we need a 32 bits discrete logistic map employed for the diffusion, which is expressed as follows[43],

$$\mathrm{X_{k+1}} = f_{\mathrm{l}}(\mathrm{X_k}) = \begin{cases} \left\lfloor \dfrac{\mathrm{X_k} \times (2^{\mathrm{N}} - \mathrm{X_k})}{2^{N-2}} \right\rfloor, & \text{if } \mathrm{X_k} \neq [3 \times 2^{\mathrm{N}-2}, 2^{\mathrm{N}}] \\ 2^N - 1, & \text{if } \mathrm{X_k} = [3 \times 2^{\mathrm{N}-2}, 2^{\mathrm{N}}] \end{cases} \tag{23}$$

where $\mathrm{X_{k+1}}$ stands for the new output calculated from its previous one $\mathrm{X_k}$; N is the number of bits representing the integer output of the discrete logistic map. In our proposed cryptosystem, since a 32-bit discrete logistic map is applied, we have $N = 32$.

To determine the optimal value of the rounds r needed for the encryption scheme to pass successfully the security tests, we evaluate first the confusion and diffusion performance by calculating the Hamming distance(HD) between

two ciphered images corresponding to original images with one bit difference. The calculation is given in the form below,

$$\mathrm{HD}\left(C_1, C_2\right) = \frac{1}{lb} \sum_{k=1}^{lb} C_1\left[k\right] \oplus C_2\left[k\right] \tag{24}$$

In the above equation, C_1, C_2 represent two cipher images; lb is the bit length of the image which is calculated by $lb = N_{pix} \times l \times L$. The N_{pix} is the number of pixels in the image; l is equal to 1 if a grey image is encrypted and equals 3 for a colored image; L denotes the number of bits for each pixel.

Three images of different types (grey or colored) with different sizes and features have been tested. Twenty pairs of C_1 and C_2 have been obtained for each plain image by randomly changing one bit of a pixel in the original image. Then, equation (24) is employed to calculate the HDs of the images. The average HD over these 20 pairs of cyphered images shows that when $r = 2$, the HD is already close to 50%. The result indicates that the probability of a bit change is 0.5, which is the optimal value meaning the diffusion is effective.

4.3 Decryption scheme of the proposed cryptosystem

The diagram for decryption scheme is given in Fig.8. It is the reversed process of the encryption scheme.

For each round, the decryption starts from the last block c_{B_n-1} to the first block c_0. For block $c_l(l = 1, 2, ...B_n - 1)$, XOR operation is first operated between the cyphered pixels and the output of logistic map. The input $x_l(0)$ for the first 4 pixels $c_l(k), (k = 1, ..., 4)$, is the decimal value of the 32-bit string converted from the last 4 pixels of the previous cyphered block c_{l-1}; and the inputs for the rest of the cyphered pixels are obtained from performing XOR operation between the 4 current cyphered pixels and 4 previously XORed pixels. Dynamic DNA encoding method applying DNA encoding rules $R_d(k)$ obtained from *KDNAd* is employed to turn the XORed pixels' values into DNA bases. Same way as for the encryption process, a DNA base matrix of size 64×64 is constructed. After the matrix has been acquired, the modified 2D cat map with Kp_m generated by FCPRNG and DNA complementary rules permutates the bases. The dynamic DNA decoding process with decoding rules $R_e(k)$ is then employed, after which XOR operations between obtained sequence and the cyphered pixels of previous blocks c_{l-1} proceed.

For the first block, c_0 for round $r(\ r \neq 1)$, the first input of the logistic map is acquired correspondingly to the last four pixels of the whole image. The final XOR operation over the block is done between the obtained block and the decyphered c_{B_n-1} of last round $r - 1$. Whereas for $r = 1$, the first input $x_l(0)$ is given by Kd_m from FCPRNG, and the XOR is done with initial vector IV.

To prove that our proposed FCPRNG and cryptosystem are reliable, a series of well-recognized tests and indicators have been adopted, and the performance has been analyzed in the next section.

Algorithm 2 Decryption steps

Generate the IV values to decrypt the first block B_1 for $r = 1$
for $r = rg : 1$ **do**
  for $j = Bn : 1$ **do**
    if $j = 1$ and $r = 1$ **then**
      Get $x_l(0)$ from KMp
    else if $r \neq$ and j=1 **then**
      $x_l(0)$ equals the 32 bits decimal value converted from the binary string consisted of $[D_{B_n}(b_s - 3), D_{B_n}(b_s - 2), D_{B_n}(b_s - 1), D_{B_n}(b_s)]$
    else
      $x_l(0))$ equals the 32 bits decimal value converted from the binary string consisted of $[D_j(b_s - 3), D_j(b_s - 2), D_j(b_s - 1), D_j(b_s)]$
    end if
    Calculate $s(0) = f_l(x_l(0))$ using Equation.23
    Convert D_j to string Dy_j consisted of 32-bits decimal values
    for $t = 1 : b_s/4$ **do**
      if $t = 1$ **then**'
        Calculate $s(1) = f_l(s(0))$ using Equation.23
      else
        Calculate $s(t) = f_l(Dy_j(t-1))$
      end if
      $x(t) = Dy_j(t)$ XOR $s(t)$
    end for
    Get$Dy_j^{'}$ from converting the string of x to 8 bits value
    Get $R_d(k)$ by converting 3 bits in $KDNAd$ to decimal value
    Encode $Dy_j^{'}$ to $Dy_{j,DNA}^{'}$ applying $R_d(k)$
    Reshape $Dy_{j,DNA}^{'}$ to a matrix of DNA bases MDy_{DNA} of size $\sqrt{4b_s} * \sqrt{4b_s}$
    for $i = 1 : \sqrt{4b_s}$ **do**
      for $l = 1 : \sqrt{4b_s}$ **do**
        Calculate (i_{new}, l_{new}) using Equation.21
        $DMy_{\mathrm{DNA}}^{'}(i_{new}, l_{new}) = MDy_{\mathrm{DNA}}(i, l)$
      end for
    end for
    Reshape $DMy_{\mathrm{DNA}}^{'}$ to a DNA bases string $Dy_{j,\mathrm{DNAnew}}^{'}$
    Apply the DNA complementary rules to change $Dy_{j,\mathrm{DNA}}^{'}$ to $Dy_{j,\mathrm{DNA}}$
    Get $R_e(k)$ by converting 3 bits in $KDNAe$ to decimal value
    Decode $Dy_{j,\mathrm{DNA}}$ by the rule $R_e(k)$ given in Table.1 to Dy_j
    if $r = 1$ and $j = 1$ **then**
      $iv(k) = IV(k)$
    else if $r \neq 1, j = 1$ **then**
      $iv(k) = P_{B_n}(k)$
    else
      $iv(k) = P_{j-1}(k)$
    end if
    Calculate $P_j(k) = Dy_j(k)$ XOR $iv(k)$
  end for
end for

