An Image encryption using pseudo random bit generator based on a non-linear dynamic chaotic system

ALGIMANTAS ČITAVIČIUS, AUDRIUS JONAVIČIUS
Department of Electronics and Measurement Systems
Kaunas University of Technology
Studentu 50, Kaunas
LITHUANIA
algimantas.citavicius@ktu.lt

Abstract: - This paper deals with the image encryption scheme based on non-linear dynamic chaotic system in stream cipher architecture. The narrowing of the numbers domain interval to [0.2, 0.8] and a double precision floating point format to encode float numbers to binary format were used for pseudo random stream generator. The results of the NIST statistical test suite and security analysis are given. The results show very good cryptographic properties and high security.

Key-Words: - Image encryption, Non-linear chaotic dynamic system, Stream generator, Security analysis, Cryptography.

1 Introduction

New rapid developments in the telecommunication technologies present new challenges for protecting the information from unauthorized access. It has intensified the research activities in the field of cryptography to fulfill the strong demand of the new secure cryptographic techniques [1]. Researchers from the nonlinear dynamics community have noticed a relationship between chaos and cryptography [2, 3]. Such properties as ergodicity, sensitivity to initial conditions/system parameters and mixing property are among them [4]. Chaotic systems can provide fast and secure data protection, which is crucial for multimedia applications [5, 6, 7]. So, the idea to use a non-linear chaotic dynamic system for data encryption is perspective.

In this paper the protection of images is in interest, while traditional ciphers like IDEA, AES, DES are not suitable for real time image encryption as these ciphers require a large computational time and high computing power [6, 7].

Most of image encryption designs are implemented in the form of block cipher although stream cipher can provide better security. In this paper an image encryption system, where stream generator is based on non linear dynamic chaotic system, is proposed. The initial conditions are generated and regularly updated using an external secure 128 bit key.

The test suite from National Institute of Standard and Technology (NIST) [4] was chosen to test sequences generated by proposed stream cipher. This suite is composed of several different well known tests. Each of them is applied to the same sequence of n bits and gives a P-value i. e. the probability that the sequence under the test is random. If a P-value for a test is determined to be equal to 1, then the sequence appears to have perfect randomness. A P-value of zero indicates that the sequence appears to be completely non random. It is well known that many tests present some flaws. A statistical test is formulated to test a specific null hypothesis (H_0). The null hypothesis under test is that the sequence being tested is random against the alternative hypothesis (H_a) for which the sequence is not random. We can commit two errors:

1. Reject H_0 when the sequence is generated by a perfect random generator (Type I error);
2. Accept H_0 when the sequence is generated by a generator that is non random (Type II error).

A more intensive test, involving a number N of different sequences generated by the stream generator under test, was done to overcome the impasse.

The paper is organized as follow. Section 2 describes in detail the design of image encryption scheme. The analysis using NIST statistical test suite is given in section 3, the security analysis is given in section 4 and section 5 concludes the paper.

2 Image encryption and decryption scheme

Proposed image encryption and decryption scheme is presented in Fig. 1.

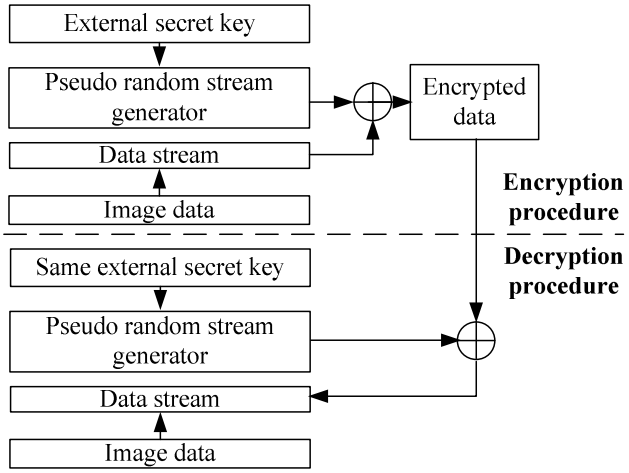


Fig. 1. Image encryption/decryption scheme

The most important part in this scheme is the stream generator. Proposed stream generator consists of two main parts:

1. The auxiliary generator (AG), based on two prime-modulus multiplicative congruential generators (PMMLCG). Parameters of PMMLCG were chosen as suggested in [11];

2. The non-linear dynamic chaos system (NLDCS).

For description of the NLDCS the following formula was proposed:

$$x_{n+1} = (1+\beta)(1+1/\beta)^{\beta} \cdot x_n (1-x_n)^{\beta}. \qquad (1)$$

Here β is an integer in the range $1 \le \beta \le 4$, $x_n \in X = (0,1)$. The set X can be interpreted as a set of float numbers in a computer presentation. The theoretical background for this construction is presented in [8]. A functional model of this generator is presented in Fig. 2.

