P- ISSN 1991-8941 E-ISSN 2706-6703 *Journal of University of Anbar for Pure Science (JUAPS)* Open Access
2012,(6), (2) :79-84

New Construction of a Chaotic Generator on the Lorenz Attractor

Adda Ali-Pacha* Naima Hadj-Said* Mohamed Sadek Ali-Pacha* Abdallah M'Hamed**

*University of Sciences and Technology of Oran USTO, ALGERIA
**Institute of Telecommunications Evry- France.

ARTICLE INFO

Received: 00 / 00 /00
Accepted: 00 / 00 /00
Available online: 9/12/2012
DOI: 10.37652/juaps.2012.63243

Keywords:
Cryptography;
Chaos;
stream cipher,
Attractor,
Generator,
Lorenz,
Ali-Pacha.

ABSTRACT

Let it be known that the chaotic phenomena can be obtained from relatively simple systems that are governed by a small number of variables. The system will then be deterministic, although its behaviour is unforeseeable. The chaotic generator hereby suggested is implemented under the 7.0 version of MATLAB software. It makes use exclusively, of the fundamental properties of chaotic systems; that are sensitivity to initial conditions and equations of strange attractor. All is done in order to set up systems with protected transmissions. As a matter of fact and in the long term, the unforeseeable behaviour of such systems is very much related to the extreme sensitivity of initials conditions. Another fundamental property is that the chaotic system is characterized by a strange attractor, within the space of state.

Introduction

The major interest of pseudo random generators of genuine crypto-graphically continuations [Schneier, 96], is that they are perfect for coding. This type of generator generates a flow of bits (stream keys: Codon) k_0, k_1, k_2 , …, k_{m-1} ; which is a continuation of known length , zeros and ones bits. The first term k_0 is called the seed. The recurrence general algorithm:

1. Choose k_0 in M
2. Set $k_{n+1} = f(k_n)$ for n>0

Where f is an adequate application of M within

$$c_i = m_i \oplus k_i \qquad \text{(eq. 1)}$$

This flow is combined by mean of exclusive or, to the flow of data bits (plaintext) m_1, m_2, m_3… m_i; in order to produce the bits flow of transmitted data:

At the receiver, the bits of received data (cryptograms) are combined by mean of exclusive or, to an identical flow of codons, so as to find the bits of transmitted data:

$$m_i = c_i \oplus k_i = (m_i \oplus k_i) \oplus k_i = m_i \qquad \text{(eq. 2)}$$

Flow is said to be random, for this continuation is arbitrary. However, when the continuation comes to an end, the generator does not stop operating.

* Corresponding author at: University of Sciences and Technology of Oran USTO, ALGERIA-.E-mail address: drtaghreed2@gmail.com

The already transmitted sequence is once again reproduced (periodical generating). Hence the pseudo-random attribute (See Figure 1).

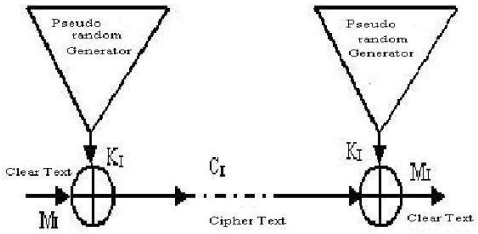

Figure 1: Stream Cipher configuration

The system safety depends entirely on the internal details of the codons generator. If the seed is not changed, the same sequence will always be obtained. This happens to be useful, for the data recovery; thus the seed can also determine the key of the sequence. Placing, the concepts of chaos at the disposal of telecommunications means building a chaotic generator on the basis of strange attractor in order to produce a flow of codon chaotic.

The majority of chaotic attractors (Lorenz attractor in this case) are defined by an irresolvable system of differential equations. To obtain their representations, one need to use approximation methods, whose exacts solutions can never be

obtained; for that reason, one needs an infinite precision. Thus, this implies that the approximations will have a cyclic trajectory because of the finite precision required. The two dimensional graph, representing each differential equation separately seems to present irregularities.

The idea is to use thee irregularities of a differential equation (for instance dz/dt), for the Stream keys generation. Therefore, this work consists of implementing of Ali-Pacha generator[Ali-Pacha and all, 2007].