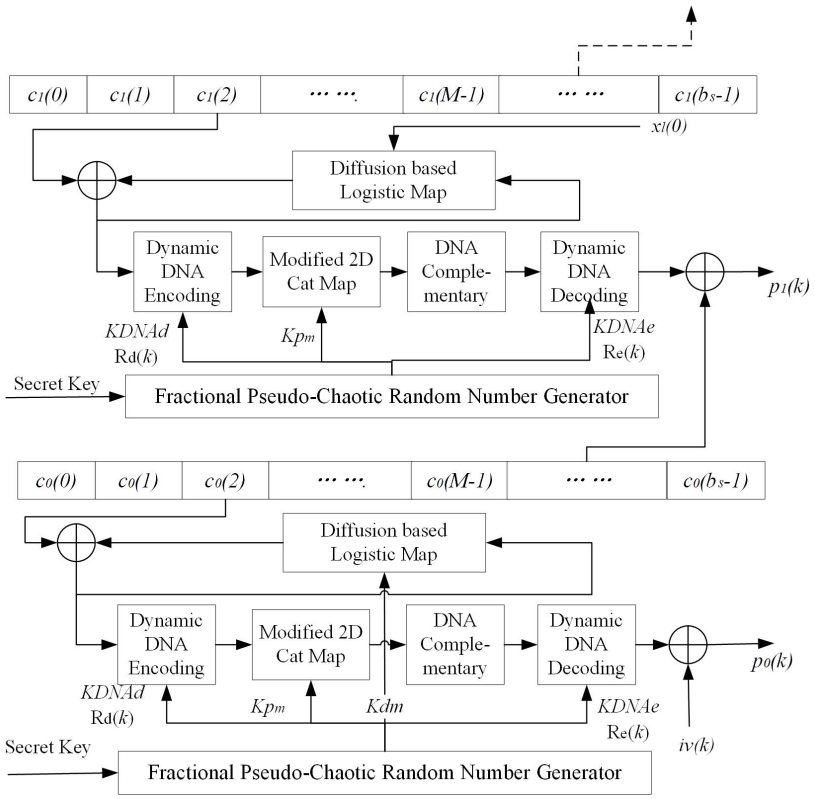

Fig. 8 Decryption structure of the cryptosystem

5 Performance and Security analysis

The following performance and security analyses are divided into two parts. We will 1) discuss the performance of the proposed FCPRNG structure through statistical analysis and NIST(National Institute of Standard and Technology)test[17]; 2) the whole cryptosystem's security is evaluated through various tests on the encrypted images.

5.1 FCPRNG statistical analysis

5.1.1 Histogram and Chi square test

The histogram is often used to exhibit the distribution of the data. For a well-designed random number generator, its outputs should be uniformly distributed. Therefore, hereafter, we employ the histogram and Chi-square test to evaluate the distribution of the proposed FCPRNG's outputs. One hundred sequences with one million, 1000000 bits) are generated with 100 pairs of different secret keys(3125000 samples). The distribution of these outputs

is shown in Fig.9. It can be observed that these 3125000 values are almost uniformly distributed, which meets the uniformity requirement of the pseudo random generator.

Apart from the histogram, which can visually illustrate the uniform distribution of the outputs, the Chi-square test is also employed. To do the test, a null hypothesis is to be established, and a significance level α signifying the probability of rejecting the null hypothesis while it is true, is chosen. If one obtains a test statistic (experimental value) smaller than the critical Chi-square value(CV) under the given degree of freedom of the samples for the Chi-square test with a significance level of α, then the null hypothesis is considered to be true and validated. The experimental value V for the Chi-square test is calculated through the following formula,

$$V = \sum_{i=0}^{N_c-1} \frac{(O_i - E_i)^2}{E_i} \tag{25}$$

In equation (25), N_c denotes the number of classes, O_i is the number of samples in the i-th class, and E_i represents the number of expected samples for a uniform distribution.

For our uniformity test, we assume the null hypothesis H_0 states that the outputs of the proposed FCPRNG are uniformly distributed (numbers of samples in each of the 1000 classes are identical). With a $\alpha = 0.05$, the critical Chi-square value χ_c^2 can be obtained which is equal to 1073.6473. The calculated experimental value V with $N_c = 1000$, $E_i = 3125000/N_c = 3125$ equals to 999. The fact that $V < \chi_c^2$ leads to the acceptance of H_0, which in turn confirms that the FCPRNG outputs have a uniform distribution.

It is worth mentioning that, the same chi-square test is also employed to test the uniformity of the ciphered images' pixel values latter in the cryptosystem's security analysis. But the number of classes N_c and degree of freedom is different, since there are only 256 different possibilities for pixel values (0-255).

5.1.2 NIST test suite

To evaluate the pseudo-randomness of the FCPRNG outputs, we employ the NIST test suite. The NIST test is a suite of tests consisting of 15 randomness tests. For the outputs to be pseudo-random, each of the tests should have a p-value greater than 0.01 and a proposition greater than 96 (Detailed information of this test suite and the criteria can be found in [17]).

To do the test, a sequence of 100000000 bits from 100 distinct sets of secret keys (100 sequences of 31250 samples) is needed. The tests' results are given in Table.2. With all the p-values greater than 0.01, and propositions greater than 96.000, the pseudo randomness of the generated outputs is certified.

After having demonstrated the performances of the newly designed FCPRNG, now let's investigate the performances of the whole cryptosystem.

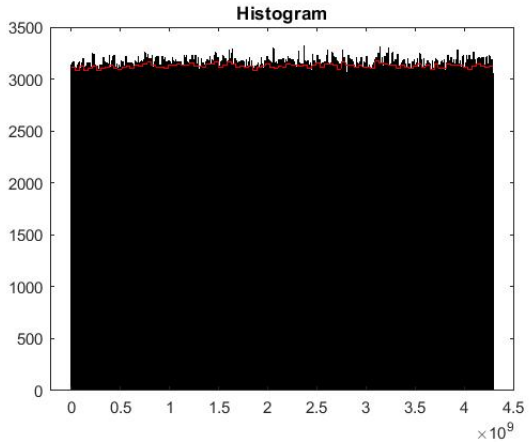

Fig. 9 Histogram of 31250000 output samples

Table 2 NIST test suite results

Test	P-value	Prop	Test	P-value	prop
Frequency test	0.834	99.000	**Block-frequency test**	0.115	100.000
Cumulative-sums test	0.698	99.000	**Runs test**	0.192	98.000
Lognest-run test	0.290	100.000	**Rank test**	0.276	98.000
FFT test	0.016	100.000	**Nonperiodic-templates**	0.532	98.885
Overlapping-templates	0.740	99.000	**Universal**	0.290	97.000
Appropximty entropie	0.419	100.000	**random-excursions-variant**	0.240	99.138
Serial test	0.382	99.000	**Linear-complexity**	0.437	99.000
random-excursions	0.437	99.000			

5.2 Cryptosystem security analysis

To evaluate the security and the performance of the proposed encryption algorithm, we encrypted several benchmark images in black and RGB colored by our proposed encryption algorithm. Some well-recognized tests against security attacks, such as statistical attacks, differential attacks, and so on, have been adopted, and the results will be reported in the following.

