Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 28 (2025), No. 7, pp. 2645–2670
DOI : 10.47974/JDMSC-1847

Design and analysis of a pseudo random number generator based on the logistic-sine system and a nonlinear feedback shift register : Application to image encryption

Hadjer Bourekouche *
Samia Belkacem †
Noureddine Messaoudi §
Department of Engineering of Electrical Systems
Faculty of Technology
University of Boumerdes
35000 Boumerdes
Algeria

Abstract

This study aimed to improve the output quality and statistical characteristics of random output sequences by combining the output of an 8-bit Fibonacci nonlinear feedback shift register (NLFSR) of degree three with a sequence of keys generated by a coupled logistic-sine system (LSS). Subsequently, the proposed pseudorandom number generator PRNG-LSS-NLFSR was investigated using a simple image-encryption application. Theoretical randomness evaluation through the National Institute of Standards and Technology (NIST) and other statistical test of its application by means of the histogram, entropy analysis, correlation test, and other tests confirm that the proposed PRNG-LSS-NLFSR is simple, has random and nonlinear features, and displays a strong cryptographic security aspect.

Subject Classification: *(2010) 11T71, 68P25, 94A60, 68R01.*

Keywords: *NLFSR, Logistic-sine system, Random number generator, PRNG-LSS-NLFSR, NIST, Image encryption, Security analysis.*

* *ORCID-ID :* **0000-0003-3516-290X**
† *ORCID-ID :* **0000-0003-0912 3392**
§ *ORCID-ID :* **0000-0002-3406-4120**

* *E-mail:* **h.bourekouche@univ-boumerdes.dz (Corresponding Author)**
† *E-mail:* **s.belkacem@univ-boumerdes.dz**
§ *E-mail:* **n.messaoudi@univ-boumerdes.dz**

1. Introduction

Images are the most crucial multimedia material that must be preserved and shared over networks, which makes their security a major concern. Current image encryption algorithms include block ciphers, genetic algorithms, deoxyribonucleic acid (DNA) encoding, and convolution neural network (CNN)[1] . Among these techniques, chaos-based image encryption is deemed to be one of the most practical and outstanding encryption methods because it improves the quality of chaotic systems. Key streams are often produced by chaotic systems in chaos-based image encryption [2], which have remarkable characteristics of determinism and sensitivity to initial conditions[3].

Radom number generators, such as linear congruential generators (LCGs), cellular automata (CA), chaotic generators, DX generators, and DL generators [4], are essential in the design of encryption and decryption algorithms. For example, Pseudo-Chaotic Number Generators (PCNGs)[3] can produce pseudo-random numbers with a high accuracy. For instance, the logistic map (LM) is the best demonstration to show how a basic recursive nonlinear dynamic equation is adequate to give birth to complicated chaotic behavior and create a high-quality sequence of random numbers. However, for the chaotic sequence, we just use the parameter r and the initial state X_0 as initial keys, which limits logistic map applications [5]. As a result, the sequences created by the logistic map cannot be replicated [6], and the key generated is relatively small [5], which means that the map needs to be connected to another generator as well as a deterministic random number generator.

In addition to PCRNGs, pseudorandom number generators based on feedback shift registers (FSRs) are frequently used [7]. This is mainly because they feature fast and compact implementations, and can ensure extremely long periods. Based on the type of feedback loop used, this class of FSRs can be divided into two categories [8]. The LFSRs are prone to numerous threats [9], [10], and because of their linear nature, they are unsuitable for generating high-quality random number sequences when used as key-stream generators, encryption secrets can be readily deciphered [11]. The NLFSR is an extension of the LFSR, the past state is a non-linear function of the present state, making it a suitable choice for designing numerous cryptographic algorithms, because the random sequence produced by a linear feedback shift register cannot satisfy the unpredictability requirement for secure paradigms [12]. NLFSR sequences are often recognized as being more resistant to cryptanalytic attacks [13], but they are generally inefficient to use directly as PRNGs for cryptography applications. General information about the current designs for chaotic PRNG, along with FSR applications in designing random sequences, is presented in Table 1.

The main contributions of this study are as follows: Given the capacity of the NLFSR to create extremely nonlinear pseudorandom numbers and LSS

employment in highly safe encryption algorithms owing to its chaotic nature, our proposed technique produces a random sequence with a longer period and more complicated architecture. We demonstrate the applicability of the proposed PRNG-LSS-NLFSR by emphasizing some of its intriguing statistical aspects through a simple image-encryption application based on the XOR operation. We also provide the complete results of the statistical testing performed on the PRNG-LSS-NLFSR using the most exacting tests of randomness in the NIST suite to identify the particular properties anticipated for pseudorandom sequences.

The paper is structured as follows: In Section 2, summary information relevant to the NLFSR and LSS theory is provided. Section 3 discusses the conceptual architecture of the proposed PRNG-LSS-NLFSR and encryption/decryption algorithm. In Section 4, the statistical test suite for the proposed PRNG-LSS-NLFSR and the experimental results of its application are presented and discussed. In Section 5, a comparative study is presented to determine whether the adoption of the proposed system affects important outcomes.

Table 1
Recent works on chaos/FSR-based pseudorandom numbers generators

Ref	Generators	Model & Details	Control parameters	Tested through	Pros	Implementation	Application purpose
[14]	Pseudo randomly enhanced logistic map (PRNG–PELM)	The enhanced logistic map used to generate a sequence of key: $X_{i+1} = mod((aX_i(1 - X_i)(100000), 1)$ Length of generated sequence:8bits	(a, X_0) $3.999 < a < 4$ $0 < X_0 < 1$	-NIST 800-22 - TestU01	-Robustness against differential attacks; high pseudo-randomness; -Secure, effective, high speed; -Requires less arithmetic operation; -High processing output.	MATLAB	Cryptography
[15]	The XOR enhanced logistic map (XELM-PRBG) The XOR enhanced Henon map (XEHM-PRBG)	An XOR operation performed on the mantissas of the state variables of the chaotic maps improves the 1-D logistic map and the 2-D Henon map. For each bit of the PRBG, the method employs one XEHM cascaded with two XELMs. The two XELMs are: $\begin{cases} x_{i+1} = \Psi[4\mu_1 x_i(x_i - 1)] \\ y_{i+1} = \Psi[4\mu_2 y_i(y_i - 1)] \end{cases}$ After that, the 2-dimensional XEHM is created by cascading these two XELMs	$(x_0, \mu_1, \mu_2, \alpha, \beta)$ $x_0 = 0.21;$ $\mu_1 = 0.91;$ $\mu_2 = 0.961;$ $\alpha = 4.7;$ $\beta = 1.3$	-NIST 800-22	-A desired chaotic behavior over a significantly larger range of system variables; -Secure against the differential attacks; -Increased key-space.		Cryptographic systems

