A New Key Stream Generator Based on 3D Henon map and 3D Cat map

Dr. EkhlasAbass Albhrany[1] **TayseerKaram Alshekly[2]**

Abstract-

The key is the important part at any security system because it determines whether the system is strength or weakness. This paper aimed to proposed new way to generate keystream based on a combination between 3D Henoun map and 3D Cat map. The principle of the method consists in generating random numbers by using 3D Henon map and these numbers will transform to binary sequence. These sequence positions is permuted and Xoredusing 3D Cat map. The new key stream generator has successfully passed theNIST statistical test suite. The security analysisshows that it has large key space and its very sensitive initial conditions.

Keywords: chaotic function,pseudorandom sequences, random bit generator, 3D Henon map, 3D Cat map, NIST test suite.

1.Introduction

Generating random numbers are mainly used to generate secret keys or random sequences and it can be carried out by many different techniques such as chaotic maps. In recent years many pseudo random numbers or sequences generators are proposed based on chaotic maps. In [1] the Chaotic Linear Congruentlygeneratorcalled CLCG is proposed as a pseudo random number generator and shows that what chaotic features of the Discrete Logistic Map are utilizable for creating pseudo random numbers in cryptographic perspective. The reference [2] proposed new method based on two main operations one to create chaotic values utilizing two logistic maps and the second to transform them into binary words utilizing random S-Box tables. In [3] new pseudo random number generator is proposed based on the Chaotic Henonmap. This paper hows that what chaotic Henon Map features are Exploited for using as pseudo random numbers generatorin cryptographic. Three logistic maps in [4] are cumulated in the algorithmic to produce anovel pseudo- random bit generator and 32 arbitrary bits block is createdat each iteration. By mixing chaotic maps produced from an input main vector, a new pseudo-random number generator (PRNG) is proposed [5]. The algorithm uses a chaotic function for permutations where it computes and indexes new positions based on linear congruence's. Three 1D chaotic maps in [6] are used to create key stream and the key stream correlation propertiesare studied and analyzed. The Jacobian elliptic chaotic maps and standard map are used to create new pseudorandom number generatorby producing binary sequence from cn and sn types of elliptic chaotic maps [7]. This binary sequence location isdiffused utilizing standard map.

In this paper newkey stream generator is proposed based on a combination between 3D Henon map and 3D Cat map and it is called ***Chaotic Key Stream Generator*** (CKSG). The basic idea of CKSG is to iterate the 3D Henon map many times until the three outputsproduce three binary sequences of same length. The three sequences are permuted and Xored using 3D Cat map and produce one sequence of 72-bits length. The choice of using 3D Henon map and 3D Cat map is to enlarge the system complexity and makes difficulties for an attacker to concentrate sensitive information from the output. The proposed generator has a number of advantages which are: high sensitiveness to initial values, high degree of randomness and great throughput.

Whatever is left of paper is sorted out asSection 2displays the description of the chosen chaotic maps. Section 3displays a detailed description of proposed CKSG. The statistical analysis is discussed in Section 4. The security analysis is given in section 5 before conclusions.

2. chaotic maps

Chaos-based encryption innovation has been examined for decades and it has turned into a significance branch of cryptography. Some of specific properties of chaos system that make it good choice for cryptography aresensitivity of initial parameter, unpredictable, and so on.., while it is deterministic, so it very appropriate to be utilized as generator of key stream for stream cipher system[6]. In proposed CKSG, two chaotic maps which are 3D Henon map and 3D Cat map.

2.1 3D Henon map

3D Henon map is a simple three dimensional quadric autonomous system that can be represented by the equation [8]:

$$\begin{aligned} x_{i+1} &= a - y_i^2 - bz_i \\ y_{i+1} &= x_i \\ z_{i+1} &= y_i \end{aligned} \qquad (1)$$

The system is discrete over time, its state portrayed by$(x_i,y_i,z_i)$where the iteration number *is* represented by *i*. In any new iteration *i+1*, the state $(x_{i+1},y_{i+1},z_{i+1})$ is computedby using(x_i,y_i,z_i).This paper usesa = 1.76 and b =0.1. The generator initial value would be a set of incessant values (x_i,y_i,z_i)selected within the range [-1.18 , 1.85] for each output state $x_{i+1},y_{i+1\,or}\ z_{i+1}$[8].