5.2.1 Histogram and Chi-square test results

For a cryptosystem to resist a statistical attack, one basic requirement must be met first, which is that the ciphered image should possess a uniform distribution in terms of its pixel values. In Fig. 10, histograms of colored images 'Lena' and 'Goldhill' are illustrated. One can easily observe that the histograms of the original plain images in the three color layers follow certain patterns over the pixel values (0 to 255) but don't satisfy the conclusion for uniform distribution. In comparison, those of the ciphered images possess much uniformly distributed pixel values in all color layers.

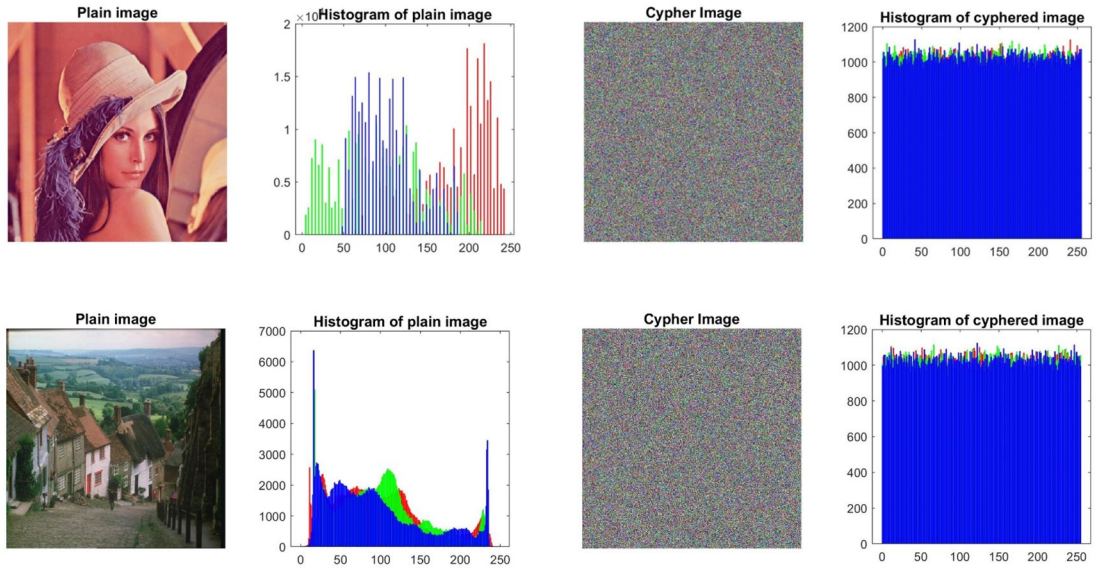

Fig. 10 Histogram of plain and cypher colored images

The histograms of the grey images 'Airfield' and 'Black' are also given in Fig. 11. Same observations can be made, which signifies that the pixel values of the ciphered image satisfy the essential requirement of having a uniform distribution.

The Chi-square test is also adopted here to test the uniformity of the distribution of the cyphered pixel values. Different than the previous Chi-square test discussed for FCPRNG performance analysis in section 5.1.1, the number of classes N_c here for equation (25) is equal to 256 since there are in total 0 to 255, 256 possible pixel values; and $E_i = ImSize/N_c$. The critical value χ_c^2 for the test with $\alpha = 0.05$ and degree of freedom 255 $(N_c - 1)$ can be found equal to 293.2478. The experimental Chi-square values for several images are given in Table.4. With all the experimental values smaller than 293.2478, the cyphered images obtained after encryption are uniformly distributed.

5.2.2 Correlation analysis

The correlation between pixels is another feature which has to be tested to evaluate the ability of the encryption scheme to resist statistical attacks, which calculates the strength of correlation between adjacent pixels. A secure cryptosystem should break the high correlation between the pixels of the plain image. The correlation is calculated by the formula below,

$$\rho_{xy} = \frac{\sum_{i=1}^{N_p} \left[(x_i - \overline{x}) (y_i - \overline{y}) \right]}{\sqrt{\sum_{i=1}^{N_p} (x_i - \overline{x})^2} \sqrt{\sum_{i=1}^{N_p} (y_i - \overline{y})^2}} \tag{26}$$

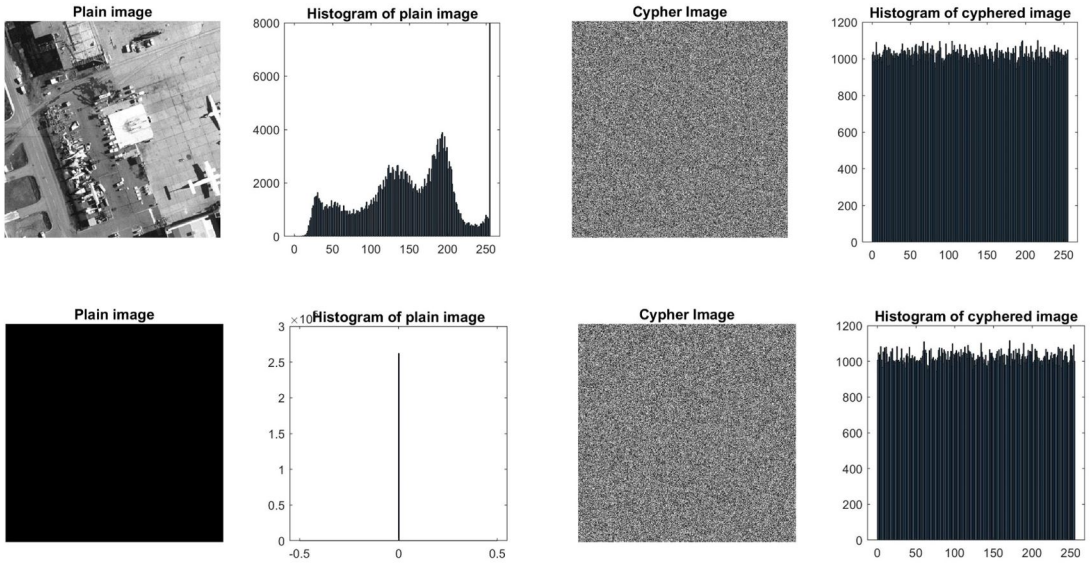

Fig. 11 Histogram of plain and cypher colored images

In equation (26), N_p is the number of pairs of adjacent pixels, which are randomly selected for the analysis, x_i and y_i represent the pixel values in the image, and $\overline{x}, \overline{y}$ are the corresponding expected values.

For our analysis, 8000 different pairs of pixels have been chosen in the direction of horizontal, vertical, and diagonal for several benchmark images. For each image, the correlation coefficients for different color layers are evaluated over 100 different cipher images. These cipher images have been obtained through encrypting the images with one bit difference from the original image in their pixel value of a random position.

The average correlation coefficients are tabulated in Table.3. In all cases, the correlation coefficients in every color layer and each direction of the ciphered images have a value close to zero. This implies that the encryption scheme is highly resistant to statistical-based attacks. We also illustrated the correlation in the three directions for both plain and ciphered images of colored image 'Pepper' and the grey image 'Boat' in Fig.12 and Fig.13, respectively. One can easily observe that in the plain images, the correlation between pixels is evident, while in the ciphered images, same as given by the coefficients, the high correlation is broken.