		as follows: $\begin{cases} z_{1,i+1} = \Psi[1 - \alpha x^2{}_{i+1} + y_{i+1}] \\ z_{1,i+1} = \Psi[\beta x_{i+1}] \end{cases}$					
[16]	Modified Discretized Chaotic Map (MDCM)	Modified Discretized Chaotic Map used to generate a sequence of key: $x[n+1] = [\lambda * x[n] * \left(1 + \frac{1}{\lambda} - x[n]\right)] * mod(1)$	(x_0, n, λ)	-NIST	-Uniformly distributed produces discrete numbers throughout the range (0, 1). -Statistically independent produces binary numbers of each other .	MATLAB	-Digital processing technology - Cryptography
[17]	Chaotic PRNG Based on NLFSR and 1D piecewise linear one-dimensional map (PW)	The 10-bit Fibonacci NFSR of parallel type with feedback function: $x_0 = (x_9 \oplus x_1 \oplus x_4) \oplus (x_3 . x_7)$ The piecewise linear one-dimensional map used: $x_{n+1} = \begin{cases} -2x_n + \frac{1}{2} & 0 \le x_n \\ 2x_n & \frac{1}{4} \le x_n \\ 2(1 - x_n) & \frac{1}{2} \le x \\ 2\left(x_n - \frac{1}{2}\right) - \frac{1}{2} & \frac{3}{4} \le x \end{cases}$	x_0	-NIST	-Random numbers generated with high output quality; -Noisy-like output signal.	ISE Project Navigator	
[18]	Pseudorandom bit generator based on LFSRs and a discrete chaotic map (DCM)	Two LFSRs with lengths of 31 and 33 bits are connected through XOR operation with discrete chaotic maps defined as: $Z_{i+1} = l[l^{-1}(z_i) \circ f(l^{-1}(z_i), l^{-1}(c))]$	$(z_i, c) \in \{1,2, \dots m! - 1\}$	-NIST, - TestU01, - DIEHARD	-Infinite key space; -Resistance to digital degradation caused by using continuous chaotic maps in digital systems	MATLAB (R2016a) s	Cryptosystems
[19]	Pseudo-random number generators of chaotic sequences (LSP-PRNG) based on LFSR, Logistic, Skew-Tent, PWLCM map,3D-Chebyshev Map	32-bit LFSR coupled with logistic map, 3D-Chebyshev Map, PWLCM, Skew-Tent map through Xor operation feedback function of the LFSR: $Q(x) = x^{32} + x^{22} + x^2 + x + 1$	$(XL0, XS0, XP0, P_s, P_p)$	-NIST	-Resistance against algebraic attacks -High level of security -Good hardware metrics -Resistance to cryptanalysis attacks	Xilinx PYNQ-Z2	Cryptographic systems

2. Methods for RNG: Theoretical Foundations

In this section, the main properties of the NLFSR and LSS used for designing our generator are reviewed.

A. *Nonlinear linear feedback shift register (NLFSR)*

The NLFSR of n binary storage elements (represented by x_i, $i \in [0, n-1]$) is an enhanced variant of the LFSR where the present state is a nonlinear feedback function of prior states [20] [22] [21] (see Figure 1). The bit values x_i^t, $i \in [0, n-1]$ stored in all the stages form the NLFSR's internal state[21], as revealed by $X^t = \{x_0^t, x_1^t, \ldots, x_{n-1}^t\}$ at clock t.

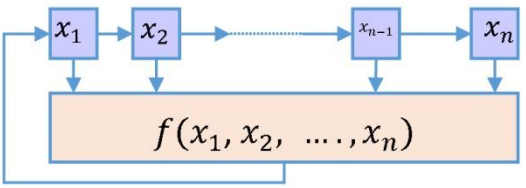


Figure 1
General diagram of the Feedback Shift register

There are two main ways in which an NLFSR can be constructed: the Fibonacci and Galois configurations[22] (see Figure 2). In the Fibonacci setup, the feedback is applied at the final step, in contrast to the Galois setup, where feedback is applied at every step [23].

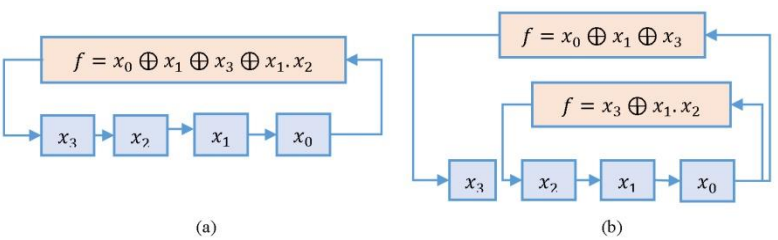


Figure 2
4bit-NLFSR configurations : (a) Fibonacci; (b) Galois

The feedback function or connection polynomial are additional names for the feedback path [24], and the degree of the feedback function utilized to create a series of keys is vital to the system security and must be properly set [25].

For instance, Dubrova et al.[12] developed the following form of the NLFSR Fibonacci feedback functions of degree 3, with optimum periods of 2^{n-1} [26], for $4 \leq n \leq 25$[55]:

$$f(x_0, x_1, \ldots\ldots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \oplus x_c.x_d.x_e \quad (1)$$

Although a, b, c, d, and e are members of the set {1, 2… n – 1}. The potential valid feedback polynomials for the 8 bit NLFSR of degree 3 proposed in [26] are as follows:

8(0,2,7,1,2,7)	8(0,6,7,2,3,4)	8(0,4,5,2,5,7)
8(0,2,7,2,3,5)	8(0,6,7,2,3,6)	8(0,4,5,3,4,7)
8(0,2,7,2,3,7)	8(0,6,7,3,6,7)	8(0,4,5,4,5,7)
8(0,2,7,2,4,5)	8(0,6,7,4,5,6)	8(0,4,7,1,3,7)
8(0,2,7,2,5,6)	8(0,6,7,1,2,3)	8(0,4,7,1,4,5)
8(0,2,7,2,6,7)	8(0,6,7,1,5,6)	8(0,4,7,1,6,7)
8(0,2,7,3,4,7)		8(0,4,7,2,4,6)
		8(0,4,7,3,4,5)
		8(0,4,7,4,6,7)

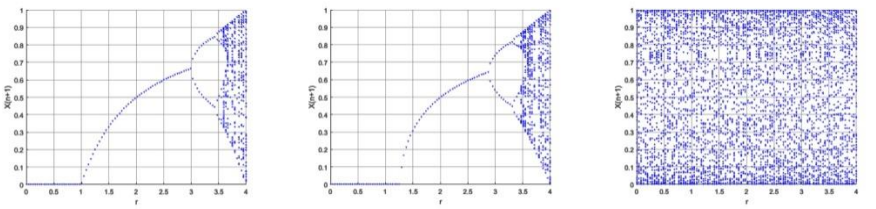
B. *Logistic-Sine system (LSS)*

Coupled chaotic sources are highly sensitive to circuit noise and system parameters. Therefore, combining the logistic and sine maps forms the 1D LSS [27] demonstrated in [28]. Mathematically, the LSS equation is given by [29]

$$X_{n+1} = \mathcal{F}_{LSS}(r, X_n) = \left[\mathcal{L}(r, X_n) + S\left((4-r), X_n\right)\right] mod 1$$
$$= [rX_n(1 - X_n) + (4 - r)\, sin(\pi X_n)] mod 1 \quad (2)$$
$$r \in (0;\ 4]$$

The chaotic character of the LSS is the greatest across the majority of regions [0–4]; however, it is not uniformly chaotic throughout [30]. Interestingly, it differentiates between periodic, quasiperiodic, and chaotic behaviors at various r values. Furthermore, it was proven in [30] that these maps exhibit chaotic behavior, but not uniform; certain places exhibit quasi-periodic behavior when $r \epsilon [3.4 - 3.59] \cup [3.8 - 3.89]$ and are chaotic elsewhere.

The bifurcation diagrams and Lyapunov exponent (LE)[31] of the logistic map, sine map, and LSS are shown in Fig. 3 and Fig. 4, which show the dispersion (or lack thereof) of iterations for a range of r-values.



(a) (b) (c)

Figure 3

Bifurcation diagram: (a) logistic map; (b) sine map; (c) LSS system

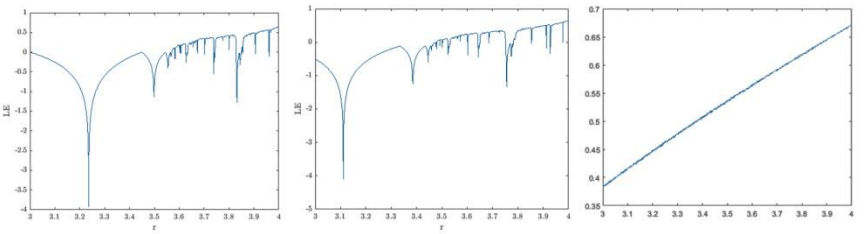


Figure 4
Lyapunov exponent: (a) logistic map ;(b) sine map; (c) LSS system

3. The Construction Method

The construction method consisted of two steps.

