Efficient Pseudo-chaotic Number Generator for Cryptographic Applications

Zongchao Qiao[1], Ina Taralova[1] and Safwan El Assad[2]

Laboratoire des Sciences du Numérique de Nantes[1]
LS2N, UMR CNRS 6004
Ecole Centrale de Nantes
Nantes, France

Institut d'Electronique et des Télécommunications de Rennes[2]
IETR, Université de Nantes/Polytech Nantes
Nantes, France

Abstract

A secure and efficient pseudo-random number generator is crucial for the security of a stream cipher or block cipher in a cryptosystem. This paper is devoted to the design and analysis of a new robust Pseudo-Chaotic Number Generator (PCNG) based on discrete chaotic maps over a finite field, and a multiplexing mechanism. The proposed PCNG contains three discrete chaotic maps: Piece-Wise Linear Chaotic Map (PWLCM), skew tent and logistic map. XOR operators are applied on these maps to form three optional intermediate outputs that are selected by the multiplexing mechanism to produce the final pseudo-chaotic sequence. The XOR operators and multiplexing mechanism can increase the dynamic pseudo-chaotic properties significantly. The PCNG is designed over a finite 2^{32} field, has a simple structure and can be easily implemented. It has been implemented on a new stream cipher. Security analyses and statistical experiments have been carried out and the results have proven that the proposed PCNG can generate effective pseudo-random numbers and the stream cipher exhibits good security properties with large secret key space and high key sensitivity. Therefore, the proposed PCNG can be successfully applied in cryptosystems.

1. Introduction

With the rapid development of new internet technology and communication networks, huge amounts of various digital data, for instance text message, image, audio signal and video with confidential information are exchanged via insecure network channels. Therefore, a good cryptosystem is required, including stream cipher and block cipher, etc., that can resist attacks to ensure the information security.

The cryptographic keys are crucial in cryptosystems. According to Kerckhoffs' principle, the security of a cryptosystem should depend only on its key [1]. Thus, the key must exhibit random properties and can control the generation of random numbers which are used to be the dynamic keys for the encryption process [2]. Compared to the True Random Number Generators (TRNGs) that come from the natural phenomena, the Pseudo-Random Number Generators (PRNGs) are easier to generate and reproduce pseudo-random numbers under control [3]. Over the past decades, chaos-based PRNGs, namely Pseudo-Chaotic Number Generators (PCNGs), have been verified to be efficient in producing pseudo-random numbers due to their excellent properties, such as, deterministic character, non-periodicity, high sensitivity to initial conditions and parameters, etc. [4]. If the initial conditions and the parameters of the PCNGs are considered to be the secret keys, several secret keys are able to generate abundant numbers of highly secure key-dependent pseudo-random values that will be used as dynamic keys in the cryptosystem. This latter point can not only greatly save storage memory but also boost the cryptosystem's secret key sensitivity.

In general, low dimensional chaotic maps have the advantages of simple structure and easy implementation, but when they are discretized in finite precision, they may lead to low periodicity with dynamical degradation. Indeed, most classical chaotic maps, defined over real numbers, such as Logistic, Tent, Henon, Chen etc., are not secure for encryption purposes [5], due to the lack of floating point numbers calculation accuracy and round-off errors, which appear when the above maps are numerically implemented. To solve these problems and improve the security, some efficient methods for PCNGs have been recently proposed.

A chaotic coupling method with topology network was explored to design two robust PCNGs based on tent and logistic maps [6]. R. Lozi brought out a weak coupling approach to hide the chaotic functions and

enhance the chaotic generators [7]. R. Hamza proposed a secure PCNG based on a combination of the three coordinates of the Chen chaotic orbits and it adopted the approach of cascading and mixing the orbit samples to overcome the degradation problem during the finite precision computations [8]. A new form of the power-exponential structure was presented in [9], and it can integrate 1-Dimension chaotic maps to achieve good chaotic behavior. In [10], the authors implemented a robust PCNG for a block cipher by connecting skew tent map and PWLCM map in parallel, and in this PCNG, a Linear Feedback Shift Register (LFSR) was designed to ensure a very large periodicity for all generated sequences. The authors of [11] introduced two stream ciphers based on two PCNGs which used a predefined week coupling matrix and a binary diffusion matrix respectively. A generalized fractional order logistic map was proposed in [12] to design a chaotic generator for cryptosystem. Reference [13] adopted chaotic coupling and multiplexing technique to achieve good PCNGs.

