



US 20160380760A1

(19) **United States**

(12) **Patent Application Publication**
CAMPOS CANTON et al.

(10) **Pub. No.: US 2016/0380760 A1**

(43) **Pub. Date: Dec. 29, 2016**

(54) **PSEUDO-RANDOM BIT GENERATOR
BASED ON MULTIM-MODAL MAPS**

Publication Classification

(71) Applicant: **Instituto Potosino de Investigación
Científica y Tecnológica A.C.**, San
Luis Potosi (MX)

(51) **Int. Cl.**
H04L 9/00 (2006.01)
H04L 9/06 (2006.01)
G06F 7/58 (2006.01)

(72) Inventors: **Eric CAMPOS CANTON**, San Luis
Potosi (MX); **Moises GARCIA
MARTINEZ**, San Luis Potosi (MX)

(52) **U.S. Cl.**
CPC **H04L 9/001** (2013.01); **G06F 7/582**
(2013.01); **H04L 9/0668** (2013.01)

(73) Assignee: **Instituto Potosino de Investigación
Científica y Tecnológica A.C.(40%)**,
San Luis Potosi (MX)

(57) **ABSTRACT**

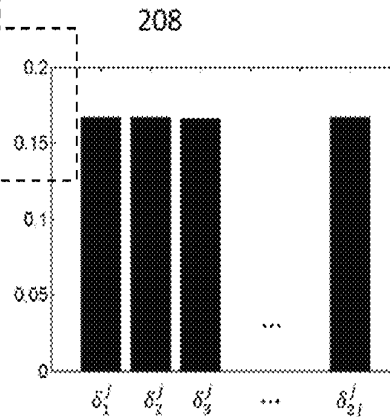
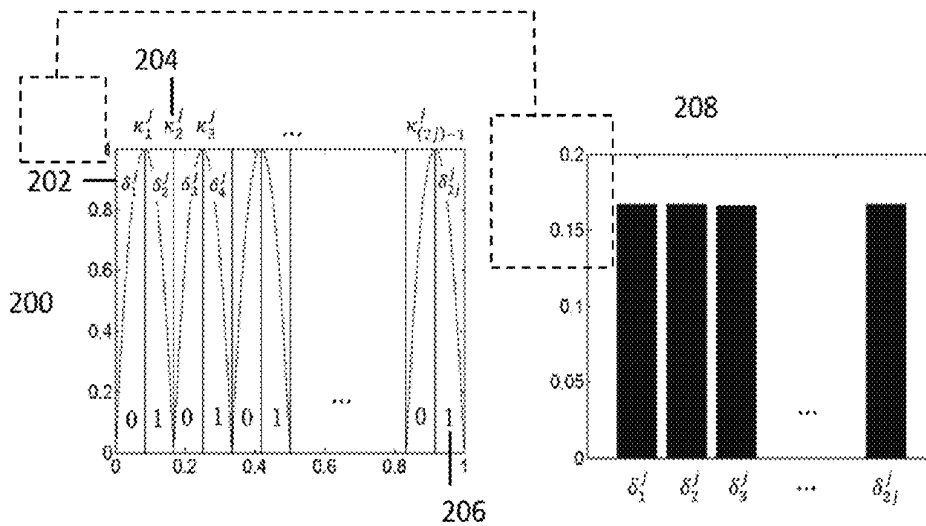
(21) Appl. No.: **15/190,616**

The present invention is related with a computer-implemented method for generating a pseudo-random bit generator including the steps: a. Defining a multi-modal map by the equation: $f_p = \beta(d_{r+1} - x)(x - d_r)$, $x \in \mathbb{X}_r$; b. Set the value of $k \in \mathbb{N}^+$, and obtaining the values of β_j , for j =from at least 1, to the final value k by the following equations: $\beta_1 = 4k$; $\beta_j = (j)(\beta_1)$; for $2 \leq j \leq k$ and taking the values of β_j and split the space into $2j$ regions δ_{1^j} , to δ_{2^j} which are determined by values k_1^j to $k_{(2^j)-1}^j$.

(22) Filed: **Jun. 23, 2016**

Related U.S. Application Data

(60) Provisional application No. 62/184,600, filed on Jun. 25, 2015.



