International Journal of Electrical and Computer Engineering (IJECE)
Vol. 8, No. 6, December 2018, pp. 4577 – 4583
ISSN: 2088-8708, DOI: 10.11591/ijece.v8i6.pp4577-4583

Domain Examination of Chaos Logistics Function As A Key Generator in Cryptography

Alz Danny Wowor[1] **and Vania Beatrice Liwandouw**[2]
[1]Department of Informatics Engineering, Satya Wacana Christian University, Salatiga, Indonesia
[2]Department of Computing Science, Radboud University, Nijmegen, The Netherland

Article Info

Article history:
Received September 28, 2017
Revised May 09, 2018
Accepted May 22, 2018

Keyword:
Domain Examination
Logistic Function
Chaos
Cryptography

ABSTRACT

The use of logistics functions as a random number generator in a cryptography algorithm is capable of accommodating the diffusion properties of the Shannon principle. The problem that occurs is initialization x_0 was static and was not affected by changes in the key, so that the algorithm will generate a random number that is always the same. This study design three schemes that can providing the flexibility of the input keys in conducting the examination of the value of the domain logistics function. The results of each schemes do not show a pattern that is directly proportional or inverse with the value of x_0 and relative error x_0 and successfully fulfill the properties of the butterfly effect. Thus, the existence of logistics functions in generating chaos numbers can be accommodated based on key inputs. In addition, the resulting random numbers are distributed evenly over the chaos range, thus reinforcing the algorithm when used as a key in cryptography.

Corresponding Author:
Alz Danny Wowor
Affiliation
Faculty of Information Technology, Satya Wacana Christian University,
Jl. Diponegoro 52-60, Salatiga 50711, Indonesia
+62852 0071 0079
alzdanny.wowor@staff.uksw.edu

1. INTRODUCTION

The logistic function $f(x) = rx(1-x)$ or in the iterative form $x_{i+1} = rx_i(1-x_i)$ was usually used as a generator in the generation of chaos-based random numbers [1]. This function is able to accommodate the diffusion properties of Shannon's principle on cryptography algorithms, since they have a sensitivity to initial values.

Research [2], [3], [4], uses numbers as static initialization on logistics functions in cryptography algorithms. This means that changes to the key don't have an effect on random numbers, thus the algorithm will still acquire a random number that is always the same. Research [5], also uses numbers as input values that directly fill in by user, nevertheless becomes inefficient since the user needs more input other than the key and plaintext. This research not only combines algorithms such as [6], [7], [8], but designs a new algorithm to obtain an unique key. Additionally, the function will generate a random number, if the initialization domain r and x_0 as seed is limited to a certain value, $0 < x_0 < 1$, and $r = 4$.

The value of r is constant, making the strength of the algorithm rests on the value of x_0. Moreover it needs an examination process by using any scheme that can increase x_0 complexity space, on the contrary remain in logistics function domain. This study provides the flexibility of key inputs that are efficient and can generate different random numbers of logistics functions.

2. PROPOSED EXAMINATION SCHEME

The examination process of the logistics function domain conducted to a wide variety of inputs that allow it to be used as a key. Suppose the key $k_1, k_2, \ldots, k_8$ is the result of a conversion of eight ASCII characters. Key input is set up to eight characters, considering the user's ability to recall key and also consider the complexity of the key guessing space when the use of the characters too little. Each key input of less than eight characters, then do the padding process with character "§" that equivalent to 167 in ASCII.

This study provides three algorithms that are used for the examination process of initialization value in domain $0 < x_0 < 1$, as shown in the general scheme in Figure 1. The ratio of r_a is designed as a comparison of two values to accommodate the domain from initialization. Let $r_a = p_a/q_a$ where $q_a > p_a$, for $a = 1, 2, 3$ and $p_a, q_a \in R$.