- Design of the PRNG-LSS-NLFSR
- Design of the encryption and decryption algorithm based on the PRNG-LSS-NLFSR model

A. *Design of the PRNG-LSS-NLFSR*

The LSS, NLFSR, and XOR operations are the three fundamental building blocks used to design the final key-sequence generator. The operation of the global system is explained using the structural diagram shown in Figure 5.

PRNG-LSS

The one-dimensional LSS system properties are used to create the initial pseudo-sequence, whereas using equation (4), it is possible to build an endless chaotic number sequence. Noting that little perturbation of the initial condition and control parameter may completely impact the random behavior of the system. Because the control parameter should be set to be in the chaotic range, the r value that we designate as secret key1 is set according to a microscopic analysis of its chaotic behavior in the range [3-4] done by Muthu et al.[30], where three distinct behaviors of LSS at various r values are observed. Based on [30], the best chaotic behavior of the LSS was observed in the range $r \in$[3,6–3,69]. Therefore, we can utilize the starting value X_0=0.1 and r to produce a sufficiently lengthy chaotic sequence. Before that, enhancement of the LSS behavior was necessary. Since the chaotic sequence generator generates values between 0 and 1 with 10^{-15} decimals, we use equation (5) to enhance the properties of the LSS and produce better pseudorandom sequences (see Figure 6 that represent an enhancement version of the LSS through LE), and second, to convert each value of the chaotic sequence X_n to an 8-bit integer. Finally, the generated sequence is converted into binary data.

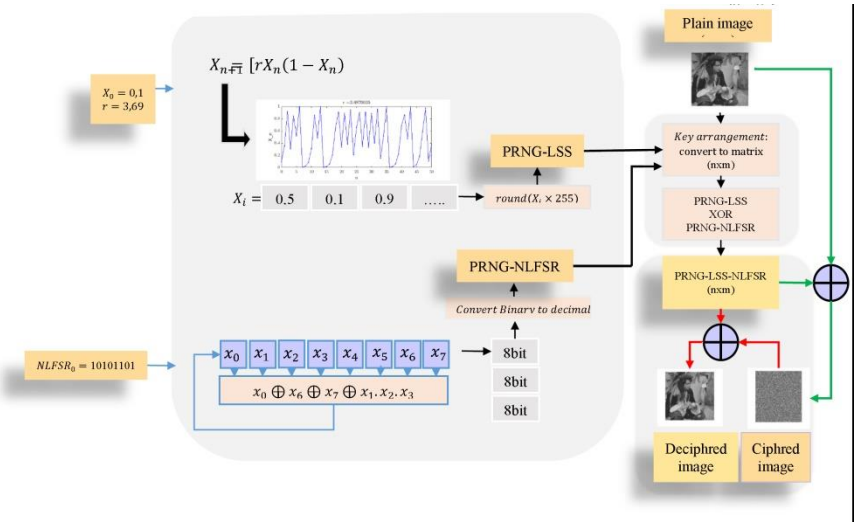


Figure 5
Structural diagram of the proposed PRNG-LSS-NLFSR

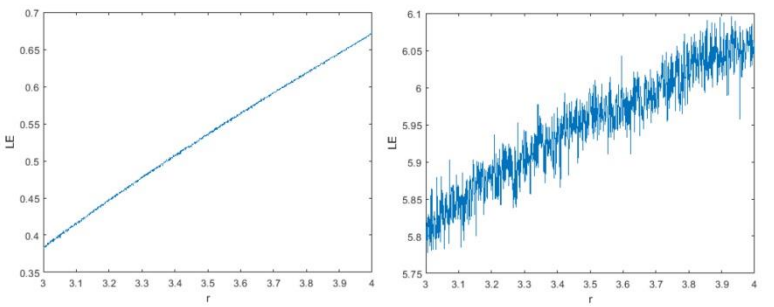


Figure 6
Lyapunov exponent: (a) LSS; (b) enhanced LSS

$$PRNG - LSS = round(X_i \times 255) \quad (3)$$

PRNG-NLFSR

The second pseudo-sequence is generated using an 8-bit Fibonacci NLFSR of degree three defined by:

$$f(x_0, x_1, \ldots\ldots, x_{n-1}) = x_0 \oplus x_6 \oplus x_7 \oplus x_1.x_2.x_3 \quad (4)$$

The generated random sequence is periodic with a period of 2^8-1=255. Note that the XOR($\oplus$) operation in equation (6) is a linear function whereas the

AND (.) is a nonlinear function; thus, taps $x_0, x_6,$ and x_7 are XORed with the nonlinear function $x_1.x_2.x_3$ to obtain the output bit and feedback to the rightmost bit.

Because the number of 0s and 1s in the initial state of the NLFSR should be as equal as feasible, the initial state of the NLFSR is set to=[10101101]. The output sequence is obtained from the tap in the seventh register.

PRNG- LSS-NLFSR

The final sequence is achieved by performing a bitwise XOR operation between the random outputs of PRNG-LSS and PRNG-NLFSR of equal length.

A. *Design of the image encryption algorithm based on the PRNG-LSS-NLFSR*

In this section, we describe the application of the developed PRNG-LSS-NLFSR method for image encryption. The procedure that will be followed comprises two stages. In the first phase, a key arrangement is performed to adapt the produced key to a size similar to the source image, where we reshape the generated pseudosequence vector to an M × N matrix named K by flowing the row-major ordering. Given an M × N source image represented by a matrix P at the grey scale level, where, the grey level of each pixel is represented with entries of 0 or 255. The encryption and decryption mechanisms are outlined in the following steps:

Encryption

The input image P to be encrypted is joined with a keystream matrix K of equal length obtained using the approach provided in the preceding section, and the encrypted matrix $E = K \oplus P$ that may be safely transmitted in a secure communications scheme is produced by employing the frequently used XOR method between the plain image P and secret key K.

Decryption

The decryption process is performed using the reverse technique of the encryption phases, in which we perform $\hat{E} = E \oplus K$ on the encrypted image E to obtain the decrypted image $\hat{E}$.

4. Tests and evaluation

The running key generator described in Section 3 simulates a sequence used to encipher the plain image. The security of such a synchronous key generator depends on the "randomness" of the sequence produced. In this section, the proposed PRNG-LSS-NLFSR is subjected to several parametric tests, which are subsequently used to validate its suitability for cryptographic applications.

A. *Test of randomness - NIST suite test*

The NIST has set rules for random number production to recognize the utility of random number generators, especially in fields such as cryptography. The specifications of the 15 statistical tests are available at [32]. Table 2 lists the NIST SP-800 test results obtained after processing 1000 binary sequences by using the GitHub Python implementation[1].

Conclusion: The sequence is considered as random since the P-value for each of the 15 tests was $\geq$ 0.01, indicating that our proposed PRNG-LSS-NLFSR is indeed a highly randomness properties.

B. *Key sensitivity analysis*

Correlation tests based on Pearson's correlation coefficient[33] and the Hamming distance[34] were used to examine the consistency between the key sequences created in order to guarantee the sensitivity of the keys [35].