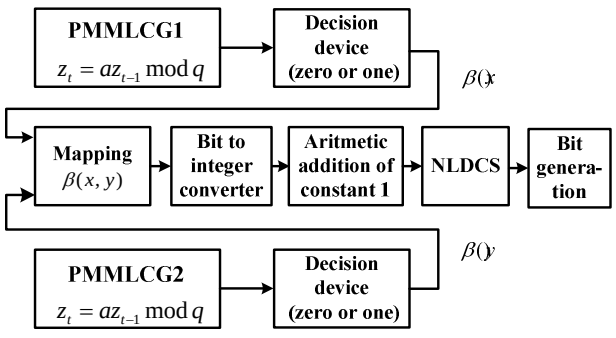


Fig. 2. Functional model of proposed stream generator

Two prime-modulus multiplicative congruential generators (PMMLCG1 and PMMLCG2) represent AG. Heaving adequately chosen the parameters (initial seeds are different for both generators) PMMLCG generates pseudo random numbers, which are divided by modulo *m*, so the normalization is achieved. Zeros and ones as described in [10] are received. After that, each *n* bit sequence is mapped to *n* bit of other PMMLCG. As parameter β value must be integer value in the range $1 \le \beta \le 4$, mapped bits are then transformed to decimal code by using "Bit to integer converter", and, additionally, the constant equal to 1 is added. Next, β value is transmitted to NLDCS. NLDCS generates a set of float numbers in interval (0;1). Numbers domain interval was narrowed to [0.2;0.8] to get pseudo random bits and, according to IEEE 754 standard a double precision floating point format was used in order to encode float numbers to binary format (Fig. 3).

Sign (1 bit)	Exponent (11 bit)	Mantissa (52 bit)

Fig. 3. IEEE 754 Double Floating Point Format

Thus 64 bits from one float number were obtained. Actually, to get more randomness, the highest 16 bits (2 bytes) were removed from 64 bits because they represent sign, exponent and mantissa highest bits that are usually not random (to increase bits randomness it is recommended to remove 3 or 4 bytes, but in this case the performance will decrease because 40 or 32 bits respectively from one float number will be received). Bit generation algorithm after the delivery of numbers to NLDCS is presented in Fig. 4.

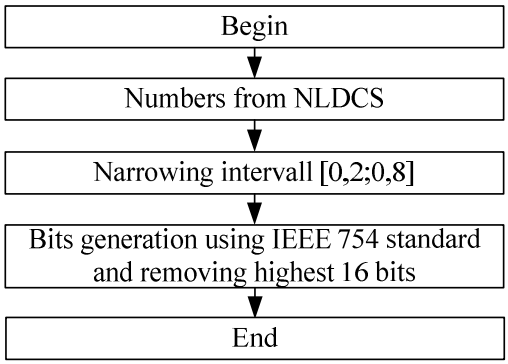


Fig. 4. Bit generation algorithm

2.1 Image encryption

As mentioned above, the purposed image encryption procedure uses 128 bit external secret key.

1. External secret key is divided into 8 blocks, referred to as sessions' keys:

$$K = k_1, k_2, k_3, ...k_{16}. \quad (2)$$

Here k_i is group of two alphanumeric characters in hexadecimal (0-9 and A-F), that represents a session key.

2. An initial condition for nonlinear dynamic chaotic system (2) is calculated as follows:

$$x_0 = \frac{B_1 \cdot 2^0 + B_2 \cdot 2^1 + ... + B_{127} \cdot 2^{126} + B_{128} \cdot 2^{127}}{2^{128}}. \quad (3)$$

Here B_i are binary digits (0 or 1). After that initial condition is updated as follows:

a) First, the session keys are converted to decimal code and this shows the number of iteration, than the x_{n+1} (1) parameter must be updated. For example, if a first session key is equal to $(k_1)_h$='AA', then $(k_1)_d$=170. The x_{n+1} parameter must be updated after 170 iterations. After 16 session keys the parameter x_{n+1} is updated again from the first session key.

b) Second, the appropriate session key is converted to binary and intermediate parameter x'_n is calculated as fallows:

$$x'_n = \frac{B_1 \cdot 2^0 + B_2 \cdot 2^1 + ... + B_7 \cdot 2^6 + B_8 \cdot 2^7}{2^8 - 1}. \quad (4)$$

c) Third, x_{n+1} (1) parameter is updated according to this equation:

$$x_{n+1} = (x_n + x'_n + x_n) \bmod 1. \quad (5)$$

3. The initial condition for PMMLCG1 is calculated as follows:

$$X_{0(\mathrm{PMMLCG1})} = \sum_{i=1}^{32} B_i \cdot 2^{i-1} \bmod (2^{31} - 1), \quad (6)$$

and for PMMLCG2 as:

$$X_{0(\mathrm{PMMLCG2})} = \sum_{i=97}^{128} B_i \cdot 2^{i-1} \bmod (2^{31} - 1). \quad (7)$$

Here B_i are binary digits (0 or 1). PMMLCG has reiteration property. Therefore, after some period parameters must be updated, thus:

a) The sessions keys are grouped in to four groups: $m_1 = \{k_1, k_2, k_3, k_4\} ... m_4 = \{k_{13}, k_{14}, k_{15}, k_{16}\}$.

b) Then each PMMLCG generator reaches the total number of iteration, which is equal to 2^{30}, the parameters are updated as follows:

$$X_{\mathrm{n+1(MLCG1,2)}} = \left(\sum_{i=-31+(32 \cdot m)}^{32 \cdot m} B_i \cdot 2^{i-1} + X_n \right) \bmod (2^{31} - 1). \quad (8)$$

Here B_i are binary digits (0 or 1) and m – groups of session keys and it increases by 1, then PMMLCG reaches the total number of iteration. When the last m group is reached, the parameter $X_{\mathrm{n+1(MLCG1,2)}}$ is updated again from the first group m_1.