Chaos

The discovery of the chaotic systems reconciles the apparently paradoxical concepts of chaos and determinism[GLEICK, 1989]. As a matter of fact, systems described by (very simple) equations comply with perfectly deterministic laws and yet, their behaviour is fully unforeseeable. This un-prediction is not random, but, it is due to sensitivity to initials conditions.

- They are deterministic, because objective and precisely measurable and locatable effects determine the continuation of events.
- They are chaotic, because we ignore at all what will occur, despite of the knowledge we have, of all data which determine the events.

Hence, the behaviour of the chaotic systems is unforeseeable i.e. complex, non-periodical, irregular, erratic and of random appearance; even though it is set up from a deterministic mathematical model.

Sensitivity to Initial Conditions

As a matter of fact and in the long run, the unforeseeable behaviour of the dynamic system is closely related to extreme sensitivity to initial conditions, which is a fundamental characteristic of dynamic systems.

It should be emphasised here that a system will react in a completely different way according to initial condition. Consequently, such system, even when all its components are known, it is totally unforeseeable.For it is sensitive to minor initial disturbances[Ali-Pacha and all, 2004]. This sensitivity can be quantified by the positive exponents of Lyapunov.

Strange Attractor

One of the most interesting discoveries of the last decade was that of strange attractor [Devaney, 2004]. These geometrical objects resulting from evolution of chaotic systems characterizing it.

In the plan, they are made out of an infinite continuation of points x_0, x_1, x_2, x_3 ..., x_n …, all depend on the initial value x_0. As the number of points increases, an image is formed in the plan and becomes progressively clearer (Figure 2).

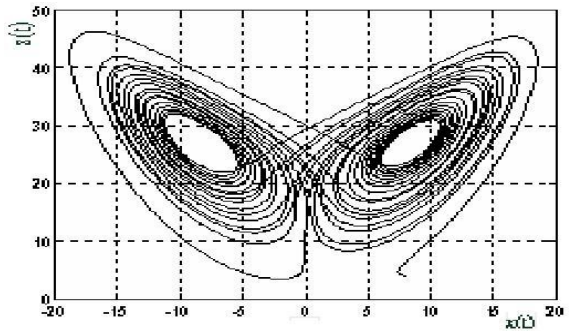

Figure 2: Lorenz attractor in 3D.

This image is neither a curve nor a surface; it is in fact, an intermediate object made up of points with free spaces in between.

The object is described as strange because of its dotted structure and its fractal nature. A different value of x_0 leads to a very other continuation which after a short phase, draws the same image (the state of the system is then steady).

- Anywhere and anytime, one always finds itself on the attractor; it is the foreseeable side of the evolution.
- Where does one find itself exactly on the attractor? It is impossible to answer the question; this is then the unforeseeable side of the evolution.

One calls attractor this form which appears in a repetitive matter (as illustrated in figure 2), independently of the initial conditions or the trajectories.

Lorenz Attractor

The Lorenz attractor is quite an interesting example [Strogatz, 1994] of a strange attractor and of a differential system with a chaotic behaviour for certain values of parameters. Although at the beginning, its inventor had no intention to create a chaotic phenomenon. As a result, this happens to be a

completely random event which has never the same behaviour.

The equations of system of Lorenz are:

$$\begin{cases} \frac{dx}{dt} = s*(y-x) \\ \frac{dy}{dt} = r*x - y - x*z \\ \frac{dz}{dt} = r*y - b*z \end{cases} \quad \text{(eq. 3)}$$

s, r and b are positive real values. Beyond a critical value of the parameter r, the behaviour of the system becomes chaotic. We notice that we are dealing with a nonlinear equation due to the existence of terms like x*z. This system of equation has no analytical solution the case general.

However, resolution of this system (once the 3 parameters s, b, and r parameter are fixed), can only be achieved using numerical approximation method (Euler method or Runge-Kuttad method). We adopted the latter one. There are several alternatives of this method (order 3, 4, 5...), we will be interested in the method of order 4, which calculates the value of the function in four intermediate points.

Bearing in mind that the number of values to be calculated is significant (about 100.000), we implement our model through MATLAB software [Chalabi, 2006], using a Pentium III portable computer.Table 1 shows the evolution of amplitudes of x, y and z in terms of time; for:

1. s = 10,
2. r = 28,
3. b = 8/3,
4. the step h=0.005 and
5. Following initial conditions x_o = 8; y_o = 3, and z_o = 4.