Table 2
NIST test suite

n	Test	P-value	Conclusion
1	Frequency Test (Monobit)	0.589159	Pass
2	Frequency Test within a Block	0.921522	Pass
3	Run Test	0.476752	Pass
4	Longest Run of Ones in a Block	0.671805	Pass
5	Binary Matrix Rank Test	0.419681	Pass
6	Discrete Fourier Transform (Spectral) Test	0.741104	Pass
7	Non-Overlapping Template Matching Test	0.165002	Pass
8	Overlapping Template Matching Test	0.232730	Pass
9	Maurer's Universal Statistical test	-1.0	Fail
10	Linear Complexity Test	0.415656	Pass
11	Serial test	0.756912 0.734652	Pass Pass
12	Approximate Entropy Test	0.049818	Pass
13	Cummulative Sums (Forward) Test	0.499073	Pass
14	Cummulative Sums (Reverse) Test	0.931289	Pass
15	Random Excursions Test	0.761365	Pass

Pearson's correlation coefficient

The Pearson correlation coefficient[36] given by equation (7) is a statistical indicator of the strength of the linear connection between paired data [35].

[1] https://github.com/stevenang/randomness_testsuite.git

Where str and str1 represent two sequences specified by $Str = [x_1, .. x_n]$ and $Str1 = [y_1, .. y_n]$ respectively.

$$R_{(str,str1)} = \frac{\sum_{i=0}^{n-1}(x_i - \bar{x}).(y_i - \bar{y})}{[\sum_{i=0}^{n-1}(x_i - \bar{x})^2]^{1/2}.[\sum_{i=0}^{n-1}(y_i - \bar{y})^2]^{1/2}} \quad (5)$$

$\bar{x}$ and $\bar{y}$ mentioned in equation (8) represent the mean values of Str and Str1 respectively.

$$\begin{cases} \bar{x} = \sum_{i=0}^{n-1} \frac{x_i}{n} \\ \bar{y} = \sum_{i=0}^{n-1} \frac{y_i}{n} \end{cases} \quad (6)$$

Hamming distance

The average Hamming distance for two binary sequences given by the following equation [35]:

$$d(s_1^b, s_2^b) = \sum_{j=0}^{m-1}(x_j \oplus y_j) \quad (7)$$

where x_j and y_j are the elements of s_1^b and s_2^b respectively.

Table 3
Pearson's correlation and the hamming distance test results

	K1 vs K2	K1 vs K3	K2 vs K3
Pearson's correlation coefficients	-0.00209	0,00106	0,00170
Hamming distance	0.99615	0.99594	0.99606

In this test, three major binary keystream sequences, K1, K2, and K3, were generated using slightly different starting values.

K_0 was obtained using the following initial values:

$r_0 = 3.699000000000011$, $X_0 = 0.100000000000011$, NLFSR =10101101.

K_1 was obtained using the following initial values:

$r_1 = 3.699000000000099$, $X_1 = 0.100000000000099$, , NLFSR =10101100.

K_3 was obtained using the following initial values:

$r_0 = 3.699000000000059$, $X_0 = 0.100000000000067$, NLFSR =10101001.

The correlation results explained by Pearson's correlation coefficients with the Hamming distance test [35] of the three key sequences were constructed as described in Table 3. This means that one set of sequences is redundant if two sequences have a Pearson correlation and Hamming distance of 1 or -1.

In our case, the relationship between K1, K2, and K3 is assigned a value between -1 and 1, which might imply that the three sets of keys are uncorrelated, and can be interpreted as the three sequences of keys having different information contents. The suggested PRNG-LSS-NLFSR appears to be highly sensitive to the system parameters and initial values (keys), and it has the capacity to create a large number of key sequences without any discernible

associations between them. This capability may be helpful for many cryptographic applications.

A. *Statistical analysis*

In this section, histogram analysis, along with the correlation coefficient tests in three directions [37] and entropy, are the three metrics used to assess the efficacy of the proposed image encryption algorithm against statistical assaults.

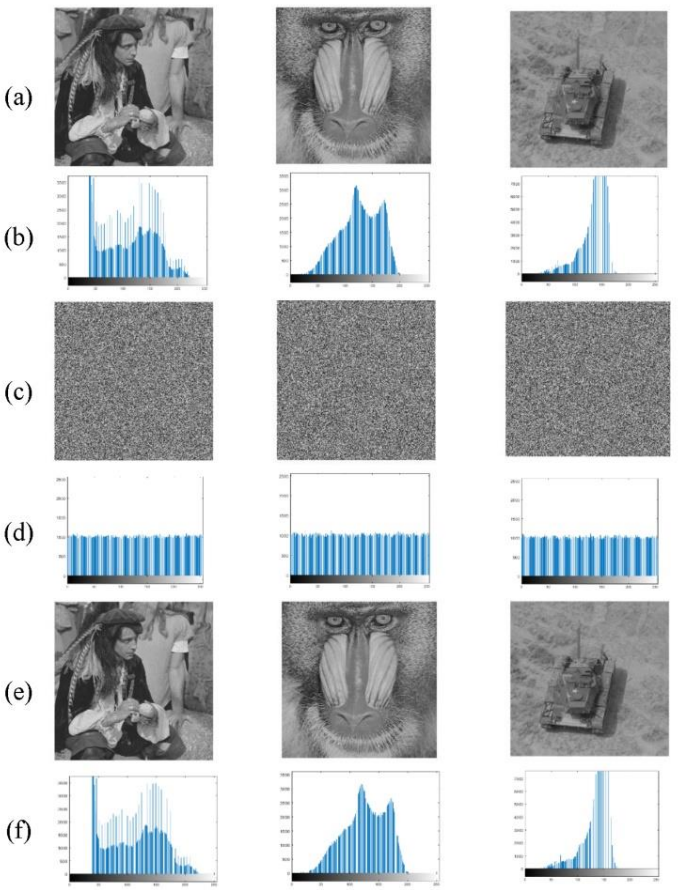


Figure 7
Histogram analysis: (a) original images; (b) histograms of original images; (c) encrypted images; (d) histogram of encrypted images; (e) decrypted images; (f) histograms of decrypted images

Histogram analysis

The term "image histogram" refers to the bars that represent each individual byte in an image. The regularity of the pixels suggests a strong encryption strategy that can withstand all statistical assaults, in contrast to the jagged edges of these bars, which reflects a poor encryption technique [38]. Figure 7 shows the histogram analysis results of the 512 × 512 standard Man, Baboon, and Tank images from the USC-SIPI image database[2][39]. With a flat histogram distribution of the pixel values of the encrypted images, it make it difficult for an attacker to deduce pixel values or secret keys.

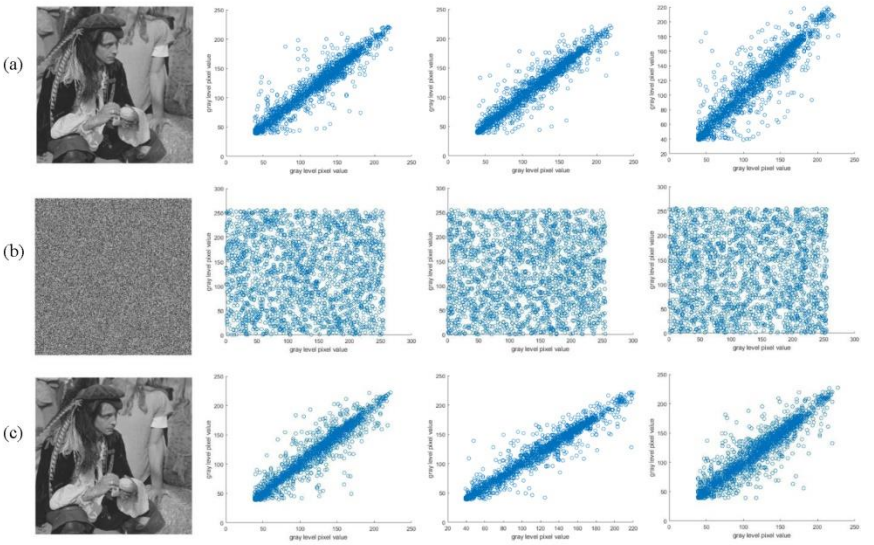


Figure 8
Plots of 2048 pairs of images:(a) original image; (b) encrypted image; (c) decrypted image

Correlation coefficient evaluation

The correlation coefficients for each pair in the three directions are calculated as follows[40] [41]:

$$r_{xy} = cov(x,y)/\sqrt{D(x)D(y)} \tag{8}$$
$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$$

[2] https://sipi.usc.edu/database/

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

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

where the grayscale values of the two neighboring image pixels are defined by x and $y, N\ is$ the total number of duplets (x, y) [42].