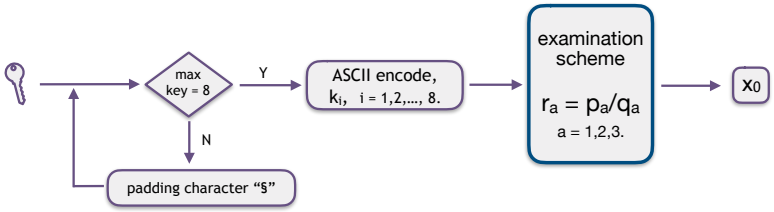

Figure 1. General Scheme of x_0 Initialization Value Determination.

2.1. The First Scheme

Given each $k_i \in Z_{256}$ for $i = 1, 2, \ldots, 8$ is the decimal base number of ASCII conversion results. To be able to make changes at every turn of the input, given index value $d_i \in Z_{256}$ for $j = 1, 2, \ldots, 8$ as a constant value which is multiplied by the value k_i. Scheme-1 is determined by using $r_1 = p_1/q_1$ where,

$$p_1 = (k_1)^2 + k_7 d_7 + k_3 d_3 + k_4 d_4 + k_5 d_1 + k_6 d_6 + k_8 d_8 + k_1 d_5 \tag{1}$$

$$q_1 = \sum_{i=1}^{8} k_i/8 \cdot (k_7 + d_2) k_1 d_3 + k_4 d_5 + k_6 d_8 + k_3 d_1 + k_5 d_6 + k_8 d_7 (k_2 + d_4) \tag{2}$$

Determination of p_1 value is obtained from the sum of multiplication k_i and d_i which is performed based on position, it is only for k_1 and d_5 there is crossing position and at k_1 squared. Multiplication of position difference is also done to acquire the q_1 value, however in sum with the average value of each k_i. The multiplication combination is done as a variation to gain a unique ratio value, taking into account the requirement $q_1 > p_1$.

2.2. Second scheme

Scheme-2 also uses the same decimal number and index value, where each k_i, $d_i \in Z_{256}$ for $i, j = 1, 2, \ldots, 8$. The ratio value is determined by the equation $r_2 = p_2/q_2$,

$$p_2 = (k_1 + k_2 + k_3 + k_4 + k_5 + k_6)/6 = \sum_{i=1}^{6} k_i/6 \tag{3}$$

$$q_2 = k_1 d_1 + k_2 d_2 + k_3 d_3 + k_4 d_4 + k_5 d_5 + k_6 d_6 + k_7 d_7 + k_8 d_8 = \sum_{i=1}^{8} k_i d_i \tag{4}$$

The p_2 value is obtained using the average of the first six values of k_i, whereas to earn the q_2 value is the sum of the multiplication of k_i and d_i values according to the order of each value.

2.3. Third scheme

Every $k_i, d_j \in Z_{256}$ for $i, j = 1, 2, \ldots, 8$. Scheme-3 is obtained based on $r_3 = p_3/q_3$, where the determination of the numerator and denominator is given in Equation (5) and Equation (6). The p_3 value is the

average of the multiplication k_i with d_i only at the first value, the third value, and the sixth value with each index.

$$p_3 = (k_1 d_1 + k_2 + k_3 d_3 + k_4 + k_5 + k_6 d_6)/6 \tag{5}$$

$$q_3 = 3(k_1 d_1) + 7(k_2 d_2) + 11(k_3 d_3) + 13(k_4 d_4) + 17(k_5 d_5) + 23(k_6 d_6) + 6(k_7 d_7) + 27(k_8 d_8) \tag{6}$$

The value of q_3 is the sum of the k_i and d_i multiplications based on the index with the constants selected by different increments.

3. RESULT AND DISCUSSION

3.1. Analysis of Examination Process

Domain examination of the logistics function is performed based on the three schemes given in the previous section. Referring to the general scheme in Figure 1, then testing related variations of inputs that allow to be used as a key. Each possible key is an ASCII character whose decimal basis is in the range 0 to 255. The problem that occurs is not all numbers have a character. Consequently the test for the lowest number can't start from decimal 0 instead in decimal 32 which is proportional to the space character, whereas testing for the largest number at decimal 255 is equivalent to the character . In addition to character testing for minimum and maximum decimals, key tests are also tested with one bit difference, so that it can be seen how sensitive each scheme is to generating initialization values. Table 1 is the simulation result of each scheme in obtaining the value of x_0.

