INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD CULHUACÁN

Sección de Estudios de Posgrado e Investigación

“Cifrador Caótico de Bloques Usando el Mapeo Logístico”

T E S I S

QUE PARA OBTENER EL GRADO DE MAESTRO EN CIENCIAS DE INGENIERÍA EN MICROELECTRÓNICA

P R E S E N T A:

ING. ERIC IBARRA OLIVARES

ASESOR:

DR. RUBÉN VÁZQUEZ MEDINA

SIP-14

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México D. F., siendo las 18:00 horas del día 16 del mes de junio del 2009 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de SEPI-ESIME-CULH. para examinar la tesis de titulada:

"Cifrador Caótico de Bloques usando Mapeo Logístico"

Presentada por el alumno:

IBARRA	OLIVARES	ERIC
Apellido paterno	Apellido materno	Nombre(s)

Con registro: A 0 7 0 1 8 6

aspirante de:

Maestría en Ciencias de Ingeniería en Microelectrónica

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***SU APROBACIÓN DE LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de tesis

Dr. Rubén Vázquez Medina

Dr. Miguel Cruz Irisson

Dr. Gonzalo Isaac Duchén Sánchez

Dr. Volodymyr Ponomaryov

Dr. José Alejandro Díaz Méndez

INSTITUTO POLITÉCNICO NACIONAL ESTADOS UNIDOS MEXICANOS
S. E. P.
SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION
ESIME CULHUACAN

EL PRESIDENTE DEL COLEGIO

Dr. Héctor Manuel Pérez Meana

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 16 del mes JUNIO del año 2009, el (la) que suscribe Ing. Eric Ibarra Olivares alumno (a) del Programa de Maestría en Ciencias de Ingeniería en Microelectrónica con número de registro A070186, adscrito a SEPI ESIME-CULH., manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Rubén Vázquez Medina y cede los derechos del trabajo intitulado CIFRADOR CAÓTICO DE BLOQUES USANDO EL MAPEO LOGÍSTICO, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección eric.ibarraolivares@gmail.com Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Eric Ibarra Olivares

Nombre y firma

ÍNDICE

ÍNDICE DE FIGURAS

PRESENTACIÓN DE LA TESIS

La Teoría del Caos se ha extendido en usos y aplicaciones a muchas ramas de la ciencia y ha sido una alternativa en la búsqueda de la seguridad de la información encontrando bastante aceptación en al área de la criptografía, particularmente, en los procesos de cifrado por bloques.

En este trabajo se presenta la realización de un algoritmo de cifrado por bloques cuya función de transformación corresponde a la función logística y cuya estructura general está definida por una red de Feistel desbalanceada, es decir, se presenta la realización y evaluación de *un algoritmo de cifrado por bloques caótico*.

Este algoritmo de cifrado se ha evaluado empleando conceptos de la teoría de la información como son la entropía del mensaje de entrada y de salida, la información mutua y la distribución estadística. Para valorar la aleatoriedad manifiesta en la distribución estadística se hace uso de las pruebas estándares de valoración de aleatoriedad del NIST[1]. Finalmente, se hace una comparación del algoritmo desarrollado con otros algoritmos de cifrado de bloque de uso comercial.

[1] El conjunto de pruebas estadísticas del NIST es un paquete estadístico que consta de 16 pruebas que fueron desarrolladas para probar la aleatoriedad de secuencias binarias de tamaño arbitrario producidas por hardware o software criptográfico basado en números aleatorios o seudo aleatorios.

RESUMEN

Esta tesis muestra el diseño, la construcción y evaluación de un algoritmo de cifrado de bloques basado en la función logística como sistema dinámico caótico. El algoritmo opera como cifrador de bloques de texto claro, con bloques de tamaño 64 bits, y una llave de 64 bits de longitud. La función no lineal que se usa en la red desbalanceada de Fesitel emplea como función básica la función logística. Posteriormente, se calcula la entropía del archivo a la entrada (plaintext) y se compara con la entropía del archivo a la salida (ciphertext) como medida de difusión en el proceso de cifrado, según se explica en el capítulo IV. Finalmente se compara la fortaleza del algoritmo propuesto y la de otros algoritmos de cifrado de bloques (DES, TRIPLE DES, AES, BLOWFISH, etc.) con base en el criterio de seguridad de Shannon.