Inspired by idea of the multiplexing technique, in our previous paper [14], we proposed a simple and efficient PCNG working over a finite 2^{32} field based on three discrete chaotic maps, that is, PWLCM, skew tent map and logistic map. Only four XOR operators have been adopted in this scheme to mix these 1st-order maps to form the intermediate chaotic outputs as well as the decision samples, which operate cooperatively under the control of multiplexing mechanism to generate the final pseudo-chaotic sequence for the encryption.

To evaluate and verify the cryptographic property of this PCNG when implemented into encryption application, in this paper, a stream cipher based on the secure PCNG has been introduced. The PCNG is used to provide the dynamic key stream for the stream cipher. The security tests on the PCNG and the stream cipher are carried out in the following sections.

The paper is organized as follows: the proposed PCNG is described in Section 2. Section 3 analyzes the performance of the PCNG in terms of computational performance and statistical analysis. We present the cryptanalytic analysis results of the stream cipher based on the proposed PCNG in Section 4, including key space analysis, uniformity test, entropy test, correlation analysis and key sensitivity analysis. Finally, we draw the conclusion in Section 5.

2. Proposed pseudo-chaotic number generator (PCNG)

The scheme of the proposed PCNG is shown in Figure 1. It operates in a fixed finite precision of N=32 bits based on the operations of XOR (denoted by $\oplus$) and multiplexing mechanism technique on three classical chaotic maps: *Fp[Xp(n-1)]*, *Fs[Xs(n-1)]* and *Fl[Xl(n-1)]* namely the discrete functions of PWCLM, skew tent map and logistic map. All the maps are chaotic but exhibit poor dynamic properties for encryption purposes when taken alone. They are defined by Equation (1), (2) and (3) respectively.

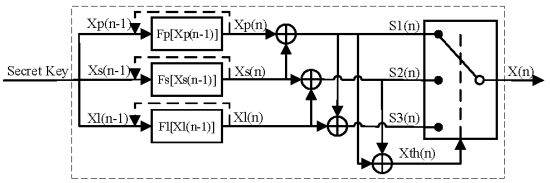


Figure 1. Proposed PCNG

$$Xp(n) = Fp\left[Xp(n-1)\right]$$
$$= \begin{cases} \left\lfloor 2^N \times \dfrac{Xp(n-1)}{Pp} \right\rfloor & if \;\; 0 < Xp(n-1) \le Pp \\ \left\lfloor 2^N \times \dfrac{Xp(n-1) - Pp}{2^{N-1} - Pp} \right\rfloor & if \;\; Pp < Xp(n-1) \le 2^{N-1} \\ \left\lfloor 2^N \times \dfrac{2^N - Pp - Xp(n-1)}{2^{N-1} - Pp} \right\rfloor & if \;\; 2^{N-1} < Xp(n-1) \le 2^N - Pp \\ \left\lfloor 2^N \times \dfrac{2^N - Xp(n-1)}{Pp} \right\rfloor & if \;\; 2^N - Pp < Xp(n-1) \le 2^N - 1 \\ 2^N - 1 - Pp & otherwise \end{cases} \quad (1)$$

$$Xs(n) = Fs[Xs(n-1)]$$
$$= \begin{cases} \left\lfloor \dfrac{2^N \times Xs(n-1)}{Ps} \right\rfloor & if \;\; 0 < Xs(n\text{-}1) < Ps \\ \left\lfloor \dfrac{2^N \times (2^N - Xs(n-1))}{2^N - Ps} \right\rfloor & if \;\; Ps < Xs(n\text{-}1) < 2^N \\ 2^N - 1 & otherwise \end{cases} \quad (2)$$

$$Xl(n) = Fl\left[Xl(n-1)\right]$$
$$= \begin{cases} \left\lfloor \dfrac{Xl(n-1) \times \left[2^N - Xl(n-1)\right]}{2^{N-2}} \right\rfloor & if \;\; Xl(n-1) \ne \dfrac{3}{4} \times 2^N, 2^N \\ 2^N - 1 & otherwise \end{cases} \quad (3)$$

