Pseudo-Random Key Stream Generation Algorithm for Encryption Purposes

Zongchao Qiao, Ina Taralova
Laboratoire des Sciences du Numérique de Nantes
LS2N, UMR CNRS 6004
Ecole Centrale de Nantes
Nantes, France

Safwan El Assad
Institut d'Electronique et des Technologies du numéRique
IETR, UMR CNRS 6164
Université de Nantes/Polytech Nantes
Nantes, France

Abstract

For both chaos-based stream ciphers and chaos-based block ciphers, key streams have a crucial influence on their security. A well designed pseudo-chaotic number generator (PCNG) that exhibits both chaotic properties and pseudo-randomness is a good candidate for creating the cryptographic key stream for encryption purposes. PCNGs are based on multiple chaotic maps. Since the majority of the chaotic maps are defined using real numbers, most of the proposed PCNGs use floating-point notations. However, this data type, especially the double-precision notation, has disadvantages of high computation cost and inefficient resource utilization. Also, the quantification errors may undermine the reliability of the produced key stream. To overcome these drawbacks, a key stream generation algorithm using a PCNG scheme is proposed in this paper. The PCNG is based on reformulated skew tent maps over a 32-bit integer field. It not only reduces the resource utilization from the hardware perspective, but also ensures the key stream performance over various operation platforms. Furthermore, the proposed PCNG uses a parameter changeable strategy, which can help to expand the key space, and thus increases the immunity against the brute-force attack. The quality of the key stream produced by the PCNG has been tested in a stream cipher. The analysis and the obtained test results have demonstrated that the proposed PCNG is secure and reliable to generate cryptographic key streams for encryption purposes.

1. Introduction

Nowadays, rapid development of technology and huge amount of information transmission make information security an important issue [1]. Effective and dependable cryptographic techniques are increasingly demanded. Chaotic systems possess the high sensitivity to initial conditions, random-like behavior and complex dynamics, which is highly consistent with the cryptographic requirements of the traditional cryptography according to Shannon's theory of information security. Over the years, chaos-based cryptosystems have become an important encryption strategy in cryptography [2].

Chaos-based symmetric-key cryptosystems can be classified as block ciphers and stream ciphers. The block cipher uses chaotic dynamics to achieve high confusion and diffusion cryptographic objectives by encrypting the plaintext block by block. The key stream required by the encryption algorithm is produced by a chaotic system. While the stream ciphers encrypt the plaintext by masking it using exclusive-or (XOR) operation between a key stream which is also generated by a chaotic system [3,4].

The key stream is crucial for the security of a chaos-based cryptosystem. In particular, the security of a stream cipher relies heavily on the cryptographic quality of the used key stream. A well-designed pseudo-chaotic number generator (PCNG) can provide pseudo-chaotic numbers as the key stream that exhibits enhanced chaos properties and pseudo-random features. In other words, the PCNG is a pseudo-random number generator (PRNG) which is based on chaotic maps, and consequently, chaotic dynamics contributes to the high sensitivity of the PRNG to its seed (i.e. the secret key that is composed of initial conditions and parameters of the adopted chaotic maps in the PCNG).

However, since the exact chaotic behavior exhibits in an assumed continuous field with infinite precision, it is inevitable that the implementation of chaotic maps on digital devices with finite precision leads to dynamical degradation [5]. Due to the quantization,

truncations or round-offs in digital implementations, the adopted chaotic maps using real numbers may lose chaotic features [6,7]. What's worse, they may be locked into periods or even fixed points. As a result, the PCNG has a high risk of losing randomness, which damages the reliability of the produced key streams and leads to a security breach. In addition, from the hardware perspective, the computation of floating-point numbers (especially the double-precision notation) has the disadvantages of slow data transfer and inefficient resource utilization when compared to the fixed-point numbers and integer numbers [8].

To solve these problems, fixed-point solutions have been investigated in [8,9] and a binary PCNG based on a crossed-coupled skew tent scheme with the fraction length of 32 bits has been proposed in [8]. Also, the PCNGs based on the chaotic maps redefined over an integer field with fixed finite precisions proposed in [10,11] have overcome the dynamical degradation problem as well.

In the existing open literature, it has been widely admitted that the coupling of chaotic maps can enhance the chaotic properties effectively [12-15]. But most of them work in the real number set. Due to the effects of finite precision, they cannot be copied but have a positive significance on the design of PCNG using finite precision. Inspired by the idea of ultra-weak coupling structure over a real number domain presented in [16], using redefined integer chaotic maps, we have proposed a coupling scheme which minimizes the dynamical degradation and amplifies the chaotic features effectively. Based on the coupling scheme, a PCNG has been developed in [17]. The key space is limited to 2^{132}, which is large enough to resist the brute-force attack. But similar to the most of the proposed cryptosystems that have the limited key space, equivalent keys or weak keys may lie in the key space, which will reduce the key space and thus decrease the cryptosystem's resistance to the brute-force attack. Hence, a larger key space is expected to enhance the security [13].