After stream generator has been initiated, the image is converted to binary data stream and encryption is done using XOR operation. For decryption procedure, the same external key must be used.

3 The NIST statistical test suite

NIST Test Suite is a statistical package consisting of 15 tests [9]. To perform stream generator analysis with NIST Test suite and get more reliable results for our research on the stream generator, 100 (sample size N = 100) different binary sequences have been generated. A significance level α = 0.01, as suggested by NIST, has been used for the analysis of P-values obtained from various statistical tests.

NIST recommends two strategies to perform the analysis of the generator. First, to check if the P-values are uniformly distributed in the interval [0, 1] with a goodness of fit test, and second, to calculate proportion of sequences passing a test and compare it with the expected value [9].

The distribution of P-values for a large number of binary sequences (N=100) has been examined to check the uniform distribution of P-values for each test. The computation is as follows [9]:

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - N/10)}{N/10}, \quad (9)$$

where F_i *is* the number of occurrences that the P-value is in the i-th interval and N denotes the sample size (in our case $N=100$). The P-value$_T$ of the P-values is calculated using formula (10):

$$P - value_T = igamc\left(\frac{9}{2}, \frac{\chi^2}{2}\right), \quad (10)$$

where *igamc* is the incomplete Gamma function. If $P - value_T \geq 0{,}0001$ then the P-values are considered to be uniformly distributed. The results of each statistical test are presented in Table 1.

Table 1. The results of NIST Statistical Tests*

Nr.	Statistical Test	P_value$_T$
1	Frequency (Monobit)	0.040108
2	Block Frequency (M =128)	0.102526
3a	Cumulative Sums (Forward)	0.289667
3b	Cumulative Sums (Backward)	0.115387
4	Runs	0.699313
5	Longest Runs of Ones	0.759756
6	Binary Matrix Rank (M = 32)	0.289667

7	Spectral DFT	0.678686
8	Non-periodic Templates ($m=1$)	0.115387
9	Overlapping Templates	0.883171
10	Maurer's Universal ($L=7,Q=1280$)	0.798139
11	Approximate Entropy	0.224821
12	Random Excursions (x = +3)	0.445459
13	Random Excursions Variant (x=-3)	0.108791
14	Linear Complexity (M = 1024)	0.616305
15a	Serial (m = 16)	0.637119
15b	Serial (m = 16)	0.350485

* Number of sequences N=100, size of sequence n=10^6 bits.

Graphical representation of the results is presented in Fig. 5. Results form Table 1 and Fig. 5 show that requirements for the uniform distribution are met. So, conclusion was made that bits generated by proposed generator are uniformly distributed.

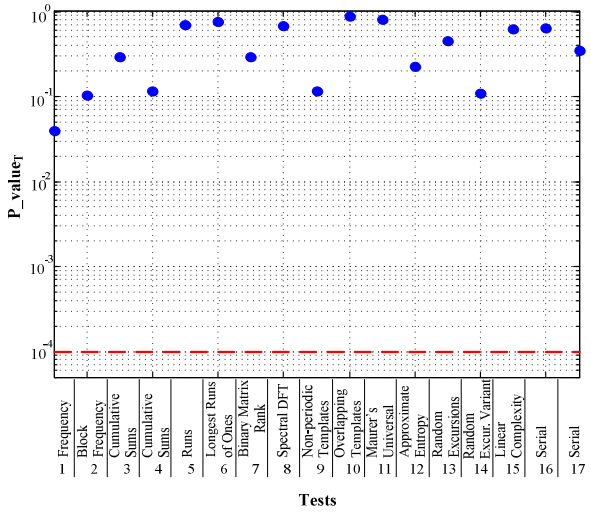


Fig. 5. P-value$_T$ for each NIST statistical test. Dashed line represents the threshold value of P-value$_T$

The range of acceptable proportion was calculated to calculate proportions of the sequences passing the tests. The range of acceptable proportion is determined by the following formula:

$$\hat{p} \pm 3 \cdot \sqrt{\hat{p}(1-\hat{p})/N} \,. \qquad (11)$$

Here $\hat{p} = 1-\alpha$, and N is a number of generated sequences. In our case N=100, so the range of acceptable proportion is 0.99±0.02985 ([1.01985; 0.96015]). The results of proportions of the sequences passing the tests are presented in Table 2.