Table 4
Horizantal, vertical, and diagonal corr values of 2048 pairs of original, encrypted and decrypted images

Images		H	V	D
Man (512x512)	Plain image	0.963498	0.969473	0.944761
	Encrypted image	-0.012503	0.023462	0.008635
	Decrypted image	0.958535	0.974236	0.943747
Baboon (512x512)	Plain image	0.866207	0.763390	0.720104
	Encrypted image	-0.007877	0.001621	0.018385
	Decrypted image	0.862310	0.760213	0.728583
Tank (512x512)	Plain image	0.951603	0.928100	0.913960
	Encrypted image	0.009304	0.013036	-0.01144
	Decrypted image	0.942851	0.928486	0.890130
Lena (256x256)	Plain image	0.945365	0.973780	0.908599
	Encrypted image	0.014755	0.000263	0.015830
	Decrypted image	0.9320	0.973427	0.911418
Cameraman (256x256)	Plain image	0.9456160	0.952193	0.908356
	Encrypted image	0.0076287	-0.030052	-0.016688
	Decrypted image	0.9356073	0.9539645	0.9090709

The correlation coefficients between the neighboring pixels of the Man, Baboon, Tank, Lena, and Cameraman images were calculated, listed, and shown in *Figure 8* and *Table 4*. As result, correlations in the three directions are extremely low and almost zero in all the directions. This proves that the proposed cryptosystem is efficient against statistical attacks.

Information entropy (IE)

The entropy $H(S)$ of a source S defined by its P(S_i) and given by equation (11) [43] is calculated to determine the randomness of the pixel values in the image[44]. A high entropy score of 7.9+ means that the image is very random.

$$H(S) = -\sum_{i=0}^{2N-1} P(S_i)\, log_2(S_i) \quad (9)$$

The IE values are listed in Table 5 and remain tightly close to 8. Consequently, it is exceedingly difficult to extract visual information from ciphered images.

Table 5
Entropy score results

Entropy	SIZE	Original	Encrypted	Decrypted
Man	(512x512)	7.192588	7.999307	7.192588
Baboon	(512x512)	7.139098	7.999209	7.139098
Tank	(512x512)	5.495739	7.999293	5.495739
Lena	(256x256)	7.243229	7.997453	7.243229
Cameraman	(256x256)	7.009716	7.997207	7.0097162

A. *Differential attack*

In this test, Man, Baboon, Tank, Lena, and Cameraman are used as the original images for testing. We computed the values of the number of pixels change rate (NPCR) [45], unified average change in intensity (UACI) and the peak signal-to-noise ratio (PSNR)[46] via equations (12),(13) and (14) [47] to determine how a 1-bit change in the plain image would affect the associated cipher images[48].

$$NPCR = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{L} D(i,j,k)}{W.H.L} \times 100\% \tag{10}$$

$$D(i,j,k) = \begin{cases} 0 & C1(i,j,k) = C2(i,j,k) \\ 1 & C1(i,j,k) \neq C2(i,j,k) \end{cases}$$

where C1 and C2 are either plain or cipher images created using the original and pixel-modified keys, respectively[49]. The height, length, and width of the image are denoted by H, L, and W respectively.

$$UACI = \frac{1}{W.H.L}\left[\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{L}\frac{|C1(i,j,k)-C2(i,j,k)|}{2^{Q}-1}\right] \times 100\% \tag{11}$$

where Q is the number of bits corresponding to the red, green, and blue channels[49].

$$PSNR = 20\ log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \tag{12}$$

$$MSE = \frac{1}{W.H.L}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{L}[I(i,j,k) - K(i,j,k)]^2 \tag{13}$$

The MSE is the mean squared error. i, j, and k are the pixel positions [49].

The NPCR is shown to have a maximum predicted value of 99.61% when two random images are considered, while the UACI's maximum anticipated value is approximately 33% [50]. Table 6 shows that the values for the NPCR are close to the aforementioned predicted values, obtained UACI values are nearly equal to the defined critical values. Hence, the proposed

encryption technique guarantees necessary resistance to any type of differential cryptanalysis.

Table 6
NPCR, UACI, PSNR and MSE results

	NPCR	UACI	PSNR	MSE
Man	0.999596	0.2893528	44.210997	7.961727e+03
Baboon	0.999596	0.2715081	45.664880	6.795722e+03
Tank	0.996501	0.263013	46.412942	6.235254e+03
Lena	0.996566	0.278823	45.024871	7.296694e+03
Cameraman	0.973888	0.311146	42.601568	9.365429e+03

B. *Secret key sensitivity analysis*

To verify the scheme against the sensitivity of the key, a key is taken and an image is encrypted. The value of the key, or more specifically, one bit from the bitstream used for encryption is changed, and an image is encrypted again [51]. The same analysis was performed during the decryption process. The output images were compared to determine if there were any matches. To perform this analysis, the first key set (referred as K_1) is:

$$r_0 = 3.699, X_0 = 0.1, \text{ NLFSR} = 10101101.$$

K_1 was used to encrypt the original image Figure 9(a). The encrypted image is E_1 (Figure 9(b)).

To create a different key set designated as K_2, we apply a small change to $r_0 = 3.999000000000001$ while maintaining X_0 unchanged. The same manner with K_2 to obtain (E_2) in Figure 9(c).

The simulation results and the pixel-to-pixel difference $|E_1 - E_2|$ [52] clarify that the encrypted image changes significantly as a consequence of tiny changes in the original image and key.

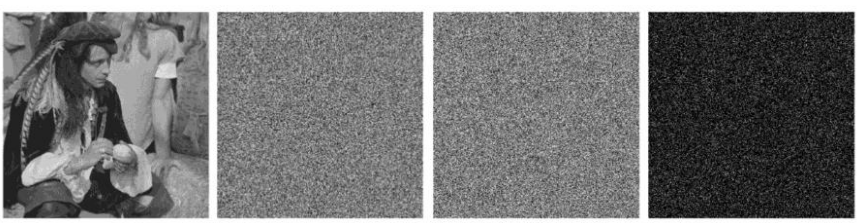


Figure 9
Key sensitivity analysis: (a) Original image;
(b) Encrypted image E1; (c) E2; (d)|E1-E2|

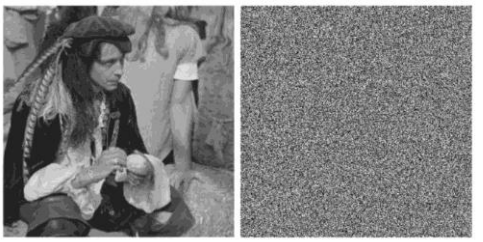


Figure 10
Key sensitivity analysis: (a) Decrypted with K1; (b) Decrypted with a wrong key (K2)

Figure 10(a) and (b) show the decryption results, where only the correct key (K1) allows for full reconstruction of the plain image. However, even a slight modification in the security key (K2) leads to image decryption failure (Figure 10(b)).

C. *Brute-force attack*

A brute-force attack is used to assess the security of the cipher[53]. Given that the actual number is represented in computers using double precision floating point numbers in the algorithm, where r is estimated by 10^{-15} according to IEEE-754 standards[54]. The key space used by the proposed method represents the sum of the keys utilized as a part of the key-generator system (see Table 7).