Therefore, in this paper, we explore a new key space expandable PCNG scheme based on the integer skew tent maps. The parameters of the skew tent maps are changed by iteration. This scheme not only expands the key space of the PCNG leading to an enhanced immunity against the brute-force attack, but also increases the system's complexity and security performance. The PCNG is a key stream generation algorithm for encryption purposes. To evaluate the cryptographic performance, the generated key stream is tested in a stream cipher.

This paper is organized as follows. The proposed key stream generation algorithm, i.e. PCNG scheme, is described in Section 2. Performance analysis of the PCNG is discussed in Section 3. In Section 4, a stream cipher is constructed based on the PCNG and the cryptographic performance of the generated key stream is evaluated. Finally, Section 5 concludes this work.

2. Scheme of the proposed pseudo-chaotic number generator

The proposed PCNG scheme is shown in Figure 1.

The kernel of the PCNG is the operation of chaotic maps coupling. It uses a coupling matrix (**A**) to couple two skew tent maps with changeable parameters aiming to fulfill very long period orbits to overcome the effect of finite precision and improve the complexity of the PCNG and the uniform distribution property of the produced key stream.

The coupling matrix (**A)** is defined as follows [17]:

$$\mathbf{A} = \begin{bmatrix} 17-\varepsilon & \varepsilon \\ 2\varepsilon & 31-2\varepsilon \end{bmatrix} \quad (1)$$

where ε is the coupling parameter in Ne bits, and $\varepsilon \in \mathbb{N}_+ \cap [1, 2^{Ne}-1]$, $Ne = 4$ bits. Only one coupling parameter (ε) designed here is to ensure a high sensitivity of the produced key stream to ε, and to avoid the linear correlation in **A**.

The chaotic maps coupling process is described as follows:

$$\begin{bmatrix} X1(n) \\ X2(n) \end{bmatrix} = \mathbf{A} \times \begin{bmatrix} F1[X1(n-1)] \\ F2[X2(n-1)] \end{bmatrix} \quad (2)$$

where $X1(n-1), X2(n-1)$ are the previous states of the current states $X1(n)$, $X2(n)$ in the sequences $X1, X2$; $F1, F2$ represent the employed skew tent maps (F).

Among the most frequently-used 1-dimensional discrete chaotic maps, the skew tent map exhibits good uniformity and rich dynamical behavior in the definition range of its parameter (P) [17]. The skew

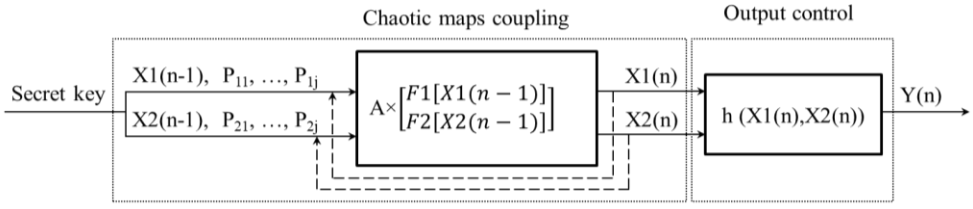


Figure 1. Proposed PCNG scheme for generating key streams

tent map (F) reformulated over an N-bit (N=32) integer field can be described as follows:

$$X(n+1) = F[X(n)] = \begin{cases} \left\lfloor 2^N \times \frac{X(n)}{P} \right\rfloor, & 0 < X(n) < P \\ \left\lfloor 2^N \times \frac{2^N - X(n)}{2^N - P} \right\rfloor, & P < X(n) < 2^N \\ \lfloor 2^N - 1 \rfloor, & otherwise \end{cases} \quad (3)$$

where the state ($X(n)$) and the parameter (P) are integers ranging in [1, 2^N-1].

For $F1$ and $F2$, their corresponding initial conditions are $X1(0)$ and $X2(0)$. Their parameters are $P_{11}, P_{12}, \dots, P_{1j}$ and $P_{21}, P_{22}, \dots, P_{2j}$ respectively (j is the number of parameters used in each skew tent map, and $j = 1,2,3,4, \dots$), and they are used one by one in $F1$ and $F2$. That is, when $j = 1$, P_{11} is the only parameter of $F1$ and P_{21} is the only parameter of $F2$; when $j = 2$, P_{11}, P_{12} are used in turn in $F1$, and P_{21}, P_{22} are used in turn in $F2$; similarly, when $j = 4$, $P_{11}, P_{12}, P_{13}, P_{14}$ and $P_{21}, P_{22}, P_{23}, P_{24}$ are used in turn in $F1$ and $F2$.