Table 2. The results of proportions of the sequences passed the tests**

Nr.	Statistical Test	Proportion
1	Frequency (Monobit)	0.9700
2	Block Frequency (M =128)	1.0000
3a	Cumulative Sums (Forward)	0.9800
3b	Cumulative Sums (Backward)	0.9700
4	Runs	1.0000
5	Longest Runs of Ones	1.0000
6	Binary Matrix Rank (M = 32)	1.0000
7	Spectral DFT	0.9900
8	Non-periodic Templates (m=1)	0.9900
9	Overlapping Templates	1.0000
10	Maurer's Universal (L=7,Q=1280)	0.9900
11	Approximate Entropy	0.9900
12	Random Excursions (x = +3)	1.0000
13	Random Excursions Variant (x=-3)	0.9900
14	Linear Complexity (M = 1024)	0.9900
15a	Serial (m = 16)	0.9900
15b	Serial (m = 16)	0.9900

** Number of sequences N=100, size of sequence n=10^6 bits.

Graphical representation of the results is shown in Fig. 6.

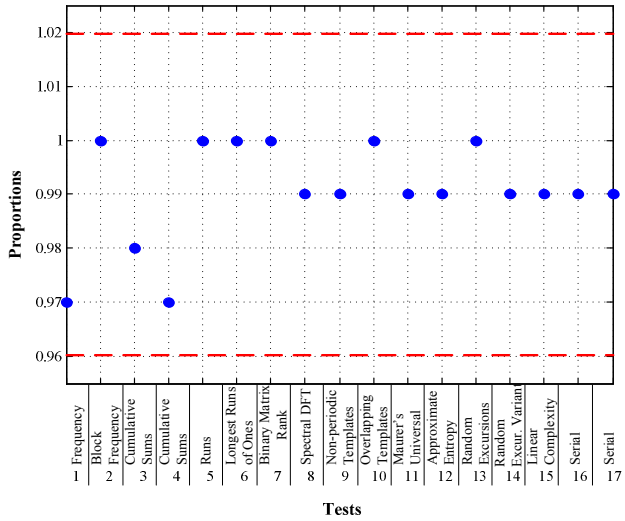


Fig. 6. Proportions of the sequences passing the tests. Dashed line represents the range of acceptable proportion

From a computed proportion for each NIST statistical test a conclusion was made that sequences generated by a proposed generator are inside acceptable proportion range.

4 Security analysis

A good encryption procedure should be robust against cryptoanalytic, statistical and brute force attacks. In this section the security analysis of proposed scheme is discussed to show that it is secure against most common attacks.

4.1 Visual analysis

To perform a visual analysis the several images there encrypted and original and encrypted images were compared. An example of this visual analysis is presented in Fig. 7.

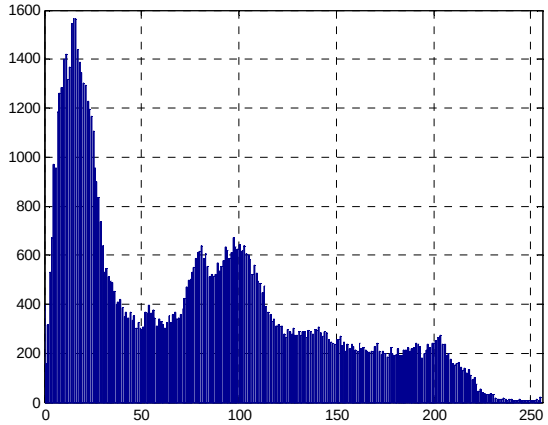
The comparison of original and encrypted images shows, that in encrypted image no visual information is observed.

a) original image b) encrypted image

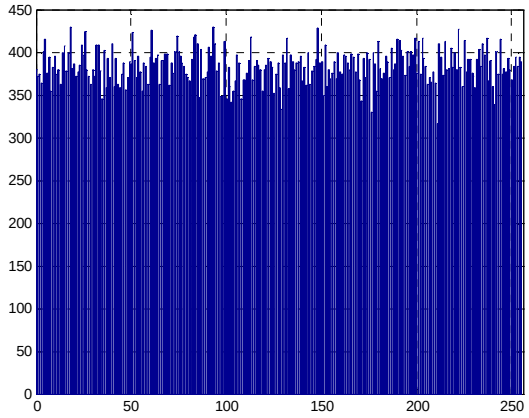
Fig. 7. Image encryption using external secret key '76E55262F2956970ADE550BFE057ED42'

4.2 Histogram analysis

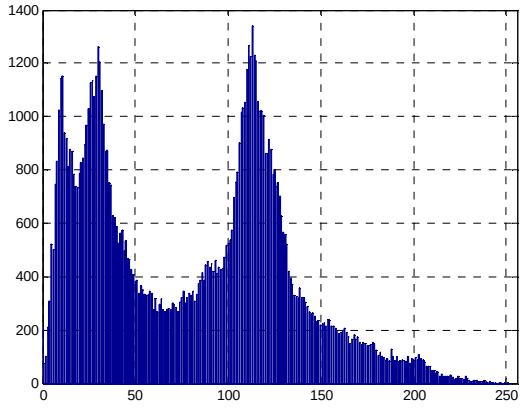
An image histogram illustrates how pixels in an image are distributed at each color intensity level. The histograms of several encrypted and its original images have been analyzed. The histograms analysis of image presented in Fig. 7 for red, green and blue channels are shown in Fig. 8.