The total key-space is $\approx 255 \times 262 \times 10^3 \times 10^{15} \times 10^{15} \approx 10^{37}$ which is roughly equal to 2^{126} and greater than 2^{100}, which is sufficient to fend against brute force assaults.

Table 7
The initial conditions and parameters that form the secret key

Keys	Symbol	Definition
Key 1	NLFSR	The sequences generated by the NLFSR ($(2^8\text{-}1) \approx 255$)
Key 2	X_0	Initial condition of the LSS chaotic map
Key 3	r	Bifurcation parameter
Key 4	*n*	Iteration number (depend on the input image(512x512))

5. Comparison and discussion

The fundamental concept of the PRNG-LSS-NLFSR is to carefully arrange a number of 1D chaotic maps with the NLFSR inside the same algorithm to boost the security level compared to existing PRNGs such as [55] and [28] and to achieve the following properties in Table 8:

Table 8
Improved results on the proposed PRNG-LSS-NLFSR

	LSS	NLFSR	PRBG-LSS-NLFSR
Good randomness	×	-	×
Aperiodicity	×	-	×
Non-linearity	-	×	×
High speed	×	×	×
Ease of implementation	×	×	×

The LSS is the core aspect of our proposed generator; therefore, it has many more benefits than simply using a logistic map or a sine map. By construction, the PRNG-LSS-NLFSR generated sequence is aperiodic. This involves the aperiodic behavior of the described PRNG-LSS-NLFSR.

The NLFSR was used for the following reasons:

- To eliminate the linearity drawback of the LFSR and introduce the concept of nonlinearity.
- An NLFSR has the advantages of high speed and ease of implementation.
- NLFSRs are more immune to algebraic attacks and are more cryptographically secure than LFSRs.

Our sequence-based cryptosystem is cryptographically safe from several selected attacks. Table 9 compares the outcomes of the security tests conducted on the proposed PRNG-LSS-NLFSR with previously mentioned techniques to validate its use in the cryptography field.

Consequently, the correlation coefficient is almost zero, the NPCR and UACI values we obtained were satisfactory because they fit within the range of comparable research. The entropy values of encrypted images reflect highly visually distorted images. Moreover, the PRNG-LSS-NLFSR has a large keyspace with an additional ability to expand it to withstand brute-force and statistical attacks as shown in Table 10.

Table 9
Comparison of the proposed and existing image encryption schemes

Image	Refe-rences	Entropy		NPCR	UACI	Correlation		
		Plain-image	Encrypted image			H	V	D
Lena (512 x 512)	**Proposed**	7.44506	7.9993	99.9596	28.6517	-0.0165637	-0.0052927	0.0125289
	Ref [56]	-	7.9733	-	-	−0.00522	−0.012447	0.002597
	Ref [57]	7.4455	7.9994	99.6055	33.4325	−0.0001	−0.0004	−0.0008
	Ref [19]	-	7.9997	99.6159	33.4637	−0.00770	0.00437	−0.03169
Lena (256 x 256)	**Proposed**	7.24322	7.9974	99.656	27.882	0.01475	0.00026	0.01583
	Ref [58]	7.4442	7.9972	99.5850	33.4918	0.0007	0.0049	0.0030
	Ref [59]	-	7.9997	99.5979	29.6782	0.0105	0.0052	−0.0023
	Ref [60, p. 4]	-	7.9974	-	-	-0.0107	0.0014	-0.0066
Cameraman	**Proposed**	7.009716	7.997207	99.65667724	31.11466950	0.0076287	-0.030052	-0.016688
	Ref [58]	7.0097	7.9973	99.6231	33.7219	0.0035	0.0007	0.0025
	Ref [56]	-	7.996721	-	-	0.013029	−0.001695	−0.001437
	Ref [57]	7.0480	7.9993	-	-	-	-	-
Baboon (512 x 512)	**Proposed**	7.13909	7.99920	0.99959	0.2715081	-0.00787	0.00162	0.018385
	Ref[19]	-	7.99976	99.6056	33.4751	−0.01895	−0.00289	−0.00559

Table 10
Comparison of key space between the proposed and existing image encryption schemes

Study	Control parameter & initial condition	Precision	Key space
Proposed	$(X_0, r, NLFSR_0)$	10^{-15}	2^{126}
[15]	$(x_0, \mu_1, \mu_2, \alpha, \beta)$	64-bit double precision	$2^{62+4\times64} = 2^{318}$
[18]	(Z_0, c)	-	$3.4 \times 10^{38} \cong 2^{186}$
[14]	(x_0, a)	10^{-15}	2^{128}
[56]	(x_0, y_0, a, M, T)	10^{-14}	$0.4 \times 10^{5\times14} \approx 2^{231}$

6. Conclusion

A new PRNG-LSS-NLFSR structure consisting of an 8-bit Fibonacci NLFSR and coupled LSS was presented in this paper. One of the key reasons for using NLFSR and coupled LSS as a cryptosystem is to accomplish computational objectives by enhancing randomness and generating a non-periodic sequence. The proposed PRNG-LSS-NLFSR was successfully applied and tested for image encryption after passing fifteen rigorous NIST tests. It provides high cryptographic performance, fulfills statistical security requirements, and offers various intriguing characteristics such as sufficient key space and strong levels of security. Future work will consider the application of this technique to field-programmable gate arrays (FPGAs) to perform a corresponding search.

References

[1] V. Himthani, V. S. Dhaka, M. Kaur, and P. Hemrajani, "V-Net architecture based advanced visually meaningful image encryption technique," *J. Discrete Math. Sci. Cryptogr.,* vol. 25, no. 7, pp. 2183–2194 (Oct. 2022), doi: 10.1080/09720529.2022.2133255.

[2] R. Ge, G. Yang, J. Wu, Y. Chen, G. Coatrieux, and L. Luo, "A Novel Chaos-Based Symmetric Image Encryption Using Bit-Pair Level Process," *IEEE Access,* vol. 7, pp. 99470–99480 (2019), doi: 10.1109/ACCESS.2019.2927415.

[3] Z. Qiao, I. Taralova, and S. El Assad, "Efficient Pseudo-chaotic Number Generator for Cryptographic Applications," *Int. J. Intell. Comput. Res*., vol. 11, no. 1, pp. 1041–1048 (Dec. 2020), doi: 10.20533/ijicr.2042.4655.2020.0126.

[4] H.-C. Tang and C. S. Chen, "Spectral test of DX and DL multiple recursive random number generators," *J. Discrete Math. Sci. Cryptogr.*, vol. 21, no. 1, pp. 171–178 (Jan. 2018), doi: 10.1080/09720529.2017.1338602.

[5] M. K. Khairullah, A. A. Alkahtani, M. Z. Bin Baharuddin, and A. M. Al-Jubari, "Designing 1D Chaotic Maps for Fast Chaotic Image Encryption," *Electronics,* vol. 10, no. 17, Art. no. 17 (Jan. 2021), doi: 10.3390/electronics10172116.

[6] S. Kanamaru, Y. Shimada, K. Fujiwara, and T. Ikeguchi, "Performance evaluation of chaotic random numbers generated from responses of integer logistic maps," *Nonlinear Theory Its Appl. IEICE*, vol. 12, no. 3, pp. 489–499 (2021), doi: 10.1587/nolta.12.489.

[7] V. Kanth, T. Martinsen, and P. Stanica, "The Self-Shrinking Conflation Generator: A Proposed Improvement to the Self-Shrinking Generator," *Eur. J. Pure Appl. Math.*, vol. 15, no. 4, Art. no. 4 (Oct. 2022), doi: 10.29020/nybg.ejpam.v15i4.4504.

[8] J. Walczak and R. Stępień, "Discrete models of the NLFSR generators," Comput. Appl. Electr. Eng., vol. Vol. 9, 2011, Accessed: Jan. 15 (2023). [Online]. Available: http://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-58e76496-f009-450f-b2b1-d0121a67f8f7.