The 3D Henon map has reversibility and simple geometric insight. In addition, it will decrease the errors in calculation because of its coefficient is an integer.

2.2 3D Cat map

Arnold cat map is a special chaotic map. It has important property which isthe permutation of the location of bits [9]. Many authors have proposed the improved 3D Arnolds Cat Map. Hongjuan Liu et al. [10] enhanced 3D Cat Map by presenting four control parameters a, b, c and d as shown in the following equation:-

$$\begin{aligned} \mathrm{x}_1 &= (\mathrm{x}_0 + \mathrm{y}_0\mathrm{a}\ )\mathrm{mod\ N} \\ \mathrm{y}_1 &= (\mathrm{x}_0\mathrm{b} + \mathrm{y}_0(\mathrm{ab}+1)\mathrm{mod\ N} \\ \mathrm{z}_1 &= (\mathrm{x}_0\mathrm{c} + \mathrm{y}_0\mathrm{d} + \mathrm{z}_0)\mathrm{mod\ N} \end{aligned} \qquad (2)$$

Here x_0, y_0, z_0 represents the original location of bits and x_1, y_1, z_1represents theposition of the bits after the Cat Map transformation and the$(\mathrm{a}, \mathrm{b}, \mathrm{c}\ \mathrm{and}\ \mathrm{d})$ represent the control parameters which are any integer number.

[1]Department of Computer Science, Mustansiriyah University, Baghdad, Iraq, email: akhlas_abas@yahoo.com

[2]Department of Computer Science, Mustansiriyah University, Baghdad, Iraq, email: tayseer.karam@yahoo.com

3. The Proposed chaotic key stream generator (CKSG)

The proposed CKSG comprises of the following main steps:-

- **Step 1:** Input the initial value (x_0,y_0,z_0)and the control parameters values which are a =1.76 and b =0.1 to the Henon map equation (1). Theinitial values (x_0,y_0,z_0) are floating point numbers with precision of10^{-16}.
- **Step 2:**The Henon map is iterated100 times and overlook the outcomes, so as toremove the chaotic transient effect.
- **Step 3:** Do the following steps:
  - A. Iterate the Henon maps one time.
  - B. Convert the three floating point outputs to binary sequence by repeating multiplying the number after decimal by 2 until it becomes 1.0.
  - C. If the three binary sequences have the same length which is 80 bits then continue otherwise return to step A.
- **Step 4:**Cut 24- bits the three resulted sequence in the following way:cut the first 8 bits from the first binary sequence,cut 8 bits from the middle of second binary sequence and finally cut the last 8 bits from third binary sequence.
- **Step 5:**These 24-bits are divided into 4 6-bits integer numbers that serves as the four control parameters of the Cat map equation (2).
- **Step 6:**The remaining 72 bits in each binary sequence are Xordin the following way:
  - A. Iterate the Cat map one time and results three different locations in rang [1..72].
  - B. The bit in first location in first binary sequence is Xored with the bit in second location in second binary sequence and with the bit in third location in third binary sequence.
  - C. These two steps A and B are repeated for all 72-bit binary sequences and result one 72-bits binary sequence.
- **Step 7:** Repeat from step 3 until it reached to the desired number of bits.

4. Performance analysis

The output sequences must have a high degree of security, randomness and be absolutely decorrelated from each other. Some cryptographic tests must be performed to gauge the degree of security and analyze the performance of the proposed CKSG.

4.1 Statistical Analysis

The output binary sequences should show independently a high degree of randomness and be decorrelated with each other, whatever the initial values. So that a statistical analysis should be coordinated to exhibit the quality and nature of the pseudorandom binary sequences.

