(11) **MX 2016008395 A**

IMPI
INSTITUTO MEXICANO DE LA PROPIEDAD INDUSTRIAL

(12) **SOLICITUD de PATENTE**

(43) Fecha de publicación: **26/12/2016** (51) Int. Cl: **G06F 7/58** (2006.01)

(22) Fecha de presentación: **23/06/2016**
(21) Número de solicitud: **2016008395**

(30) Prioridad(es): **25/06/2015 US 62/184,600**

(71) Solicidante:
INSTITUTO POTOSINO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA, A.C.
Camino a la Presa San Jose No. 2055 78216 SAN LUIS POTOSI San Luís Potosí MX

(72) Inventor(es):
ERIC CAMPOS CANTON
Circuito Andalucia 116 SAN LUIS POTOSI San Luís Potosí 78433 MX
Moises GARCIA MARTINEZ

(74) Representante:
NORMA ISABEL GARCIA CALDERON
Camino a la Presa San Jose 2055 SAN LUIS POTOSI San Luís Potosí 78216 MX

(54) Título: **GENERADOR DE BITS PSEUDOALEATORIOS BASADOS EN MAPAS MULTIMODALES.**

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

(57) Resumen

La presente invención está relacionada con un método implementado por computadora para generar un generador de bits pseudoaleatorio que comprende los pasos de: a. Definir un mapa multimodal para la ecuación: fß= ß(dr + 1 - X)(X-dr), x ? £r b. Determinar el valor de k e n+ c. Obtener los valores de ßj, para j=1,... k, por las siguientes ecuaciones: ß1 = 4k; ßj = (j) (ß1); para 2 = j = k y tomando los valores de ßj y separar el espacio de fase en regiones 2j sj1, asj2j los cuales son determinados por valores kj1 akj (2j)-1. En modalidad preferida, dicho método comprende además los pasos de: iterar el sistema xjn= fßj (xn) y dependiendo en cual región evoluciona la representación del valor de 0 o 1 de la secuencia binaria cj, genera un número k de series de tiempo caóticas por xjn= fbj (xn) y cada una produce una secuencia binaria cj, mezclando cada una de dichas secuencias binarias para generar una secuencia final z como sigue: z=c1+c2+...ck. La invención comprende además una cadena cifrada derivada a partir de un generador de bits pseudoaleatorio, dicha cadena de cifrado comprende c1=p1+z1+IV cj=Pi+Zi+Ci-1 donde cj es el iésimo pixel de la imagen cifrada c112, Pi es el iésimo pixel de la imagen aumentada P, en donde el valor de P1= IV(1,1), Zi ES LA CADENA CLAVE 106, IV E [0,1,...,255] es un vector de inicialización 108, y + es la operación XOR bit por bit. En una modalidad adicional, la presente invención incluye un generador de bits pseudoaleatroio que comprende un mapa caótico multimodal como se define en: fß= (dr+1-x) (x-dr), x E Jr donde el valor de k E N+, obtener los valores de ßj, para j= de al menos 1, al valor final de k por las siguientes ecuaciones: ß1= 4k; ßj= (j) (ß1); para 2 = j = k en donde dicho mapa obtiene resultados satisfactorios sobre pruebas estadísticas.

(57) Abstract

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?=?(dr+1?x)(x?dr), x ? ?r; b. Set the value of k ? +, and obtaining the values of ?j, for j=from at least 1, to the final value k by the following equations: ?1=4k; ?j=(j)(?1); for 2? j?k and taking the values of ?j and split the space into 2j regions ?1j, to ?2jj which are determined by values k1j to k(2j)?1j.

GENERADOR DE BITS PSEUDOALEATORIOS BASADOS EN MAPAS MULTIMODALES

CAMPO DE LA INVENCION

La presente invención se refiere al campo de la informática segura y, más particularmente, a un generador de secuencias binarias pseudo-aleatorio el cual es producido por medio de sistemas dinámicos en tiempo discreto multimodales unidimensionales llamados mapeos o mapas k-modales. Además, se presenta el impacto de usar estas secuencias pseudo-aleatorias en cifrado en flujo resultado en una mejor calidad de encriptación la cual está correlacionada con el número de modas del mapeo caótico. Finalmente, el algoritmo propuesto puede encriptar por medio de añadir un vector estocástico a la imagen plana, esto es, el mismo conjunto de llaves genera una imagen cifrada diferente cada vez que la imagen plana es encriptada.

Esta invención comprende al menos una aplicación de un algoritmo basado en matemáticas del caos (sistemas dinámicos caóticos), las cuales se correlacionan con el número de modales en un mapa caótico. Entre las principales aplicaciones se encuentran la optimización de la calidad de encriptación de objetos que emplean secuencias de un cifrador de cadenas de bits provienen desde un generador pseudoaleatorio de ellas donde, dichas secuencias son obtenidas desde sistemas dinámicos discretos unidimensionales multimodales llamados mapas k-modal.

La aplicación en la optimización de la calidad de encriptado, se realiza añadiendo un vector estocásticamente obtenido y aplicado al plano de la imagen (función probabilística uniforme) provocando que un mismo conjunto de imágenes de la función integrada en el algoritmo aplique cifrando diferente cada vez que es reproducido.

ANTECEDENTES

Un sistema dinámico en tiempo discreto es definido como $x_{n+1} = f(x_n), n = 0,1,2,\dots,N$, en donde $x_n \in \Re$ y x_0 es la condición inicial, dicho sistema dinámico es referido generalmente como mapa o mapeo, ya que es completamente determinado por su lado derecho. Los mapas más simples son los llamados mapas unimodales, tales como los mapas logístico y casa de campaña, mientras que su generalización, los llamados mapas multimodales o k-modal que pueden presentar comportamientos dinámicos aún más ricos.

Definición 1: un mapa $f: \Im \to \Im$ es denominado como k-modal, si es continuo en $\Im$ y tiene k puntos críticos denotados por $c_0, c_1, \dots, c_{k-1}$ en $\Im$. Más aún, existen intervalos $\Im_i, i = 0, \dots k-1$, $\cup_{i=1}^{k} \Im_{i-1} = \Im$, tales que $\forall\, i = 0, \dots, k-1$, contienen $c_i \in \Im_i$ y $f_\beta(c_i) > f_\beta(x)\ \forall\, x \in \Im_i$ y $x \neq c_i$, donde β es un parámetro.

El caso k= 1 se referirá simplemente como el llamado mapa unimodal.

Más específicamente, el mapa está definido en $\Im := [0,1]$ y, recordando que un punto crítico c de un mapeo continuo suave por partes $f(x): \Im \to \Im$, es dado por un $c \in \Im$ como f es diferenciable entonces $f'(c) = 0$.

Un punto crítico c ocurre para $f'(c) = 0$ o cuando $f'(c)$ no existe. Pero los mapas continuos suaves siempre presentan $f'(c) = 0$. La anterior definición no limita una función a tener únicamente K puntos críticos. Sin embargo únicamente considera aquellos que son máximos locales en un sub-intervalo.