5.2.3 Avalanche

As mentioned in the previous section, theoretically, a one-bit change in the plaintext should lead to changes to 50% of the ciphertext for a well-designed cipher scheme. To prove the sensitivity of our proposed encryption algorithm in this sense, we calculate the Hamming distances(HD) between the pixels of

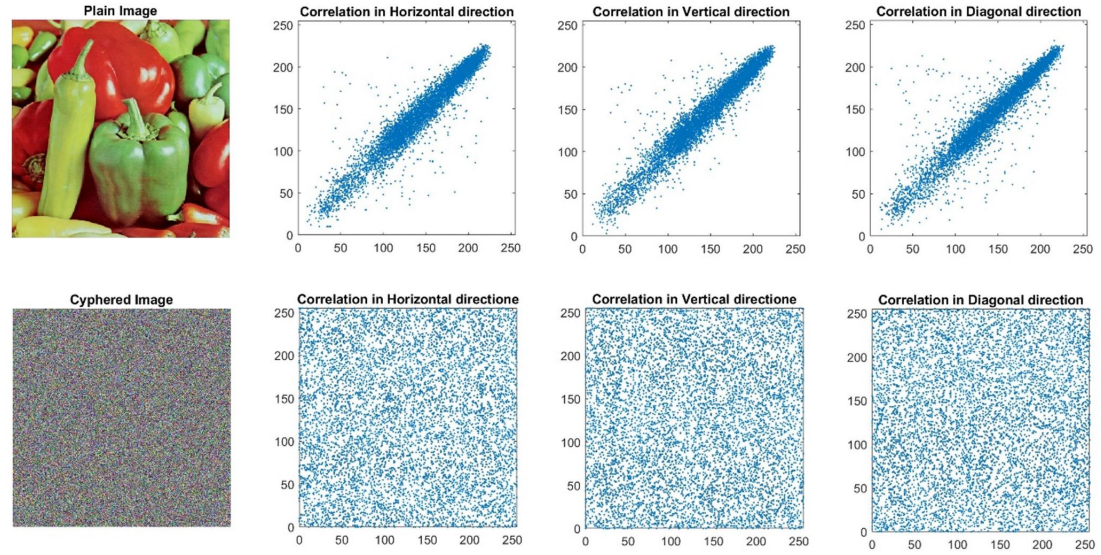

Fig. 12 Correlation results in 3 directions of colored image 'Pepper'

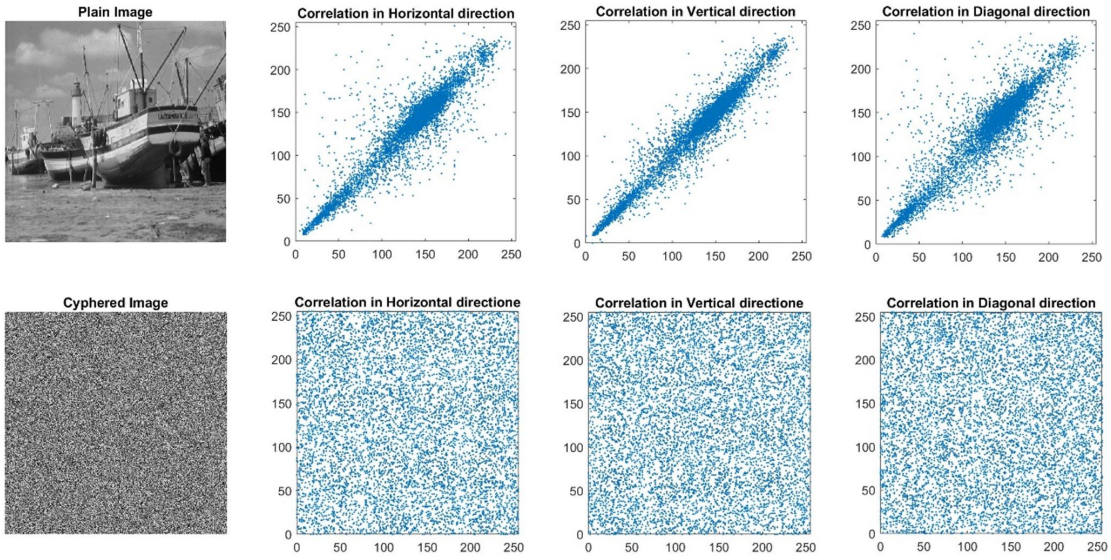

Fig. 13 Correlation results in 3 directions of grey image 'Boat'

Table 3 Correlation coefficient of several images in different directions

Image	Size	Direction	Plain Image			Cyphered Image		
			Correlation coefficient			Correlation coefficient		
			Red	Green	Blue	Red	Green	Blue
Baboon	256*256*3	Horizontal	0.9543	0.8845	0.9288	-0.0015	0.0006	-0.0007
		Vertical	0.9343	0.8561	0.9281	-0.0010	0.0011	-0.0031
		Diagonal	0.9175	0.8125	0.8924	-0.0004	0.0001	-0.0018
Lena	512*512*3	Horizontal	0.9753	0.9666	0.9337	-0.0004	0.0007	0.0005
		Vertical	0.9852	0.9803	0.9558	0.0026	-0.0002	0.0005
		Diagonal	0.9653	0.9528	0.9177	0.0030	0.0023	0.0007
Goldhill	512*512*3	Horizontal	0.9778	0.9819	0.9845	0.0022	0.0005	0.0006
		Vertical	0.9763	0.9850	0.9864	0.0008	0.0009	0.0007
		Diagonal	0.9604	0.9700	0.9733	-0.0007	0.0004	-0.0048
Airfield	512*512	Horizontal		0.9399			-0.0005	
		Vertical		0.9418			0.0011	
		Diagonal		0.9053			0.0009	

the cipher images encrypted from the plain images with only one-bit change through equation (24). The mean HD over 100 cyphered images with one-bit difference in a random pixel for several tested images are in Table.4. It can be seen that all the HD values are close to 50%.

5.2.4 Information entropy

For image encryption scheme evaluation, the Global Shannon Entropy (GSE) can be employed to evaluate the randomness of the image pixel value. The calculation takes the entire image into account and is formulated as,

$$H(C) = \sum_{i=0}^{Q-1} Pro(c_i) \times \log_2 \frac{1}{Pro(c_i)} \tag{27}$$

where Q is equal to the pixel value classes(256) and $Pro(c_i)$ stands for the number of occurrences of the pixel value c_i in the range of 0 to 255.

For a robust encryption algorithm, the ideal value of GSE for a cipher image with the 8-bit grey level is equal to 8. The mean entropy of the ciphered images for the benchmark images is given in Table.4. The results indicate that the proposed encryption scheme gives rise to randomly distributed image pixel values.