1. Randomness test

This analysis isapplied through statistical tests suite NIST [11]. This testing is accomplished on binary sequences created from close seed values. The testing was performed by creating 1000 of various binary sequences each of which has length 10000bits. Each sequence is produced utilizing various initial values. The initial values are dispersed in the interval[-1.18….1.85]. All 1000 sequences that produced by the proposed CKSG are testedutilizing the NIST statistical package. NIST consists of 15 tests. In each test, a set of p-value is calculated and compared to a fixed signicance level α= 0.01(i.e., just 1% of the generated sequences are anticipated to be not passed). If the p-value ≥ α, a statistical test is passed and fails otherwise.For multiple sequences that tested at the same time, a ratio**η**is defined each test as the proportion of sequences passing successfully. The ratio **η** is compared to an acceptable ratiowhere scope of satisfactory proportions isdictated by utilizing the confidence interval defined as

$$\mathrm{p} \pm 3\sqrt{\mathrm{p}(1-\mathrm{p})/\mathrm{n}} \qquad (3)$$

Where p = 1- α = 0.99 and n is the number of sequences. The NIST tests are performed on two types of sequences: individual sequences, concatenate sequence.

The randomness of proposed CKSG is analyses by partitioning theNIST tests into two groupsbasedon the length of generated sequences. The tests that must be performed on the sequences of length $>= 100$ bits represent the first group while tests that must beperformed on the sequences of length $>= 100$ represent the second group. So, 1000 individual sequences are generated each of which has 10000 bits. The individual sequences are tested by using the first test group. And twoconcatenate sequences are generated each of which created by concatenating 500 individual sequences (i.e. each concatenate sequence has length of 5000,000 bits). The concatenate sequences are tested by using first and second test groups.

For individual sequences, the ratio**η** is:$0.99 \pm 3\sqrt{0.99 \times 0.01/1000} = 0.99 \pm 0.00944$. This means that the ratio should be in the confidence rang [0.99944..0.98056]. Table 1 and Table 2 shows the results of NIST tests that performed on the two groups.

2. Correlation test

Correlation test is to review the correlation between the generated key stream sequences. This can be achieved in two ways: -

A. Pearson's correlation coefficient is computed between each pair of generated sequences to analyses the correlation between them [12]. Let atwo of sequences specified by:$\mathrm{Str1} = [\mathrm{x1}, \ldots, \mathrm{xN}]$ and$\mathrm{Str2} = [\mathrm{y1}, \ldots, \mathrm{yN}]$. Hence, the corresponding correlation coefficient is [12]:

$$C_{str\,1,str\,2} = \frac{\sum_{i=0}^{N-1}(x_i - \bar{x}).(y_i - \bar{y})}{[\sum_{i=0}^{N-1}(x_i - \bar{x})^{\ 2}]^{\frac{1}{2}}.[\sum_{i=0}^{N-1}(y_i - \bar{y})^{\ 2}]^{\frac{1}{2}}} \qquad (4)$$

Where the mean values of Str1 and Str2 are: $\bar{x} = \sum_{i=0}^{N-1} x_i / N$And $\bar{y} = \sum_{i=0}^{N-1} y_i / N$.

Pearson's correlation coefficient is computed between each two of the 1000 generated sequences to find the correlation between them. Figure 1shows the histograms of the coefficient results. The histogram demonstrates that the processed coefficients are near 0. This implies around 99.99% of the coefficients have a place with [-0.045, 0.045] and the correlation between the generated sequences is little.

B. Hamming distance is computedin order to find the correlation based straight on the bits of sequences. Given two binary sequences $S = [s_0, \ldots\ldots, s_{M-1}]$and $S' = [s'_0, \ldots\ldots\ s'_{M-1}]$of same length (M). The Hamming distance is the quantity of locations where they vary, i.e., the quantity of locations where one has a 0 and the other a 1 [13]. The distance is given as:

$$d(S, \acute{S}) = \sum_{j=0}^{M-1} (s_j \oplus \acute{s}_j). \qquad (5)$$

The binary sequences are truly randomwhen the normally distance is around M/2, which gives an attribution (i.e. d(S, S') / M) of about 0.50. The bits of 1000 generated sequences are tested using Hamming distance andFigure 2 shows a histogram of all resulted valuesof Hamming distance. The distributions demonstrate that every one of the attribution are around 50% and 99.98% of the coefficients are belonging to [0.482, 0.52]. This testing gives another sign about the decorrelation between the created sequences.