The final output (Y) is controlled by an output control function, i.e. $h\big(X1(n), X2(n)\big)$, which is designed for promoting the unpredictability and randomness of the key stream generation. It is achieved by

$$Y(n) = h\big(X1(n), X2(n)\big) = Scir(X1(n), q1, l) \oplus Scir(X2(n), q2, r) \quad (4)$$

where $Scir$ represents the circular shift operation; $Scir(X1(n), q1, l)$ means $q1$-bit left circular shift on the binary sequence $X1(n)$, and $Scir(X2(n), q2, r)$ means $q2$-bit right circular shift on the binary sequence $X2(n)$; $\oplus$ stands for the XOR operator. Here, we choose $q1 = 5, q2 = 3$ for the following tests.

Note that, the proposed PCNG scheme (Figure 1) is not restricted to the working mode described above. It can be considered as a flexible framework which can be used to produce much more different pseudo-random sequences. The flexibility can be seen in the following aspects: (1) any positive integer number of j can be chosen to design the PCNG; (2) it holds not only for the skew tent map, the coupling scheme is also suitable for coupling different piecewise linear chaotic maps; (3) for the output control, other control methods are also applicable (e.g. output X1(n), X2(n) alternatively).

3. Performance analysis of the PCNG

Pseudo-chaotic numbers generated by the PCNG is the key stream for chaos-based cryptosystems. A secure encryption scheme requires that the PCNG exhibits good cryptographic and statistical performances.

For this, for different j (j =1,2,3,4) of Figure 1, 100 secret keys are produced randomly by function "*randi*" in MATLAB. Each secret key is used to generate $10^6 + 3125000$ pseudo-chaotic numbers, and the first 10^6 numbers are regarded as transient and removed. Thus, for each j, 100 sequences with length of 3125000 are tested in this section. (Each sequence has $3125000 \times N\ bits = 100 \times 10^6\ bits$.)

All testes are conducted in MATLAB (R2017a) on a personal computer of Intel(R) Core(TM) i5-5200U CPU in Windows 10, 2.19 GHz processor, 8.00 GB RAM. The test results are summarized in Table 1.

3.1. Cryptographic analysis

3.1.1. Key space analysis. The brute-force attack is a standard attack which aims to find the correct secret key by searching all possible keys (combinations). A large key space is necessary to prevent this kind of attack. It is considered to be secure if the key space is larger than 2^{128} [18]. It is clear that the larger the key space is, the stronger the resistance of the encryption system to the brute-force attack.

The secret key of a chaos-based cryptosystem is comprised of the initial conditions and parameters of the adopted chaotic maps, and the coupling parameter (ε) in the PCNG. Thus, the key size for different j is calculated as:

$$|K_j| = |X1(0)| + |P_{11}| \times j + |X2(0)| + |P_{21}| \times j + |\varepsilon| \text{ bits} \quad (5)$$

where the size of the initial conditions ($|X1(0)|, |X2(0)|$) and parameters ($|P_{11}|, |P_{21}|$) for $F1$ and $F2$ are N bits (N=32), and $|\varepsilon| = Ne$ bits ($Ne = 4$).

Thus, if $j = 1$, $|K_{j=1}| = |X1(0)| + |P_{11}| + |X2(0)| + |P_{21}| + |\varepsilon| = 132$ bits. If $j = 4$, $|K_{j=4}| = |X1(0)| + |P_{11}| + |P_{12}| + |P_{13}| + |P_{14}| + |X2(0)| + |P_{21}| + |P_{22}| + |P_{23}| + |P_{24}| + |\varepsilon| = 324$ bits.

We have calculated the key space for $j = 1,2,3,4$

Table 1. PCNG test results

PCNG	Cryptographic tests		Statistical tests ("✓" means "pass the test")		
	Key space	D_H	Histogram	χ^2_{exp}	Phase space portrait
$j = 1$	2^{132} ✓	50.0004 ✓	✓	1004.19 ✓	✓
$j = 2$	2^{196} ✓	49.9999 ✓	✓	1003.21 ✓	✓
$j = 3$	2^{260} ✓	49.9998 ✓	✓	995.26 ✓	✓
$j = 4$	2^{324} ✓	50.0002 ✓	✓	991.84 ✓	✓

and the results have been recorded in Table 1. The key space of the PCNGs is 2^{132}, 2^{196}, 2^{260}, and 2^{324} respectively, which can protect the cryptosystem from the brute-force attack.