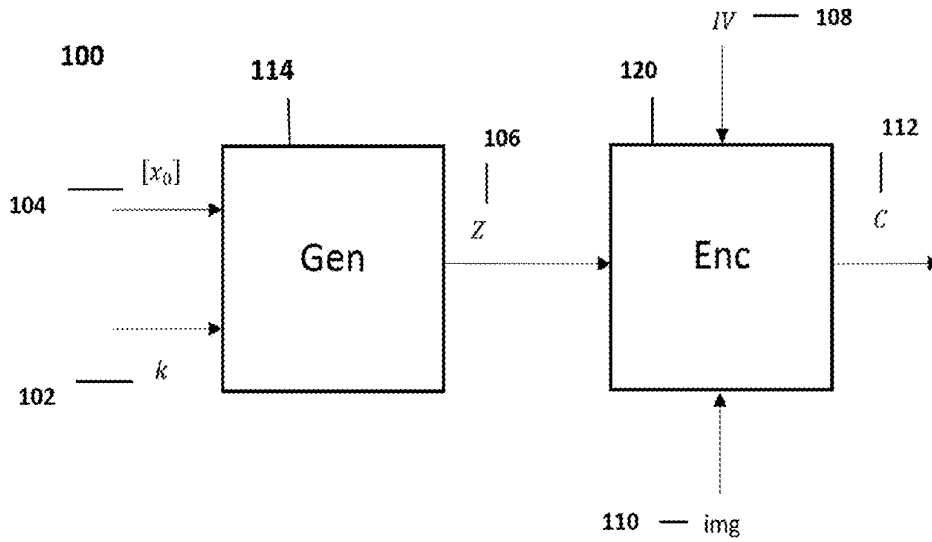


FIG. 1

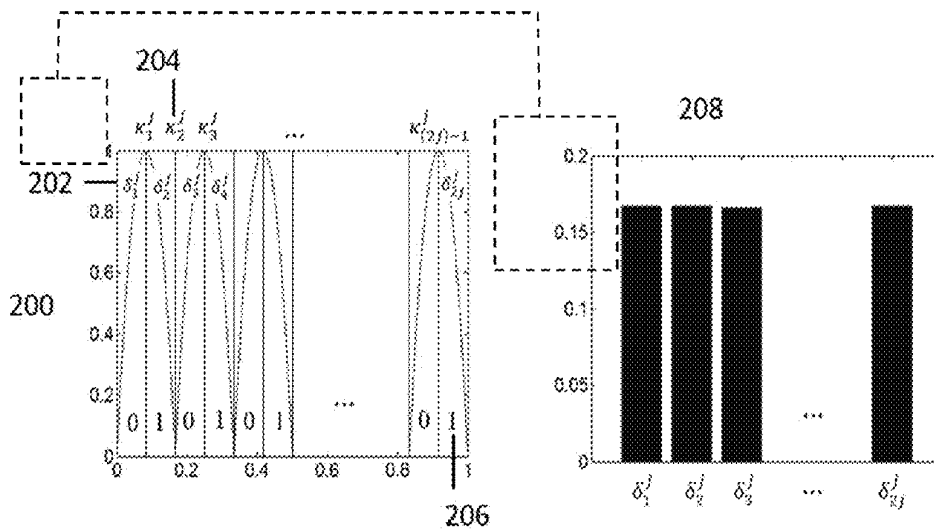


FIG. 2

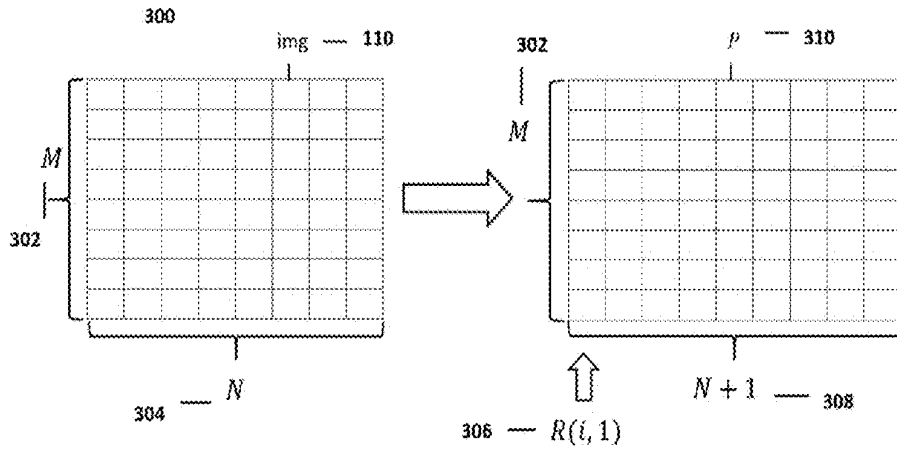


FIG. 3

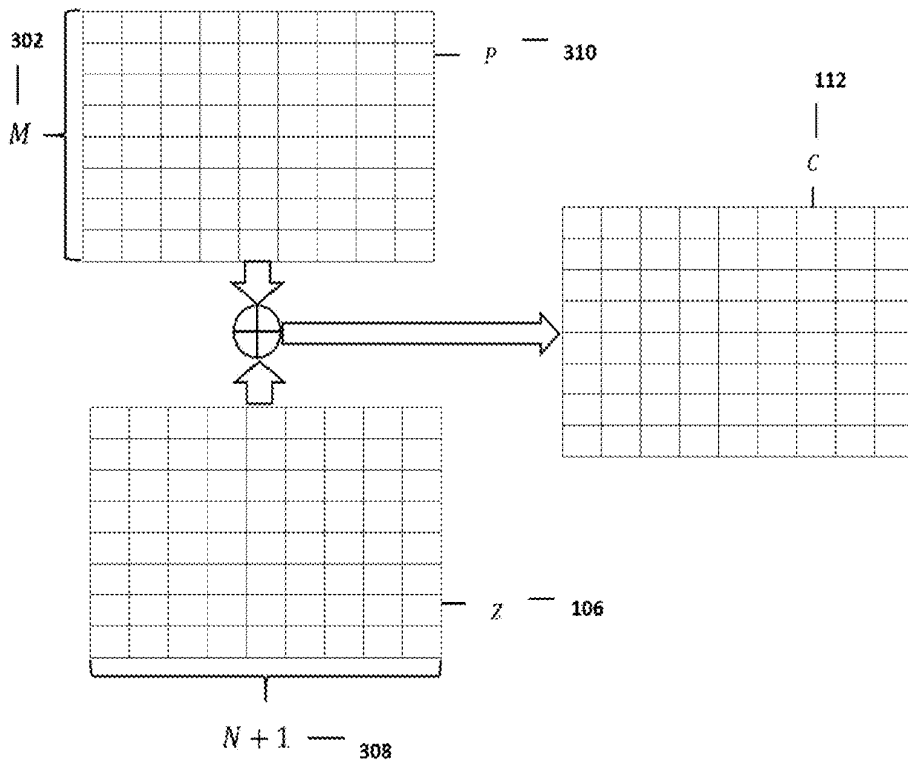


FIG. 4



FIG. 5

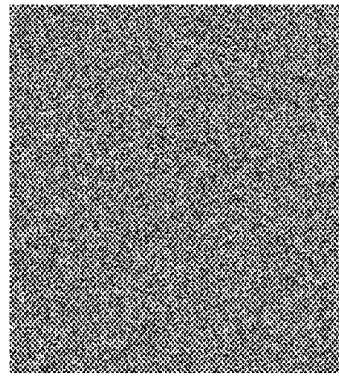


FIG. 6

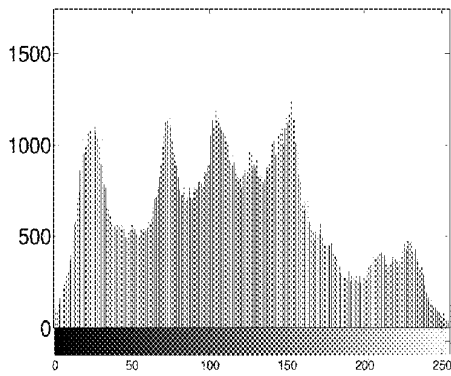


FIG. 7

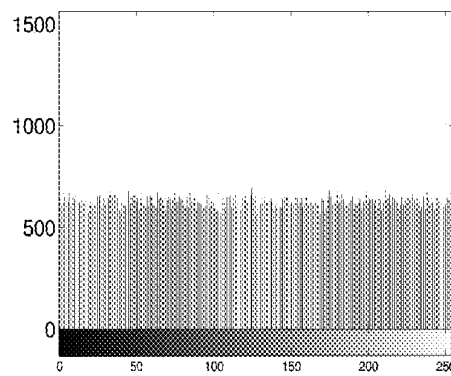


FIG. 8

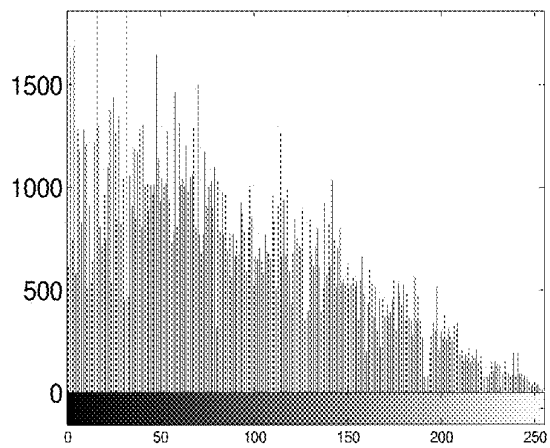


FIG. 9

**PSEUDO-RANDOM BIT GENERATOR
BASED ON MULTIM-MODAL MAPS**

CROSS-REFERENCE TO RELATED
APPLICATIONS

[0001] This application claims the benefit of priority to U.S. Provisional Application No. 62/184,600, filed Jun. 25, 2015, the contents of which are incorporated herein by reference.

FIELD OF THE INVENTION

[0002] The instant invention refers to the secure informatics field, particularly, to a pseudo-random bit generator which is produced through dynamic systems in discrete time unidimensional multimodal, called k-modal maps. Furthermore, the impact of using these pseudo-random sequences in stream ciphering resulting in a better ciphering quality correlated with the modal number of the chaotic mapping, is presented. The proposed algorithm can cipher through adding a stochastic vector to the plane image, that is, the same set of keys generates a ciphered image each time different from the ciphered plane image. Also, the present invention relates to the field of computer dynamics and, more particularly, to structures dynamically configurable of linear basic logic gates; which they are able to change their response in accordance with the threshold reference signals, to compensate these signals in order to produce different logic gates. These dynamic logic gates would support the development of logic chips for next-generation computers and their applications in new electronic devices.

[0003] This invention comprises at least one application of an algorithm based on chaos mathematics (chaotic dynamical systems), which are correlated to number of modals in a chaotic map. The main applications are optimizing quality encryption from objects employing sequences from a bit string cipher, which come from a pseudorandom generator, where such sequences are obtained from a multimodal uni-dimensional discrete dynamical systems called k-modal maps.