Definición 2. Generador de Números Aleatorios Real (Por sus siglas en inglés True Random Number Generator, TRNG) es un dispositivo

físico diseñado para producir secuencias aleatorias de números sin ser reproducibles.

Definición 3. Generador de números pseudoaleatorios (Por sus siglas en inglés Pseudo-Random Number Generator, PRNG) es un algoritmo que produce una sucesión de números con una excelente aproximación a un conjunto de números aleatorios real.

Definición 4. Generador de números pseudo-aleatorios criptográficamente seguros (Por sus siglas en inglés Cryptographically Secure Pseudo-Random Number Generator, CSPRNG) es un Generador de números pseudo-aleatorios (PRNG) con características que lo hacen adecuado para su uso en criptografía.

Ahora, analicemos algunas patentes que han aplicado este tipo de conocimientos significativos en el desarrollo de nuevas tecnologías:

Típicamente un generador de bits pseudo-aleatorio está basado en un Registro de desplazamiento con retroalimentación lineal (por sus siglas en inglés LFSR de linear feedback shift register), este generador consiste de un grupo de elementos biestables (flip-flops por su denominación en inglés) y puertas exclusivas OR, que permiten a algunos bits en el registro ser verdaderos; es decir, es un registro de desplazamiento en el cual la entrada es un bit proveniente de aplicar una función de transformación lineal a un estado anterior.

El valor inicial se denomina semilla y, como la forma de operar el registro es determinista, la secuencia de valores producidos está completamente determinada por el estado actual o el estado anterior. La secuencia tiene un periodo de repetición, es decir que la secuencia vuelve a generarse y se repite indefinidamente. Cuando el periodo de repetición es máximo, ese LFSR tiene interés criptográfico.

Una característica principal de la invención propuesta, es que describe un algoritmo para generar secuencias de bits pseudo-aleatorias donde el periodo

de repetición es mucho mayor en comparación con los LFSR; es decir, en términos holísticos, el generador está basado en un sistema dinámico de tiempo discreto cuya aplicación es más versátil que la anteriormente descrita.

En el estado de la técnica encontramos algunas patentes relacionadas con las aplicaciones que se contienen en la presente solicitud, tales como:

La patente **US8738675B2** (Generador de números aleatorios usando el caos en tiempo continuo) se refiere a un método de generación de números aleatorios basados en osciladores caóticos de tiempo continuo; donde la compensación de la frecuencia es empleada para maximizar la calidad estadística de una secuencia de salida donde los circuitos propuestos se pueden realizar en circuitos integrados y el uso de caos continuo en la generación de números aleatorios con un rendimiento muy alto;

Otra invención se refiere a la patente **CN101364171B** (Generador dinámico de números aleatorios reales) que se refiere a un generador, el cual emplea datos introducidos a validadores de Von Neumann para ajustar los bits de salida por medio de algoritmos de conversión del caos a través de componentes discretos y cifrado en bloques; usando especialmente el índice Lyapunov para la transformación del bloque, como manera de controlar el caos dentro de un rango establecido y el empleo de las llamadas S-caja que influye en la transformación de los bloques de cifrado;

La patente **US6954770** (Generador de números aleatorios) que comprende un oscilador con una señal de salida que depende de una fuente aleatoria *-que no es basada en teoría de caos-*, un dispositivo de muestreo para muestrear la señal de salida del oscilador para obtener una salida del oscilador muestreado, y un registro de reloj de frecuencia fija manejando al registro de desplazamiento de realimentación lineal (LFSR) acoplado en comunicación con el dispositivo de muestreo a través de una puerta digital para recibir la salida del oscilador muestreado, y proveer un número aleatorio en una salida del LFSR. Además, el generador de números aleatorios puede estar conformado por función de

mezclado opcional acoplada en comunicación al LFSR para leer el número aleatorio, y para insertar el número aleatorio en un algoritmo para obtener un número aleatorio robusto; no obstante, esta tecnología depende únicamente de los dos osciladores y las posibles respuestas que mantiene solo presenta estas dos opciones de salida.

Las patentes anteriores, son útiles en la criptografía para la generación de llaves/claves/datos de insumo en sistemas de cifrado simétrico. Un conjunto de patentes que están relacionadas con arquitecturas de computación caóticos para puertas lógicas basadas en elementos no lineales, aunque la presente invención da a conocer estructuras dinámicamente configurables de puertas lógicas básicas lineales son: **US8091062**, **US7973566**, **US7924059**, **US7863937**, **US7415683**, **US7096437**, **US7453285**, **US7925814**, **US7925131** y la solicitud **US 2010/0219858**.

Todas las patentes anteriores representan antecedentes ya sea de entrada, de aplicación o de salida de los datos, pero el tipo de invenciones relacionadas con la que se contiene en la presente solicitud, son aquellas que se relacionan con generadores de números aleatorios, donde su salida puede ser reproducida cuando se conoce su condición inicial, existe una variedad de este tipo de sistemas, por ejemplo: **US8023649B2** y **US7925014B2** las cuales proponen algoritmos de generadores pseudo-aleatorios basados en autómatas celulares.

El generador pseudo-aleatorio que se propone en esta patente se basa en el uso de sistemas dinámicos debido a las propiedades intrínsecas de estos sistemas como lo son la sensibilidad a las condiciones iniciales, comportamiento complejo pero determinista, ergodicidad, etc.

Incluye implementaciones de sistemas dinámicos de tiempo continuo por medio de hardware, similares a los de las patentes **US2014101217A1**, **EP2758510A1** y **US7170997B2**, donde este tipo de sistemas requiere de métodos numéricos para obtener su evolución en el tiempo si se implementa por medio de software,

sin embargo, si se implementan en circuitos electrónicos o en FPGA's es posible obtener su evolución.

En el estado de la técnica hay generadores pseudo-aleatorios basados en sistemas dinámicos de tiempo discreto por ejemplo la patente **US2006251250A1**, propone un generador pseudo-aleatorio basado en el mapeo de henon y el mapeo de lozi, el método de generación consiste en iterar el sistema y cada iteración se convierte a un número entero y por último cada iteración se convierte en un bit por medio de la aplicación de módulo 2; para generar el arreglo de bits posibles solo como 1 o 0; cuya dimensión es de 1XN.

La patente **US6064738A** utiliza un mapeo de dos dimensiones (Baker map) para construir un arreglo de bits con dimensiones N x M, con esto se realiza una permutación de pixeles, para cambiar de lugar los pixeles es importante tener en cuenta que al realizar este paso el histograma de la imagen no cambia, adicionalmente se hace una extensión del mapeo a tres dimensiones y posteriormente se sustituye el valor de los pixeles de tal forma que en este paso si cambia la distribución de pixeles, tal como si se tratara de una función f(x,y) donde $x \in N \; e \; y \in M$.