Table 1. Key Variations Test on Each Scheme.

Test	Input Keys	Scheme-1	Scheme-2	Scheme-3
1	ZZZZZZ	0.002416239247	0.020477815700	0.082448573324
2	ZZZZZY	0.002417771534	0.020467836257	0.082481509183
3	fti	0.002441029054	0.024268127467	0.082049869712
4	ftj	0.002439355077	0.024284705051	0.082201209246
5	$4LaT1g4	0.006268369980	0.025188594809	0.095909955612
6	$4LaT1g3	0.006272557467	0.025266127990	0.096168154762
7	(space 8 character)	0.018665158371	0.027777777778	0.096054888500
8	~~~~~~~~	0.003244514742	0.027777777778	0.096054888508
9	ÿÿÿÿÿÿÿÿ	0.001456374283	0.027777777778	0.096054888508

Testing with one bit difference (numbers 1 through 6) shows the changes in values that begin to occur in the 4th or 5th mantissa of the x_0 value in each schema. Accordingly, the process of domain examination for each scheme succeeds to generate different initialization. This condition corresponds to the need of logistics functions in obtaining random numbers based on chaos. Tests with the same eight-character input based on the smallest decimal, the medium decimal, and the largest decimal are given successively in the numbers 7, 8, and 9 in Table 1. The initialization values in scheme-2 and scheme-3 obtain the same value, although the input is very different. This condition occurs since the determination ratio of r_2 and r_3 using the average process to get the numerator, while for the denominator using the addition and multiplication of characters with the index value with the same position. On the other hand, scheme-1 uses a combination of multiplicity of position difference, squared and mean process. In that case, scheme-1 keeps generating different initialization values.

3.2. Relative Error Test

Relative error testing [9] was conducted to see whether linear character reduction would yield also linear results on the x_r values with proportional or inversely proportional.

Use $E_R = |c_a - p_a| / c_a \cdot 100\%$, the key ÿÿÿÿÿÿÿÿ is chosen as the reference value c_a (number 1 in Table 2), and the approximation value as the key input p_a less than eight characters ÿ (iterated from number 2 to number 8).

Table 2. Relative error test on key variation differences.

Test	Input keys	Scheme-1	Scheme-2	Scheme-3	Relative Error		
					Scheme-1	Scheme 2	Scheme-3
1	ÿÿÿÿÿÿÿÿ	0.00145637428	0.027777777778	0.09605488850			
2	ÿÿÿÿÿÿÿ	0.00151184278	0.030084945729	0.10267089320	3.81	8.31	6.89
3	ÿÿÿÿÿÿ	0.00237830973	0.032442748091	0.10498487627	63.30	16.79	9.30
4	ÿÿÿÿÿ	0.00247676219	0.032778687034	0.10922797121	70.06	18.00	13.71
5	ÿÿÿÿ	0.00260130457	0.032743277230	0.11525621373	78.62	17.88	19.99
6	ÿÿÿ	0.00272929162	0.032262996942	0.11525043975	87.40	16.15	19.98
7	ÿÿ	0.00287551307	0.031283195241	0.10350303974	97.44	12.62	7.75
8	ÿ	0.00146481070	0.029781420765	0.09849935979	0.58	7.21	2.54

The relative error results in each scheme do not show a proportional or inversely proportional pattern. So, it will complicate cryptanalyst to see the pattern of input changes on the same character.

3.3. Linear Regression Test

One-to-one correspondence between the input-output is also important so that cryptanalyst difficult to reconstructing scheme and make prediction of the key used as input. This relationship can be seen through a linear regression test based on the rate of change on the resulting value of x_0.

Figure 2, is the result of each scheme visualized using the Scatter plot. The diagram of each scheme has no linear relationship, because when the curve matching process is used, the coefficient of determination (R^2) is close to zero. This test illustrates that any changes to the key characters is done patterned, will not provide an initial value x_0 linearly patterned, either proportional or inversely.