where *Pp* and *Ps* are the parameters for PWLCM and skew tent map which are in the range of [1, 2^{N-1}-1] and [1, 2^N-1] respectively; the initial condition of these maps are *Xp(0), Xs(0), Xl(0)* which are all in the range of [1, 2^N-1]. *S1(n), S2(n), S3(n)* are three intermediate outputs that come from the following expressions (4):

$$\begin{aligned} S1(n) &= Xp(n) \oplus Xs(n) \\ S2(n) &= Xs(n) \oplus Xl(n) \\ S3(n) &= Xl(n) \oplus S1(n) = Xp(n) \oplus Xs(n) \oplus Xl(n) \end{aligned} \quad (4)$$

The usage of the convertible XOR operators can improve the chaotic characteristics effectively in comparison with the individual chaotic maps. The final output *X(n)* is controlled by the nonlinear multiplexing mechanism that is designed to increase the scheme complexity and enhance the randomness. The multiplexing mechanism includes a decision sample

Xth(n) with two thresholds *Th1* and *Th2*, where *Xth(n)* can be considered as a dynamic parameter to switch between *S1(n), S2(n), S3(n)*:

$$X(n)=\begin{cases} S1(n) \ \ if \ \ 0 < Xth(n) < Th1 \\ S2(n) \ if \ \ Th1 \le Xth(n) < Th2 \\ S3(n) \ otherwise \end{cases} \quad (5)$$

where $Xth(n) = S1(n) \oplus S2(n) = Xp(n) \oplus Xl(n)$ and $Th1 = 0.8 \times 2^N$, $Th2 = 0.9 \times 2^N$.

3. Performance analysis of PCNG

In this section, we discuss the performance of the proposed PCNG in terms of computational performance and statistical analysis. Histogram, Chi-square test and NIST test have been adopted to analyze the statistical properties. All these tests are used to explore and verify the cryptographic and randomness performances of the proposed PCNG.

All simulations are conducted in MATLAB (R2017b) on a computer of Intel Core i-7-3770 CPU in Windows 7 Professional, 64-bit operating system with 3.4GHz processor, 8 GB RAM.

3.1. Computational performance

We give below the results in terms of average *Bit Rate* (Mbps) and average *NCpB* (Number of needed Cycles to generate one Byte). For that, we generate *100* chaotic sequences with length of *31250* samples in each sequence using *100* different secret keys. Then the average generation time of these *100* sequences is calculated. The *Bit Rate* and *NCpB* are given by the following relations:

$$Bit\ Rate(Mbps) = \frac{Generated\ data\ size(Mbits)}{Average\ generation\ time(s)} \quad (6)$$

$$NCpB = \frac{CPU\ speed(Hz)}{Bit\ Rate(Byte/s)} \quad (7)$$

The obtained results are shown in Table 1:

Table 1. Time consuming results

Bit Rate(Mbps)	NCpB
17.679	1539

3.2. Statistical analysis

The PCNG is responsible for providing key stream in a block cipher or stream cipher. The key stream must exhibit randomness property to ensure that the attackers cannot find the rule of the key stream and never be able to derive the secret key. Thus, firstly, the basic rule of PCNG in statistical analysis is that the generated chaotic sequences should have a uniform distribution. And, then the chaotic sequences should pass the randomness test.

3.2.1. Histogram and Chi-square test. We can see the generated chaotic sequence is uniformly distributed by its histogram in Figure 2, where 10^7 chaotic samples *X* have been plotted in *1000* statistical classes and the red line in the figure shows the average values in every *10* classes (an interval).

When we zoom in a part of Figure 2, for instance, the range of [3×10^9 , 3.2×10^9], and plot its histogram in Figure 3, we can see that the partial histogram is qualitatively similar to the whole histogram of Figure 2. Note that, in Figure 3, to reveal the partial histogram more clearly, we divide each class of Figure 2 into *10* classes, which is equivalent to the original sequence in Figure 2 being drawn in the histogram of *10000* classes, and then a part of it has been magnified.