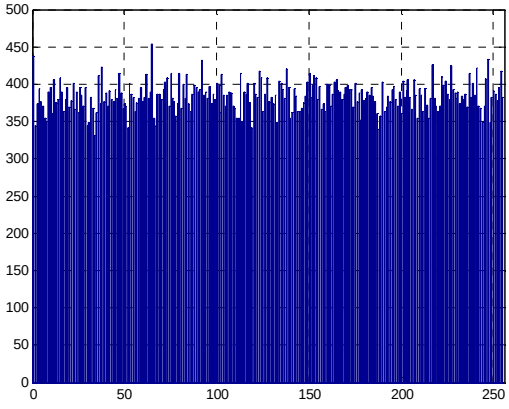
a) Histogram of red channel in original image



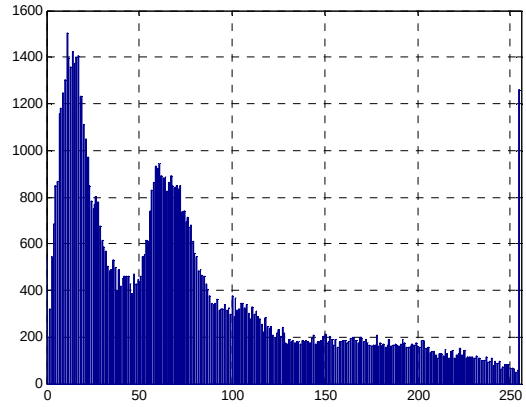
b) Histogram of red channel in encrypted image



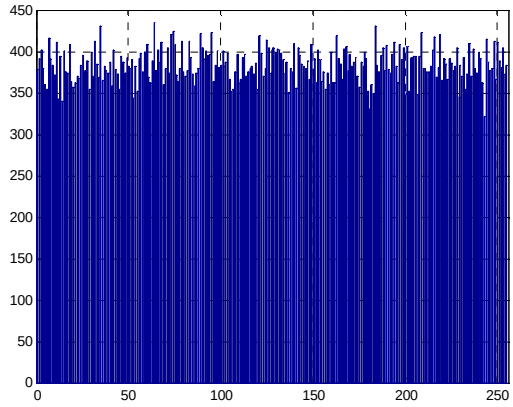
c) Histogram of green channel in original image



d) Histogram of green channel in encrypted image



e) Histogram of blue channel in original image



f) Histogram of blue channel in encrypted image

Fig. 8. Histogram analysis

Fig. 8 shows that the histograms of encrypted image are nearly uniformly distributed and significantly different from the respective histograms of the original image. Analysis of several different images has shown similar results.

4.3 Correlation coefficient analysis

To test the correlation between two adjacent pixels in original image and encrypted image, the image, presented in Fig. 7 was used and the following procedure was carried out. First, the image was converted to grayscale. Second, random selection 1000 pairs of two adjacent (in horizontal and vertical direction) pixels from an image were made. Then, referring to [6], the correlation coefficient was calculated using the following formula:

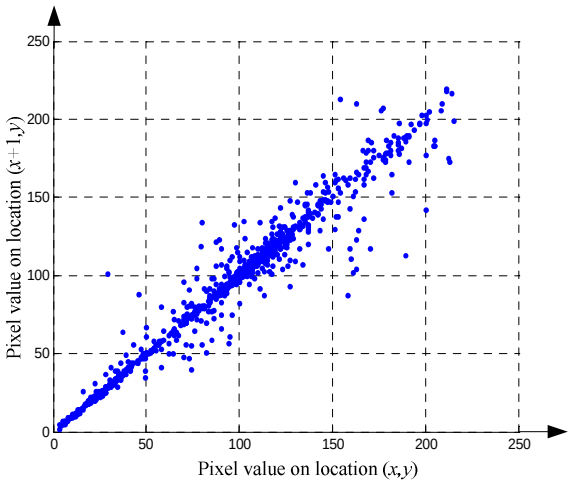
$$r_{xy} = \frac{N \cdot \sum_{i=1}^{N}(x_i \cdot y_i) - \sum_{i=1}^{N} x_i \cdot \sum_{i=1}^{N} y_i}{\sqrt{\left(N \cdot \sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2\right) \cdot \left(N \cdot \sum_{i=1}^{N} y_i^2 - \left(\sum_{i=1}^{N} x_y\right)^2\right)}}, \quad (12)$$

where x and y are the values of two adjacent pixels in the image and N is the total number of pixels selected from the image.