Their respective plotting is represented in Figures 3, 4 and 5. One notices the non-periodical aspect of the 3 curves in interval [0 - 40].

Table 1a: evolution of x, y, z component

Position	*dx/dt*	*dy/dt*	*dz/dt*
0	8.0000	3.0000	4.0000
1	7.7791	3.9273	4.0827
2	7.6134	4.8232	4.1959
3	7.4988	5.6932	4.3376
4	7.4317	6.5423	4.5069
5	7.4088	7.3751	4.7036
6	7.4274	8.1958	4.9281
7	7.4849	9.0079	5.1814
8	7.5790	9.8146	5.4647
9	7.7078	10.6190	5.7802
10	7.8695	11.4220	6.1299
11	8.0625	12.2260	6.5165
12	8.2854	13.0320	6.9428
13	8.5367	13.8400	7.4122
14	8.8153	14.6510	7.9278
15	9.1199	15.4630	8.4935
16	9.4492	16.2740	9.1128
17	9.8020	17.0840	9.7895
18	10.1770	17.8870	10.5280
19	10.5730	18.6820	11.3300
20	10.9870	19.4610	12.2020
21	11.4190	20.2210	13.1440
22	11.8670	20.9530	14.1610
23	12.3270	21.6510	15.2540
24	12.7980	22.3050	16.4230
25	13.2770	22.9060	17.6700
26	13.7600	23.4440	18.9930
27	14.2440	23.9070	20.3880
28	14.7250	24.2850	21.8520
29	15.1980	24.5660	23.3790
30	15.6600	24.7380	24.9590
31	16.1050	24.7910	26.5830

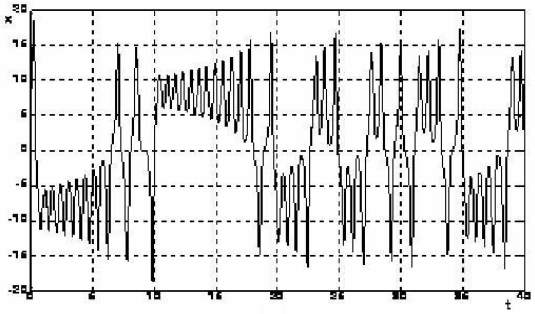

Figure 3: x- Curve attractor.

Table 1b: evolution of x, y, z component

Position	*dx/dt*	*dy/dt*	*dz/dt*
32	16.5270	24.7150	28.2380
33	16.9220	24.5020	29.9100
34	17.2830	24.1450	31.5810
35	17.6060	23.6410	33.2350
36	17.8850	22.9870	34.8510
37	18.1150	22.1870	36.4100
38	18.2910	21.2440	37.8920
39	18.4090	20.1680	39.2790
40	18.4650	18.9710	40.5520
41	18.4580	17.6660	41.6960
42	18.3860	16.2720	42.6970
43	18.2470	14.8080	43.5460
44	18.0420	13.2950	44.2360
45	17.7730	11.7530	44.7650
46	17.4410	10.2040	45.1320
47	17.0500	8.6694	45.3430
48	16.6050	7.1667	45.4040
49	16.1080	5.7131	45.3260
50	15.5670	4.3230	45.1200
51	14.9860	3.0083	44.8000

52	**14.3710**	**1.7778**	**44.3810**
53	**13.7280**	**0.6380**	**43.8770**
54	**13.0640**	**-0.4074**	**43.3030**
55	**12.3830**	**-1.3570**	**42.6740**
56	**11.6920**	**-2.2114**	**42.0020**
57	**10.9950**	**-2.9729**	**41.2990**
58	**10.2960**	**-3.6453**	**40.5770**
59	**9.6017**	**-4.2332**	**39.8450**
60	**8.9142**	**-4.7422**	**39.1100**
61	**8.2371**	**-5.1782**	**38.3810**
62	**7.5735**	**-5.5476**	**37.6620**
63	**. . .**	**. . .**	**. . .**

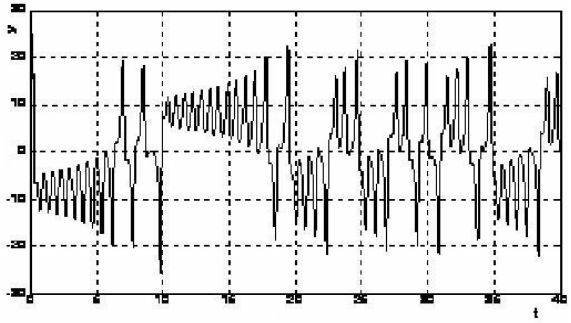

Figure 4: y- Curve attractor.