3.1.2. Key sensitivity analysis. A high sensitivity of a key stream to its secret key is necessary for a secure cryptosystem. It requires that two secret keys with a tiny (one bit) difference can give two completely different key streams. This property can be evaluated by calculating Hamming distance (D_H) given as follows.

$$D_H(Y,Z) = \frac{1}{N_b} \times \sum_{i=1}^{N_b} (Y(i) \oplus Z(i)) \quad (6)$$

where D_H between two output key streams Y and Z is calculated bit by bit; Y and Z are generated from two secret keys with only one bit difference; N_b is the number of bit in a sequence; $\oplus$ is an XOR operator. 50% is the optimal value indicating the bit change probability is 50% and the generated key stream is highly sensitive to the secret key.

At each value of j, for the 100 produced test sequences, each corresponding secret key has been randomly changed by one bit to generate another 100 sequences. Calculating 100 D_Hs between them and the average values have been recorded in Table 1, from which we can find that all the average Hamming distance values are very close to 50%. Thus, the key steams produced by the PCNG have been proven to be sensitive to its secret key.

3.2. Statistical analysis

The key stream generated by a PCNG should be uniformly distributed, which can evaluate how random the numbers are. Apart from this, a generation of a number in the key stream should be unpredictable, which means it is difficult to predict the subsequent generations if attackers are in the knowledge of the current state.

Thus, hereafter, the proposed PCNG is analyzed in terms of uniform distribution, phase space portrait and randomness test.

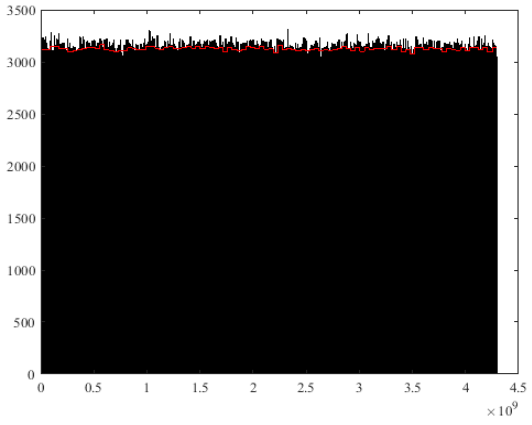


Figure 2. Histogram of a produced key stream when j=1 using 1000 classes
(the curve in red shows the averages in every 10 classes)

3.2.1. Uniform distribution test. The histogram of a produced key stream when j=1 seems to be uniformly distributed in Figure 2. Test results show that for other value of j *(j=2,3,4),* all test sequences have the similar histograms as shown in Figure 2.

In addition, χ^2 tests are applied to assert the uniformity to a more precise degree. The experimental χ^2 value (χ^2_{exp}) is calculated by the following equation.

$$\chi^2_{exp} = \sum_{i=0}^{K-1} \frac{(O_i - E_i)^2}{E_i} \quad (7)$$

where K is the number of classes ($K = 1000$); O_i is the observed number of the generated values in the i-th class; E_i is the expected number of a uniform distribution in the i-th class (here, $E_i = \frac{3125000}{K} = 3125$). The theoretical value $\chi^2_{theo}(K,\alpha)$ is obtained for a threshold of $\alpha = 0.05$, thus $\chi^2_{theo}(K,\alpha) = 1073.64$. If the calculated χ^2_{exp} is smaller than 1073.64, we can confirm the uniformity of the test sequence.

The average χ^2_{exp} values over 100 test sequences for $j = 1,2,3,4$ have been calculated in Table 1, from which we can observe that they are all smaller than 1073.64. Therefore, the PCNG can generate uniformly distributed key streams.

3.2.2. Phase space portrait. The phase space portrait of the skew tent map (Equation (3)) using a randomly generated parameter P has been plotted in Figure 3. According to Figure 3, it can be seen that the function of the used chaotic map can be discovered by plotting the output sequence in the phase space (X(n), X(n+1)). Besides, the next generation of a state in the skew tent map can be predicted. This is not secure in cryptosystems.

By contract, the output sequence of the PCNG in the phase space (Y(n), Y(n+1)) has been displayed in Figure 4, from which we can observe that, the iteration relation between the current state and the next state is in complete confusion, and it is impossible to obtain the generating chaotic function or to predict the next generation.