En la patente **US6014445A** utiliza el mapa Chebyshev para generar secuencias con valores reales y presenta cuatro métodos para "binarizar" la serie de tiempo;

a) La primera metodología se basa en el uso de un umbral (threshold)
b) La segunda metodología se toma el valor absoluto de cada iteración y se toma su representación en binario.
c) La tercera metodología es una generalización de la segunda metodología de tal forma que se realiza un escalamiento que depende del intervalo en donde evoluciona el sistema, por lo que si el intervalo es [0,1] entonces la metodología 2 y 3 son iguales.
d) En la cuarta metodología tienen M umbrales (siendo estos umbrales la llave) de tal forma que cada valor real de la serie de tiempo, es evaluada por estos umbrales por lo que para un valor real se tienen

M bits, finalmente se combinan los M bits por medio de operaciones XOR y así obtienen un 1 bit por iteración.

La patente **US2013129088A1** propone un generador pseudo-aleatorio basado en la combinación de múltiples sistemas dinámicos de tiempo discreto, específicamente utilizan el mapa cyclic shift, para esto definen k mapeos idénticos los cuales se iteran de forma independiente, posteriormente se combinan por medio de la operación XOR de tal forma que para obtener una iteración n se requiere previamente haber iterado los k mapeos.

Finalmente, ha habido propuestas de generadores basados en los siguientes tipos:

Primero. En mapas caóticos unimodales, algunas de las patentes que divulgan un generador de bits pseudoaleatorios utilizando solo un mapa logístico;

Segundo. En una la cadena de bit generada a través de un sistema caótico era potencialmente inseguro debido a que la salida puede fugar alguna información acerca del sistema caótico.

Tercero. En el uso de dos mapas caóticos completamente diferentes los cuales son combinados para obtener una secuencia compleja de bits.

Objetivo de la invención:

El generador de bits pesudoaleatorio propone ser basado en al menos un par de mapas caóticos definidos por partes, los cuales son iterados independientemente y las cadenas de bits son generadas al comparar las salidas de estos mapas caóticos y donde en esta invención se emplea sólo un mapa caótico multimodal que puede reemplazar a los dos mapas caóticos unimodales para la generación de secuencias pseudoaleatorias.

En esta patente se propone el uso de sistemas dinámicos multi-modales (k-modal) de tiempo discreto, con esto es posible obtener k diferentes series de

tiempo con la ventaja de tener solamente un mapeo, la forma de convertir la serie de tiempo real a binario es por medio de 2k umbrales, con esto cada modal es capaz de producir un cero o un uno, finalmente se combinan las k series binarias por medio de la operación XOR.

DESCRIPCION

Los detalles característicos de este novedoso **método para generar secuencias binarias a través de un generador de bits pseudoaleatorios basados en caos,** se muestran claramente en la siguiente descripción y en las figuras que se acompañan, así como una descripción de aquella, donde se siguen los mismos signos de referencia para indicar las partes y figuras mostradas:

Breve descripción de las figuras:

Figura 1 Es una vista del diagrama general de bloque del generador de bits pseudoaleatorios basados en caos, que implementa el método para generar secuencias binarias.

Figura 2 muestra un diagrama de bloque que describe el generador pseudoaleatorio basado en mapas multimodales, que implementa el método para generar secuencias binarias.

Figura 3 es un diagrama de bloque que describe un preproceso para encriptación, como un ejemplo de aplicación del método para generar secuencias binarias a través de un generador de bits pseudoaleatorios basados en caos.

Figura 4 es un diagrama de bloque que describe el proceso de encriptación, como un ejemplo de aplicación del método para generar secuencias binarias a través de un generador de bits pseudoaleatorios basados en caos.

Figura 5 es una imagen convencional que cuenta con la distribución de los pixeles a cifrar y que se prepara para emplear el método

para generar secuencias binarias a través de un generador de bits pseudoaleatorios basados en caos.

Figura 6 es la vista del histograma de la imagen convencional de la figura 5, que cuenta con la distribución de los pixeles a cifrar y que se prepara para emplear el método para generar secuencias binarias a través de un generador de bits pseudoaleatorios basados en caos.

Figura 7 es la representación de la imagen convencional de la figura 5, con el cifrado con una llave x_1 de las obtenidas en el método para generar secuencias binarias a través de un generador de bits pseudoaleatorios basados en caos.

Figura 8 es la vista del histograma de la representación de la imagen convencional de la figura 7, con el cifrado con una llave x_1 de las obtenidas en el método para generar secuencias binarias a través de un generador de bits pseudoaleatorios basados en caos.

Figura 9 es la vista del histograma obtenido de las diferencias de los histogramas cifrados en las figuras 6 y 8.

Descripción general de la novedad de la invención:

El método para generar secuencias binarias a través de un generador de bits pseudoaleatorios basados en caos, a grandes rasgos consiste en el empleo de al menos un par de parámetros de un mapa multimodal o k-modal, en nuestro caso empleamos mapas k-modales, obtenidos de la definición previa donde es posible generar familia de mapas, donde la novedad radica en que los generadores pseudoaleatorios existentes, emplean la combinación de al menos dos sistemas dinámicos diferentes; en la presente invención el generador emplea un solo sistema con diferente valor en un parámetro, el cual es capaz de producir al menos dos modos de oscilar un sistema (familia de sistemas)

para combinarlos entre ellos y eficientar el cifrado empleando al menos una llave por cada modo del sistema generado y aplicarlo en múltiples procesos como el encriptado de imágenes, por ejemplo.

Este generador de bits pseudoaleatorio y el cifrado en flujo son útiles para utilizarse en imágenes, documentos, encriptación de audio y/o video, archivos multimedia, Sistemas de Control Industrial (ICS) de seguridad y otras aplicaciones de seguridad.

OBSERVACIÓN: Los ejemplos presentados en este documento son meramente enunciativos, más no limitativos, por lo que solamente se enlistan para ejemplificar la aplicación del método y la novedad de su implementación en unidades de procesamiento lógico.

Con base en las figuras anteriores, el generador de bits pseudoaleatorios basados en caos se implementa en una unidad de procesamiento lógico con los siguientes bloques:

BLOQUE 1 (GEN) .- GENERACIÓN DE SECUENCIAS. Durante esta etapa, se genera un conjunto de k secuencias independientes, representado por $\{x_1, x_2, x_3, x_4, \ldots x_k\}$ y donde cada secuencia es producida empleando el mismo sistema dinámico empleando una unidad de procesamiento lógico (computador, tableta electrónica, etc.), posteriormente se combinan las secuencias k para obtener una sola secuencia (Z) y esta será la salida del generador de bits pseudoaleatorios basados en caos (Bloque de salida 1);

Donde el generador es implementado en la unidad de procesamiento lógico para generar al menos un mapa multimodal definido por el número k que define los números máximos de modas en una familia $\mathfrak{F}$ y el intervalo $\mathfrak{I} = [a, b]$ es dividido entre los k subintervalos $\mathfrak{I}_0 =$

$[d_1, d_2), \ldots, \Im_{k-1} = [d_{k-1}, d_k]$, entonces el Sistema f_β es una función por partes basada en k mapas unimodales. La familia parametrizada $\mathfrak{F}$ es definida por la siguiente función por partes

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

Donde $d_r = {}^{r}/_{k}, (r = 0,1,2, \ldots, k-1), k$ es el número de modas, $\beta = \beta(k, \gamma)$ es el parámetro de bifurcación, $\gamma = {}^{1}/_{k}$ es la capacidad de carga. Para obtener el valor máximo de β con k modas hay una relación directa, $\beta_{max} = {}^{4k}/_{\gamma}$.