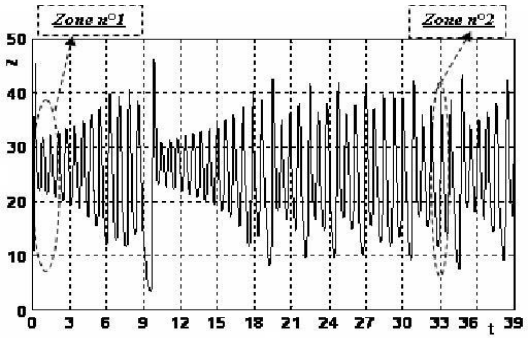

Figure 5: z- Curve attractor.

This confirms the "chaotic" behaviour of the system solutions. The implemented program allows us to write down in a file, the values of the 3 continuations given above, according to the initial conditions and the variables defined earlier.

Chaotic Generator with based on the Lorenz Attractor

The suggested generator in enabled us to give a first approximate solution of what we expected for such a chaotic generator. In this work, another solution could be possible since we program with the Matlab software. Hence, the exact solution of a differential equation of the type (eq. 3), is a continuous function. Computers however, can only provide a finite number of numerical values.

It all starts, with a preliminary choice of finite number of points x_i along the interval [a - b]. This is called a discretization or a grid of the geometrical interval (i.e. the segment [a – b]).

We shall restrict calculation to an approximate calculation of solution in these points. The choice of point's x_i is obviously crucial to the validity of the obtained numerical solution. The grid permits us to represent the solution in a precise manner. As this solution is initially unknown, we proceed by techniques of adaptation of grid, as a matter of fact. We determine a first solution through a grid network.

From this first calculation we can deduce significant variation of the solution. We can then improve resolution of the grid network in these zones in order to make it simple.

Thus, this work consists of approximating the curve Z(t) = dz/dt for example, by using an interpolation polynomial. Mathematical formula of this curve materializes the generator itself.

Unfortunately, neither the programming language (Matlab) nor the machine (microcomputer) gives such a mathematical polynomial formula in all the range. For that, we recommend to split up the curve into finites distinct intervals (zones), while seeking for each section of curve (zone) its corresponding interpolation polynomial.

Let us study two finished intervals of the layout as it is shown in figure 5. One limited oneself in this example on an interval of [0 - 2] for zone 1 and the interval [33 - 35] for zone 2 this to decrease the errors due to the interpolation polynomial.

4.1 Polynomial Interpolation for Area 1

Let us zoom zone 1 in Figure 6a.

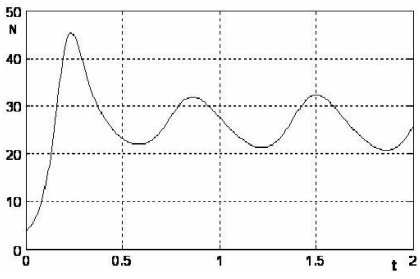

Figure 6a: Polynomial Interpolation-1

We interpolate the curve Z(t) in order to obtain an approximating curve which possesses the same shape as in Figure 6b.

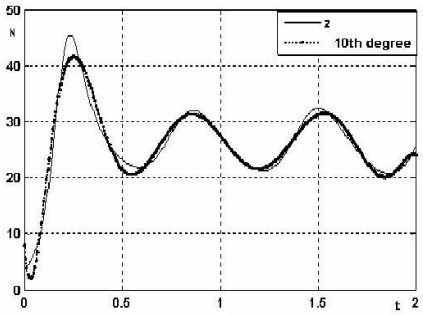

Figure 6b: Polynomial Interpolation-1

By selecting only the approximated curve, the programming language Matlab enables us to find the coefficients of the polynomial of interpolation P_1 (X) of this curve in interval[0 - 2], (e=10):

$$P_1(x) = 5.4\,e^2x^{10} - 6.7\,e^3x^9 + 3.6\,e^4x^8 - 1\,e^5x^7 + 1.8\,e^5x^6 - 2\,e^5x^5 + 1.3\,e^5x^4 - 4.8\,10^4x^3 + 8.5\,e^3x^2 - 4.1\,e^2x + 8$$

The greatest value of the polynomial $P_1(x)$ in the range [0 - 2] is $Z_{max} = 41.5997$.