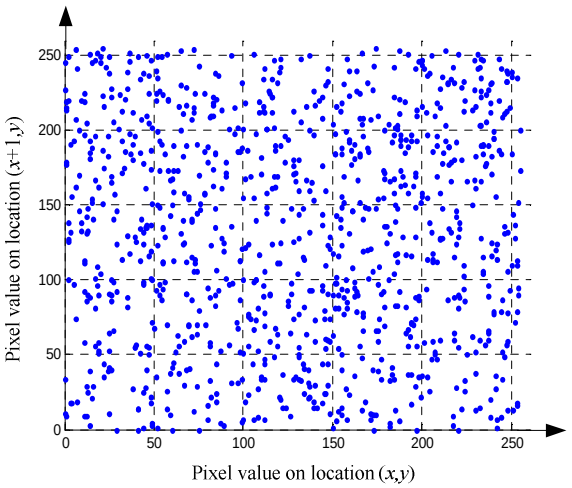
In Fig. 9 the distribution of two adjacent pixels in both directions (vertical and horizontal) respectively in original and encrypted image is shown.

By comparing distributions in original and encrypted images it is clear, that pixels in original image are highly concentrated, but pixels in encrypted image are uniform distributed.

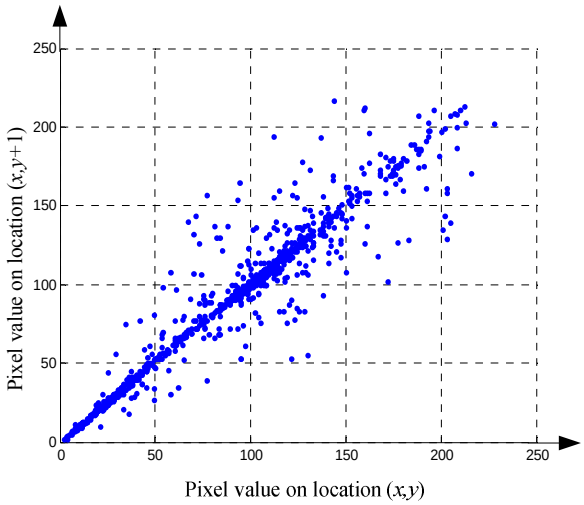
.



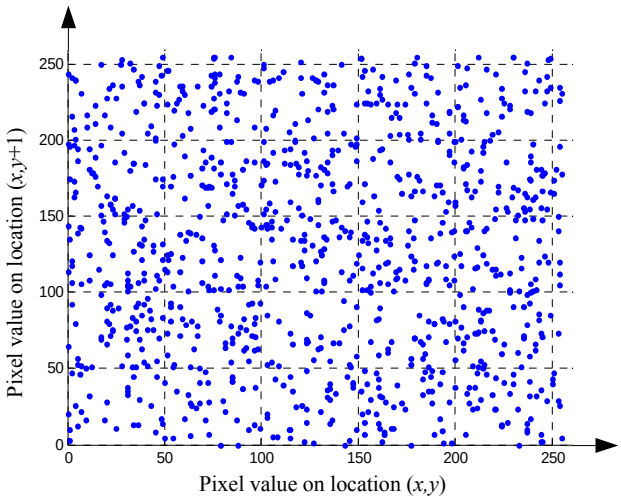
a) The distribution of two horizontally adjacent pixels in the plain image



b) The distribution of two horizontally adjacent pixels in the encrypted image



c) The distribution of two vertically adjacent pixels in the plain image



d) The distribution of two vertically adjacent pixels in the encrypted image

Fig. 9. The distribution of two adjacent pixels in the plain image and encrypted images

The correlation coefficient results are shown in the Table 3.

Table 3. The results of correlation coefficient analysis

Direction	**Correlation coefficient of original image (Fig. 7, grayscale)**	**Correlation coefficient of encrypted image (Fig. 7, grayscale)**
Horizontal	0.97815	0.00828
Vertical	0.95618	-0.03290

Table 3 clearly illustrates, that two adjacent pixels in the original image in both directions are highly correlated. Correlation coefficients are 0.97815 and 0.95618 in horizontal and vertical directions respectively. But correlation coefficient in encrypted image is negligible, thus these correlation analyses prove that the proposed encryption procedure satisfies zero co-correlation.

The correlation coefficient study was extended by using the proposed encryption scheme. The USC-SIPI image database was used, which is a collection of digitized images available and maintained by the University of Southern California primarily to support research in image processing, image analysis, and machine vision. The database is divided into four different categories based on the basic character of the pictures [6]. The image database is freely available at http://sipi.usc.edu/database/.

To achieve our goal the second external secret key was used ‘8F02F124C7AD0C59CA2412DC1490E9FD‘ and correlation coefficient was measured:

1. between original and its corresponding encrypted image;
2. for the two adjacent pixels in the encrypted image.

The results are presented in Table 4.

By comparing our results with results presented in [6] between original and its corresponding encrypted images it is seen, that our measured correlation coefficients are generally smaller.