Y tomando los valores de β_j y separar el espacio en regiones $2j$ $\delta_1^j, a\ \delta_{2j}^j$ los cuales son determinados por valores $\kappa_1^j\ a\ k_{(2j)-1}^j$.

En una modalidad preferida, dicho método comprende además los pasos de: iterar el sistema $x_n^j = f_{\beta_j}(x_n)$ y dependiendo en cual region evoluciona la representación del valor de 0 o 1 de la secuencia binaria ζ^j, genera un número k de series de tiempo caóticas por $x_n^j = f_{\beta_j}(x_n)$ y cada una produce una secuencia binaria ζ^j, mezclando cada una de dicha secuencia binaria para generar una secuencia final Z como sigue:

$$Z = \zeta^1 \oplus \zeta^2 \oplus \ldots \zeta^k.$$

La invención, comprende además un cifrado en flujo derivado a partir de un generador de bits pseudo-aleatorio, dicho cifrado en flujo está dado por

$$C_1 = P_1 \oplus Z_1 \oplus IV$$
$$C_i = P_i \oplus Z_i \oplus C_{i-1}$$

Donde C_i es el iésimo pixel de la imagen cifrada C **112**, P_i es el iésimo pixel de la imagen aumentada P, en donde el valor de $P_1 = IV(1,1), Z_i$ es la

cadena de llave o clave **106**, $IV \in \{0,1,\ldots,255\}$ es un vector de inicialización **108**, y $\oplus$ es la operación XOR bit por bit.

En una modalidad adicional, la presente invención incluye un generador de bits pseudoaleatorio que comprende un mapa caótico multimodal como se define en:

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

Donde $d_r = \mathrm{r/k}, (\mathrm{r} = 0, 1, 2, \ldots, \mathrm{k} - 1)$, k es el número de modas, $k \in \mathbb{N}^+$, Para obtener los valores de β_j, para $j = 1, \ldots, k$, se usan las ecuaciones siguientes:

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

$$\beta_j = (j)(\beta_1); \text{ para } 2 \leq j \leq k. \qquad (3)$$

Y tomando los valores de β_j y separar el espacio en regiones $2j$ $\delta_1^j, a\ \delta_{2j}^j$ los cuales son determinados por valores $\kappa_1^j\ a\ k_{(2j)-1}^j$.

En una modalidad preferida, dicho método comprende además los pasos de: iterar el sistema $x_n^j = f_{\beta_j}(x_n)$ y dependiendo en cual región evoluciona la representación del valor de 0 o 1 de la secuencia binaria ζ^j, genera un número k de series de tiempo caóticas por $x_n^j = f_{\beta_j}(x_n)$ y cada una produce una secuencia binaria ζ^j, mezclando cada una de dicha secuencia binaria para generar una secuencia final Z como sigue:

$$Z = \zeta^1 \oplus \zeta^2 \oplus \ldots \zeta^k. \qquad (4)$$

Para construir el generador a partir de cualquier valor k 102. Comprende las siguientes etapas:

A. Definir un mapa k-modal.
B. Elegir k condiciones iniciales y diferentes parámetros de bifurcación de β.
C. Generar números reales de una secuencia caótica.

D. Proveer una partición del espacio de fase donde evoluciona el mapa k-modal.
E. Generar secuencias binarias por medio de la partición del espacio de fase.
F. Seleccionar secuencias pseudo-aleatorios de bits que pasen la suite de pruebas estadísticas del NIST.

BLOQUE 2 (ENC).- VALIDACION DE LA SECUENCIA COMBINADA. Durante esta etapa es importante evaluar la salida del generador para estar seguros de que la secuencia es criptográficamente segura, para esto utilizamos una suite de pruebas estadísticas propuesta por el NIST (10) (Instituto Nacional de Estandares y Tecnología), esta suite consiste de 15 pruebas independientes.

En el presente bloque se lleva a cabo la aplicación y para su mejor comprensión, se explicará en el proceso de encriptación por medio de un cifrado en flujo.

A manera de ejemplo una imagen en escalas de grises (véase figura 5), en este bloque nos aseguramos de que cuando la imagen es cifrada aún con la misma clave, la imagen cifrada será diferente cada vez. En algunos cifrados como el AES[1] cada vez que se cifra una imagen con

[1] Advanced Encryption Standard (AES), también conocido como Rijndael (pronunciado "Rain Doll" en inglés), es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos. El AES fue anunciado por el Instituto Nacional de Estándares y Tecnología (NIST) como FIPS PUB 197 de los Estados Unidos (FIPS 197) el 26 de noviembre de 2001 después de un proceso de estandarización que duró 5 años. Se transformó en un estándar efectivo el 26 de mayo de 2002.

la misma llave, el resultado siempre será el mismo, esto es $C_1 = C_2$. Si se resta el histograma de C_1 y C_2 el resultado será cero, esto indica que se trata de la misma imagen.

Por otro lado, al usar el algoritmo de cifrado de la presente invención, se observa que al restar el histograma de C_1 y C_2 se obtiene el histograma de la figura 9; esto se debe a que las imágenes $C_1 \neq C_2$, bajo la condición de que fueron cifradas con la misma llave x_1.

Es preciso hacer notar que el proceso inverso (desencriptación) siempre produce como salida la imagen original sin importar cual realización se obtuvo en el proceso de cifrado, por eso se dice que el proceso de descifrado es determinista; en otras palabras el proceso de encriptación será probabilístico mientras que el proceso de desencriptación será determinístico.

Para evaluar las propiedades del proceso de encriptación utilizamos pruebas estadísticas y mostramos que cuando incrementamos el número de modas (k) entonces aumentará el número de cambio en el valor de los pixeles en la imagen cifrada, esto puede ser medido por la calidad de encriptación.

Entonces el segundo bloque de la presente invención es un método para encriptar que incluye un proceso probabilístico que comprende los pasos de:

A. Tomar la salida del primer bloque como la entrada del segundo bloque;
B. Proveer una imagen en escala de grises del tamaño (N)(M);
C. Elegir un vector de iniciación IV (tamaño (N)(1) (vector estocástico);

Desde 2006, el AES es uno de los algoritmos más populares usados en criptografía simétrica.

D. Generar la imagen aumentada de tamaño (N)(M+1) dada por el vector estocástico y la imagen;
E. Mezclar la secuencia de bits pseudoaleatorias generada en el primer bloque con la secuencia de bits dada en la imagen aumentada.

EJEMPLO 1

Con la finalidad de brindar soporte técnico a la novedad de la invención y su mecanismo de implementación, se ejemplifica el método para generar secuencias binarias a través de un generador de bits pseudoaleatorios basados en caos, para el valor de k = 3.