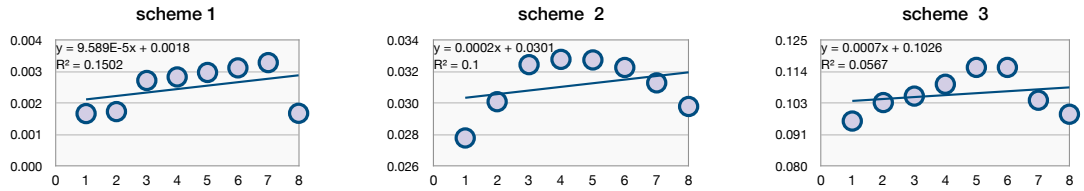

Figure 2. Diagram Scatter Initialization Value x_0 based on Key Pattern Changes.

3.4. Butterfly Effect Test

The butterfly effect test is used to indicate the change in bits in the key input, whether it gives a large change to the output. Suppose that as a comparator key ZZZZZZ and ZZZZZY are selected which has a difference of one bit. The result of the two keys with scheme-1 is obtained by a random number of the first 500 iterations, shown in Figure 3.

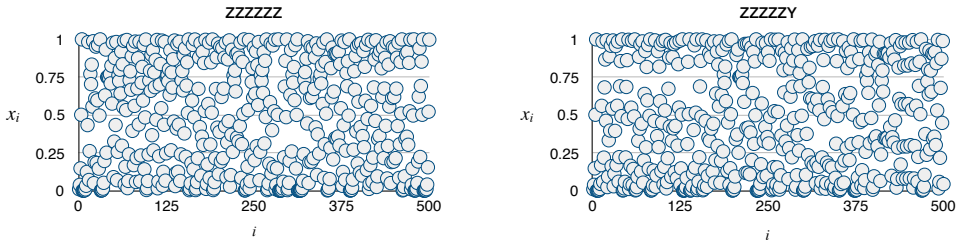

Figure 3. Results of Random Numbers with Different Inputs 1 Bit for Scheme-1.

Visually, the random number generated is very different although the difference in initialization value x_0 is only $0.0000151538 \approx 1.538 \times 10^{-6}$ or in an absolute relative of 0.063%. A minor change in input and a major change in output proves that the 1st scheme has successfully fulfilled the butterfly effect.

The initial value difference of x_0 is 0.0487% in scheme-2. But the rate of change occurs very significantly that appears in the Scatter diagram in Figure 4. So, the scheme-2 has also fulfilled the butterfly effect test.

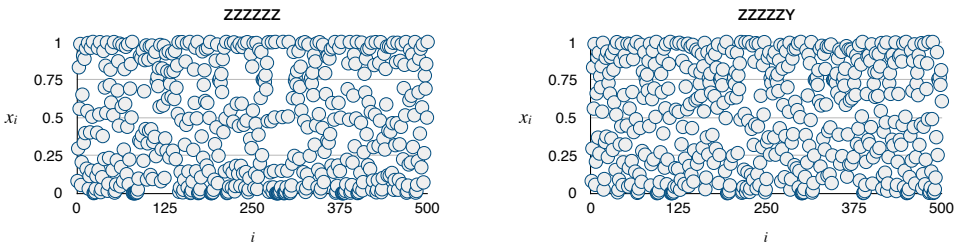

Figure 4. Results of Random Numbers with Different Inputs 1 Bit for Scheme-2.

Significant changes also occur at random values with scheme-3, as shown in Figure 5, although the difference is 0.0399% at input value x_0. So, the scheme-3 also meets the properties of the butterfly effect.

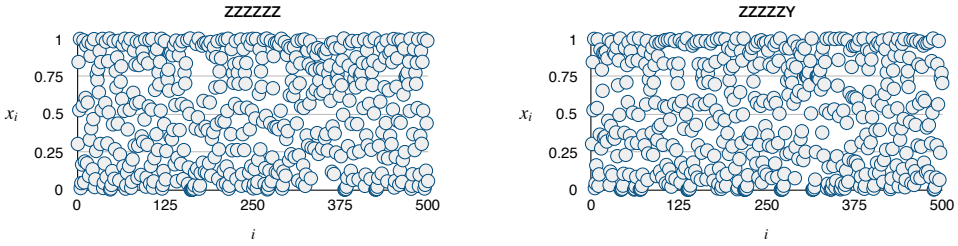

Figure 5. Results of Random Numbers with Different Inputs 1 Bit for Scheme-3.