4.2 Security analyses

All the basic purposes of the cryptosystem and cryptographic necessities should be considered when the examination of theproposed CKSG [7]. The analyzed points are: -

Table 1: Results of first group of the NIST tests on the 1000 generated sequences for individual and concatenate sequences. The ratio η of p-value concerns individual sequences while the p-value.

Test name	η of Individual	Final Result	p-value of concatenate	Final Result
Frequency	0.991	Success	0.905307983776307 0.289191814914286	Success
Block Frequency	0,993	Success	0.225447062078015 0.68924610362277	Success
Runs	0.994	Success	0.853810801639485 0.0412925005184941	Success
Longest Run	0.99	Success	0.409529448519039 0.0882155857706728	Success
Rank	0.993	Success	0.48828080757995 0.144378747864366	Success
FFT	0.987	Success	0.0325054727221126 0.892272584017501	Success
Non-Overlapping	0.999	Success	0.673184428623356 0.0420308945936199	Success
Serial (1)	0.99	Success	0.359692175657996 0.831578626667663	Success
Serial (2)	0.993	Success	0.30241034593743 0.83070114469873	Success
Cumulative Sums	0.9994392	Success	0.871868696399201 0.316994598434403	Success

Table 2: Results of second group of the NIST tests on the 1000 generated sequences for concatenate sequences .The p-value concerns the concatenate sequences

Test name	p-value of concatenate	Final Result
Overlapping	0.815126105580905	Success
	0.834413786808821	Success
Universal	0.374868375109138	Success
	0.384221340553717	Success
Linear Complexity	0.2763623967443990	Success
	0.1666329034656860	Success
Approximate Entropy	0.0795227617270783	Success
	0.124208144417423	Success
Random Excursions (8 p-values)	0.05979924688738	Success
	0.41532049476826	Success
	0.55331522338838	Success
	0.97974056949489	Success
	0.35870585263769	Success
	0.71569400477744	Success
	0.98079793904025	Success
	0.08372913676105	Success
	0.37204390949148	Success
	0.54596512512388	Success
	0.73140941364088	Success
	0.59560523763353	Success
	0.53458744509288	Success
	0.83795752596742	Success
	0.30247755507913	Success
	0.19874577385017	Success
Random E-Variant (18 values)	0.668734316427883	Success
	0.621168259454406	Success
	0.671069436710653	Success
	0.72162046590831	Success
	0.516274324411201	Success
	0.533431066542473	Success
	0.396143909152074	Success
	0.583882420770365	Success
	0.689564900574569	Success
	0.342781711147911	Success
	0.119545252882496	Success
	0.114573741133914	Success
	0.104590551119221	Success
	0.119008828347636	Success
	0.197160533396295	Success
	0.177690646908093	Success
	0.106136748337259	Success
	0.178881439504483	Success
	0.686486257281681	Success
	0.273303853579324	Success
	0.177274696250261	Success
	0.244643937323319	Success
	0.218178203164651	Success
	0.368874221649744	Success
	0.512925852937571	Success
	0.547282873911836	Success
	0.812080422204426	Success
	0.647528665710641	Success
	0.533314928628673	Success
	0.799853385685005	Success
	0.835724372797886	Success
	0.490920729780798	Success
	0.405061053016087	Success
	0.539393062681328	Success
	0.646933474779644	Success
	0.660580505963299	Success