ABSTRACT

This work presents the design, construction and evaluation of a block's cipher based on logistic equation as chaotic dynamical system. The algorithm operates on 64-bit plaintext blocks, and the key is 64 bits long. A nonlinear function known as logistic equation is used as round transformation. As the algorithm is a Feistel network, it can be used for both encryption and decryption. Afterward, the entropy of the input file known as plaintext is calculated and compared with the entropy of the exit file as process's measure diffusion. Finally compares the strength of proposed algorithm and others like DES, TRIPLE DES, AES, BLOWFISH, etc., with a based approach to Shannon' security.

DEDICATORIA

AGRADECIMIENTOS

A Dios:
En él está la sabiduría y el poder;
Suyo es el consejo y la inteligencia.

Al IPN:
Por darme la oportunidad
de formarme como investigador.

Al CONACYT:
Por facilitar los recursos necesarios
para que esta tesis se pudiera llevar a cabo.

CAPÍTULO I: MARCO DE REFERENCIA

Resumen

Este capítulo establece el marco teórico de referencia de esta tesis, ya que presenta la definición del problema específico que se ha planteado para la construcción de un algoritmo criptográfico de bloque. Se establecen las razones por las que es importante resolver el problema definido y se destacan las posibles ventajas. Seguido a lo anterior, se define el objetivo general y los objetivos específicos que se tendrán que cumplir al término de esta tesis. Posteriormente se presenta una breve introducción a la Teoría del Caos y una reseña de los principales trabajos que se han reportado en la literatura especializada relacionados con la aplicación de dicha teoría en procesos de cifrado.

Definición del problema

Con la proliferación de las redes de comunicaciones, la criptografía asume especial importancia, ya que es la ciencia que contribuye a la protección de la confidencialidad e integridad de la información durante su comunicación en canales y redes bajo condiciones hostiles de inseguridad. Comúnmente la criptografía se usa para la protección de datos que deben ser comunicados y/ó almacenados, lo que permite proteger las comunicaciones clasificadas, tales como las transferencias electrónicas de fondos bancarios.

Las actuales técnicas criptográficas normalmente se basan en la teoría de números o en algoritmos algebraicos. La teoría del caos es otro paradigma que parece prometedor. El caos es una rama del campo de la dinámica no lineal y ha sido ampliamente estudiado, encontrando un sin número de aplicaciones en diferentes áreas de la ciencia como por ejemplo; la economía, al estudiar el comportamiento de los mercados financieros [Nieto de alba, 2008], la medicina, en el estudio del sistema inmunitario humano [A.L. Goldberg, 2008], la biología, en el estudio de encimas y hormonas sujetas a la dinámica caótica [R.M., May, 2008], etc. Un gran número de aplicaciones en sistemas reales se desarrollan y estudian con base en sistemas dinámicos y teoría del caos como es el caso de los osciladores caóticos [B. Rubén, C. Isaac, Campos. Eric, 2006]. El comportamiento caótico es un sutil comportamiento de un sistema no lineal que parece ser aleatorio. Sin embargo, esta aleatoriedad no tiene un origen estocástico, es puramente derivado de la definición de un proceso determinista aunque muy sensible a las condiciones iniciales del sistema, de acuerdo con la definición dada por Devaney[2].

[2] Devaney's Definition of Chaos: Sea X un espacio métrico. Un mapeo continuo $f: X \rightarrow X$ se dice ser caótico en X si

1. f es transitiva
2. Los puntos periódicos de f son densos en X
3. f presenta dependencia sensitiva a las condiciones iniciales

En esta tesis se aborda el problema de entender y aplicar las herramientas de mecánica estadística como el diagrama de bifurcación, la distribución estadística y el exponente de Lyapunov para diseñar, construir y evaluar un algoritmo criptográfico caótico de bloques basado en el modelo propuesto por Ljupco Kocarev a partir de la discretización y escalamiento del Mapeo Logístico.