In order to have normalised amplitudes, we must divide the polynomial $P_1(x)$ by $Z'_{max} = 41.6$.

Figure 6c, shows the plotting of the interpolation polynomial $P_2(x)$ corresponding to zone 1 in the range [0 - 2] *(e=10),*

$$P_2(x) = 13x^{10} - 1.6\,e^2x^9 + 8.6\,e^2x^8 - 2.5\,e^3x^7 + 4.3\,e^3x^6 - 4.7\,e^3x^5 + 3.1\,e^3x^4 - 1.1\,e^3x^3 + 21x^2 - 9.9x + 0.19 \quad \text{(eq. 4)}$$

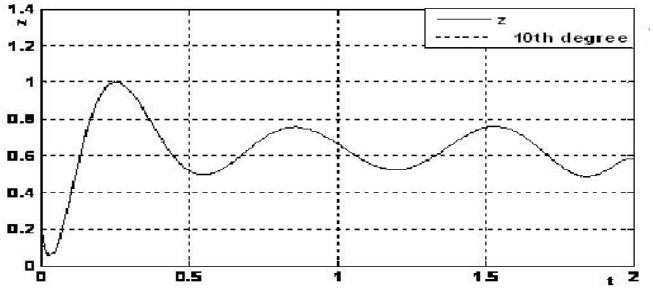

Figure 6c: Polynomial Interpolation-1

polynomial Interpolation for Area 2

Zooming the zone 2 which ranges in the interval [33 - 35], as depicted in Figure 7a:

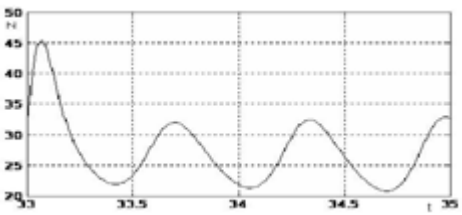

Figure 7a: Polynomial Interpolation-2

To find its interpolation polynomial it is necessary to consider the initial conditions of position 33 of table 1, hence:

$$\begin{cases} x(1) = 16.9220, \\ y(1) = 24.5020 \\ z(1) = 29.9100 \end{cases} \quad \text{(eq. 5)}$$

We interpolates curve Z(t) of the zoomed figure in order to obtain an approximate curve which has the same shape (see Figure 7b).

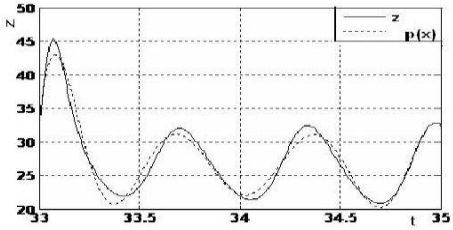

Figure 7b: Polynomial Interpolation-2

By selecting only the approximated curve, the programming language Matlab enables us to find the coefficients of the interpolation polynomial $P_1(x)$ of this curve in the interval of [33 - 35] (e=10):

$$P_1(x) = 5.8e^2x^{10} - 5.8e^3x^9 + 2.5e^4x^8 - 5.5e^4x^7 + 7e^4x^6 - 4.8e^4x^5 + 1.3e^4x^4 + 2.2e^3x^3 - 2e^3x^2 + 2.6e^2x + 34 \quad \text{(eq. 6)}$$

The greatest value of the polynomial $P_1(x)$ in the interval [33 -35] is Z_{max} =43.0019. We divides the polynomial $P_1(x)$ by Z'_{max} =43.002 in order to have normalised amplitudes.