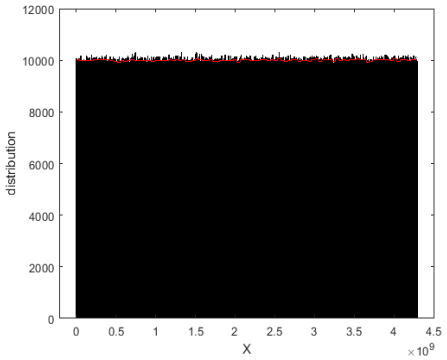


Figure 2. Histogram of the chaotic sequence X and average value per interval (in red)

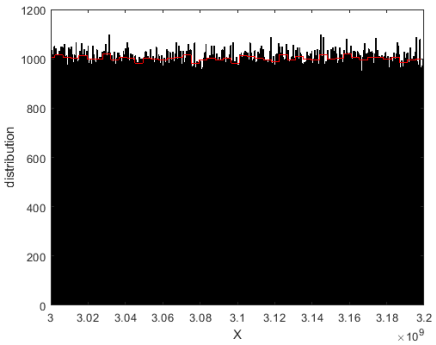


Figure 3. Histogram of a part of chaotic sequence X and average value per interval (in red)

Furthermore, to assert its uniformity more precisely, the Chi-square test is used and its experimental value is calculated as below:

$$\chi^2_{\exp} = \sum_{i=0}^{N_c-1} \frac{(O_i - E_i)^2}{E_i} \quad (8)$$

where N_c is the number of classes chosen (here N_c = *1000*), O_i is the number of observed samples in the *i-th* class and E_i is the expected number of samples in a uniform distribution. Here, we generate *3125000* chaotic samples, hence E_i=*3125000/1000*. The experimental value χ^2_{exp} of Chi-square equals to *961.0874* which is smaller than the theoretical value

$\chi^2_{th}(N_c - 1, \alpha)$ =*1073.6427* obtained for a threshold α = *0.05* of Chi-square distribution confirming the uniformity of the generated chaotic sequence.

3.2.2. Uniformity test in binary level. In addition, the calculation of the number of bits *0* and *1* in the binary conversion of the output sequence *X* provides another perspective to analyze its randomness. Convert the sequence *X* from decimalism to binary elements first, then separate it into *100* bit streams. So, each bit stream contains $3125000 \times 32/100 = 10^6$ bits. After that, we calculate the number of *0* and *1* in each bit stream. The result shown in Figure 4 presents that the proportion of bit *0* and bit *1* are symmetrically distributed around the optimal value *50%*. Meanwhile, the mean value of *100* proportions of bit *0* and bit *1* among all bitstreams are respectively *49.993%* and *50.007%*.

Hence, the proposed PCNG can pass the statistical analysis not only in decimal level, but also in binary level. It shows highly similar properties with random numbers.

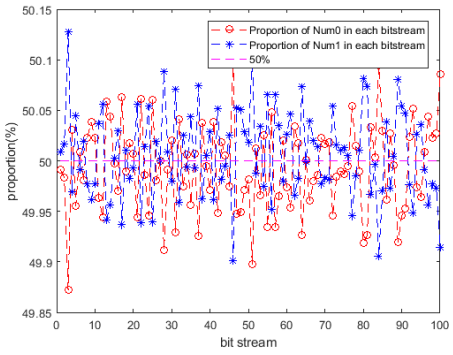


Figure 4. Proportion of bit '0' and '1' in binary bit stream

3.2.3. NIST test. NIST (National Institute of Standard and Technology) test is a suite of tests which are the widely used tool to measure the sequences for randomness [15]. In the NIST results report, *P-value* larger than β = *0.01* means that the sequence would be random with a confidence of *(1- β) =99%* [16]. Here, we apply the NIST test on the produced sequence (3125000×32 *bits* = 100×10^6 *bits)*. The results in Table 2 show that the chaotic sequence has passed the NIST test successfully.

4. Security analysis of a stream cipher

The proposed PCNG is implemented in a design of a robust stream cipher. This stream cipher is achieved by using XOR operation to mask the plaintext *128* bits by *128* bits with the key stream provided by the PCNG in Cipher Block Chaining (CBC) mode. This section analyzes the security performances of the stream cipher.