[0004] The implementation in optimizing the quality encryption is performed by adding a stochastic vector to the plain image (probabilistic uniform function), this is, and the same set of keys generates a different cipher image each time it is used.

BACKGROUND OF THE INVENTION

[0005] A discrete-time dynamical system is given as follows: $x_{n+1}=f(x_n)$, $n=0,1,2, \dots, N$, where $x_n \in \mathfrak{X}$ and x_0 is the initial condition, such dynamical system is usually referred to as map, as it is fully determined by its right hand side. The simplest maps are the so-called uni-modal maps like the logistic and tent maps, while their generalization, the so-called multi-modal or k-modal maps may present even more rich dynamical behaviors.

[0006] Definition 1: A map $f: \mathfrak{X} \rightarrow \mathfrak{X}$ is called as the k-modal one, if it is continuous on \mathfrak{X} and it has k critical points denoted by c_0, c_1, \dots, c_{k-1} in \mathfrak{X} . Moreover, there exist intervals \mathfrak{X}_i , $i=0, \dots, k-1$, $\mathfrak{X}_{i-1} \xrightarrow{k} \mathfrak{X}_{i-1} = \mathfrak{X}$, such that $\forall i=0, \dots, k-1$, it holds $c_i \in \mathfrak{X}_i$ and $f_{\beta}(c_i) > f_{\beta}(x) \forall x \in \mathfrak{X}_i$ and $x \neq c_i$, where β is a parameter.

[0007] The case $k=1$ will be further simply referred as to the so-called uni-modal map.

[0008] To be more specific, $\mathfrak{X} := [0,1]$ and recall that a critical point c of the continuous piecewise smooth map $f(x): \mathfrak{X} \rightarrow \mathfrak{X}$ is $c \in \mathfrak{X}$ where f is differentiable and $f'(c)=0$.

[0009] A critical point c occurs for $f'(c)=0$ or $f'(c)$ does not exist. But continuous smooth maps always present $f'(c)=0$. The above definition does not constraint a function to have only k critical points. However it only considered those that are local maximum on a subinterval.