The three schemes have succeeded in fulfilling the properties of the butterfly effect, thus the existence of the logistics function in generating chaos numbers can be accommodated based on key inputs. Each scheme can be used as a complement to a cryptography algorithm to meet the diffusion properties of Shannon's principle.

3.5. Analysis Of The Algorithm Ability

Each scheme is tested in correlation [10], to detect the connectedness of the random number generated based on the input. While MAPE [11], it is used to find out how massive the difference of random value of key change. Used three variations of the input [12], the first is the same character input, a second input alphabetic characters that revolve around the 26-character alphabet. While the last test, used alphabet, symbols, and numbers.

Calculation of correlation in Table 3 show that there are two values on scheme-1 and scheme-3 which correlation is negative, besides the rest is positive. Cryptographically, the negative value is not too influential, hence it seen how close the value to zero indicating the unrelated two random numbers are generated. In the context of the relations, this same analogy can be used to test the difference of two random numbers generated.

Table 3. 1 Bit differences test with key variations.

Test	Input keys	Correlation value			MAPE		
		Scheme-1	Scheme 2	Scheme-3	Scheme-1	Scheme-2	Scheme-3
1	ZZZZZZ - ZZZZZY	0.1739823	0.04831424	−0.0685288	23.9966	3486.022	13.192
2	fti - ftj	0.0220125	0.05469541	0.00900384	612.172	210.2993	46.6942
3	$4LaT1g4 - $4LaT1g3	−0.011209	0.02911591	0.00971799	64.9543	38.1335	380.158

Overall the correlation value generated by each scheme is within the range of 0.00 - 2.99. Based on [13], the interval shows the strength of a very weak relationship. This condition provides information that 1 bit key input difference, can generate different random numbers on each scheme.

In addition to the correlation and MAPE analysis, we also tested the distribution of random number data using box-plot diagrams.

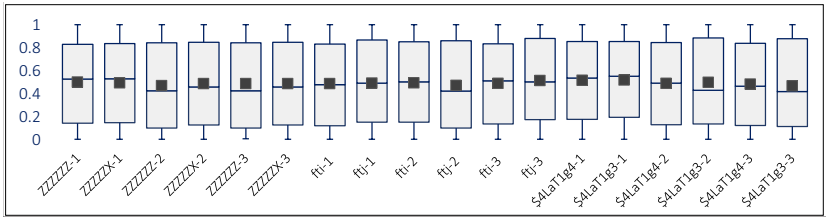

Figure 6. Box Plot Random Number of Each Scheme Output with Various Input Variations

Based on Figure 6, each box has almost the same size in which the upper and lower whisker lines vary slightly, but the maximum value is always close to one and the minimum is near zero. The distribution of data in the chaos range will strengthen the cryptography algorithm if used as a key, this condition will certainly complicate cryptanalyst to be able to search for infinitely many numbers although limited.

4. CONCLUSION

Each designed scheme is capable of providing key input flexibility that can execute domain logistics function values. A 1 bit difference in the key character affects every random number generation, so each key will generate a different random number sequence. Under the key input conditions of the same eight characters, the 1st scheme is better at generating different initialization values than the scheme-2 and schema-3. In addition, the one character reduction of the eight identical characters in the key input does not show a proportional or reverse pattern with the initial x_0 values and the relative error x_0. The resulting random numbers distributed evenly over the chaos range will amplify the algorithm when used as a key in cryptography. This condition will certainly complicate cryptanalyst to be able to search for infinitely many numbers although limited. The three schemes have succeeded in fulfilling the nature of the butterfly effect, thus the existence of the logistics function in generating chaos numbers can be accommodated based on key inputs. Each scheme can be used as a complement to a cryptography algorithm to satisfy the diffusion properties of the Shannon principle.