Table 2. P-value and proportion results of NIST test

Test	*P-value*	Proportion (%)
frequency	0.936	100.000
Block-frequency	0.817	99.000
Cumulative-sums	0.117	99.500
Runs	0.350	99.000
Longest-run	0.163	97.000
Rank	0.475	100.000
FFT	0.554	99.000
Non-overlapping template	0.511	98.845
overlapping template	0.637	99.000
universal	0.335	98.000
approximate entropy	0.063	99.000
random-excursions	0.411	98.790
random-excursions-variant	0.371	99.283
serial	0.232	99.000
linear-complexity	0.740	100.000

4.1. Key space analysis

A large secret key space of a cryptosystem is necessary to resist the brute-force attack and it is considered to be secure if the key space is equal or greater to 2^{128} [17].

The secret key of this stream cipher depends on the input values of the proposed PCNG which contains first the initial conditions for the three chaotic maps: *Xp(0), Xs(0), Xl(0)* and then the control parameters *Pp* and *Ps* for PWLCM and skew tent map. Thus, the key size is :

$$|K| = |Xp(0)| + |Xs(0)| + |Xl(0)| + |Pp| + |Ps| = 159 bits$$

where $|Xp(0)| = |Xs(0)| = |Xl(0)| = |Ps| = 32 bits$, and $|Pp| = 31 bits$.

Therefore, the key space of the proposed stream cipher is 2^{159}, which is large enough to resist a brute-force attack.

4.2. Histogram and Chi-square test

We tested 5 images with different sizes and features. The ciphered image should be uniformly distributed to resist the statistical attack. We analyze the distribution of plain and ciphered image of "Lenna"

and "Baboon" in Figure 5. Figure 5 (b), (f) show the histograms in RGB plan of plain images of Figure 5 (a), (e). Their ciphered images Figure 5(c), (g) are uniformly distributed in every plan, which are shown in Figure 5 (d), (h).

In addition, the Chi-square test is applied by Equation (8) but with different parameters: $N_c=256$, $E_i= ImageSize/N_c$, $\alpha=0.05$, which give the theoretical value $\chi^2_{th}(255,0.05) = 293.2478$. For each image, *100* different secret keys have been applied to repeat this test. Table 3 shows the average experimental Chi-square test results χ^2_{exp}, which confirms the uniformity of the ciphered images.

4.3. Entropy test

The information entropy is used to evaluate uncertainty and randomness properties in a message. The image pixel values are in the range of [0, 255]. In a robust cipher algorithm, the occurrence probability of any pixel value should be the same or almost the same. The random behavior of the ciphered image can be evaluated using the information entropy given by:

$$H(C) = \sum_{i=0}^{Q-1} Pro(c_i) \times \log_2 \frac{1}{Pro(c_i)} \quad (9)$$

where *H(C)* is the entropy of the ciphered image *C*, $Pro(c_i)$ is the occurrence number of c_i in each level *(i=0,1,2…255)*, and $Q=256=2^8$ is the number of levels.

Therefore, ideally, each level should have equal occurrence probability $Pro(c_i) = \frac{1}{Q} = 2^{-8}$. In this case, the information entropy is maximal:

$$H(C) = \sum_{i=0}^{255} 2^{-8} \times log_2 256 = 8.$$

We have calculated the information entropy for each plain image (*H(P)*) and the average entropy for the ciphered image (*H(C)*) over *100* entropy results accomplished by *100* secret keys. From the obtained results shown in Table 3, we remark that all average information entropy of the ciphered images is close to the above mentioned ideal value.

Table 3. Results of the Chi-square and entropy test

Image	χ^2_{exp}	Entropy: H(P)	Entropy: H(C)
airfield	254.1341	7.1206	7.9993
baboon	253.0482	7.7073	7.9991
boat	257.3041	7.1914	7.9993
lenna	251.4138	5.6822	7.9998
pepper	252.2190	7.6698	7.9998

4.4. Correlation analysis

Image has an intrinsic feature that is the high correlation between pixels. A secure cryptosystem should break this relationship. To test the correlation between two adjacent pixel, 8000 pairs of adjacent pixels have been selected randomly in horizontal (Hor-D), vertical (Ver-D) and diagonal (Dia-D) directions respectively from the plain image and its corresponding ciphered image. Then the correlation coefficient (ρ_{xy}) of each pair is calculated by Equation (10).

$$\rho_{xy} = \frac{\sum_{i=1}^{N_p}\left[\left(x_i-\bar{x}\right)\left(y_i-\bar{y}\right)\right]}{\sqrt{\sum_{i=1}^{N_p}\left(x_i-\bar{x}\right)^2}\sqrt{\sum_{i=1}^{N_p}\left(y_i-\bar{y}\right)^2}} \quad (10)$$

where $N_p = 8000$ is the number of the randomly selected pairs of adjacent pixels; x_i, y_i are pixel values of *i-th* pair, and $\bar{x}, \bar{y}$ are the mathematical expectations.