[0010] Definition 2. Random Number Generator (RNG) is a computer or hardware device designed to produce sequences of numbers without an order apparently.

[0011] Definition 3. Pseudo-Random Number Generator (PRNG) is an algorithm that produces a sequence of numbers with an excellent approximation to a set of random numbers.

[0012] Definition 4. Cryptographically Secure Pseudo-Random Number Generator (CSPRNG) is a pseudorandom number generator (PRNG) with features that make it suitable for use in cryptography.

[0013] Some patents that have implemented this type of significant expertise in the development of new technologies, are depicted herein below:

[0014] Typically a Pseudo-Random Bit Generator is based on a Linear Feedback Shift Register (LFSR), this generator consists of a group of flip-flops and exclusive OR, which allow some bits in the register to be true; in other words allowable in the register and selected bits are used, for example, scientific instruments used to measure the intensity of the solar radiation reflected by the earth in various directions and spectral bands; digital cameras are conventionally gathered in spectral bands at different angles for generating images with different wave lengths; the critical points of the above definition, allowing to generate an array representing entries image generation, the function of various wavelengths for these purposes.

[0015] A main feature of the proposed invention is that describes an algorithm for generating pseudo-random bits sequences, the usefulness of these sequences includes creating encryption keys and generating a chain-key, which can be equated to addresses spectral bands like LFRS operate; ie holistically, the generator is based on a dynamic discrete time system whose implementation is more versatile than that described above.

[0016] In the prior art are some patents related to applications that are contained in this application, such as:

[0017] The U.S. Pat. No. 8,738,675B2 patent (Random Number Generator using chaos in continuous time) relates to a method of generating random numbers based on continuous-time chaotic oscillators; where compensation frequency is employed to maximize the statistical quality of an output stream where the proposed circuit can be realized in integrated circuits and the use of continuous chaos in generating random numbers with a very high yield; another invention relates to Patent CN101364171B (real random numbers dynamic generator) relating to a generator, which employs data entered to Von Neumann validators to adjust the output bits by means of chaos conversion algorithms through discrete components and block cipher; especially using the Lyapumov index for the transformation of the block, as a way to control the chaos within a set range and the use of so called S-box that influences the transformation of encrypted blocks;

[0018] The patent U.S. Pat. No. 6,954,770 (Random Number Generator) comprising an oscillator with an output

signal dependent on a random source which is not based on chaos theory, a sampling device for sampling the output signal of the oscillator to obtain a sampled oscillator output, and a fixed frequency clock driven by linear feedback shift register (LFSR) communicatively coupled to the sampling device through a digital gate to receive the output of the sampled oscillator, and to provide a random number in a LFSR output. Furthermore, the random number generator may comprise an optional communicatively coupled to the LFSR, to read the random number mixing function, and to insert the random number in an algorithm in order to obtain a robust random number; however, this technology depends only on the two oscillators and possible responses that keep only presents these two output options.

[0019] The above patents are useful in cryptography for generating keys/cipher/data input in symmetric ciphers. A set of patents that are related architectures chaotic computing logic gates based on nonlinear elements, while the present invention provides dynamically configurable structures linear basic logic gates are: U.S. Pat. No. 8,091,062, U.S. Pat. No. 7,973,566, U.S. Pat. No. 7,924,059, U.S. Pat. No. 7,863,937, U.S. Pat. No. 7,415,683, U.S. Pat. No. 7,096,437, U.S. Pat. No. 7,453,285, U.S. Pat. No. 7,925,814, U.S. Pat. No. 7,925,131 and US 2010/0219858 applications.

[0020] All of the above patents represent antecedents either input data, operation data or output data, but the type of related inventions contained in this application are those that relate to random number generators, where his output could be reproduced when the initial condition is known. A variety of such systems, eg U.S. Pat. No. 8,023,649B2 and U.S. Pat. No. 7,925,014B2 which proposed algorithms pseudo-random generators based on cellular automata.

[0021] The pseudo-random generator proposed in this patent is based on the use of dynamic systems due to the intrinsic properties of these systems as are the sensitivity to initial conditions, complex and deterministic behavior, ergodicity, etc.

[0022] It includes implementations of dynamic continuous time systems, similar to those of the US2014101217A1, EP2758510A1 and U.S. Pat. No. 7,170,997B2 patents, which required the use of numerical methods for their evolution in time when they are implemented by software, however when implemented with electronic circuits or Field Programmable Gate Array (FPGA) is possible to obtain its evolution through hardware.

[0023] In the prior art there are pseudo-random generators based on dynamic discrete-time systems such as US2006251250A1 patent proposes a pseudo-random generator based on mapping systems Hénon and Lozi, the generation method consist on repeatedly iterate the system to generate the array of bits 1 or 0 whose possible dimension is $1 \times N$, each iteration is converted to integer and finally using module 2 each iteration becomes a bit.

[0024] The S6064738A patent uses a mapping of two-dimensional (Baker Mapping) to build an array of bits with dimensions $N \times M$, which performs a permutation of pixels, to exchange pixel places is important to note that performing this step the histogram of the image does not change, in addition, an extension of the mapping to three dimensions is made and then the value of the pixels is replaced so that in this way the distribution of pixels is changed, as if it were a function $f(x, y)$ where $x \in N$ and $y \in M$.

[0025] In U.S. Pat. No. 6,014,445A Patent Chebyshev map used to generate sequences with real values and has four methods in order to "binarizing" time series;

[0026] a) The first method is based on the use of a threshold

[0027] b) The second methodology the absolute value is considered for each iteration, and then it takes its binary representation.

[0028] c) The third methodology it is a generalization of the second methodology such that a scaling-dependent interval where the system evolves is performed, so if the interval is $[0,1]$ then the methodology 2 and 3 are equal.