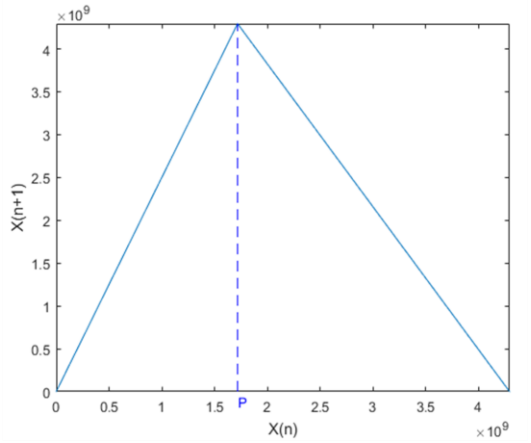


Figure 3. Skew tent map over the 32-bit integer field

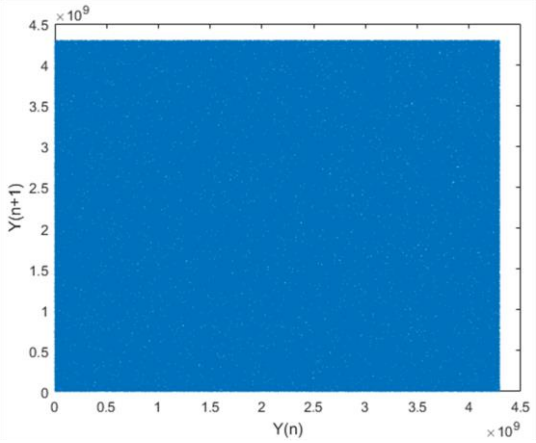


Figure 4. Produced key stream (Y) in the phase space (j=1)

(test sequences for j=1,2,3,4 have the similar diagram)

3.2.3. NIST test. NIST (National Institute of Standard and Technology) test is a suite of randomness tests that is commonly applied to quantify the randomness level of a pseudo-random sequence [18]. It contains 15 independent statistical tests for evaluating the deviations of the test sequence from random behavior. In the NIST standard, if the test sequence is in the length of 100×10^6 bits, the P-value ≥ 0.01 means that the sequence would be considered to be random with a confidence of 99%. And proportions should be larger than 96%.

The produced sequences at each value of j have been tested using NIST test. The results shown in Table 2 have demonstrated that they have passed the NIST test successfully, which has verified that the proposed PCNGs can generate successfully pseudo-random key streams.

4. Performance analysis of a stream cipher

To further verify the cryptographic properties, the produced key streams are evaluated in a stream cipher. A secure stream cipher is a one-time pad cipher whose secret key is used only once. The security performances of a stream cipher include the aspects of a large key space (to guarantee that a large amount of different key streams can be generated), a high sensitivity of the ciphertext to the secret key (to resist the differential attack), and random features of the

Table 2. NIST test results

Test	$j=1$		$j=2$		$j=3$		$j=4$	
	P-value	Proportion	P-value	Proportion	P-value	Proportion	P-value	Proportion
Frequency	0.419	100.000	0.898	99.000	0.699	99.000	0.249	99.000
Block-frequency	0.237	99.000	0.350	99.000	0.384	100.000	0.616	100.000
Cumulative-sums	0.683	99.000	0.961	99.000	0.622	98.000	0.545	99.500
Runs	0.637	97.000	0.319	99.000	0.956	98.000	0.122	98.000
Longest-run	0.851	98.000	0.086	100.000	0.720	98.000	0.304	98.000
Rank	0.154	99.000	0.851	100.000	0.596	99.000	0.760	100.000
FFT	0.335	100.000	0.494	100.000	0.514	100.000	0.554	98.000
Non-overlapping templates	0.469	98.959	0.482	99.115	0.512	98.980	0.488	98.865
Overlapping-templates	0.145	98.000	0.319	98.000	0.182	99.000	0.972	99.000
Universal	0.276	98.000	0.868	100.000	0.658	99.000	0.817	99.000
Approximate entropy	0.637	97.000	0.851	99.000	0.978	100.000	0.249	100.000
Random-excursions	0.412	99.231	0.402	98.654	0.315	98.684	0.436	98.654
Random-excursions-variant	0.406	99.915	0.389	98.889	0.331	98.733	0.444	99.915
Serial	0.667	98.500	0.852	98.500	0.456	99.500	0.887	99.000
Linear-complexity	0.798	99.000	0.335	98.000	0.946	100.000	0.575	100.000

Table 3. Test results of the stream cipher

Image	Size	Key sensitivity test		Statistical test		
		NPCR(%)	UACI(%)	χ^2_{exp}	Entropy: H(P)	Entropy: H(C)
Airfield	512×512	99.6105	33.4616	256.8519	7.1206	7.9993
Lena	512×512×3	99.6096	33.4648	255.7571	5.6822	7.9998
Fruits	480×512×3	99.6097	33.4476	256.0049	7.5190	7.9997

cipher text (to resist the statistical attack). The proposed PCNG has an expandable large key space, which has been discussed in Section 3.1.1. In the following, we will analyze the rest of the security performance. The PCNG with j=4 is used to do the following tests. We select three frequently-used test images, namely, Airfield, Lena, Fruits that have different sizes and features. The length of the key stream depends on the size of the test image. Note that, the first 100 numbers in each produced key stream are regarded as transient state and have been discarded.