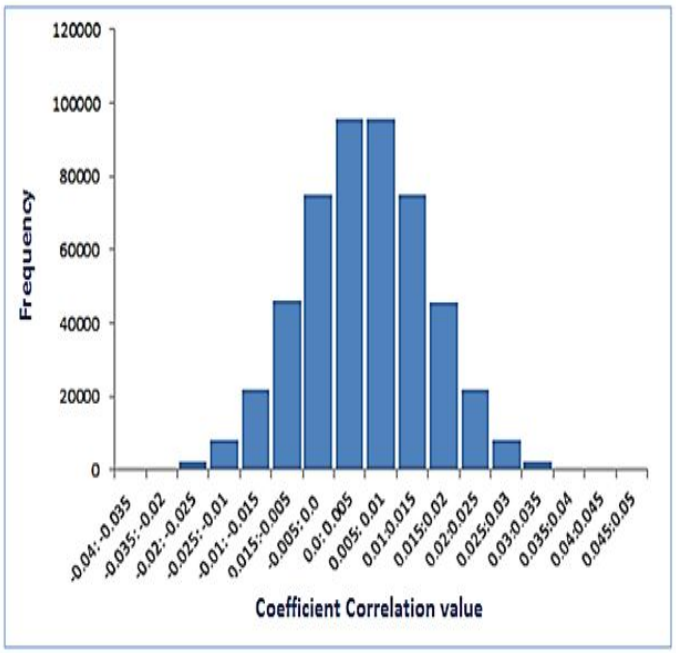

Figure 1: Histogram of Pearson's correlation coefficient values on interval [-0.04, 0.05] for the 1000 sequences.

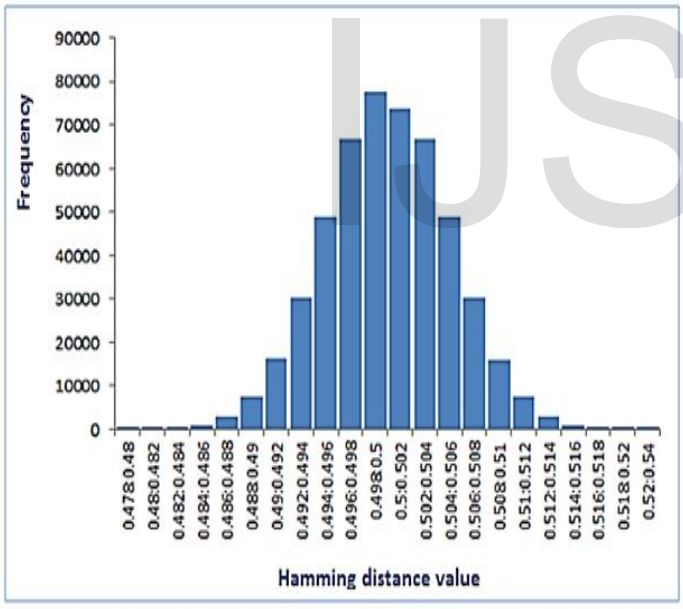

Figure 2: Histogram of Hamming distance on interval [0:478; 0:540] for the 1000 sequences.

1. Key space analyses

Keeping in mind to make brute-force attacks infeasible, the proposed CKSG should to have a big key space. The size of key space that is littler than 2^{128} is not secure adequately. Here, the key space is consist of the Henon map parameters (x_0, y_0, z_0)which are floating point numbers, where $x_0, y_0, z_0 \epsilon$ [-1.18,1.85]. If each of x_0, y_0, z_0 has precision of 10^{-16}, the size of key space for initial values is 2^{160} $((10^{16})^3)$. Moreover the Cat map control parameters a, b, c & d ϵ [0 ..63] are integer number where every parameter has 64 (2^6) possible keys. So the total space of keys is $2^{160}+(2^6)^4=2^{184}$.

2. Key sensitivity analyses

The sensitivity to the change of key parameters is imperative for any key stream generator [7]. The sensitivity on the key is fundamental element for the pseudo-random generation. In other words, a little different on initial values, the output binary sequences should be\totally uncorrelated.

To guarantee the sensitivity of the key, twotestsareperformed utilizing Pearson's correlation coefficients and Hamming distance. Four large binary key stream sequences S1, S2, S3, S4 are generated from a little various initial values in which each one haslength of N =1000,000 bits.

S1 is generated by using the initial values X_0=0.7435681012324872, Y_0=-1.1217564322658924, Z_0=1.6910245678205533.
S2 is generated by using the initial values X_1=0.7435681012324873, Y_1=-1.1217564322658923, Z_1=1.6910245678205534.
S3 is generated by using the initial values X_2=0.7435681012324874, Y_2=-1.1217564322658922, Z_2=1.6910245678205535.
S4 is generated by using the initial values X_3=0.7435681012324875, Y_3=-0.1217564322658921, Z_3=1.6910245678205536.

Table 3 shows the results of Pearson's correlation coefficients and Hamming distance which demonstrate that the generated sequences are uncorrelated from each other.