[0029] d) In the fourth methodology it is considered M thresholds (these thresholds are the keys) so that each real value of the time series is evaluated by these thresholds and from a real value are obtained M bits, finally combine these M bits by operations XOR and obtain 1 bit per iteration.

[0030] The US2013129088A1 patent proposes a pseudo-random generator based on the combination of multiple discrete time dynamical systems, specifically using the cyclic shift map, for this defining k identical maps which are iterated independently then combined by XOR, so that for any iteration n it is previously required to iterate the k maps.

[0031] Finally, there have been proposals generators based on the following types:

[0032] First, using chaotic uni-modal maps, some of the patents disclose bit-stream pseudorandom generator using only a logistic map;

[0033] Second, using a bit string generated through a chaotic system, which is potentially unsafe because the output may leak some information about the chaotic system.

[0034] Third, by the use of two chaotic maps which are combined to obtain a complex bit-stream.

Objective of the Invention

[0035] The generator proposed pseudo-random bits is based on at least one pair of chaotic maps linear fragmented, which are iterated independently and are generated bit-stream by comparing the outputs of these chaotic maps and where a chaotic map only multimodal that can be used replace the two chaotic maps uni-modal for generating pseudo-random sequences.

[0036] This patent proposes the use of discrete-time dynamic multimodal systems (k-modal), which makes it possible to obtain k different time series with the advantage of using only a map, converting the time series real is through $2k$ thresholds, making each modal capable of producing a zero or a one, and finally the k binary strings are combined by XOR.

DESCRIPTION

[0037] The characteristic of this novel method for generating binary sequences through a pseudo-random bit-stream generator based on chaos are clearly shown in the following description and the accompanying figures, and a description of that, where follow same reference signs to indicate the parts and figures shown:

BRIEF DESCRIPTION OF THE FIGURES

[0038] FIG. 1 is a general block diagram view of the pseudo-random bit generator based on chaos, which implements the method for generating binary sequences;

[0039] FIG. 2 shows a block diagram describing the pseudo-random generator based on multimodal maps, which implements the method for generating binary sequences;

[0040] FIG. 3 is a block diagram for describing a pre-process for encryption, as an application example of the method for generating binary sequences through a pseudo-random bit generator based on chaos;

[0041] FIG. 4 is a block diagram describing the encryption process, as an application example of the method for generating binary sequences through a pseudo-random bit generator based on chaos;

[0042] FIG. 5 It is a conventional image that has the distribution of pixels to encrypt and preparing to use the method to generate binary sequences through a pseudo-random bit generator based on chaos;

[0043] FIG. 6 is the representation of the conventional image of FIG. 5 with encryption with a key x_1 , those obtained in the method for generating binary sequences through a pseudo-random bit generator based on chaos;

[0044] FIG. 7 is the view of the histogram of the conventional image of FIG. 5, which it has the distribution of pixels to encrypt and preparing to use the method to generate binary sequences through a pseudo-random bit generator based on chaos;

[0045] FIG. 8 is the view of the histogram of the conventional image of FIG. 7 with encryption with a key x_1 , those obtained in the method for generating binary sequences through a pseudo-random bit generator based on chaos; and

[0046] FIG. 9 is the view of the histogram obtained from differences in encrypted histograms in FIGS. 7 and 8.

GENERAL DESCRIPTION OF THE NOVELTY OF THE INVENTION

[0047] The method for generating binary sequences through a chaos-based pseudo-random bit generator, broadly it involves the use of at least one pair of multimodal maps or k-modal, obtained from the previous definition of at least two uni-modal maps (map family), where the novelty is that the pseudo-random generators existing use combining at least two dynamic systems; in the present invention the generator employs a single system, which is capable of producing at least two systems (system family) for combining between them and more efficient encryption using at least one key for each generated system and apply multiple processes as for example the encrypted image.

[0048] This pseudo-random bit generator and the stream cipher are suitable for use in imaging, document, audio and/or video encryption, multimedia files, security Industrial Control Systems (ICS), and other security appliances.

[0049] NOTE: The examples presented herein are merely declarative, but not limiting, so listed only to illustrate the application of the method and the novelty of its implementation in logical processing units. Based on the above figures, the pseudo-random bit generator chaos-based implement the following blocks in a logical processing unit:

[0050] BLOCK 1 (Gen).-SEQUENCE GENERATION. During this stage, a set of k independent sequences is generated $\{x_1, x_2, x_3, x_4, \dots, x_k\}$ where each sequence is produced using the dynamic system by a logic unit processing (computer, electronic tablet, etc.), then the k sequences are combined to obtain a single sequence (Z) and this will be the generator output of pseudo-random bits based on chaos (Output block 1);

[0051] Where the generator is implemented in logic processing unit to generate at least a multi-modal map is defined by the number k defines the maximal numbers of modals in a family \mathfrak{S} and the interval $\mathfrak{S}=[a,b]$ is divided between k subintervals $\mathfrak{S}_0=[d_1, d_2), \dots, \mathfrak{S}_{k-1}=[d_{k-1}, d_k]$, then f_{β} the system is a piecewise function by k uni-modal maps. The parameterized family Als: is defined by the following piecewise function:

$$f_{\beta}=\beta(d_{r+1}-x)(x-d_r), x \in \mathfrak{S}_r \quad (1)$$

[0052] Where $d_r=r/k$, ($r=0,1,2 \dots, k-1$), k is the number of modals, $\beta=\beta(k,\gamma)$ is the bifurcation parameter, $\gamma=1/k$ is the carrying on capacity. To obtain the maximum value of β with k modals there is a direct relationship, $\beta_{max}=4k/\gamma$.

[0053] And taking the values of β_j and split the space into 2j regions δ_1^j , to δ_{2j}^j which are determined by values to κ_1^{j-1} to $\kappa_{(2j)-1}^j$.