Table 4. Correlation coefficients for the two adjacent pixels in the original and encrypted images using external secret key ‘8F02F124C7AD0C59CA2412DC1490E9FD‘

File name	**File description**	**Size**	**Type**	**Correlation coefficient between the image and corresponding encrypted image**	**Correlation coefficient for the two adjacent pixels in the encrypted image**
4.1.01	Girl	256x256	Color	-0.002003	-0.002003

4.1.02	Couple	256x256	Color	-0.000069	-0.000069
4.1.03	Girl	256x256	Color	-0.001044	-0.001044
4.1.04	Girl	256x256	Color	-0.001389	-0.001389
4.1.05	House	256x256	Color	-0.001789	-0.001789
4.1.06	Tree	256x256	Color	0.000734	0.000734
4.1.07	Jelly beans	256x256	Color	0.001216	0.001216
4.1.08	Jelly beans	256x256	Color	-0.000293	-0.000293
4.2.01	Splash	512x512	Color	-0.006527	-0.006527
4.2.02	Girl(Tiffany)	512x512	Color	0.006660	0.006660
4.2.03	Baboon	512x512	Color	-0.000542	-0.000542
4.2.04	Girl(lenna)	512x512	Color	0.000292	0.000292
4.2.05	Airplane(F-16)	512x512	Color	0.000040	0.000040
4.2.06	Sailboat on lake	512x512	Color	0.005071	-0.003402
4.2.07	Peppers	512x512	Color	0.000802	0.000802
House	House	512x512	Color	-0.003728	-0.003728
5.1.09	Moon surface	256x256	Gray	0.000351	0.000351
5.1.10	Aerial	256x256	Gray	-0.000988	-0.000988
5.1.11	Airplane	256x256	Gray	0.014382	-0.000591
5.1.12	Clock	256x256	Gray	0.003262	0.003262
5.1.13	Resolution chart	256x256	Gray	0.001469	0.001469
5.1.14	Chemical plant	256x256	Gray	0.001863	0.001863
5.2.08	Couple	512x512	Gray	0.001017	0.001017
5.2.09	Aerial	512x512	Gray	-0.000989	-0.000989
5.2.10	Stream and bridge	512x512	Gray	-0.004599	-0.004599
7.1.01	Truck	512x512	Gray	0.000690	0.000690
7.1.02	Airplane	512x512	Gray	0.012373	0.000500
7.1.03	Tank	512x512	Gray	-0.002512	-0.002512
7.1.04	Car and APCs	512x512	Gray	0.000090	0.000090
7.1.05	Truck and APCs	512x512	Gray	0.002983	0.002983
7.1.06	Truck and APCs	512x512	Gray	0.001820	0.001820
7.1.07	Tank	512x512	Gray	-0.000323	-0.000323
7.1.08	APC	512x512	Gray	0.000344	0.000344
7.1.09	Tank	512x512	Gray	0.001190	0.001190
7.1.10	Car and APCs	512x512	Gray	0.000465	0.000465
boat.512	Fishing Boat	512x512	Gray	-0.000996	-0.000996
elaine.512	Girl(Elaine)	512x512	Gray	-0.000042	-0.000042
gray21.512	21 level step wedge	512x512	Gray	-0.001543	-0.001543
numbers.512	256 level test	512x512	Gray	-0.000749	-0.000749
ruler.512	Pixel ruler	512x512	Gray	0.002519	0.002519
5.3.01	Man	1024x1024	Gray	0.000227	0.000227
5.3.02	Airport	1024x1024	Gray	0.000531	0.000531
7.2.01	Airplane	1024x1024	Gray	-0.000312	-0.000312
Testpat.1k	General test pattern	1024x1024	Gray	0.001253	-0.005286

4.4 Sensitivity analysis

Encryption procedure must be key sensitive. So one bit difference in the key should provide significantly different results. To perform an analysis the lena.bmp image (512x512) presented in Fig. 10 was chosen and for key sensitivity testing the following procedure was carried out:

1. An original image was encrypted by external secret key ‘C9A4864CDC074CCF857248E75736991E‘ (first step).

2. The same original picture was encrypted with slightly different external key with only one bit difference (second step)

'B9A4864CDC074CCF857248E75736991E' (the most significant bit was changed);

3. Again, the same original picture was encrypted with slightly different external key, where the last significant bit was changed 'C9A4864CDC074CCF857248E75736991F' (third step);

4. Three encrypted images were compared.

For comparison correlation coefficients between three encrypted pictures were calculated using method described above. The results are shown in Table 5. Table 5 clearly renders that no correlation exists between three encrypted images even with slightly different secret keys.

Table 5. Correlation coefficient analysis between three encrypted images

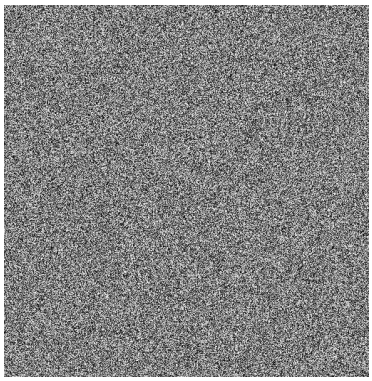
Image 1	**Image 2**	**Correlation coefficient**	**NPCR, %**
Encrypted image (first step)	Encrypted image (second step)	-0.001508	99.561
Encrypted image (First step)	Encrypted image (third step)	0.011862	97.936
Encrypted image (second step)	Encrypted image (third step)	-0.007157	99.254

The number of pixel change rate (NPCR) of two encrypted images with only one bit difference in the keys was measured. Results are presented in Table 5. Almost the same NPCR results we obtained with different encrypted images. According to our results, the conclusion was made, that proposed scheme is very sensitive to small changes in the external secret key.