4.1. Key sensitivity analysis

To measure the cryptosystem's sensitivity to the secret key, two common methods, i.e., Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI), have been adopted. They are defined as follows.

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

$$D[u,v,w] = \begin{cases} 0, if\ C_1[u,v,w] = C_2[u,v,w] \\ 1, if\ C_1[u,v,w] \neq C_2[u,v,w] \end{cases} \quad (8)$$

$$UACI = \frac{1}{255 \times M_1 M_2 M_3} \times \sum_{u=1}^{M_1}\sum_{v=1}^{M_2}\sum_{w=1}^{M_3} |C_1[u,v,w] - C_2[u,v,w]| \times 100\% \quad (9)$$

where C_1 and C_2 are ciphered images whose corresponding secret keys are just one bit different. The size of the test image is $M_1 M_2 M_3$ (M_1 rows, M_2 columns, M_3 color layers). The notations *u,v,w* indicate that 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* color layer.

For each test image, C_1 and C_2 are encrypted from two secret keys that are only LSB (Least Significant Bit) different (at a randomly chosen initial condition or parameter). We calculate the average values of NPCR and UACI using 100 secret keys (randomly created by MATLAB) and their corresponding LSB different secret keys. The obtained results for each test image have been shown in Table 3. They are very close to the optimal values of NPCR and UACI that are 99.6094% and 33.4635% respectively, which has demonstrated that the stream cipher is extremely sensitive to its secret key.

4.2. Uniform distribution analysis

The distribution property of the test images can be observed by their histograms shown in Figure 5. Figure 5 (b), (f) are the histograms in RGB color layers of the plain images (Airfield and Lena) shown in Figure 5 (a), (e). Their ciphered images shown in Figure 5(c), (g) whose pixel values are uniformly distributed in every color layer, which can be seed in Figure 5 (d), (h).

In addition, χ^2 test is applied using Equation (7) with different parameters: N_c =256, $E_i = M_1 M_2 M_3 / N_c$. $M_1 M_2 M_3$ is the size of the test image that has the same definition as explained after Equation (8). $\alpha = 0.05$. The theoretical value is $\chi^2_{th}(255, 0.05) = 293.2478$. For each test image, 100 different secret keys have been generated to encrypt the test image into the ciphered image. Then, for each ciphered image, we have calculated the experimental χ^2 value (χ^2_{exp}). The average χ^2_{exp} can be found in Table 3, which has confirmed the uniformity of the ciphered images.

4.3. Entropy test

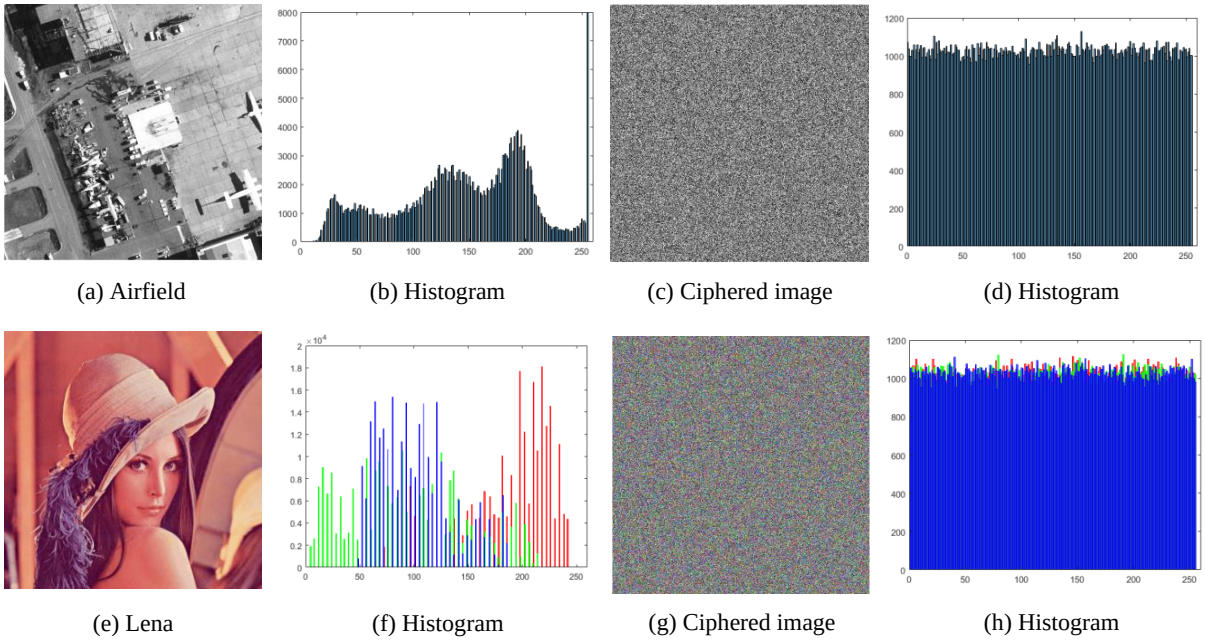
Entropy test is used to evaluate the uncertainty and randomness in a message. The entropy value can be calculated by:

$$H(C) = \sum_{i=0}^{Q-1} Pro(c_i) \times \log_2 \frac{1}{Pro(c_i)} \quad (10)$$

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

In a robust cipher algorithm, the occurrence probability of any pixel value should be the same or almost the same. Hence, 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 results obtained in Table 3, we remark that, compared to the entropy of the plain images, the entropy has been highly raised in the ciphered images and all average



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

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

Figure 5. Plain and ciphered images and their corresponding histograms

information entropy of the ciphered images is close to the optimal value.

4.4. Correlation analysis

It is an intrinsic feature that adjacent pixels in an image show high correlations. A secure cipher should break this relationship.

For this, 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

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

where $N_p = 8000$ is the number of 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 has shown the results obtained in each direction. More visually, Figure 6 gives the distribution of the adjacent pixels in the plain image "Fruits" and in its ciphered image separately. Table 4 and Figure 6 have revealed that the adjacent pixels are highly correlated to each other in the plain image and the stream cipher can break this correlation effectively.

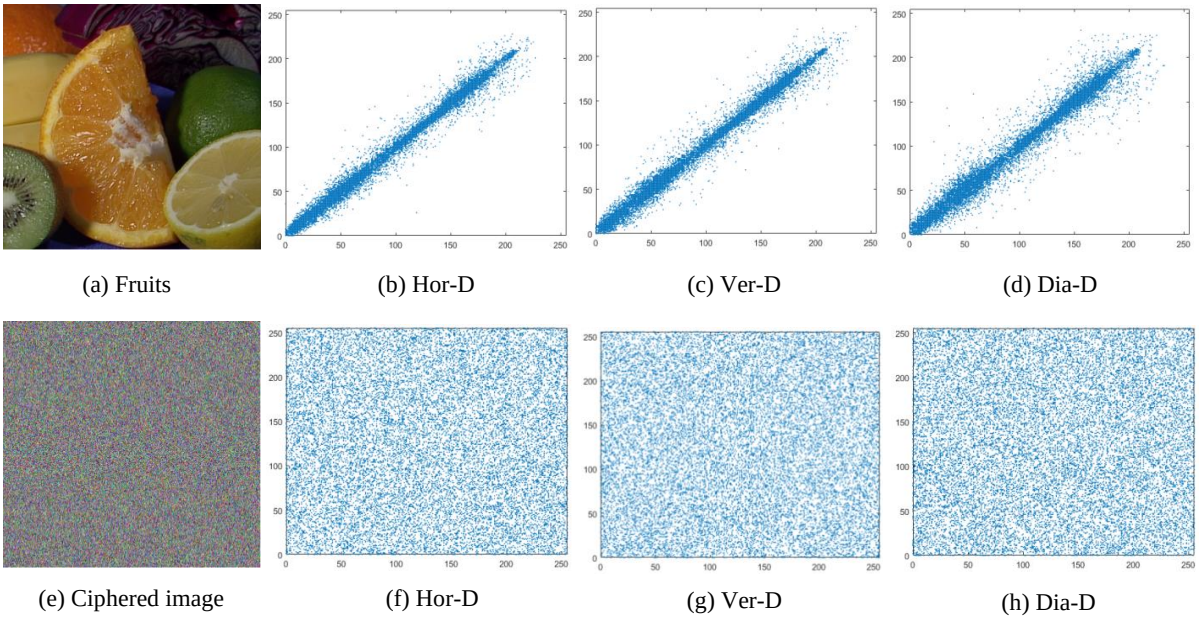
5. Conclusion

In this paper, we have proposed a pseudo-random key stream generation algorithm using a reliable PCNG scheme which is based on a coupling structure and a key space expandable strategy. The chaotic maps coupling structure couples two skew tent maps with changeable parameters, which enlarges the key space and improves the inner dynamics of the PCNG effectively. The output control operation enhances the system's complexity and unpredictability.