[0054] In a preferred embodiment, said method further comprises the steps of: iterating the system $x_n^j=f_{\beta_j}(x_n)$ and depending on which region evolves the represent the value of 0 or 1 of the binary sequence ζ^j ,

[0055] generate a k number of chaotic time series by $x_n^j/32$ $f_{\beta_j}(x_n)$ and each one produces a binary sequence ζ^j , mixing each of said binary sequence for generating a final sequence Z is as follows:

$$Z=\zeta^1 \oplus \zeta^2 \oplus \dots \zeta^k$$

[0056] The invention, further comprises a stream cipher derived from the pseudo-random bit generator, said stream cipher comprising:

$$C_1=P_1 \oplus Z_1 \oplus IV$$

$$C_i=P_i \oplus Z_i \oplus C_{i-1}$$

[0057] where c_i is the ith pixel of the cipher image c **112**, P_i is the ith pixel of the augmented image P, wherein the value of $P_1=IV(1,1)$, Z_i is the key stream **106**, $IV \in \{0,1, \dots, 255\}$ is an initialization vector **108**, and \oplus is the operation XOR bit by bit.

[0058] As a further embodiment, the instant invention includes a pseudo-random bit generator comprising a multimodal chaotic map defined as:

$$f_{\beta}=\beta(d_{r+1}-x)(x-d_r), x \in \mathfrak{S}_r$$

[0059] set the value of $k \in \mathbb{N}^+$.

[0060] obtaining the values of β_j , for j=from at least 1, to the final value k by the following equations:

$$\beta_1=4k; \quad (2)$$

$$\beta_j=(j)(\beta_1); \text{ for } 2 \leq j \leq k \quad (3)$$

[0061] And taking the values of β_j and split the space into 2j regions δ_1^j , to δ_{2j}^j which are determined by values κ_1^j to $\kappa_{(2j)-1}^j$.

[0062] In a preferred embodiment, said method further comprising the steps of: iterating the system $x_n^j=f_{\beta_j}(x_n)$ and depending on which region evolves the represent the value of 0 or 1 of the binary sequence ζ^j , generate a k number of chaotic time series by $x_n^j=f_{\beta_j}(x_n)$ and each one produces a binary sequence ζ^j , mixing each of said binary sequence for generating a final sequence Z is as follows:

$$Z=\zeta^1 \oplus \zeta^2 \oplus \dots \zeta^k \quad (4)$$

[0063] To build the generator from any k **102** value, comprising the following steps:

[0064] A. Define a k-modal map.

[0065] B. Choose k initial conditions and k different parameters of bifurcation.

[0066] C. Generate real numbers of a chaotic sequence.
[0067] D. Provide a partition of the phase space where the k -modal map evolves.
[0068] E. Generate binary sequences by means of the phase space partition.
[0069] F. Select pseudo-random bit sequences that pass the NIST suite of statistical test.
[0070] BLOCK 2 (Enc) .-COMBINED SEQUENCE VALIDATION. During this stage it is important to evaluate the output of the generator and be sure that the sequence is cryptographically secure, so this a suite of statistical tests proposed by NIST (10) (National Institute of Standards and Technology), used this suite consists of 15 independent tests.
[0071] In this block takes you run the application and for better understanding, the encryption process will be explained by a cipher chain, as an example in a grayscale image (see FIG. 5), in this block we guarantee that when the image is cipher even with the same key the cipher image will be different each time. Some ciphers like AES¹ each time an image is encrypted with the same key, the result is always the same, this is $C_1=C_2$. If the histograms C_1 and C_2 are subtracted, the result is zero, this indicates that this is the same image.

¹ Advanced Encryption Standard (AES), also known as Rijndael (pronounced "Rain Doll" in English) is a scheme for block cipher adopted as a standard by the US government encryption. The AES was announced by National Institute of Standards and Technology (NIST) FIPS PUB 197 US (FIPS 197) on Nov. 26, 2001, after a standardization process that lasted 5 years, it became a standard effective on 26 May 2002. Since 2006, AES is one of the most popular algorithms used in symmetric cryptography.

[0072] Furthermore, when using the encryption algorithm of the present invention, it is noted that by subtracting C_1 and C_2 histograms, the histogram FIG. 9 is obtained; this is because images $C_1 \neq C_2$, under the condition that were encrypted with the same key x_1 .

[0073] It should be noted that the reverse process (decryption) always produces as output the original image, no matter what embodiment was obtained in the encryption process, for it is said that the decryption process is deterministic; in other words the process of encryption will be probabilistic while the process of decryption will be deterministic.

[0074] In order to evaluate the properties of the encryption process we use a statistical tests and show that when we increase the number of modals (k) then it will increase the number of change in the value of the pixels in the cipher image, this can be measured by the encryption quality.

[0075] Then the second block of the present invention is a method for encryption that includes a probabilistic process comprising the steps of:

[0076] A. Take the output of the first block like the input of the second block.

[0077] B. Provide a grayscale image of size $N \times M$.

[0078] C. Choose an initiation vector IV (size $N \times 1$) (stochastic vector).

[0079] D. Generate the augmented image of size $N \times (M+1)$ given by the stochastic vector and the image.

[0080] E. Mix the pseudo-random bit sequence generated in the first block with the sequence of bits given by the augmented image.

EXAMPLE

[0081] In order to provide technical support to the novelty of the invention and its implementation mechanism, we exemplify the process for $k=3$.

[0082] A. So the first step is given by $k=3$.

[0083] B. According to the second step, $\beta_1=12$, $\beta_1=24$, $\beta_d=36$.

[0084] C. The third step is to take β_1 for obtaining the sequence ζ^1 , therefore the space is divided in two regions δ_1^1 and δ_2^1 splitted by $\kappa_1^1=1/6$. These regions represent a 0 and a 1, respectively, the regions and the histogram of the two regions would be like the present in FIG. 2.