A. El primer paso es dado por k=3.

B. El Segundo paso, $\beta_1 = 12$, $\beta_2 = 24$, $\beta_3 = 36$.

C. El tercer paso es tomar β_1 para obtener la secuencia ζ^1, por lo tanto el espacio es dividido en dos regiones δ_1^1 y δ_2^1 separadas por $\kappa_1^1 = 1/6$. Estas regiones representan un valor binario, es decir un 0 o 1, respectivamente las regiones y el histograma de las dos regiones será como la que se presenta en la figura 2.

D. Ahora, se necesita computar ζ^2 y ζ^3, posteriormente, se continua definiendo el mapa dado por β_2, que representa un mapa bimodal con cuatro regiones $\delta_1^2, \dots, \delta_4^2$ separadas por $\kappa_1^2 = 0.1521, \kappa_2^2 = 0.40174, \kappa_3^2 = 0.5954$. Ahora bien, para el último caso, considerando β_3 se subdivide el espacio en seis regiones $\delta_1^3, \dots, \delta_6^3$ las cuales son determinadas por los valores $\kappa_1^3 = 0.1465, \kappa_2^3 = 0.396, \kappa_3^3 = 0.639, \kappa_4^3 = 0.8335, \kappa_5^3 = 0.9572$.

E. Finalmente, las secuencias son mezcladas por $Z = \zeta^1 \oplus \zeta^2 \oplus \zeta^3$ donde $\oplus$ es la operación XOR. Hemos computado algunos valores que κ puede tomar, diferentes valores de k son mostrados en la Tabla 1.

Tabla 1 Resumen de valores k

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

Para caracterizar el generador propuesto y demostrar que es seguro para su uso en criptografía, se deben analizar con una variedad de pruebas estadísticas; en estas pruebas estadísticas las secuencias generadas poseen características específicas similares a secuencias aleatorias verdaderas.

Para lograr esta meta hay diversas opciones disponibles para analizar la aleatoriedad de los generadores de bits pseudo-aleatorios, sin embargo la prueba más utilizada y estándar es definida por el Instituto Nacional de Normas y Tecnología (NIST por sus siglas en inglés, National Institute of Standards and

Technology), que contiene un número suficiente de pruebas estadísticas independientes y detecta cualquier desviación de la aleatoriedad.
Los resultados se muestran en las Tablas 2 y 3.

Tabla 2. Parte 1 de los resultados de las suites de pruebas estadísticas.

Nombre de prueba	Resultado Uni-modal	Resultado Bi-modal	Resultado Tri-modal	Resultado Quad-modal
Prueba de Frecuencia	Éxito	Éxito	Éxito	Éxito
Prueba de frecuencia con un bloque	Falló	Éxito	Éxito	Éxito
Prueba de corrida	Éxito	Éxito	Éxito	Éxito
Prueba de Carrera más larga	Éxito	Éxito	Éxito	Éxito
Matriz Binaria	Éxito	Éxito	Éxito	Éxito
Transformada de Fourier discreta	Éxito	Éxito	Éxito	Éxito
Prueba de sobrelapado de templado	Falló	Éxito	Éxito	Éxito
Prueba estadística Universal de Maurer	Éxito	Éxito	Éxito	Éxito
Entropía aproximada	Falló	Éxito	Éxito	Éxito
Complejidad linear	Éxito	Éxito	Éxito	Éxito

Tabla 3. Parte 2 de los resultados para las suites de pruebas estadísticas

Nombre de la prueba	Resultado Uni-modal	Resultado Bi-modal	Resultado Tri-modal	Resultado Quad-modal
Prueba serial 1	Éxito	Éxito	Éxito	Éxito
Prueba serial 2	Éxito	Éxito	Éxito	Éxito
Prueba de sumas acumulativas				
a) hacia delante	Éxito	Éxito	Éxito	Éxito
b) hacia atrás	Éxito	Éxito	Éxito	Éxito
Prueba de no sobrelapamiento del templado				
a)	Éxito	Éxito	Éxito	Éxito
b)	Éxito	Éxito	Éxito	Éxito
c)	Éxito	Éxito	Éxito	Éxito
d)	Éxito	Éxito	Éxito	Éxito
e)	Falló	Éxito	Éxito	Éxito
Pruebas de excursiones de aleatoriedad				
a) -4	Éxito	Éxito	Éxito	Éxito
b) -3	Éxito	Éxito	Éxito	Éxito
c) -2	Éxito	Éxito	Éxito	Éxito
d) -1	Éxito	Éxito	Éxito	Éxito
e) 1	Éxito	Éxito	Éxito	Éxito
f) 2	Éxito	Éxito	Éxito	Éxito
g) 3	Éxito	Éxito	Éxito	Éxito
h) 4	Éxito	Éxito	Éxito	Éxito

Prueba de excursiones de variante al azar				
a) -9	Éxito	Éxito	Éxito	Éxito
b) -8	Éxito	Éxito	Éxito	Éxito
c) -7	Éxito	Éxito	Éxito	Éxito
d) -6	Éxito	Éxito	Éxito	Éxito
e) -5	Éxito	Éxito	Éxito	Éxito
f) -4	Éxito	Éxito	Éxito	Éxito
g) -3	Éxito	Éxito	Éxito	Éxito
h) -2	Éxito	Éxito	Éxito	Éxito
i) -1	Éxito	Éxito	Éxito	Éxito
j) 1	Éxito	Éxito	Éxito	Éxito
k) 2	Éxito	Éxito	Éxito	Éxito
l) 3	Éxito	Éxito	Éxito	Éxito
m) 4	Éxito	Éxito	Éxito	Éxito
n) 5	Éxito	Éxito	Éxito	Éxito
o) 6	Éxito	Éxito	Éxito	Éxito
p) 7	Éxito	Éxito	Éxito	Éxito
q) 8	Éxito	Éxito	Éxito	Éxito
r) 9	Éxito	Éxito	Éxito	Éxito

El segundo bloque 120 de la figura 1 comprende el algoritmo de encriptación de la información por medio de un cifrado en flujo, en las entradas de este bloque tenemos la secuencia pseudo-aleatoria Z 106, el vector de iniciación IV 108 y la información la cual en este caso es una imagen en escalas de grises denotada como img 110. En la salida de este bloque tenemos la imagen cifrada c 112.

El primer paso es un pre-procesamiento de la imagen completa, a una imagen aumentada, esto se muestra en el bloque 300 de la figura 3, en este punto la

imagen completa 110 de la figura 1 tiene los pixeles M **302** x N **304,** ahora se añade una columna de valores aleatorios $IV(i,1)$ **306 con** i = 1, ...,M. Los valores de IV pertenecen a $\{0,1,\ldots,255\}$ entonces el tamaño de la imagen aumentada P 310 aumentará a M **302** x (N+1) **308**.