[9] A. A. Kuznetsov, O. V. Potii, N. A. Poluyanenko, Y. I. Gorbenko, and N. Kryvinska, "Research of Second-Order Properties of NLFSR. Comparative Analysis of M-NLFSR and M-LFSR," in Stream Ciphers in Modern Real-time IT Systems: Analysis, Design and Comparative Studies, A. A. Kuznetsov, O. V. Potii, N. A. Poluyanenko, Y. I. Gorbenko, and N. Kryvinska, Eds., in Studies in Systems, Decision and Control. , Cham: Springer International Publishing, pp. 419–465 (2022). doi: 10.1007/978-3-030-79770-6_15.

[10] M. Garland, S. Le Grand, J. Nickolls, J. Anderson, J. Hardwick, S. Morton, E. Phillips, Y. Zhang, and V. Volkov, "*Parallel computing experiences with CUDA*," IEEE Micro, vol. 28, no. 4, pp. 13–27 (Jul. 2008), doi: 10.1109/MM.2008.57.

[11] J. C. Cerda, C. D. Martinez, J. M. Comer, and D. H. K. Hoe, "An efficient FPGA random number generator using LFSRs and cellular automata," in 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 912–915 (Aug. 2012). doi: 10.1109/MWSCAS.2012.6292169.

[12] E. Dubrova, "A List of Maximum Period NLFSRs." 2012. Accessed: May 28 (2023). [Online]. Available: https://eprint.iacr.org/2012/166

[13] D. Zhao, H. Peng, L. Li, S. Hui, and Y. Yang, "Novel way to research nonlinear feedback shift register," *Sci. China Inf. Sci.*, vol. 57, no. 9, pp. 1–14 (Sep. 2014), doi: 10.1007/s11432-013-5058-4.

[14] M. A. Murillo-Escobar, C. Cruz-Hernández, L. Cardoza-Avendaño, and R. Méndez-Ramírez, "A novel pseudorandom number generator based on pseudorandomly enhanced logistic map," Nonlinear Dyn., vol. 1, no. 87, pp. 407–425 (Sep. 2016), doi: 10.1007/s11071-016-3051-3.

[15] M. Sharma, R. K. Ranjan, and V. Bharti, "A pseudo-random bit generator based on chaotic maps enhanced with a bit-XOR operation," *J. Inf. Secur. Appl.*, vol. 69, no. C (Sep. 2022), doi: 10.1016/j.jisa.2022.103299.

[16] "A Modified Discretized Chaotic Map and Its Generated Pseudo Binary Random Number," *Mater. Today Proc.*, vol. 65, pp. 3806–3813 (Jan. 2022), doi: 10.1016/j.matpr.2022.06.577.

[17] M. Hemattil, A. Ahmadi, S. V. Makkil, and M. Ahmadi, "Hardware Design of Chaotic Pseudo-Random Number Generator Based on Nonlinear Feedback Shift Register," 2018 IEEE 61st Int. Midwest Symp. Circuits Syst. MWSCAS, pp. 980–983 (Aug. 2018), doi: 10.1109/MWSCAS.2018.8624001.

[18] H. S. Alhadawi, M. F. Zolkipli, S. M. Ismail, and D. Lambić, "Designing a pseudorandom bit generator based on LFSRs and a discrete chaotic map," Cryptologia, vol. 43, no. 3, pp. 190–211 (May 2019), doi: 10.1080/01611194.2018.1548390.

[19] F. Dridi, S. El Assad, W. El Hadj Youssef, and M. Machhout, "Design, Hardware Implementation on FPGA and Performance Analysis of Three Chaos-Based Stream Ciphers," *Fractal Fract.*, vol. 7, no. 2, Art. no. 2 (Feb. 2023), doi: 10.3390/fractalfract7020197.

[20] H. Bourekouche, S. Belkacem, and N. Messaoudi, "Efficient image encryption scheme using a nonlinear shift register and chaos," *Journal of Information and Optimization Sciences,* vol. 45, no. 1, pp. 157–180 (2024).

[21] G. Yao and U. Parampalli, "Improved transformation algorithms for generalized Galois NLFSRs," *Cryptogr. Commun.*, vol. 14, no. 2, pp. 229–258 (Mar. 2022), doi: 10.1007/s12095-021-00500-3.

[22] P. K. Kumar and B. Mondal, "Lightweight Stream Cipher for Health Care IoT," in 2023 IEEE 2nd International Conference on Industrial Electronics: Developments & Applications (ICIDeA), pp. 444–449 (Sep. 2023). doi: 10.1109/ICIDeA59866.2023.10295196.

[23] G. Yao, "Transformation and Security Analysis of NLFSR-based Stream Ciphers," 2020, Accessed: Sep. 27 (2023). [Online]. Available: http://hdl.handle.net/11343/267826.

[24] S. Deb, B. Biswas, and N. Kar, "Study of NLFSR and Reasonable Security Improvement on Trivium Cipher," in Information Systems Design and Intelligent Applications, J. K. Mandal, S. C. Satapathy, M. Kumar Sanyal, P. P. Sarkar, and A. Mukhopadhyay, Eds., in Advances in Intelligent Systems and Computing. New Delhi: Springer India, pp. 731–739 (2015). doi: 10.1007/978-81-322-2250-7_73.

[25] H. Bourekouche, Contribution to chaotic encryption methods for digital data, Ph.D. dissertation, Ministry of Higher Education (2024).

[26] I. Al-Hejri and S. Almuhammadi, "Constructing New NLFSR Functions with Optimal Periods," *Int. J. Interdiscip. Telecommun. Netw. IJITN,* vol. 12, no. 2, pp. 71–80 (2020), doi: 10.4018/IJITN.2020040106.

[27] T. Hu, Y. Liu, L.-H. Gong, S.-F. Guo, and H.-M. Yuan, "Chaotic image cryptosystem using DNA deletion and DNA insertion," *Signal Process.,* vol. 134, pp. 234–243 (May 2017), doi: 10.1016/j.sigpro.2016.12.008.

[28] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.,* vol. 97, pp. 172–182 (Apr. 2014), doi: 10.1016/j.sigpro.2013.10.034.

[29] V. M. Padmapriya, B. Sowmya, M. Sumanjali, and A. Jayapalan, "Chaotic Encryption based secure Transmission," in 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), pp. 1–5 (Mar. 2019). doi: 10.1109/ViTECoN.2019.8899588.

[30] Joan. S. Muthu, A. J. Paul, and P. Murali, "An Efficient Analyses of the Behavior of One Dimensional Chaotic Maps using 0–1 Test and Three State Test," in 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), pp. 125–130 (Dec. 2020). doi: 10.1109/RAICS51191.2020.9332470.

[31] C. Pak, K. An, P. Jang, J. Kim, and S. Kim, "A novel bit-level color image encryption using improved 1D chaotic map," Multimed. Tools Appl., vol. 78, no. 9, pp. 12027–12042 (May 2019), doi: 10.1007/s11042-018-6739-1.

[32] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, N. Heckert, J. Dray, and S. Vo, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," NIST, Sep. 2010, Accessed: Feb. 07 (2023). [Online]. Available: https://www.nist.gov/publications/statistical-test-suite-random-and-pseudorandom-number-generators-cryptographic

[33] M. ParsiMehr, K. Shayesteh, and K. Godini, "The modeling and prediction of the quality of the groundwater resources in Tuyserkan plain using the optimized artificial neural network," *J. Adv. Environ. Health Res.,* vol. 8, no. 2, pp. 100–110 (Apr. 2020), doi: 10.22102/jaehr.2020.210891.1150.

[34] M. M. Al-Mhadawi, E. A. Albahrani, and S. H. Lafta, "Efficient and secure chaotic PRNG for color image encryption," *Microprocess. Microsyst.,* vol. 101, pp. 104911 (Sep. 2023), doi: 10.1016/j.micpro.2023.104911.