[0085] D. Now we need to compute ζ^2 and ζ^3 , so carry on with β_2 , we have a bimodal map with four regions $\delta_1^2, \dots, \delta_4^2$ separated by $\kappa_1^2=0.1521$, $\kappa_2^2=0.40174$, $\kappa_3^2=0.5954$. Next for the last case β_3 splits the space in six regions $\delta_1^3, \dots, \delta_6^3$ which are determined by the values $\kappa_1^3=0.1465$, $\kappa_2^3=0.396$, $\kappa_3^3=0.639$, $\kappa_4^3=0.8335$, $\kappa_5^3=0.9572$.

[0086] E. Finally, the sequences are mixed by $Z=\zeta^1 \oplus \zeta^2 \oplus \zeta^3$ where \oplus is the XOR operation. We have computed some values that K can take for different values of k and they are shown in Table 1.

TABLE 1

Summary of values κ .				
Map	$\beta = \beta_1$	$\beta = \beta_2$	$\beta = \beta_3$	$\beta = \beta_4$
$\kappa = 1$	$\kappa_1^1 = 1/2$	—	—	—
$\kappa = 2$	$\kappa_1^1 = 1/4$	$\kappa_1^2 = 0.2273$ $\kappa_2^2 = 0.6020$ $\kappa_3^2 = 0.8931$	—	—
$\kappa = 3$	$\kappa_1^1 = 1/6$	$\kappa_1^2 = 0.1521$ $\kappa_2^2 = 0.4017$ $\kappa_3^2 = 0.5954$	$\kappa_1^3 = 0.1465$ $\kappa_2^3 = 0.3960$ $\kappa_3^3 = 0.3690$ $\kappa_4^3 = 0.8335$ $\kappa_5^3 = 0.9572$	—
$\kappa = 4$	$\kappa_1^1 = 1/8$	$\kappa_1^2 = 0.1140$ $\kappa_2^2 = 0.3012$ $\kappa_3^2 = 0.4465$	$\kappa_1^3 = 0.1105$ $\kappa_2^3 = 0.2970$ $\kappa_3^3 = 0.4790$ $\kappa_4^3 = 0.6250$ $\kappa_5^3 = 0.7180$	$\kappa_1^4 = 0.1070$ $\kappa_2^4 = 0.2911$ $\kappa_3^4 = 0.4829$ $\kappa_4^4 = 0.6576$ $\kappa_5^4 = 0.8034$ $\kappa_6^4 = 0.9114$ $\kappa_7^4 = 0.9776$

[0087] In order to characterize the proposed generator and demonstrate that it is safe for use in cryptography, it must be analyzed with a variety of statistical tests. These statistical tests determine whether the generated sequences possess specific characteristics similar to truly random sequences.

[0088] To achieve this goal there are several options available for analyzing the randomness of the pseudo-random bit generators, however the most used and standard test is defined by the NIST [10] that contains a sufficient number of independent statistical tests and detects any deviation from the randomness, the results are shown in Table 2 and 3.

TABLE 2

Part 1 of results from statistical suite of tests.				
Test name	Uni-modal Result	Bi-modal Result	Tri-modal Result	Quad-modal Result
Frequency Test	Success	Success	Success	Success
Frequency Test within a Block	Fail	Success	Success	Success
Runs Test	Success	Success	Success	Success
Test for the Longest Run	Success	Success	Success	Success

TABLE 2-continued

Part 1 of results from statistical suite of tests.				
Test name	Uni-modal Result	Bi-modal Result	Tri-modal Result	Quad-modal Result
Binary Matrix	Success	Success	Success	Success
Discrete Fourier Transform	Success	Success	Success	Success
Overlapping Template Matching Test	Fail	Success	Success	Success
Maurer's Universal Statistical Test	Success	Success	Success	Success
Approximate Entropy	Fail	Success	Success	Success
Linear Complexity	Success	Success	Success	Success

TABLE 3

Part 2 of results from statistical suite of tests.				
Test name	Uni-modal Result	Bi-modal Result	Tri-modal Result	Quad-modal Result
Serial Test 1	Success	Success	Success	Success
Serial Test 2	Success	Success	Success	Success
Cumulative Sums Test				
a) Forward	Success	Success	Success	Success
b) Backward	Success	Success	Success	Success
Non-overlapping Template Matching Test				
a)	Success	Success	Success	Success
b)	Success	Success	Success	Success
c)	Success	Success	Success	Success
d)	Success	Success	Success	Success
e)	Fail	Success	Success	Success
Random Excursions Test				
a) -4	Success	Success	Success	Success
b) -3	Success	Success	Success	Success
c) -2	Success	Success	Success	Success
d) -1	Success	Success	Success	Success
e) 1	Success	Success	Success	Success
f) 2	Success	Success	Success	Success
g) 3	Success	Success	Success	Success
h) 4	Success	Success	Success	Success
Random Excursions Variant Test				
a) -9	Success	Success	Success	Success
b) -8	Success	Success	Success	Success
c) -7	Success	Success	Success	Success
d) -6	Success	Success	Success	Success
e) -5	Success	Success	Success	Success
f) -4	Success	Success	Success	Success
g) -3	Success	Success	Success	Success
h) -2	Success	Success	Success	Success
i) -1	Success	Success	Success	Success
j) 1	Success	Success	Success	Success
k) 2	Success	Success	Success	Success
l) 3	Success	Success	Success	Success
m) 4	Success	Success	Success	Success
n) 5	Success	Success	Success	Success
o) 6	Success	Success	Success	Success
p) 7	Success	Success	Success	Success

TABLE 3-continued

Part 2 of results from statistical suite of tests.				
Test name	Uni-modal Result	Bi-modal Result	Tri-modal Result	Quad-modal Result
q) 8	Success	Success	Success	Success
r) 9	Success	Success	Success	Success

[0089] The second block **120** of FIG. **1** comprises the encryption algorithm of the information by means of a stream-cipher, in the inputs of this block we have the pseudo-random sequence **Z 106**, the initialization vector **IV 108** and the information which in this case is a grayscale image denoted by **img 110**. In the output of this block we have the cipher image **c 112**.