We display this last curve in figure 7c, and we find the interpolation polynomial P_2 (X) corresponding to zone 2 of the interval of [33 - 35], (e=10):

$$P_2(x) = 13x^{10} - 1.4*e^2x^9 + 5.7*e^2x^8 - 1.3*e^3x^7 + 1.6*e^3x^6 - 1.1*e^3x^5 + 3.1*e^2x^4 + 52x^3 - 47x^2 + 6x + 0.78 \quad \text{(eq. 7)}$$

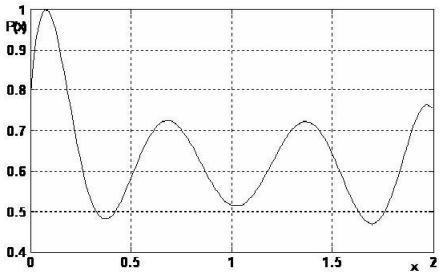

Figure 7c: Polynomial Interpolation-2

Suggested Chaotic Generator

We suggest a solution to exploit all the plotting, by splitting curve Z(t) = dz/dt in a number (preferably power of 2) of zones (of finished intervals). Each zone has its own proper initial conditions. This gives place to a polynomial of interpolation of zone. The construction of the chaotic generator proposed is done with the following model:

Initially, we choose a zone in which we associate its interpolation polynomial, and on the basis of its seed and sampling step, we determine a number (preferably power of 2) of normalised value of this polynomial. These values will then be coded according to a specific quantification.

The polynomials of interpolations of the zones are stream keys generators, they can be established in the architectures specialized, like the systolic one or the FPGA under language VHDL.

Conclusion

Thus, the chaotic generator suggested, exploits the fundamental properties of the chaotic systems, which are the sensitivity to initial conditions and the equations of strange Lorenz attractor. It is implemented by means of the version 7.0 of MATLAB software. This generator is built in order to set up secure systems in continuous transmissions.

References

[1] J. Gleick(1989), 'chaos theory', Albin Michel 1989.

[2] S. H. Strogatz (1994), "Nonlinear Systems and Chaos", Perseus publishing 1994.

[3] B. Schneier (1996)," Applied Cryptography-Protocols, Algorithms and Source Code in C", John Wiley & Sounds, Inc, New York, Second Edition, 1996.

[4] A. Ali-Pacha A, N. Hadj-Said, B. Belmekki, A. Belghoraf, «Chaotic Behaviour for the Secrete key of Cryptographic System», Chaos, Solitons& Fractals Journal , Volume 23/5 pp. 1549-1552. Available online. October 2004.

[5] Robert L. Devaney (2004), "Differential Equations, Dynamical Systems, and an Introduction to Chaos", Elsevier Academic Press, 2004.

[6] R. Chalabi, H. Hakim (2006), "Study and Implementation of Chaotic Attractor for their Applications to Cryptography" PFE - USTO 2006.

[7] A. Ali-Pacha, N. Hadj-Said, A. M'Hamed, A. Belghoraf,(2007) «Lorenz's Attractor Applied to the Stream Cipher (Ali-Pacha Generator)», Chaos, Solitons& Fractals Journal , Volume 33/5 pp.1762-1766. August 2007.

تركيب جديد لمولد عشوائي بناء على جاذب "لورينز"

عداء علي باشا ، نعيمة حاج سعيد ، محمد صادق علي باشا ، عبدالله محمد

, Email: a.alipacha@gmail.com

الخلاصة:

ينبغي ان يكون من المعلوم أن الظواهر العشوائية يمكن الحصول عليها من منظومات بسيطة نسبيا يحكمها عدد قليل من المتغيرات. وسيكون النظام الناتج محددا، غير أن تصرفاته لا يمكن التنبؤ بها. لقد تم تنفيذ المولد العشوائي المقترح هنا بواسطة برنامج "ماتلاب" النسخة السابعة. وهو يقوم بشكل كبير على استخدام الخواص الرئيسية للمنظومات العشوائية وهي الحساسية للشروط الابتدائية ومعادلات الجاذب غير المألوف. حيث تم اجراء ما يلزم لبناء نظم ذات قدرات تواصل محمية. وفي الحقيقة البعيدة المدى فإن السلوك الغير قابل للتنبؤ في هكذا منظومات يبقى عائدا من حيث المبدأ إلى الحساسية المفرطة للشروط الابتدائية. والخاصية الأساسية الأخرى هي اعتماده على خواص الجاذب غير المألوف في فضاء الحالات.