5.2.5 NPCR and UACI results

The sensitivity to the changes of the image is one general requirement for the image encryption scheme to resist differential attacks. Two commonly used indicators are the Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI). The former assesses the change rate of the number of pixels in two cipher images, whereas the latter, UACI, measures the average intensity of the differences between the plain and cipher images. The calculations of these two indicators are given below,

$$\begin{aligned} \mathrm{NPCR} &= \frac{1}{M_1 \times M_2 \times M_3} \times \sum_{u=1}^{M_1} \sum_{\nu=1}^{M_2} \sum_{w=1}^{M_3} D\left[u, \nu, w\right] \times 100\% \\ D\left[u, \nu, w\right] &= \begin{cases} 0, \text{ if } C_1\left[u, \nu, w\right] = C_2\left[u, \nu, w\right] \\ 1, \text{ if } C_1\left[u, \nu, w\right] \neq C_2\left[u, \nu, w\right] \end{cases} \end{aligned} \tag{28}$$

$$\mathrm{UACI} = \frac{1}{M_1 \times M_2 \times M_3 \times 255} \times \sum_{u=1}^{M_1} \sum_{\nu=1}^{M_2} \sum_{w=1}^{M_3} |C_1 - C_2| \times 100\% \tag{29}$$

In equation (28) and (29), C_1, C_2 represent two ciphered images encrypted from one plain image with only one-bit difference in a random pixel; $\mathrm{M}_1 \times \mathrm{M}_2 \times \mathrm{M}_3$ is the size of the image; (u, v, w) stands for the coordinate of the pixel (u-th row, v-th column and w-th color plan). Table.4 gives out the average results of NPCR and UACI for several benchmark color images. For each image, we generate 100 images with a one-bit change in a random pixel position. Knowing that the optimal values for NPCR under $\alpha = 0.05$ and UACI are equal to 99.6094% and 33.4635%, the results prove that the cryptosystem is sensitive to the changes of plain image, which indicates the high resistance to differential cryptanalysis and has good diffusion.

Table 4 Statistical and performance analysis

Image	**Size**	**Mean NPCR**	**Mean UACI**	**Mean** χ^2_{exp}	**Mean HD(%)**	**Mean Entropy**
Baboon	256*256*3	33.4780	99.610	259.3507	50.0141	7.9991
Lena	512*512*3	33.4606	99.6105	255.8031	50.0139	7.9998
Goldhill	512*512*3	33.4590	99.6094	254.6217	49.9879	7.9998
Pepper	512*512*3	33.4606	99.6083	256.0509	49.9981	7.9998
Airfield	512*512*1	33.4611	99.6078	252.9924	49.9889	7.9993
Boat	512*512*1	33.4632	99.6091	255.0193	50.0007	7.9994
Black	512*512*1	33.4579	99.6080	252.2983	49.9986	7.9993
White	512*512*1	33.4493	99.6090	251.9953	50.0097	7.9993

5.2.6 Key space

The secret keys of our proposed cryptosystem are composed of the following variables, 2 fractional orders for the Chen and Lu system(β_c and β_l); one set of control parameters and initial conditions for each of the fractional 3D systems((a_c, b_c, c_c) and $\mathrm{X}_1(0)$ for Chen, and (a_l, b_l, c_l) and $\mathrm{X}_2(0)$ for Lu system); one initial condition for the FGDHL map($X_g(0)$); 4 sets of parameters and intial conditions for the skew tent maps, and one variable tr for the numbers of generated numbers which have been cut off. The ranges of these variable for the FCPRNG are given in the Table.5. With all of these given parameters, the encryption scheme has 24 different components in its secret keys. Since the default precision of the calculation is 10^{-15} for MATLAB, which we

Table 5 Keys and their ranges

Keys	Ranges	Keys	Ranges	Keys	Ranges	Keys	Ranges
$\mathbf{x_{11}(0)}$	[-15,15]	$\mathbf{x_{12}(0)}$	[-15,15]	$\mathbf{x_{13}(0)}$	[0,30]	$\mathbf{x_{21}(0)}$	[-15,15]
$\mathbf{x_{22}(0)}$	[-15,15]	$\mathbf{x_{23}(0)}$	[0,30]	$\mathbf{Xst_1(0)}$	(0,1)	$\mathbf{p_1}$	(0,1)
$\mathbf{Xst_2(0)}$	(0,1)	$\mathbf{p_2}$	(0,1)	$\mathbf{Xst_3(0)}$	(0,1)	$\mathbf{p_3}$	(0,1)
$\mathbf{Xst_4(0)}$	(0,1)	$\mathbf{p_4}$	(0,1)	$\mathbf{Fg(0)}$	(0,0.3)	β_c	[0.75,1]
β_l	[0.75,1]	$\mathbf{a_c}$	[35,40]	$\mathbf{b_c}$	[1.5,3.5]	$\mathbf{c_c}$	[23,28]
$\mathbf{a_l}$	[35,40]	$\mathbf{b_l}$	[3,8]	$\mathbf{c_l}$	[20,25]	$\mathbf{tr}$	[1000,1500]

use, the key space of the encryption scheme can be calculated and is equal to $4.27 * 10^{358}$, which is much larger than the required key space 2^{128} for a secure cryptosystem [46]. In terms of the key space, our proposed FCPRNG outperforms the generator discussed in [47] and [48], which use adaptive Zaslavasky map and Chirikov map with key space of 2^{212} and 2^{159}, respectively. The acquired key space is also greater than almost all the encryption schemes in [49].

5.2.7 Key sensitivity tests

To resist the brute force attack, a cryptosystem must have a sufficiently large keyspace and be highly sensitive to changes in the secret key. 50 different sets of secret keys with only a one-bit difference for each of the 24 different keys components are employed to encrypt the images to evaluate the sensitivity of our proposed encryption scheme to the changes of secret keys. The mean NPCR, UACI, and avalanche(HD) results for each secret key component are given for the colored image "Baboon" in Table.6. It can be observed that the NPCR, UACI, and Hamming distance results for tested color images are all close to their optimal value 99.6094%, 33.4635%, and 50, respectively. This shows that a slight change to the secret key of the cryptosystem will impact the encryption of the image, which confirms the cryptosystem's resistance to brute force attack.

5.2.8 Time consumption

All the simulations have been conducted in MATLAB R2018b on a computer of Intel(R) Core(TM) i7-6700 CPU in Windows 10 professional, 64-bit operating system with 3.40GHz processor, 32 GB RAM. The computational time of the proposed encryption scheme for several images with different sizes is given in Table.7.

Since the encryption time of an image cryptosystem is influenced by many factors, it is highly unlikely to get the explicit comparison results directly comparing the running time in different environments. Therefore, we also adopted the Encryption Throughput(ET) and Number of needed Cycles per Byte(NCpB) to evaluate the encryption speed of the proposed cryptosystem.