For each image, 100 ciphered images have been encrypted by 100 different secret keys, and then the average correlation coefficients have been calculated for plain and ciphered images. Table 4 shows the results obtained in each direction Hor-D, Ver-D, Dia-D. Figure 6 gives the correlation of image "pepper" in the three directions for the plain and ciphered images separately. Table 4 and Figure 6 reveal that the adjacent pixels are highly correlated to each other in the plain image and the stream cipher can break this correlation effectively.

4.5. Key sensitivity analysis

A robust stream cipher should have high sensitivity to the secret key. This property can be measured by calculating the Hamming distance (HD) (Equation (11)) between two ciphered images C_1 and C_2 which have been encrypted from the same plain image but their secret keys are only one bit difference with each other.

$$HD(C_1, C_2) = \frac{1}{|lb|} \times \sum_{k=1}^{|lb|} \left(C_1[k] \oplus C_2[k]\right) \quad (11)$$

where $|lb|$ is the bit length of the image under processing.

For each test image, C_1, C_2 are encrypted with different LSB (Least Significant Bit) in their secret key. *100* different secret keys are used to repeat this experiment and the average HD over 100 HDs are shown in Table 5. As we can see, the obtained HDs are close to the optimal value 50% indicating that the probability of bit changes between each pairs of ciphered images is 50%.

We also adopted two common methods to measure the cryptosystem's sensitivity to the secret key: the Number of Pixels Change Rate (NPCR) and the

Unified Average Changing Intensity (UACI), which are defined as Equation (12), (13).

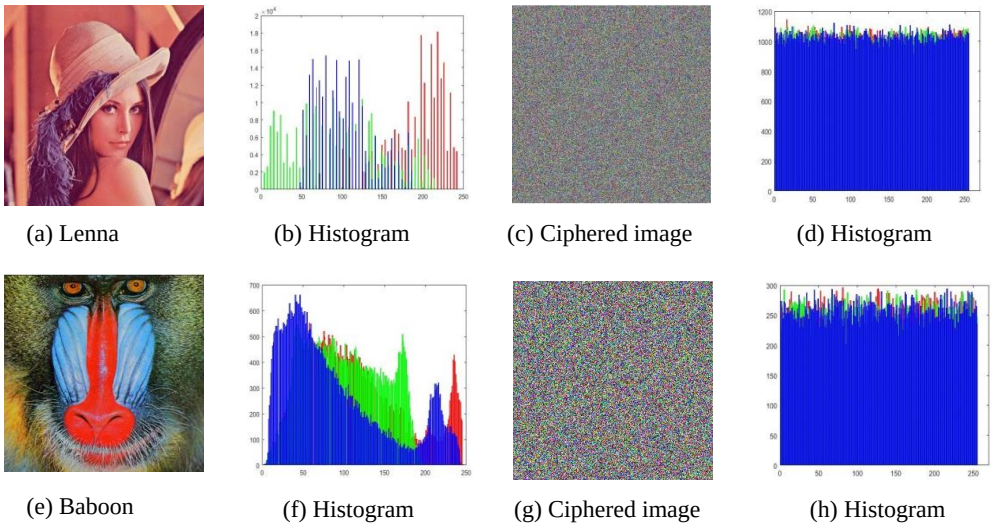
$$NPCR = \frac{1}{M_1 \times M_2 \times M_3} \times \sum_{u=1}^{M_1} \sum_{v=1}^{M_2} \sum_{w=1}^{M_3} D[u,v,w] \times 100\% \quad (12)$$

$$D[u,v,w] = \begin{cases} 0, \ if \ C_1[u,v,w] = C_2[u,v,w] \\ 1, \ if \ C_1[u,v,w] \neq C_2[u,v,w] \end{cases}$$

$$UACI = \frac{1}{M_1 \times M_2 \times M_3 \times 255} \times \sum_{u=1}^{M_1} \sum_{v=1}^{M_2} \sum_{w=1}^{M_3} |C_1 - C_2| \times 100\% \quad (13)$$

where C_1 and C_2 are the same as defined in Equation (11). The test image size is $M_1 \times M_2 \times M_3$. The *u,v,w* indicate the pixel $C_1[u,v,w]$ or $C_2[u,v,w]$ is at the position of *u-th* row, *v-th* column and *w-th* plan.