REFERENCES

[1] Devaney, R.L, 1992, *A First Course in Chaotic Dynamical Systems: Theory and Experiment*, Massachusetts: Addison-Wesley, Boston.
[2] Liwandouw, V. B., & Wowor, A. D., 2015, Kombinasi Algoritma Rubik, CSPRNG Chaos dan S-Box Fungsi Linier dalam Perancangan Kriptografi Block Cipher, *Seminar Nasional Sistem Informasi Indonesia*, Surabaya: Program Studi Sistem Informasi, ITS.
[3] Munir, R., 2011, Enkripsi Selektif Citra Digital dengan Stream Cipher Berbasiskan pada Fungsi Chaotik Logistic Map, *Seminar Nasional dan ExpoTeknik Elektro*, Universitas Achmad Dahlan.
[4] Munir, R., 2012, Analisis Keamanan Algoritma Enkripsi Citra Digital Menggunakan Kombinasi Dua Chaos Map dan Penerapan Teknik Selektif, *Jurnal Ilmiah Teknologi Informasi*, Vol. 10, No. 2, Juli: 89-95, Surabaya: ITS.
[5] Lestari, D. & Riyanto, M.Z., 2013, *Suatu Algoritma Kriptografi Stream Cipher Berdasarkan Fungsi Chaos*, Yogyakarta: MIPA Universitas Negeri Yogyakarta.
[6] Gayathri, P. & Syed Umar & Sridevi, G. & Bashwanth, N. & Royyuru Srikanth., 2017, Hybrid Cryptography for Random-key Generation based on ECC Algorithm *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 7, No. 3, June: 1293-1298.
[7] Krishna, A.R & Chakravarthy, A.S.N. & Sastry, A.S.C.S, 2017, A Hybrid Cryptographic System for Secured Device to Device Communication *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 6, No. 6, December: 2962-2970.
[8] Chandkavathe, V.M, & Bhaskar, R.S, 2016, Optimized Full Parallelism AES Encryption / Decryption, *SSRG International Journal of Electronics and Communication Engineering (SSRG - IJECE)*, Vol. 3, No. 6, June: 14-16.
[9] Chapra, S.C. & Canale, R.P., 2010, *Numerical Methods for Engineers*, Sixth Edition, New York: McGraw-Hill.
[10] Montgomery, D.C. & Runger, G.C., 2014, *Applied Statistics and Probability for Engineers*, Sixth Edition, New Jersey: John Wiley & Sons.
[11] Makridakis, S., Wheelwright, S.C., & McGree, V. E., 1999, *Metode dan Aplikasi Peramalan*, Jilid 1, Jakarta : Erlangga.
[12] Liwandouw, V.B., & Wowor, A.D., 2015, Desain Algoritma Berbasis Kubus Rubik dalam Perancangan Kriptografi Simetris, *Seminar Teknik Informatika dan Sistem Informasi*, 9 April 2015, Bandung: FTI Universitas Kristen Maranatha.
[13] Sarwono, J., 2006, *Metode Penelitian Kuantitatif dan Kualitatif*, Edisi Pertama, Yogyakarta: Graha Ilmu.

ACKNOWLEDGMENT

Thank you to Satya Wacana Christian University Research and Community Service Center (BP3M) for the research funding support through the Internal Obligatory Research scheme in the 2016 fiscal year.

BIOGRAPHIES OF AUTHORS

Alz Danny Wowor is currently a lecturer at the Faculty of Information Technology, Satya Wacana Christian University in Salatiga, Indonesia. He received bachelor and master degree in mathematics and informatics from Satya Wacana Christian University, in 2005 and 2011 respectively. His researches are in fields of Primitive Cryptography, Symmetric Cryptography: Block Cipher and Pseudorandom.

Vania Beatrice Liwandouw is a Master of Cyber Security student at Faculty of Science, Radboud University, Nijmegen, The Netherland. She received her Bachelor degree in engineering informatics at the Faculty of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia. Her research interests are in the Design and Implementation of Symmetric Cryptographic Algorithms.