Table 6 Key sensitivity analysis of different keys for image Baboon

Secret Key	UACI	NPCR	HD	Secret Key	UACI	NPCR	HD
$\mathbf{x_{11}(0)}$	33.4703	99.6105	49.9997	$\mathbf{x_{12}(0)}$	33.4765	99.6096	50.0053
$\mathbf{x_{13}(0)}$	33.4550	99.6113	49.9957	$\mathbf{x_{21}(0)}$	33.4535	99.6120	49.9975
$\mathbf{x_{22}(0)}$	33.4766	99.6087	50.0037	$\mathbf{x_{23}(0)}$	33.4635	99.6127	50.0006
$\mathbf{Xst_1(0)}$	33.4656	99.6083	49.9912	$\boldsymbol{p_1}$	33.4772	99.6110	49.9918
$\mathbf{Xst_2(0)}$	33.4498	99.6086	50.0016	$\boldsymbol{p_2}$	33.4742	99.6138	50.0041
$\mathbf{Xst_3(0)}$	33.4533	99.6106	50.0006	$\boldsymbol{p_3}$	33.4470	99.6081	49.9915
$\mathbf{Xst_4(0)}$	33.4520	99.6093	50.0059	$\boldsymbol{p_4}$	33.4533	99.6088	49.9988
$\mathbf{Fg(0)}$	33.4620	99.6139	49.9927	$\boldsymbol{\beta_c}$	33.4641	99.6072	49.9928
$\boldsymbol{\beta_l}$	33.4709	99.6074	49.9979	$\boldsymbol{a_c}$	33.4637	99.6111	49.9952
$\boldsymbol{b_c}$	33.4715	99.6075	49.9988	$\boldsymbol{c_c}$	33.4693	99.6106	50.0047
$\boldsymbol{a_l}$	33.4740	99.6086	49.9997	$\boldsymbol{b_l}$	33.4657	99.6108	50.0010
$\boldsymbol{c_l}$	33.4751	99.6111	49.9983	$\boldsymbol{tr}$	33.4626	99.6094	49.9944

The calculation formula for ET and NCpB are given in equation (30) and (31) respectively, and the results are given in Table.7.

$$ET = \frac{Image_{size}(Byte)}{Encryption_{Time}(second)} \tag{30}$$

$$NCpB = \frac{CPUspeed(Hertz)}{ET(Byte/second)} \tag{31}$$

Compared to the encryption schemes given in the table, the computational time is relatively more significant than the other cryptosystems illustrated in [50] and [51]. The generation of the FCPRNG outputs partly contributes to the greater time consumption, which the pie chart in Fig.14 explains. The pie chart displays the time consumption percentage of each scheme component for the encryption of grey images 'Lena' with size 512*512. It can be observed that the encryption process only takes a quarter of the whole cryptosystem running time. In the meantime, around three-fourths of the time (23.7615s out of 31.8463s) is spent on the generation of the pseudo-random numbers using the proposed FCPRNG.

Nevertheless, the merits of the proposed cryptosystem are significant. It is logical to remark that high security gained from more complex schemes often requires longer computational time, and there is a trade-off between the two with respect to the envisaged applications. Therefore, our proposed cryptosystem can be used for security transmission whenever time consumption is not prioritized. In addition, it can be used for secure storage of information such as medical files in hospitals, personal data (such as fingerprints), business documents and is suitable for the use of the home office.

5.2.9 Comparative analysis

In this part, we compared the performances of our proposed cryptosystem with other image encryption algorithms tested on benchmark images in terms

Table 7 Time consumption for several images

Image	Size	Encryption time(s)	ET(MBps)	NCpB
Baboon	256*256*3	22.2238	0.008	384322.7
Airfield	512*512*1	33.4181	0.008	433431.7
Lena	512*512*3	160.3160	0.005	693010.14
LenaGrey	512*512*1	31.8463	0.008	415411.29
LenaGrey Ref([50])	512*512*1	-	0.035	95367.43
LenaGrey Ref([51])	512*512*1	-	0.045	77385.32

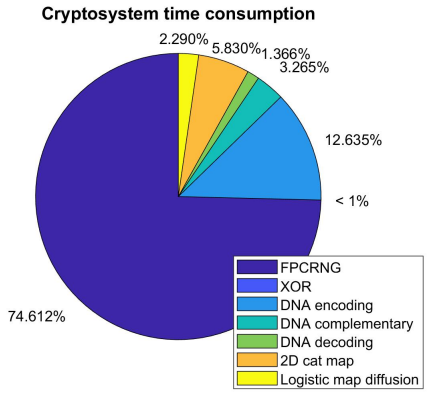

Fig. 14 Percentage of time consumption during each process

of their diffusion and confusion performances. Different works encrypting same benchmark grey and colored images 'Boat' and 'Lena' have been compared. Among them, three have employed DNA-based encryption schemes([29], [36] and [53]). The other three ([4], [52] and [54]) introduced the cryptosystems implementing chaotic systems with different encryption schemes and structures.

The features(well-recognized by the cryptography community) examined for the comparison of the confusion property are the hamming distance, the entropy test, and the correlation coefficients. The diffusion property(against the chosen-plaintext attack) which requires the cryptosystem be highly sensitive to even one bit change in the plain image or in the secret key is analysed through NPRC and UACI.

The results with respect to benchmark images, namely grey and colored 'Lena', and 'Boat', respectively, are given in Table.8 and Table.9. It can be noticed that our proposed cryptosystem achieves similar satisfactory encryption performance with respect to all the above-listed works regarding the evaluated metrics for both diffusion and confusion properties. To further analyze these characteristics, we also marked the values which are closest to the ideal values in bold cases in the comparison tables. For the confusion property, our proposed algorithm possesses closer HD and most of the correlation coefficients for the grey 'Lena' image. For benchmark 'Boat' image, our proposed cryptosystem outperforms the other works in Table.8 for all the given metrics. For the

Table 8 Comparison on confusion property

Image	Cryptosystem	HD(%)	Entropy	Correlation coefficient		
				Horizontal	Vertical	Diagonal
LenaGrey 512*512	**Proposed**	**49.9977**	7.9993	**0.0013**	**-0.00008**	**0.0011**
	Ref[52]	-	7.9977	0.0015	-0.00011	-0.0022
	Ref[29]	-	**7.9994**	0.0032	0.0016	0.0023
	Ref[4]	50.0079	7.9993	0.0018	0.0001	0.0017
Boat 512*512	**Proposed**	**50.0007**	**7.9994**	**-0.00005**	**-0.00004**	**-0.00009**
	Ref[4]	49.9978	7.9993	0.00047	0.00252	0.00124
	Ref[29]	-	7.9994	0.0003	0.0034	0.0011
	Ref[53]	-	7.9965	0.0024	0.0007	0.0040

Table 9 Comparison on diffusion property

Image	Cryptosystem		Plaintext sensitivity		Key sensitivity	
			NPRC(%)	UACI(%)	NPCR(%)	UACI(%)
LenaGrey 512*512	**Proposed**		**99.6107**	**33.4471**	**99.6093**	33.4483
	Ref[4]		99.6080	33.4925	99.6100	**33.4763**
	Ref[52]		99.6184	33.5793	-	-
	Ref[53]		99.6066	33.4977	-	-
LenaRGB 512*512*3	**Proposed**		99.6105	33.4606	**99.6095**	33.4544
	Ref[4]		**99.6097**	33.4573	99.6062	33.4672
	Ref[54]	R	99.6093	33.4678	99.6089	33.4589
		G	99.6099	33.4577	99.6089	33.4598
		B	99.6090	**33.4608**	99.6085	**33.4624**
	Ref[36]	R	99.60	33.56	-	-
		G	99.60	33.45	-	-
		B	99.61	33.49	-	-

diffusion property, we compared the plaintext and key sensitivity employing NPRC and UACI results for grey and colored 'Lena' images. As announced in section 5.2.5, the optimal values for NPRC and UACI are 99.6094% and 33.4635%, respectively. Our proposed scheme achieved better performances concerning the grey 'Lena' image encryption. As for the colored 'Lena' image, the superiority of our proposed cryptosystem is not significant, but it still possesses the advantage of acquiring greater keyspaces with similar secure encryption performance.