[35] M. M. Al-Mhadawi and A. A. Albahrani, "Hybrid Method as Pseudo-Random Bits Generator," in 2019 First International Conference of Computer and Applied Sciences (CAS), pp. 250–255 (Dec. 2019). doi: 10.1109/CAS47993.2019.9075715.

[36] M. Ťažký, L. Bodnárová, L. Ťažká, R. Hela, M. Meruňka, and P. Hlaváček, "The Effect of the Composition of a Concrete Mixture on Its Volume Changes," *Materials,* vol. 14, no. 4, Art. no. 4 (Jan. 2021), doi: 10.3390/ma14040828.

[37] J. Arif et al., "A Novel Chaotic Permutation-Substitution Image Encryption Scheme Based on Logistic Map and Random Substitution," *IEEE Access*, vol. 10, pp. 12966–12982 (2022), doi: 10.1109/ACCESS.2022.3146792.

[38] H. T. Elshoush, B. M. Al-Tayeb, and K. T. Obeid, "Enhanced Serpent algorithm using Lorenz 96 Chaos-based block key generation and parallel computing for RGB image encryption," *PeerJ Comput. Sci.*, vol. 7, pp. e812 (Dec. 2021), doi: 10.7717/peerj-cs.812.

[39] B. Ge, X. Chen, G. Chen, and Z. Shen, "Secure and Fast Image Encryption Algorithm Using Hyper-Chaos-Based Key Generator and Vector Operation," *IEEE Access*, vol. 9, pp. 137635–137654 (2021), doi: 10.1109/ACCESS.2021.3118377.

[40] M. Singh, N. Baranwal, K. N. Singh, and A. K. Singh, "Using GAN-Based Encryption to Secure Digital Images with Reconstruction through Customized Super Resolution Network," *IEEE Trans. Consum. Electron.*, pp. 1–1 (2023), doi: 10.1109/TCE.2023.3285626.

[41] S. Vishwakarma and S. Qureshi, "Secure Transmission of Video using (2,2) Visual Cryptography Scheme and Share Encryption using Logistic Chaos Method," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.,* vol. 3, no. 1, pp. 1502–1514 (Feb. 2018), doi: 10.32628/CSEIT1831357.

[42] B. Norouzi, S. Mirzakuchaki, S. M. Seyedzadeh, and M. R. Mosavi, "A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process," *Multimed. Tools Appl.,* vol. 71, no. 3, pp. 1469–1497 (Aug. 2014), doi: 10.1007/s11042-012-1292-9.

[43] A. Souyah and K. M. Faraoun, "Fast and efficient randomized encryption scheme for digital images based on Quadtree decomposition and reversible memory cellular automata," *Nonlinear Dyn.*, vol. 84, no. 2, pp. 715–732 (Apr. 2016), doi: 10.1007/s11071-015-2521-3.

[44] Z. Man, J. Li, X. Di, and O. Bai, "An Image Segmentation Encryption Algorithm Based on Hybrid Chaotic System," *IEEE Access,* vol. 7, pp. 103047–103058 (2019), doi: 10.1109/ACCESS.2019.2931732.

[45] H. Bourekouche, S. Belkacem, and N. Messaoudi, "Lightweight medical image encrypting and decrypting algorithm based on the 3D intertwining logistic map," *International Journal of Informatics and Applied Mathematics*, vol. 6, no. 2, pp. 46–62 (2024).

[46] P. Ramasamy, V. Ranganathan, S. Kadry, R. Damaševičius, and T. Blažauskas, "An Image Encryption Scheme Based on Block Scrambling, Modified Zigzag Transformation and Key Generation Using Enhanced Logistic—Tent Map," *Entropy*, vol. 21, no. 7, Art. no. 7 (Jul. 2019), doi: 10.3390/e21070656.

[47] M. Kumari and S. Gupta, "Performance comparison between Chaos and quantum-chaos based image encryption techniques," *Multimed. Tools Appl.,* vol. 80, no. 24, pp. 33213–33255 (Oct. 2021), doi: 10.1007/s11042-021-11178-3.

[48] X. Chai, "An image encryption algorithm based on bit level Brownian motion and new chaotic systems," *Multimed. Tools Appl.*, vol. 76, no. 1, pp. 1159–1175 (Jan. 2017), doi: 10.1007/s11042-015-3088-1.

[49] M. Kumari and S. Gupta, "Performance comparison between Chaos and quantum-chaos based image encryption techniques," *Multimed. Tools Appl.,* vol. 80, no. 24, pp. 33213–33255 (2021), doi: 10.1007/s11042-021-11178-3.

[50] L. Liu, Y. Zhang, and H. Zhang, "A color image encryption algorithm based on DNA computation and Chen system," *J. Phys. Conf. Ser.,* vol. 1074, no. 1, pp. 012096 (Sep. 2018), doi: 10.1088/1742-6596/1074/1/012096.

[51] Bourekouche H, Belkacem S, Messaoudi N. Impact of Confusion-Diffusion Complexity on Maintaining Data Security. In2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC) 2024 May 12 (pp. 1-6). IEEE.

[52] S. Cai, L. Huang, X. Chen, and X. Xiong, "A Symmetric Plaintext-Related Color Image Encryption System Based on Bit Permutation," *Entropy*, vol. 20, no. 4, Art. no. 4 (Apr. 2018), doi: 10.3390/e20040282.

[53] P. Li, Z. Li, W. A. Halang, and G. Chen, "A multiple pseudorandom-bit generator based on a spatiotemporal chaotic map," *Phys. Lett. A,* vol. 349, no. 6, pp. 467–473 (Jan. 2006), doi: 10.1016/j.physleta.2005.09.060.

[54] P. Li, Z. Li, W. A. Halang, and G. Chen, "A multiple pseudorandom-bit generator based on a spatiotemporal chaotic map," *Phys. Lett. A,* vol. 349, no. 6, pp. 467–473 (Jan. 2006), doi: 10.1016/j.physleta.2005.09.060.

[55] V. Patidar, K. K. Sud, and N. K. Pareek, "A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing," *Informatica,* vol. 33, no. 4, Art. no. 4 (2009), Accessed: Mar. 05, 2023. [Online]. Available: https://www.informatica.si/index.php/informatica/article/view/261.

[56] M. Elkandoz, W. Alexan, and H. Hussein, "Logistic Sine Map Based Image Encryption," (Sep. 2019). doi: 10.23919/SPA.2019.8936718.

[57] D. A. Trujillo-Toledo et al., "Real-time RGB image encryption for IoT applications using enhanced sequences from chaotic maps," *Chaos Solitons Fractals,* vol. 153, pp. 111506 (Dec. 2021), doi: 10.1016/j.chaos.2021.111506.

[58] A. H. Brahim, A. A. Pacha, and N. H. Said, "A new image encryption scheme based on a hyperchaotic system & multi specific S-boxes," *Inf. Secur. J. Glob. Perspect.*, vol. 32, no. 2, pp. 59–75 (Mar. 2023), doi: 10.1080/19393555.2021.1943572.

[59] K. Kumar, S. Roy, U. Rawat, and S. Malhotra, "IEHC: An efficient image encryption technique using hybrid chaotic map," *Chaos Solitons Fractals,* vol. 158, pp. 111994 (May 2022), doi: 10.1016/j.chaos.2022.111994.

[60] L.-H. Gong, H.-X. Luo, R.-Q. Wu, and N.-R. Zhou, "New 4D chaotic system with hidden attractors and self-excited attractors and its application in image encryption based on RNG," *Phys. Stat. Mech. Its Appl.*, vol. 591, pp. 126793 (Apr. 2022), doi: 10.1016/j.physa.2021.126793.

Received March, 2023