Objetivo general

A partir de la discretización y escalamiento del mapeo logístico, diseñar, desarrollar y evaluar un algoritmo criptográfico caótico de bloques basado en el modelo de Feistel desbalanceado propuesto por Ljupco Kocarev. Este algoritmo deberá modificar satisfactoriamente la sintáctica, la semántica y la estadística de un mensaje, de manera que lo haga incomprensible para aquellos que no son los destinatarios. Además, este algoritmo será evaluado y comparado con algoritmos de cifrado de bloques (DES, TRIPLE DES, AES, BLOWFISH, etc.) a partir del criterio de seguridad de Shannon y las pruebas de aleatoriedad del NIST.

Justificación

El campo de la investigación en seguridad informática requiere de explorar nuevas teorías, como la teoría del caos, entre otras, para mejorar los procesos de aseguramiento de las comunicaciones y por lo tanto la información.

El uso de funciones caóticas ofrece una alternativa de seguridad en el diseño y desarrollo de procesos de cifrado, ampliando así, el panorama para los investigadores interesados en la materia.

El cifrado utiliza únicamente operaciones con bytes que pueden ser fácilmente implementadas en diferentes tipos de procesadores y hardware, manteniendo de esa manera un costo bajo para su implementación.

Estado del arte

Origen de la Teoría del caos

Edward Lorenz, un meteorólogo del MIT (Massachussets Institute of Technology), trabajaba a principios de los 60 en un modelo de predicción del tiempo. Para ello había desarrollado un sistema de doce ecuaciones, que supuestamente reflejaban el comportamiento del sistema, y lo simulaba con la ayuda de una computadora. Ésta no poseía ni la velocidad ni la memoria precisa para proporcionar una simulación realista de la atmósfera y los océanos terrestres. A pesar de ello, la máquina señalaba cada minuto el paso de un día, imprimiendo una hilera de números en un papel. La máquina no predecía el tiempo, sino que predecía cómo sería probablemente el tiempo. Lorenz estudiaba las pautas que aparecían y desaparecían en la atmósfera: familia de mareas y ciclones, que obedecían siempre a reglas matemáticas aunque nunca se repitiesen. Lo esencial era ver como cambiaban estas pautas en el transcurso de las horas. Las doce ecuaciones del sistema de Lorenz eran ecuaciones que expresaban los nexos entre la temperatura y la presión, y entre la presión y la velocidad del viento, etc.

La simulación numérica del tiempo era necesaria, y para ello era imprescindible una computadora. Los vientos y las temperaturas que surgían en el papel, respondían lo que para Lorenz era una intuición: el tiempo se repetía, mostrando pautas en las que la presión subía y bajaba, y que la corriente aérea se alteraba hacia el Norte y hacia el Sur. Sin embargo al repetirse las pautas, nunca se obtenían los mismos resultados. Lorenz inventó un rudimentario procedimiento de gráficas con el fin de que las pautas se apreciasen sin dificultad. Al representar una determinada variable, ésta recorría el papel en una trayectoria sinuosa que marcaba una larga serie de colinas y valles, aunque los ciclos reconocibles que aparecían y reaparecían nunca lo hacían de la misma manera.

Un día de 1961, Lorenz quiso examinar de nuevo una determinada sucesión, y para ahorrar papel y tiempo, introdujo en la computadora los números obtenidos directamente de la impresión anterior, pero con sólo tres decimales en lugar de los seis que había estado utilizando hasta entonces. La nueva pasada tenía que haber sido exactamente igual que la anterior, pero no fue así. Al principio Lorenz examinó los dos conjuntos de números y pensó que la máquina se había averiado como pasaba frecuentemente, hasta que de repente comprendió lo que había pasado: la divergencia se había producido por el hecho de redondear los decimales en las condiciones iniciales, convencido de que la diferencia (una milésima parte) era de poca importancia. Lorenz utilizaba un sistema de ecuaciones deterministas, es decir, dado un determinado punto de partida, su predicción del tiempo se desarrollaría siempre del mismo modo. Dado un punto de partida levemente distinto, el tiempo se desarrollaría de modo ligeramente diferente, pero en el sistema de ecuaciones de Lorenz los errores ínfimos fueron catastróficos. Quiso comprobar cómo divergían las dos pautas, y copió una de ellas en una hoja transparente superponiendo ésta sobre la otra. Las dos pautas empezaban aparentemente en el mismo punto, y cada vez más divergían una de la otra en el transcurso de la gráfica, hasta que desaparecía cualquier semejanza. Lo que a

Lorenz le dejaba perplejo era, cómo teniendo un sistema determinista, el hecho de variar un poco las condiciones iniciales supusiera unos resultados tan diferentes. Siendo fiel a su instinto matemático, descartó que la máquina estuviese averiada, y empezó a comprender que algo nuevo estaba pasando, algo que nadie había descubierto hasta ahora. Sin saberlo, había nacido la Teoría del Caos.