[0090] The first step is a pre-processing of the plain image to an augmented image, this is shown by the block **300** of FIG. **3**, at this point the plain image **110** of FIG. **1** have $M \times N$ **304** pixels, now we add a column of random values **IV 306** with $i=1, \dots, M$. The values of **IV** belong to $\{0, 1, \dots, 255\}$ so the size of the augmented image **P 310** will increase to $M \times (N+1)$ **308**.

[0091] In order to cipher the image **P 310** we have a total of $(M)(N+1)$ pixels and take pixel by pixel as follows (see FIG. **4**):

$$C_1 = P_1 \oplus Z_1 \oplus IV$$

$$C_i = P_i \oplus Z_i \oplus C_{i-1} \quad (5)$$

[0092] Where C_i is the i th pixel of the cipher image **C 112**, P_i is the i th pixel of the augmented image **P 310**, note that the value of $P_1 = IV(1,1)$, Z_i is the key stream **106**, $IV \in \{0, 1, \dots, 255\}$ is an initialization vector **108**, it is used only once and \oplus is the operation XOR bit by bit. For decrypt the image we need the same key stream for this we need the values: k **102**, the initial conditions $x_{01}, x_{02}, \dots, x_{0k}$ **104** and the initialization vector (**IV**) **108** as follows:

$$P_1 = C_1 \oplus Z_1 \oplus IV$$

$$P_i = C_i \oplus Z_i \oplus C_{i-1} \quad (6)$$

[0093] Note that the decrypted image has one extra column so when the process of decrypt is completed we need to remove this column to get original size of the image (N) (**M**).

[0094] The quality of the encryption process as set herein above, can be determined as follows: Encryption quality represents the average number of changes to each gray level and it can be expressed as:

$$Q = \frac{\sum_{L=0}^{255} |H_L(C) - H_L(P)|}{256}$$

[0095] where L is the gray levels of the images, $H_L(C)$ and $H_L(P)$ are the numbers of repetition from each gray value in the original and the encrypted image respectively, we claim that if we increase the number k of one multi-modal map then the encryption quality of the cipher image will increase too.

What is claimed is:

1. A method for generating binary sequences through a pseudo-random bit generator based on chaos, comprising the following steps:

- a) generating a sequence, during this step a set of k independent sequences is generated, represented as {x1, x2, x3, x4, . . . , xk}, wherein each sequence is produced using the dynamic system by a logic unit processing (computer, electronic tablet, etc.), then the k sequences are combined to obtain a single sequence (Z) and this will be the generator output of pseudo-random bits based on chaos;

where the generator is implemented in logic processing unit to generate at least a multi-modal map is defined by the number k defines the maximal numbers of modals in a family \mathfrak{F} and the interval $\mathbb{E}=[a, b]$ is divided between k subintervals $\mathfrak{F}_0=[d_1, d_2), \dots, \mathfrak{F}_{k-1}=[d_{k-1}, d_k]$, then the system f_{β} is a piecewise function by k uni-modal maps; the parameterized family \mathfrak{F} is defined by the following piecewise function

$$f_{\beta}=\beta(d_{r+1}-x)(x-d_r), x \in [d_r, d_{r+1}) \tag{1}$$

wherein: $d_r=r/k$, ($r=0,1,2, \dots, k-1$), k is the number of modals, $\beta=\beta(k, \gamma)$ is the bifurcation parameter, $\gamma=1/k$ is the carrying on capacity; to obtain the maximum value of β with k modals there is a direct relationship, $\beta_{max}=4k/\gamma$;

- b) validating a combined sequence by evaluating the output of the generator to be sure that the sequence is cryptographically secure.

2. A pseudo-random bit generator based on chaos, implemented in units of logic processing, wherein is constructed from any k value comprising the following steps in two blocks:

wherein the block 1 includes:

- a) Defining a k-modal map;
- b) choosing k initial conditions and k different parameters of bifurcation;
- c) generating real numbers of a chaotic sequence;

- d) providing a partition of the phase space where the k-modal map evolves;
- e) generating binary sequences by means of the phase space partition;
- f) selecting pseudo-random bit sequences that pass the NIST suite of statistical test; and;

wherein the block 2 includes:

- a) taking the output of the first block like the input of the second block;
- b) providing a grayscale image of size N*M;
- c) choosing an initiation vector iv (size N*1) (stochastic vector);
- d) generating the augmented image of size N*(M+1) given by the stochastic vector and the image;
- e) mixing the pseudo-random bit sequence generated in the first block with the sequence of bits given by the augmented image.

3. The method for generating binary sequences through a pseudo-random bit generator based on chaos according to claim 1, wherein the stream-cipher by multi-modal dynamical systems comprises a pseudo-random generator block based on k multi-modal maps 110, in the inputs of this block we have the k number 102 which denotes the number of modals and k initial conditions denoted by the vector $[x_0]$ 104, and output the pseudo-random sequence Z 106.

4. The method of claim 1, wherein the validation step comprises using a suite of independent statistical tests; and compare said evaluation with a method of encryption;

where in some encrypted, the result is always the same every time an image is encrypted with the same key, this is $C_1=C_2$; if the histograms from images C_1 and C_2 are subtracted, the result is zero, this indicates that this is the same image; if images $C_1 \neq C_2$, under the condition that were encrypted with the same key x_1 ; and therefore in the reverse (decryption) always produces as output the original image, no matter what embodiment obtained in the encryption process; in other words the encryption process is probabilistic while the decryption process is deterministic.

* * * * *