The average results of NPCR and UACI over *100* different secret keys with LSB change of the parameter *Pp* given in Table 5 are close to the optimal values of NPCR and UACI that are *99.6094%* and *33.4635%* respectively, which demonstrates that the cryptosystem is sensitive to its secret key.



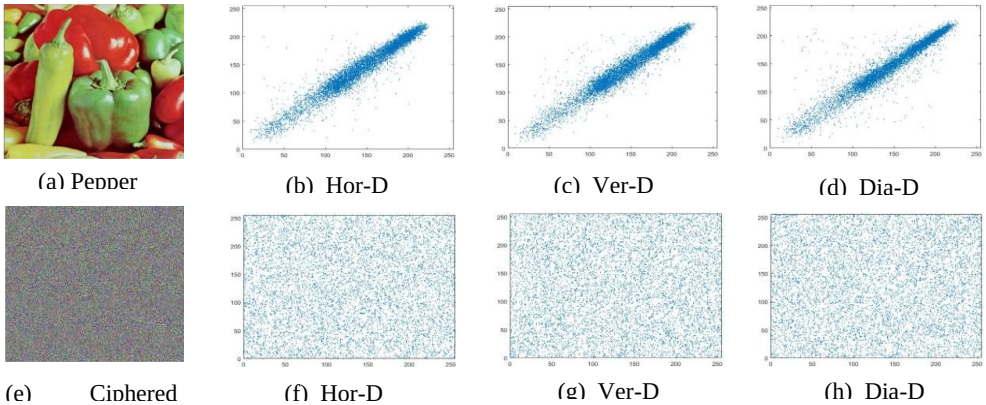
(a) Lenna (b) Histogram (c) Ciphered image (d) Histogram

(e) Baboon (f) Histogram (g) Ciphered image (h) Histogram

Figure 5. Plain and ciphered images and their histograms

Table 4. Correlation coefficient results

Image	Plain image			Ciphered image		
	Hor-D	Ver-D	Dia-D	Hor-D	Ver-D	Dia-D
airfield	0.94001	0.94226	0.90539	-0.00175	0.00108	0.00162
baboon	0.95367	0.93425	0.91758	0.00020	0.00027	-0.00014
boat	0.93812	0.97138	0.92214	-0.00024	-0.00253	-0.00058
lenna	0.97538	0.98528	0.96514	0.00019	-0.00084	0.00114
pepper	0.96172	0.96554	0.95397	-0.00174	-0.00116	0.00046



(a) Pepper (b) Hor-D (c) Ver-D (d) Dia-D

(e) Ciphered (f) Hor-D (g) Ver-D (h) Dia-D

Figure 6. Corelation between adjacent pixels of pepper in the three directions (Hor-D, Ver-D, Dia-D) in plain and ciphered image

Table 5. Hamming distance and NPCR/UACI results

Image	Size	HD (%)					NPCR (%)	UACI (%)
		Xp(0)	Pp	Xs(0)	Ps	Xl(0)		
airfield	512×512×1	49.9979	49.9973	49.9968	50.0043	49.9948	99.6059	33.4725
baboon	256×256×3	49.9985	49.9978	49.9967	50.0035	49.9914	99.6042	33.4386
boat	512×512×1	49.9979	49.9973	49.9968	50.0043	49.9948	99.6059	33.4608
lenna	512×512×3	49.9976	49.9980	49.9989	50.0008	50.0008	99.6078	33.4615
pepper	512×512×3	49.9976	49.9980	49.9989	50.0008	50.0008	99.6078	33.4565

5. Conclusions

In this paper, we developed, implemented and evaluated a novel robust PCNG based on three discrete finite-field chaotic maps: PWLCM, Skew tent map and logistic map using XOR operators and multiplexing mechanism technique. The XOR operators and multiplexing mechanism can increase significantly the complexity of the resulting mapping and enhance effectively the pseudo-chaotic properties of the final generated numbers. The proposed PCNG has a simple structure, works over a 32 bits finite precision field, and can be directly implemented in practice. We also applied this PCNG to a new stream cipher to explore its cryptographic performances in an image encryption application.