Para cifrar la imagen P 310 se tiene un total de (M)(N +1) pixeles y toma pixel por pixel como a continuación (ver figura 4):

$$\begin{aligned} C_1 &= P_1 \oplus Z_1 \oplus IV \\ C_i &= P_i \oplus Z_i \oplus C_{i-1} \end{aligned} \qquad (5)$$

Donde C_i es el iésimo pixel de la imagen cifrada C 112, P_i es el iésimo pixel de la imagen aumentada P 310, note que el valor de $P_1 = IV(1,1), Z_i$, es la cadena clave 106, $IV \in \{0,1,\ldots,255\}$ es un vector de iniciación 108, es utilizado únicamente una vez y $\oplus$ es la operación XOR bit por bit. Para desencriptar la imagen necesitamos la misma cadena de llave o clave, para esto necesitamos los valores: k **102**, las condiciones iniciales $x_{01}, x_{02}, \ldots, x_{0k}$ **104** y el vector de iniciación (IV) **108** como sigue:

$$\begin{aligned} P_1 &= C_1 \oplus Z_1 \oplus IV \\ P_i &= C_i \oplus Z_i \oplus C_{i-1} \end{aligned} \qquad (6)$$

Es importante observar que la imagen desencriptada tiene una columna extra de tal forma que cuando el proceso de desencriptación es completado se requiere remover esta columna para lograr el tamaño original de la imagen $(N)(M)$.

La calidad del proceso de encriptación como se describió anteriormente, puede ser determinado de la siguiente forma:

La calidad de encriptación representa el número promedio de cambios para cada nivel de gris y puede ser expresado como:

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

En donde L es el nivel de grises de las imágenes, $H_L(C)$ y $H_L(P)$ son los números de repetición de cada valor de gris en el original y en la imagen encriptada respectivamente, reclamamos que si se incrementa el número k de un mapa multimodal, entonces la calidad de encriptación de la imagen cifrada aumenta también.

REIVINDICACIONES

Habiendo descrito suficientemente mi invención, lo que considero como una novedad, lo contenido en las siguientes reivindicaciones:

1. Un método para generar secuencias binarias a través de un generador de bits pseudoaleatorios basados en caos, **caracterizado porque** comprende las siguientes etapas:

   a. **GENERACION DE SECUENCIAS.** Durante esta etapa, se genera un conjunto de secuencias independientes k, representado por $\{x_1, x_2, x_3, x_4, \dots x_k\}$ y donde cada secuencia es producida empleando el mismo sistema dinámico empleando una unidad de procesamiento lógico (computador, tableta electrónica, etc.), posteriormente se combinan las k secuencias para obtener una sola secuencia (Z) y esta será la salida del generador de bits pesudo-aleatorios basados en caos;

   Donde el generador es implementado en la unidad de procesamiento lógico para generar al menos un mapa multimodal definido por el número k que define los números máximos de modas en una familia $\mathfrak{F}$ y el interval $\mathfrak{I} = [a, b]$ es dividido entre los subintervalos k $\mathfrak{I}_0 = [d_1, d_2), \dots, \mathfrak{I}_{k-1} = [d_{k-1}, d_k]$, entonces el Sistema f_β es una función por partes dada por k mapas unimodales. La familia parametrizada $\mathfrak{F}$ es definida por la siguiente función por partes

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

   Donde $d_r = {}^{r}/_{k}, (r = 0,1,2, \dots, k-1), k$ es el número de modas, $\beta = \beta(k, \gamma)$ es el parámetro de bifurcación, $\gamma = {}^{1}/_{k}$ es la capacidad de carga. Para obtener el valor máximo de β con k modas hay una relación directa, $\beta_{max} = {}^{4k}/_{\gamma}$.

En una modalidad, se considera un cifrado en flujo usando sistemas dinámicos multimodales 100 como se muestra en la figura 1. El cifrado en flujo usando sistemas dinámicos multimodales comprende un bloque generador pseudo-aleatorio basado en mapas multimodales k 110, en las entradas de este bloque tenemos el número k 102 el cual denota el número de modas y las k condiciones iniciales denotada por el vector $[x_0]$ $\mathbf{104}$, y como salida la secuencia pseudoaleatoria Z 106;

b. **VALIDACION DE LA SECUENCIA COMBINADA.** Durante esta etapa es importante evaluar la salida del generador para estar seguros de que la secuencia es criptográficamente segura, para esto utilizamos una suite de pruebas estadísticas independientes; y un método de encriptamiento; donde en algunos cifrados, cada vez que se cifra una imagen con la misma llave, el resultado siempre será el mismo, esto es $C_1 = C_2$. Si se resta el histograma de C_1 y C_2 el resultado será cero, esto indica que se trata de la misma imagen; esto se debe a que las imágenes $C_1 \neq C_2$, bajo la condición de que fueron cifradas con la misma llave x_1; y por tanto, en el proceso inverso (desencriptación) siempre produce como salida, la imagen original sin importar cual realización se obtuvo en el proceso de cifrado; en otras palabras el proceso de encriptación será probabilístico mientras que el proceso de desencriptación será determinístico.

2. El generador de bits pseudo-aleatorios basado en caos, de los que se implementan en unidades de procesamiento lógico para llevar a cabo el

método de la reivindicación uno, **caracterizado porque** se construye a partir de cualquier valor k que comprende los siguientes pasos en dos bloques:

Bloque 1

A. Definir un mapa k-modal.
B. Elegir k condiciones iniciales y diferentes parámetros de bifurcación β.
C. Generar números reales de una secuencia caótica.
D. Proveer una partición de espacio de fase donde evoluciona el mapa k-modal.
E. Generar secuencias binarias por medio de la partición del espacio de fase.
F. Seleccionar secuencias pseudoaleatorios de bits que pasen la suite de pruebas estadísticas del NIST.

Bloque 2

A. Tomar la salida del primer bloque como la entrada del segundo bloque;
B. Proveer una imagen en escala de grises del tamaño (N)(M);
C. Elegir un vector de iniciación IV (tamaño (N)(1) (vector estocástico);
D. Generar la imagen aumentada de tamaño (N)(M+1) dada por el vector estocástico y la imagen;
E. Mezclar la secuencia de bits pseudo-aleatorias generada en el primer bloque con la secuencia de bits dada en la imagen aumentada.

RESUMEN

La presente invención está relacionada con un método implementado por computadora para generar un generador de bits pseudoaleatorio que comprende los pasos de:

a. Definir un mapa multimodal para la ecuación:

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

b. Determinar el valor de $k \in \mathbb{N}^+$,

c. Obtener los valores de β_j, para $j = 1, \ldots k$, por las siguientes ecuaciones:

$$\beta_1 = 4k;$$

$$\beta_j = (j)(\beta_1); \text{ para } 2 \leq j \leq k$$

Y tomando los valores de β_j y separar el espacio de fase en regiones $2j$ $\delta_1^j, a\ \delta_{2j}^j$ los cuales son determinados por valores $\kappa_1^j\ a\ k_{(2j)-1}^j$.