6 Conclusion

In this paper, we proposed a novel cryptosystem for image encryption based on a fractional chaotic pseudo-random generator(FCPRNG) and a block cipher. The 3D fractional chaotic systems Chen and Lu have been calculated on two different non-uniform grids whose grid spaces have been determined by the outputs of two distinct skew tent maps to enhance randomness for better security. The outputs of the FCPRNG are then used as the keystream for the encryption scheme. The CBC mode has been adopted for the block cipher. The confusion process of the encryption algorithm is based on permutation performing dynamic DNA encoding and decoding and 2D cat map. After that,

a discrete logistic map is employed to accomplish the diffusion process. The proposed cryptosystem has been tested through various statistical and randomness tests for security. The analysis shows that the encryption scheme can resist attacks such as chosen-plaintext attacks and brute force attacks.

To the best of our knowledge, except for our research, no precedented work has been done on the design of the PRNG based on the fractional chaotic systems, let alone the crytosytsem based on FCPRNG. The computational time of the proposed cryptosystem is relatively longer compared to encryption schemes based on classical chaotic systems, which is partly contributed by relatively greater time consumption of the calculation of 3D fractional chaotic systems. However, the trade-off is the much greater keyspace with 24 independent components in its secret keys, making the cryptosystem very unlikely to be broken. Therefore, the proposed cryptosystem can be successfully implemented whenever security plays a key role and overwhelms the other issues. The cryptosystem can also be adopted for applications for safe data storage, such as the storage for medical folders in hospitals, documents processed in the home office which are more and more frequent nowadays, etc.

Acknowledgements This work is a part of the Ph.D. thesis of Chunxiao YANG, who is supported by the China Scholarship Council (CSC).

Data availability statement

The datasets generated during and analysed during the current study are available from the corresponding author on reasonable request.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Nabe, C.: Impact of covid-19 on cybersecurity. https://www2.deloitte.com/ch/en/pages/585risk/articles/impact-covid-cybersecurity.html (2020)
2. Summerfield, R.: Cyber security and the ongoing impact of covid-19. https://www.financierworldwide.com/cyber-security-and-the-ongoing-impact-of-covid-19#.YdgOBGjMK Ul (2021)
3. Bauer, F.L.: Cryptosystem. Encyclopedia of Cryptography and Security. Springer US, boston, MA (2005)
4. Qiao, Z., El Assad, S., Taralova, I.: Design of secure cryptosystem based on chaotic componentsand AES S-Box. AE$\ddot{U}$ - International Journal of Electronics and Communications / Archivf$\ddot{u}$r Elektronik und $\ddot{U}$bertragungstechnik. 121, 153205 (2020)
5. Debnath, L.: A brief historical introduction to fractional calculus. International Journal of Mathematical Education in Science and Technology. 35(4), 487–501 (2004)
6. Machado, J. T., Kiryakova, V., Mainardi, F.: Recent history of fractional calculus. Communications in Nonlinear Science and Numerical Simulation. 16, 1140-1153 (2011)

7. Odibat, Z. M., Corson, N., Aziz-Alaoui, M., Bertelle, C.: Synchronization of chaotic fractional-order systems via linear control. International journal of bifurcation and chaos in applied sciences and engineering. 20(1), 1-15 (2010)
8. Mainardi, F.: Fractional Calculus and Waves Linear Viscoelasticity: An Introduction to Math-ematical Models. Imperial College Press, London, UK (2010)
9. Tarasov, V., Tarasova, V.: Macroeconomic models with long dynamic memory: Fractional calculus approach. Applied Mathematics and Computation. 338, 466–486 (2018)
10. Yang, C., Taralova, I., Loiseau, J.J.: Fractional chaotic system solutions and their impact on chaotic behaviour. The 14th CHAOS 2021 International Conference, Athens, Greece (turned into a virtual conference due to COVID-19) (2021)
11. Yang, C., Taralova, I., Loiseau, J.J.: Fractional chaotic system solutions and their impact onchaotic behaviour. 6th IFAC Conference on Analysis and Control of Chaotic Systems CHAOS 2021. IFAC-PapersOnLine. 54(17), 154–159 (2021)
12. Yang, C., Taralova, Loiseau, J.J., Elassad, S.: A stream cipher based on fractional pseudo chaotic random number generator. 2020 15th International Conference for Internet Technology and Secured Transactions (ICITST). 1–6 (2020). doi:10.23919/ICITST51030.2020.9351350
13. Radwan, A.G., Abd-El-Hafiz,S.K., AbdElHaleem, S.H.: Image encryption in the fractional-order domain. 2012 International Conference on Engineering and Technology (ICET). 1-6 (2012)
14. Yang, F., Mou, J., Liu, J., Ma, C., Yan, H.: Characteristic analysis of the fractional-orderhyper chaotic complex system and its image encryption application. Signal Processing 169 , 107373 (2020)
15. Yang, F., Mou, J., Liu, J., Ma, C., Cao, Y.: Dynamic analysis of an improper fractional-order laser chaotic system and its image encryption application. Optics and Lasers in Engineering 129, 106031 (2020)
16. Gao, X., Yu, J., Banerjee, S., Yan, H., Mou, J.: A new image encryption scheme based on fractional-order hyperchaotic system and multiple image fusion, Scientific Reports 11. 15737 (2021)
17. Bassham, L.E., Rukhin, A.L., Soto, J., Nechvatal, J.R., Smid, M.E., Barker, E.B., Leigh, S.D., Levenson, M., Vangel, M., Banks, D.L., et al.: Sp 800-22 rev. 1a. A statistical test suitefor random and pseudorandom number generators for cryptographic applications. National Institute of Standards & Technology (2010)
18. Gehani A., LaBean T., Reif J.: DNA-based Cryptography. Jonoska N., Păun G., Rozenberg G. (eds) Aspects of Molecular Computing. Lecture Notes in Computer Science, pp. 167–188, vol 2950. Springer, Berlin, Heidelberg (2003)
19. Clelland, C., Risca, V., Bancroft, C.: Hiding messages in DNA microdots. Nature 399, 533–534 (1999). https://doi.org/10.1038/21092.
20. Leier, A., Richter, C., Banzhaf, W;, Rauhe, H.: Cryptography with DNA binary strands. Biosystems. 57(1):13-22 (2000). doi:10.1016/s0303-2647(00)00083-6
21. Marwan, S., Shawish, A., Nagaty, K.: DNA-based cryptographic methods for data hiding in DNA media. Biosystems. 150:110-118 (2016). doi:10.1016/j.biosystems.2016.08.013
22. Kalsi, S., Kaur, H., Chang, V.: DNA Cryptography and Deep Learning using Genetic Algorithm with NW algorithm for Key Generation. J Med Syst. 42(1), 17 (2017). doi:10.1007/s10916-017-0851-z
23. Adleman, LM.: Molecular computation of solutions to combinatorial problems. Science. 266(5187), 1021-1024 (1994). doi:10.1126/science.7973651
24. Xue, X., Zhou, D., Zhou, C.: New insights into the existing image encryption algorithms based on DNA coding. PloS one, 15(10), e0241184 (2020). doi: 10.1371/journal.pone.0241184
25. Biswas, M.R., Alam, K.M.R., Akber, A., Morimoto, Y.: A DNA cryptographic technique based on dynamic DNA encoding and asymmetric cryptosystem. 2017 4th International Conference on Networking, Systems and Security (NSysS), pp. 1-8 (2017). doi: 10.1109/NSYSS2.2017.8267782
26. Yin, C.: Encoding and Decoding DNA Sequences by Integer Chaos Game Representation. J Comput Biol. 26(2), 143-151 (2019). doi:10.1089/cmb.2018.0173
27. Patro, K.A., Acharya, B.,Nath, V.: Secure, Lossless, and Noise-resistive Image Encryption using Chaos, Hyper-chaos, and DNA Sequence Operation. IETE Technical Review. 37, 223 - 245 (2019)