Moreover, we tested, that image encrypted with 'C9A4864CDC074CCF857248E75736991E' was not correctly decrypted with slight different external keys. Those results clearly show high key sensitivity of proposed scheme.

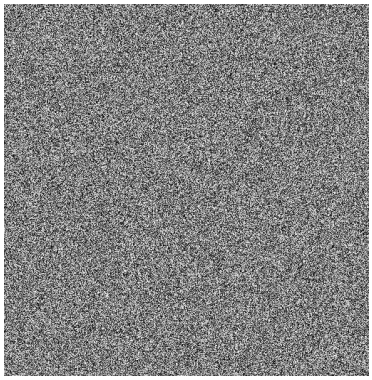
The decrypted images using different external secret keys are shown in Fig. 10.

a) Original image

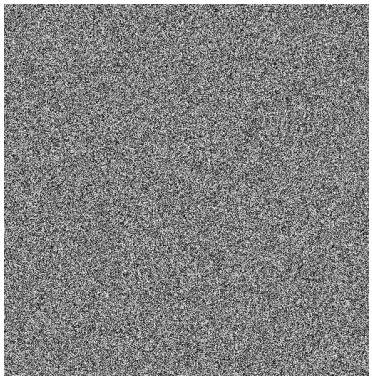


b) Encrypted image using external secret key 'C9A4864CDC074CCF85 7248E75736991E'

c) Decrypted image using external secret key 'C9A4864CDC074CCF85 7248E75736991E'



d) Decrypted image using external secret key 'B9A4864CDC074CCF857248E75736991E '



e) Decrypted image using external secret key 'C9A4864CDC074CCF857248E75736991F '

Fig. 10. Key sensitivity test

4.5 Key space analysis

Key space is the total number of different keys that can be used in the encryption. The proposed encryption scheme has 2^{128} different combinations of secret key. An image encryption system with such long key space is sufficient for practical use. Moreover, the initial conditions for proposed scheme are regularly updated and are dependent on external secret key.

5 Conclusions

This paper proposes an image encryption scheme based on non-linear dynamic chaotic system. The scheme is made in stream cipher architecture. In this paper it is shown, that proposed stream generator passes NIST tests for randomness. A detailed statistical analysis on the proposed encryption scheme is given. The experimental results showed, that proposed scheme has a large key space and high security level. Although the presented image encryption scheme has focused on an image encryption, but it can be widely applied in other information security fields.

References:

[1] Menezes A., Oorschot van P., Vanstone S, *Handbook of Applied Cryptography*, CRC Press, 1996.

[2] Lin C.F., Chung C. H., A Fast Chaos-based Visual Encryption Mechanism for Integrated ECG/EEG Medical Signals with Transmission Error, *Proceedings of 12th WSEAS International Conference on SYSTEMS*, 22-24 July 2008, Heraklion, Greece, pp. 355-360.

[3] Grigoras V., Grigoras C., Chaos Encryption Method Based on Large Signal Modulation in Additive Nonlinear Discrete-Time Systems, *Proceedings of the 5th WSEAS International Conference on Non-Linear Analysis, Non-Linear Systems and Chaos*, 16-18 October 2006, Bucharest, Romania, pp. 98-103.

[4] Patidar V., Sud K. K., A Novel Pseudo Random Bit Generator Based on Chaotic Standard Map and its Testing, *Electronic Journal of Theoretical Physics*, No. 20, 2009, pp. 327-344.

[5] Li S., Jing L., Gao X., An Image Encryption Algorithm Based on Multi-Dimensional Orthogonal Sequence, *Proceedings of the 7th WSEAS International Conference on Multimedia Systems & Signal Processing*, 15-17 April 2007, Hangzhou, China, pp. 60-65.

[6] Pareek N.K., Patidar V., Sud K.K, Image encryption using chaotic logistic map, *Image and Vision Computing*, Vol. 24, 2006, pp. 926-934.

[7] Hossam A., Hamdy K., Osama A., An Efficient Chaos-Based Feedback Stream Cipher (ECBFSC) for Image Encryption and Decryption, *Informatica*, Vol. 31, 2007, pp. 121-129.

[8] Čitavičius A., Jonavičius A., Japertas S., Unpredictable cryptographic pseudo-random number generator based on non-linear dynamic chaotic system, *Electronics and Electrical Engineering*, No. 7 (79), 2007, pp. 29-32.

[9] National Institute of Standards and Technology, *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, Special publication 800-22, Revision 1, August 2008.

[10] Ursulean R., Reconsidering the Generalized Logistic Map as a Pseudo Random Bit Generator, *Electronics and Electrical Engineering*, No 7 (56),2004, pp. 10-13.

[11] Park K. S., Miller K.W., Random dumber generators: Good ones are hard to find, *Communications of the ACM*, Vol. 31, No. 10., 1988, pp. 1192-1201.