En una modalidad preferida, dicho método comprende además los pasos de: iterar el sistema $x_n^j = f_{\beta_j}(x_n)$ y dependiendo en cual región evoluciona la representación del valor de 0 o 1 de la secuencia binaria ζ^j, genera un número k de series de tiempo caóticas por $x_n^j = f_{\beta_j}(x_n)$ y cada una produce una secuencia binaria ζ^j, mezclando cada una de dichas secuencias binarias para generar una secuencia final Z como sigue:

$$Z = \zeta^1 \oplus \zeta^2 \oplus \ldots \zeta^k.$$

La invención, comprende además una cadena cifrada derivada a partir de un generador de bits pseudoaleatorio, dicha cadena de cifrado comprende

$$C_1 = P_1 \oplus Z_1 \oplus IV$$
$$C_i = P_i \oplus Z_i \oplus C_{i-1}$$

Donde C_i es el iésimo pixel de la imagen cifrada C **112**, P_i es el iésimo pixel de la imagen aumentada P, en donde el valor de $P_1 = IV(1,1), Z_i$ es la cadena clave **106**, $IV \in \{0,1,\ldots,255\}$ es un vector de inicialización **108**, y $\oplus$ es la operación XOR bit por bit.

En una modalidad adicional, la presente invención incluye un generador de bits pseudoaleatorio que comprende un mapa caótico multimodal como se define en:

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

Donde el valor de $k \in \mathbb{N}^+$,

Obtener los valores de β_j, para $j = de\ al\ menos\ 1, al\ valor\ final\ de\ k$ por las ecuaciones siguientes:

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

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

En donde dicho mapa obtiene resultados satisfactorios sobre pruebas estadísticas.