28. Zhang, J., Hou, D., Ren, H.: Image Encryption Algorithm Based on Dynamic DNA Coding and Chen's Hyperchaotic System. Mathematical Problems in Engineering. 1-11 (2016)
29. Wu, J., Liao, X.,Yang, B.: Image encryption using 2D Hénon-Sine map and DNA approach. Signal Process., 153, 11-23 (2018)
30. Wang, X., Wang, Y., Zhu, X., Luo, C.: A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level and DNA level. Optics and Lasers in Engineering, 125, 105851 (2020)
31. Wen, W., Wei, K., Zhang, Y. et al,; Colour light field image encryption based on DNA sequences and chaotic systems. Nonlinear Dynamics. 99, 1587–1600 (2020). https://doi.org/10.1007/s11071-019-05378-8
32. Petráš, I. Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation. Springer, Berlin, Hidelberg (2011)
33. West, B.J., Bologna, M., Grigolini, P.: Physics of Fractal Operators. Springer-Verlag, New York (2003)
34. Caputo, M.: Linear models of dissipation whose q is almost frequency independent-ii. Geo-physical Journal International. 13, 529–539 (1967)
35. Watson, J.D., Crick, F.H.: Molecular structure of nucleic acids; a structure for deoxyribosenucleic acid. Nature, 737–738 (1953). doi:10.1038/171737a0.675
36. Chai, X., Fu, X., Gan, Z., Lu, Y., Chen, Y.: A color image cryptosystem based on dynamic DNA encryption and chaos. Signal Processing., 155, 44-62 (2019). https://doi.org/10.1016/j.sigpro.2018.09.029.
37. Zhang, Q., Guo, L., Wei, X.: A novel image fusion encryption algorithm based on dna sequenceoperation and hyperchaotic system. Optik- International Journal for Light and Electron Optics. 124(18), 3596–3600 (2013). doi:10.1016/j.ijleo.2012.11.018.
38. Wei, X., Guo, L., Zhang, Q., Zhang, J., Lian, S.: A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system. J. Syst. Softw. 85, 290-299 (2012)
39. Wang, Y., Lei, P., Yang, H,. Cao, H.: Security analysis on a color image encryption based on DNA encoding and chaos map. Comput. Electr. Eng. 46, C(August 2015), 433–446 (2015). doi:https://doi.org/10.1016/j.compeleceng.2015.03.011
40. Akhavan, A., Samsudin, A., Akhshani, A. Cryptanalysis of an image encryption algorithm based on DNA encoding. Optics and Laser Technology, 95, 94-99 (2017). doi:10.1016/j.optlastec.2017.04.022.690
41. Yang, C., Taralova, I., Loiseau, J.J., El-Assad, S.: Design of a Fractional Pseudo-Chaotic Random Number Generator. International Journal of Chaotic Computing, Infonomics Society. 7(1), 166-178 (2020)
42. El-Sayed, A.M., Salman, S.M. On a discretization process of fractional-order logistic differential equation. Journal of the Egyptian Mathematical Society. 22(3), 407-412 (2014). https:695//doi.org/10.1016/j.joems.2013.09.001.
43. Farajallah, M., El Assad, S., Deforges, O.: Fast and secure chaos-based cryptosystem for images, International journal of bifurcation and chaos in applied sciences and engineering. 26 (2), 1650021.1-1650021.21 (2016). doi:10.1142/S0218127416500218.
44. Farajallah, M., El Assad, S., Chetto, M.: Dynamic adjustment of the chaos-based security in real-time energy harvesting sensors. IEEE International Conference on Green Computing and Communications, Beijing, China (2013)
45. El Assad, S., Farajallah, M.: A new chaos-based image encryption system. Signal Processing:Image Communication 41, 144–157 (2016). https://doi.org/10.1016/j.image.2015.10.004.705
46. F. Özkaynak. Brief review on application of nonlinear dynamics in image encryption. Nonlinear Dynamics. 92, 305–313 (2018)
47. Tutueva, A., Nepomuceno, E., Karimov, A., Andreev, V., Butusov, D.: Adaptive chaotic maps and their application to pseudo-random numbers generation. Chaos, Solitons Fractals. 133, 109615 (2020)
48. Tutueva, A., Pesterev, D., Karimov, A., Butusov, D., and Ostrovskii, V.: Adaptive Chirikov Map for Pseudo-random Number Generation in Chaos-based Stream Encryption. 2019 25th Conference of Open Innovations Association (FRUCT). 333-338 (2019)

49. Gayathri, J., Subashini, S.: A survey on security and efficiency issues in chaotic image encryption. International Journal of Information and Computer Security, 8(4) (2016). doi: 10.1504/IJICS.2016.080427
50. Luo, Y., Zhou, R., Liu, J. et al. An efficient and self-adapting colour-image encryption algorithm based on chaos and interactions among multiple layers. Multimed Tools Appl 77, 26191–26217 (2018). https://doi.org/10.1007/s11042-018-5844-5
51. Qiao, Z., Taralova, I., El Assad, S.: A robust pseudo-chaotic number generator for cryptosystem based on chaotic maps and multiplexing mechanism. in: International Conference for Internet Technology and Secured Transactions (ICITST'2019), London, United Kingdom. (2019). doi: 10.20533/ICITST.WorldCIS.WCST.WCICSS.2019.0006
52. Zhao, CF., Ren, HP.: Image encryption based on hyper-chaotic multi-attractors. Nonlinear Dynamics. 100, 679–698 (2020). https://doi.org/10.1007/s11071-020-05526-5
53. Wang, X., Su, Y.: Image encryption based on compressed sensing and DNA encoding. Signal Processing: Image Communication. 95, 116246 (2021)
54. Huang, L., Cai, S., Xiao, M., Xiong, X.: A Simple Chaotic Map-Based Image Encryption System Using Both Plaintext Related Permutation and Diffusion. Entropy. 20(7), 535 (2018). https://doi.org/10.3390/