Table 3: Correlation coefficients between four key binary sequences produced with slightly different seeds.

Tests	S1/S2	S1/S3	S1/S4	S2/S3	S2/S4	S3/S4
Pearson Corr. Coef.	0.00151977	-0.00049284	-0.00090623	-0.00160611	-0.00028597	-0.00021140
Hamming Distance	0.499225	0.500295	0.500438	0.500822	0.500143	0.500091

3. Brute-force attack

It is an attack that can be theoretically utilized against any type of key stream generator. Such attack is generally used, when it is unrealistic to recognizeany shortcoming in thealgorithm that would make the task simpler.The methodology of the attack is easy and includes checking methodically every conceivable key until the first key is acquired.A large key space permits to discomfit such type of attack. It is by and large acknowledged that a key space of sizebigger than 2^{128} is computationally secure against such attack [4]. In the proposed CKSG, the size of the key space is around 2^{178}which clearly permits opposing the brute force-attack.

4. Differential Attacks

Such methodof cryptanalysis was presented by Biham and Shamir .Its guideline is to investigate and exploit the impact of a little variety in input pairs on the difference of corresponding output pairs. This methodpermits finding the most probable key that was utilized to generate the key stream sequence [14]. The same four large binary key stream sequences (S_1, S_2, S_3, S_4 generated from slightly different initial values) each of which has length of N =1000, 000 bitswith a specific end goal to analyses the security of the proposed CKSG algorithm against differential attack.These sequences arerepresented as pairs of 8-bits binary sequences with sizes of N =1000, 000 bits. The sum of absolute difference between a pair of sequences is calculated using the following equation[2]:

$$SAD(S_1, S_2) = \frac{1}{N}\sum_{i=1}^{N}\frac{|S_1 - S_2|}{2^8} \qquad (6)$$

The result of SAD is shown in Table 4. If SAD value is very close to ideal value of 1/3 then the test result shows that the proposed CKSG is very sensitive to the keys and it can highly resistive this kind of attack.

Table 4: sum of absolute difference four key binary sequences produced with slightly different seeds.

Sequence pairs	S1/S2	S1/S3	S1/S4	S2/S3	S2/S4	S3/S4
SAD	0.3328	0.3334	0.3330	0.3329	0.3258	0.3331

4.3 Speed analysis

Another essential side for any CKSG is the algorithm execution time. In real-time applications, the temporal limitation in the implementation of algorithm is as essential as the consequence of this algorithm. The speed performance analysis is analyzed on a personal computer withntel(R) Core(TM) 2 Duo CPU T8100@ 2.10 GHZ 2.10 GHZ. The algorithm is implemented using Visual Basic.net 2015. In this analysis, five binary sequences of different lengths are generated and their execution time is calculated. If the sequence length is increased the time is increased in a linear relationship. The Table 5 shows the performance in terms of speed of the proposed CKSG algorithm.

Table 5: speed analyses of proposed CKSG.

length in Mb	Speed in Mbps
1	43
1.5	96
2	167
2.5	257
3	371

5. Comparative study of proposed CKSG

The proposed CKSG is compared with the PRBG proposed by Reference [4] which we will named as PRNG1and PRBG proposed by Reference[15]which we will named as PRNG2. The difference between the proposed CKSG andPRBG1 and PRBG2 is made on numbers of factors which include key space, number of bits in each iteration, the result of NIST test and the result of correlation coefficients. Table 6 shows the result of comparison.

The key space of proposed CKSG is greater than that of PRNG1 and PRNG2.The number of bits in each iteration of proposed CKSG is 72 bits in each iteration. This means that the proposed CKSG is faster than PRBG1 and PRBG1. The quality of the output sequences randomness is evaluated through NIST tests. The output sequences of proposed CKSG and PRNG1 and PRNG2 are random with large values for the tests. The correlation histogram of proposed CKSG shows that the around 99.99% of the coefficients belong to [-0.045, 0.045] and the correlation between the generated sequences is very small. In PRNG1around 99:56% of the coefficients belong to [-0.1, 0.1] while in PRNG2 around 99:02% of the coefficients have an absolute value littler than 0.08. This means that the correlation between generated sequences using proposed CKSG is very low.