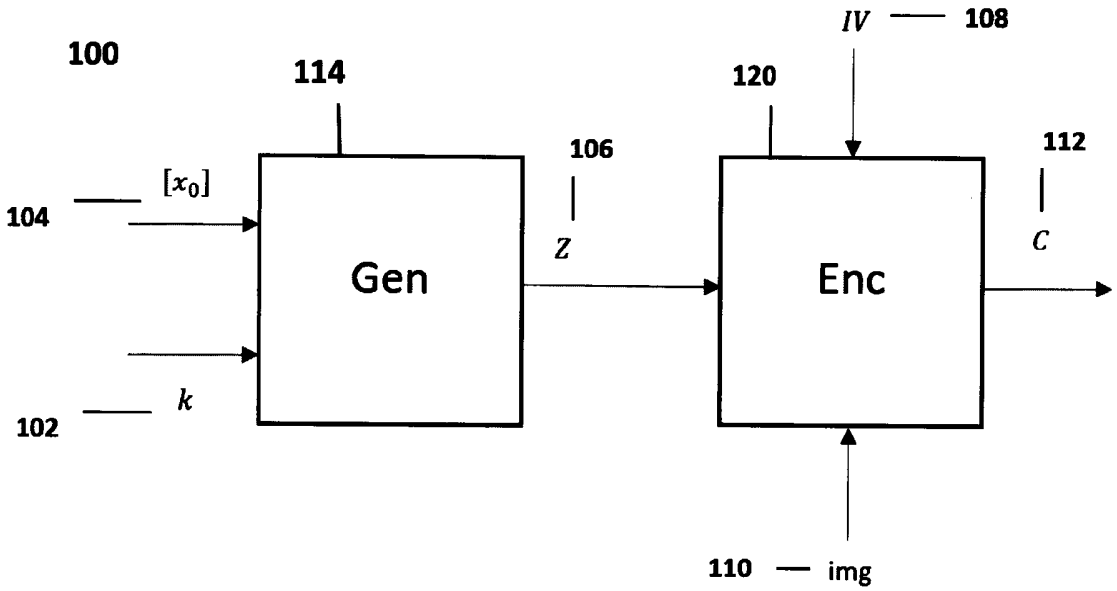

Figura 1

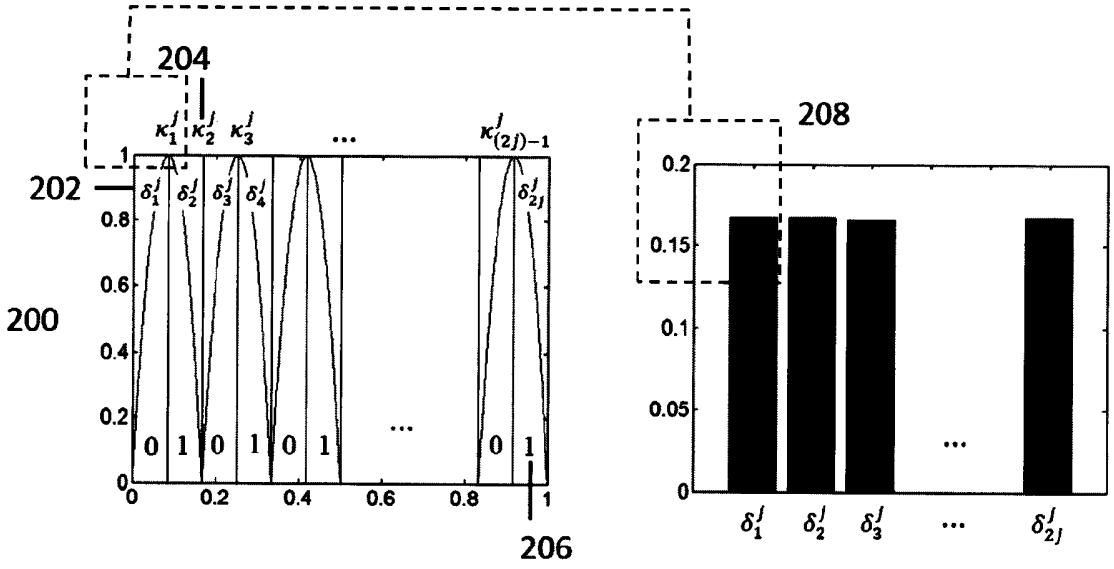

Figura 2

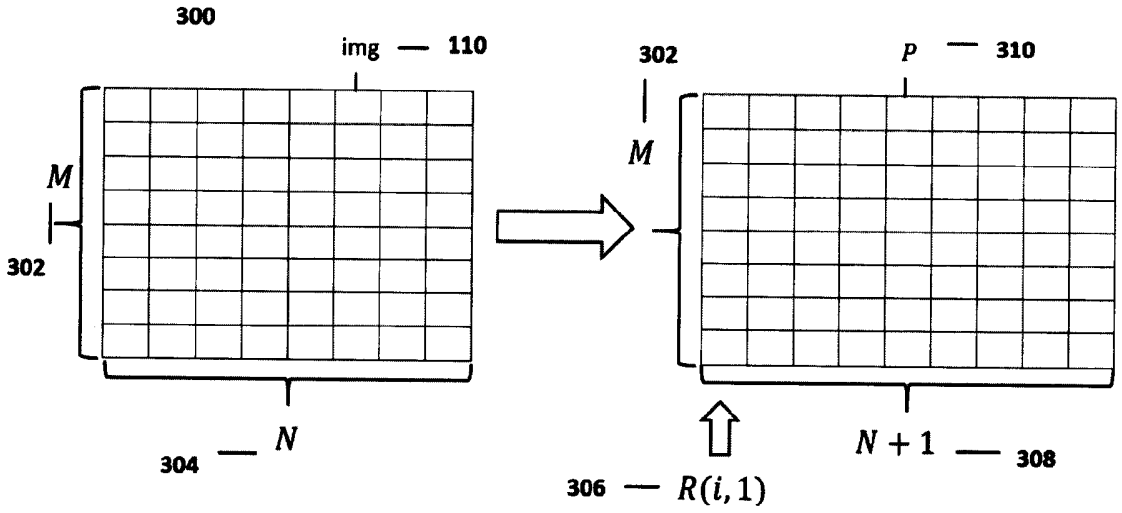

Figura 3

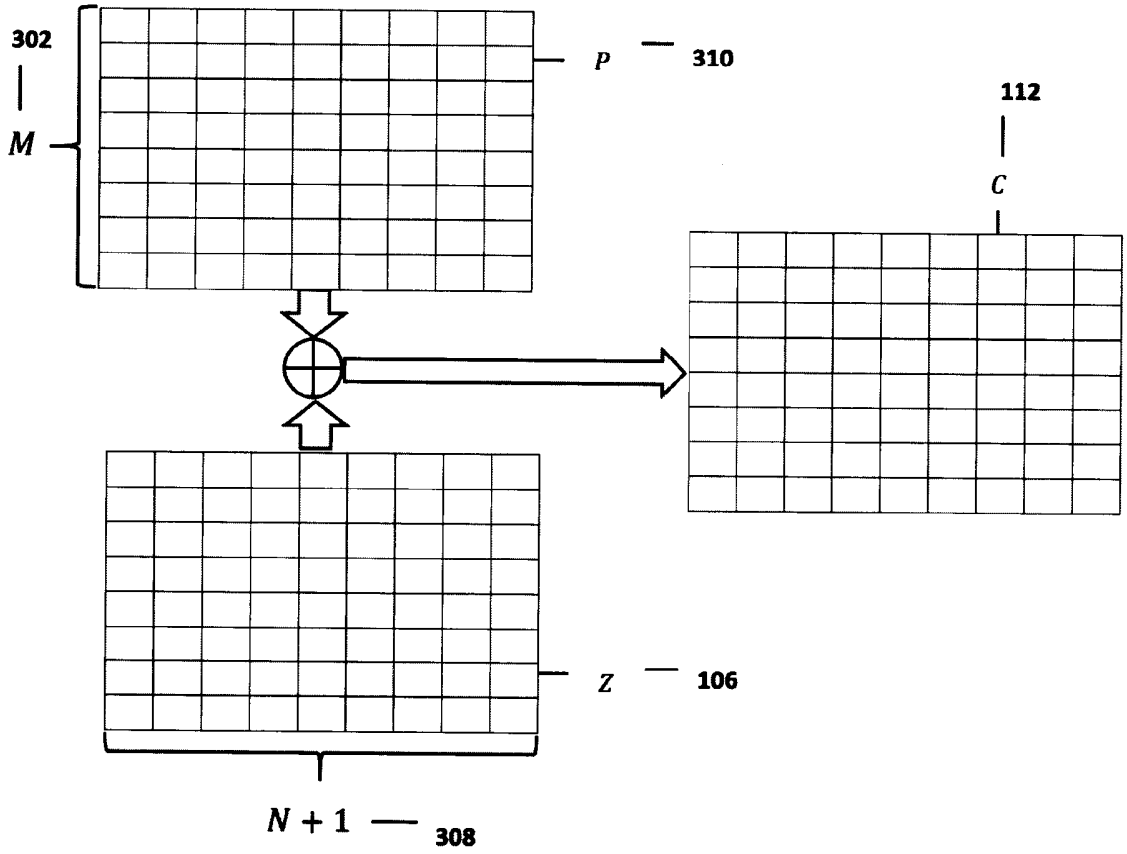

Figura 4

Figura 5

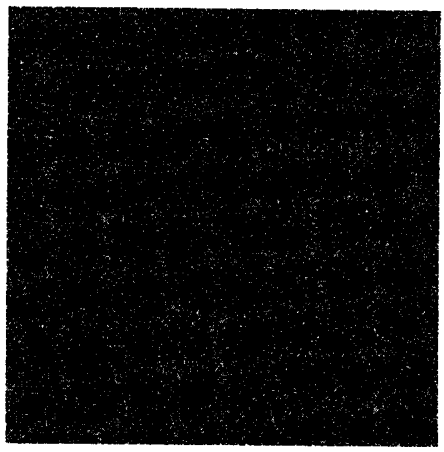

Figura 6

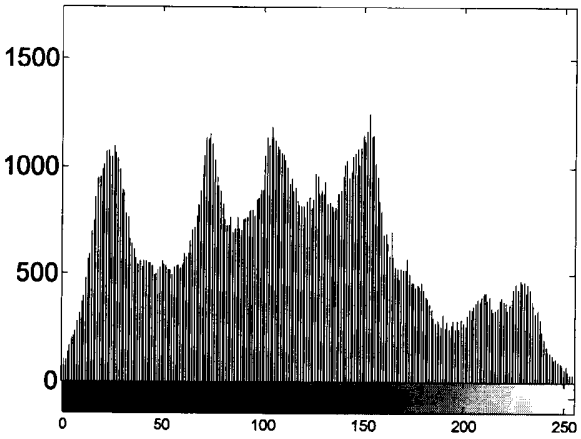

Figura 7

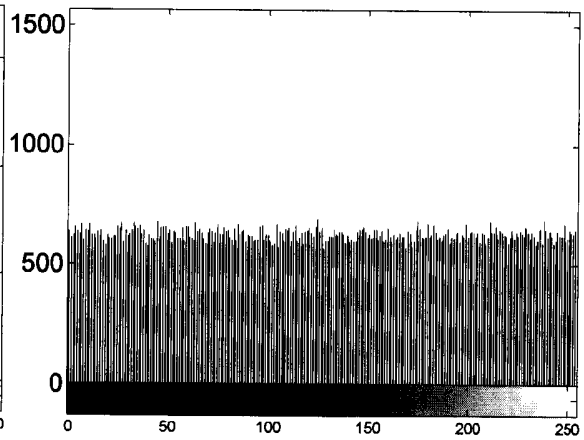

Figura 8

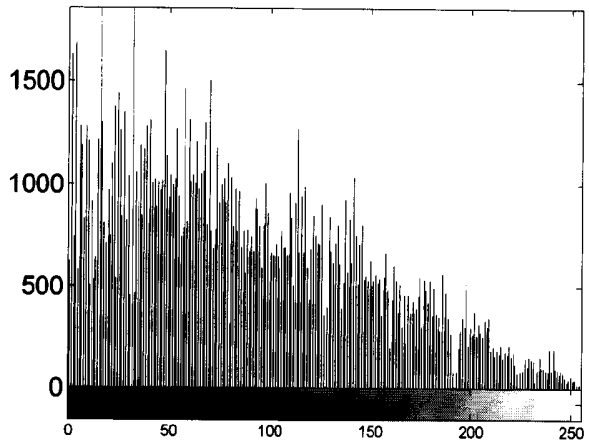

Figura 9