The produced key stream can be used in both stream ciphers and block ciphers. The obtained test results of the PCNG and the conducted simulations of the stream cipher have demonstrated that the proposed PCNG algorithm can generate cryptographic key streams with pseudo-randomness for encryption

Table 4. Correlation coefficient results

Image	Plain image			Ciphered image		
	Hor-D	Ver-D	Dia-D	Hor-D	Ver-D	Dia-D
Airfield	0.9399	0.9423	0.9050	-0.0005	0.0018	0.0009
Lena	0.9754	0.9928	0.9649	0.0008	-0.0007	-0.0006
Fruits	0.9936	0.9853	0.9867	-0.0014	-0.0022	-0.0013



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

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

Figure 6. Distribution of 8000 pairs of adjacent pixels in horizontal (Hor-D), vertical (Ver-D) and diagonal (Dia-D) directions in the plain image “Fruits” and its ciphered image

purposes. Furthermore, this PCNG employs the reformulated skew tent maps over a 32-bit integer field, which overcomes the security problems caused by applying the floating-point numbers to the finite precision hardware situations. Thus, it ensures the good performance of the produced pseudo-random numbers over different operation platforms and a high reliability of the generated key streams.

6. References

[1] X. Hu, L. Wei, W. Chen, Q. Chen, and Y. Guo, "Color Image Encryption Algorithm Based on Dynamic Chaos and Matrix Convolution", IEEE Access, 8 (2020), pp. 12452-12466.

[2] Y. Luo, S. Tang, J. Liu, L. Cao, and S. Qiu, "Image encryption scheme by combining the hyper-chaotic system with quantum coding", Optics and Lasers in Engineering, 124 (2020), pp. 105836.

[3] A.A. Rezk, A.H. Madian, A.G. Radwan, and A.M. Soliman, "Reconfigurable chaotic pseudo random number generator based on FPGA", AEU-International Journal of Electronics and Communications, 98 (2020), pp. 174-180.

[4] J. Ons, S. El Assad, M. Chetto, and R. Lozi, "Design and analysis of two stream ciphers based on chaotic coupling and multiplexing techniques", Multimedia tools and applications, 77.11 (2018), pp. 13391-13417.

[5] D. Lambić and M. Nikolić, "Pseudo-random number generator based on discrete-space chaotic map", Nonlinear Dynamics, 90.1 (2017), pp. 223-232.

[6] C. Guyeux, Q. Wang, and J.M. Bahi, "A pseudo random numbers generator based on chaotic iterations: application to watermarking", International Conference on Web Information Systems and Mining, Springer, Berlin, 2010, pp. 202-211.

[7] Z. Qiao, S. El Assad, and I. Taralova, "Design of secure cryptosystem based on chaotic components and AES S-Box", AEU-International Journal of Electronics and Communications (2020), pp. 153205.

[8] R.A. Elmanfaloty and E. Abou-Bakr, "Random property enhancement of a 1D chaotic PRNG with finite precision implementation", Chaos, Solitons & Fractals, 118 (2019), pp. 134-144.

[9] W.S. Sayed, A.G. Radwan, A.A. Rezk, and H.A. Fahmy, "Finite precision logistic map between computational efficiency and accuracy with encryption applications", Complexity, 2017 (2017).

[10] Z. Qiao, 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), London, 2019.

[11] H. Li, L. Deng, and Z. Gu, "A robust image encryption algorithm based on a 32-bit chaotic system", IEEE Access 8 (2020), pp. 30127-30151.

[12] G. Oleg, R. Lozi, and I. Taralova, "Robust PRNG based on homogeneously distributed chaotic dynamics", Journal of Physics: Conference Series, 692 (2016), pp. 012011.

[13] X. Hu, L. Wei, W. Chen, Q. Chen, and Y. Guo, "Color Image Encryption Algorithm Based on Dynamic Chaos and Matrix Convolution", IEEE Access 8 (2020), pp. 12452-12466.

[14] 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", Optics and Lasers in Engineering, 121 (2019), pp. 479-494.

[15] Z. Hua, J. Fan, B. Xu and H. Huang, "2D Logistic-Sine-coupling map for image encryption", Signal Processing,

149 (2018), pp. 148-161.
[16] R. Lozi, and I. Taralova, "From chaos to randomness via geometric undersampling", ESAIM: Proceedings and surveys, 46 (2014), pp. 177-195.
[17] Z. Qiao, I. Taralova, S. El Assad, and M. Saad, “Analysis of the logistic and skew tent map for a smart coupling over a finite field”, 13th CHAOS 2020 International Conference, (2020).
[18] M. François, T. Grosges, D. Barchiesi, and R. Erra, "Pseudo-random number generator based on mixing of three chaotic maps", Communications in Nonlinear Science and Numerical Simulation 19.4 (2014), pp. 887-895.