Table 6: Comparison between the proposed CKSG and PRBG1 and PRBG2

	key space	number of bits at each iteration	Freq.	Block Freq.	Runs	Longest Run	Rank	FFT	Non-Over-lapping	Serial (1)	Serial (2)	Cumu-lative Sums
proposed CKSG	2^{184}	72	0.991	0.993	0.994	0.99	0.993	0.987	0.999	0.99	0.993	0.9994
PRBG1	2^{147}	32	0.991	0.991	0.989	0.989	0.988	0.987	0.993	0.989	0.99	0.9910
PRBG2	2^{175}	32	0.990	0.991	0.989	0.989	0.988	0.989	0.992	0.989	0.990	0.9908

6. Conclusions

In this paper, a chaotic key stream generator based on the 3D Henon map and 3D Cat map is proposed. The initial conditions (x_0, y_0, z_0) are the input to the 3D Henon chaotic map. The Henon chaotic map is iterated to produce binary sequence of same length. 3D Cat map is used to diffuse the binary sequence. 72-bit is generated in each iterated of the proposed generator. The proposed CKSG has capability to generate a very large number of key stream sequences which can be beneficial in many applications in cryptographicsince it has numerous attributes which are the large key space size, the algorithm sensitivity to the initial values (keys), the quality of the produced key stream sequencesand thedegree of security versus several attacks.

References

[1] B.F. Vajargah, R. Asghari," Cryptographic Secure Pseudo-Random Generation: The Chaotic Linear Congruental Generator (CLCG)", Sci. Int. (Lahore), vol. 27, 2015.

[2] M. Hamdi, R. Rhouma, S. Belghith, " A Very Efficient Pseudo-Random Number Generator Based On Chaotic Maps and S-Box Tables", International Journal of Information and Communication Engineering, Vol.2, 2015.

[3] B.F. Vajargah, R. Asghari, " A Pseudo Random Number Generator Based on Chaotic Henon Map (CHCG)", International Journal of Mechatronics, Electrical and Computer Technology (IJMEC), Vol. 5, Apr. 2015, PP. 2120-2129.

[4] M. FRANCOIS, D. Defour ,C.Negre, " A Fast Chaos-Based Pseudo-Random Bit Generator Using Binary64 Floating-Point Arithmetic", Informatica, vol.38,2014,pp. 115–124.

[5] M. FRANCOIS, T. GROSGES,D. BARCHIESI1,R. ERRA," A New Pseudo-Random Number Generator Based on Two Chaotic Maps", INFORMATICA,Vol. 24,2013,pp. 181–197.

[6] Chong Fu, F. Zhang, "A Novel Stream Cipher Algorithm Based on Chaotic Maps", IEEE, 2007.

[7] L.F. Jalil , H.H. Saleh, E.A. Albhrany, " New Pseudo-Random Number Generator System Based on Jacobian Elliptic maps and Standard Map", IJCCCE, Vol.15,2015.

[8] D. Sava, Adriana Vla, R. Tataru," A new type of keystream generator based on chaotic maps: illustration on a Hénon generalized map", IEEE, 2014.

[9] S. Keshari, S. G. Modani, " Image Encryption Algorithm based on Chaotic Map Lattice and Arnold cat map for Secure Transmission", International Journal of Computer Science and Technology, Vol. 2, March 2011.

[10] P.N. Khade, M. Narnaware, " 3D Chaotic Functions for Image Encryption", IJCSI International Journal of Computer Science,Vol. 9, May 2012.

[11] Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", NIST Special Publication Revision 1a,vol. 800, April 2010, pp.131.

[12] V. Patidar, N. K. Pareek, G. Purohit and K. K. Sud, “A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption”, Optics Communications, vol. 284, no. 19, September 2011, pp. 4331-4339.

[13] W. Janke, “Pseudo random numbers: generation and quality checks”, Quantum Simulations of Complex Many-Body Systems, vol. 10, 2002,pp. 447–458.

[14] M. FRANCOIS, D. DEFOUR,” A Pseudo-Random Bit Generator Using Three Chaotic Logistic Maps”, hal.archives-ouvertes, February 6, 2013.