The obtained experimental results demonstrate that this new PCNG can generate chaotic numbers with excellent randomness characteristics. The stream cipher based on this PCNG has very good cryptographic properties. The proposed PCNG can be used not only in any design of new stream ciphers, block ciphers or other cryptosystems, but also in any other pseudo-random generator related applications.

6. References

[1] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurc. Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

[2] D. Lambić, "Security analysis and improvement of the pseudo-random number generator based on quantum chaotic map," *Nonlinear Dyn.*, vol. 94, no. 2, pp. 1117–1126, 2018.

[3] Jallouli, Ons, et al. "An efficient pseudo chaotic number generator based on coupling and multiplexing techniques." *International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2016)*. 2016.

[4] H. Karimi, S. M. Hosseini, and M. V. Jahan, "On the combination of self-organized systems to generate pseudo-random numbers," *Inf. Sci. (Ny).*, vol. 221, pp. 371–388, 2013.

[5] F. Chen, X. Liao, T. Xiang, and H. Zheng, "Security analysis of the public key algorithm based on Chebyshev polynomials over the integer ring Z_N," *Inf. Sci. (Ny).*, vol. 181, no. 22, pp. 5110–5118, 2011.

[6] O. Garasym, R. Lozi, and I. Taralova, "Robust PRNG based on homogeneously distributed chaotic dynamics," *J. Phys. Conf. Ser.*, vol. 692, no. 1, 2016.

[7] R. Lozi, "New Enhanced Chaotic Number Generators," *Indian J. Ind. Appl. Math.*, vol. 1, no. 1, p. 42, 2007.

[8] R. Hamza, "A novel pseudo random sequence generator for image-cryptographic applications," *J. Inf. Secur. Appl.*, vol. 35, pp. 119–127, 2017.

[9] M. Wang, X. Wang, Y. Zhang, S. Zhou, T. Zhao, and N. Yao, "A novel chaotic system and its application in a color image cryptosystem," *Opt. Lasers Eng.*, vol. 121, no. December 2018, pp. 479–494, 2019.

[10] S. El Assad and M. Farajallah, "A new chaos-based image encryption system," *Signal Process. Image Commun.*, vol. 41, no. 2016, pp. 144–157, 2016.

[11] O. Jallouli, S. El Assad, M. Chetto, and R. Lozi, "Design and analysis of two stream ciphers based on chaotic coupling and multiplexing techniques," *Multimed. Tools Appl.*, vol. 77, no. 11, pp. 13391–13417, 2018.

[12] S. M. Ismail et al., "Generalized fractional logistic map encryption system based on FPGA," AEU - *Int. J. Electron. Commun., vol. 80*, pp. 114–126, 2017, doi: 10.1016/j.aeue.2017.05.047.

[13] O. Jallouli, M. Abutaha, S. El Assad, M. Chetto, A. Queudet, and O. Deforges, "Comparative study of two pseudo chaotic number generators for securing the IoT," *2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016*, pp. 1340–1344, 2016.

[14] Q. Zongchao, I. Taralova, and S. El Assad. "A robust pseudo-chaotic number generator for cryptosystem based on chaotic maps and multiplexing mechanism." *International Conference for Internet Technology and Secured Transactions (ICITST'2019)*. 2019.

[15] M. Hematti, A. Ahmadi, S. V. Makki, and M. Ahmadi, "Hardware design of chaotic pseudo-random number generator based on nonlinear feedback shift register," *Midwest Symp. Circuits Syst.*, vol. 2018-Augus, pp. 980–983, 2019.

[16] A. Rukhin, J. Soto, and J. Nechvatal, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," *Nist*

Spec. Publ., vol. 22, no. April, pp. 1/1--G/1, 2010.

[17] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dyn.*, vol. 92, no. 2, pp. 305–313, 2018.