Al efecto que tienen los pequeños cambios en las condiciones iniciales, se le dio el nombre de efecto mariposa. Las predicciones meteorológicas, aunque estadística pura, eran mejor que nada. Pero transcurridos dos o tres días los pronósticos se convertían en especulaciones, y pasados seis o siete pasaban a ser despreciables. La razón de ello era el efecto de la mariposa. Cualquier predicción se deteriora (cambio en las condiciones iniciales), y los errores se multiplican, con lo que los resultados reales cambian completamente. Y todo esto debido al efecto de la mariposa, que adquirió un nombre técnico: dependencia sensitiva de las condiciones iniciales.

Lorenz abandonó las predicciones y se dedicó cada vez más a los sistemas que jamás alcanzaban estabilidad, sistemas que casi se repetían pero que nunca lo hacían. Estos sistemas abundan en la naturaleza: poblaciones de animales que se multiplican y se reducen con regularidad, o epidemias que van y vienen con puntualidad, etc. Lorenz buscó formas más sencillas de producir este comportamiento complejo, y encontró un sistema de tres ecuaciones únicas con tres variables. Eran ecuaciones no lineales, es decir, expresaban relaciones no proporcionales entre las variables. Las ecuaciones lineales son resolubles, se pueden desmontar y montar, mientras que las ecuaciones no lineales, en general, son insolubles e indesmontables. Lorenz se inspiró en la dinámica de fluidos para sus tres ecuaciones, y en concreto en el movimiento de un gas o líquido caliente, lo que se conoce técnicamente como la convección de Rayleigh-Bernard.

Consideremos una celdilla de fluido, consistente en una caja de fondo liso que se puede calentar y una tapa, también lisa, que se puede enfriar. Si esta celdilla se calienta por debajo, el líquido o gas tiende a organizarse en giros cilíndricos. El fluido caliente se eleva por un lado, pierde temperatura y desciende por el lado opuesto, según la siguiente figura (a la izquierda):

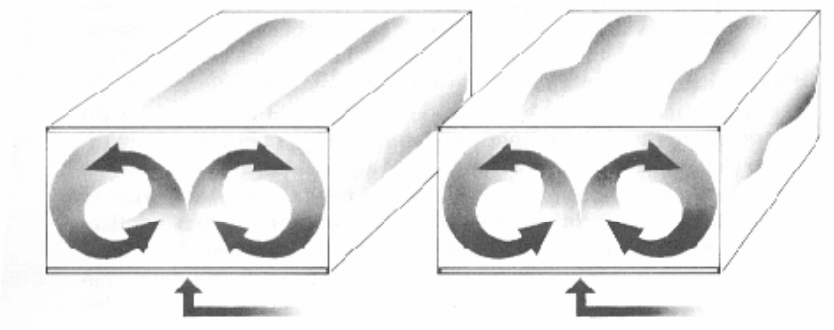


Figura 1 Convección simple de un fluido en una caja

Si se intensifica el calor aparece la inestabilidad, y los giros exhiben un temblor que recorre adelante y atrás la longitud de los cilindros. A temperaturas más elevadas, la corriente se desordena y se hace más turbulenta (Figura 1.2, derecha). Aunque no cumpliese punto por punto el modelo de la convección, el sistema de Lorenz tenía analogías importantes. Abstraían sólo un rasgo de la convección tal como se produce en el mundo: el movimiento circular del fluido caliente, que ascendía y volteaba como una noria. Precisamente a partir del anterior esquema, el siguiente sistema es descrito con precisión por las tres ecuaciones de Lorenz: la rueda de agua o la noria de Lorenz, analogía mecánica del círculo rotante de la convección:

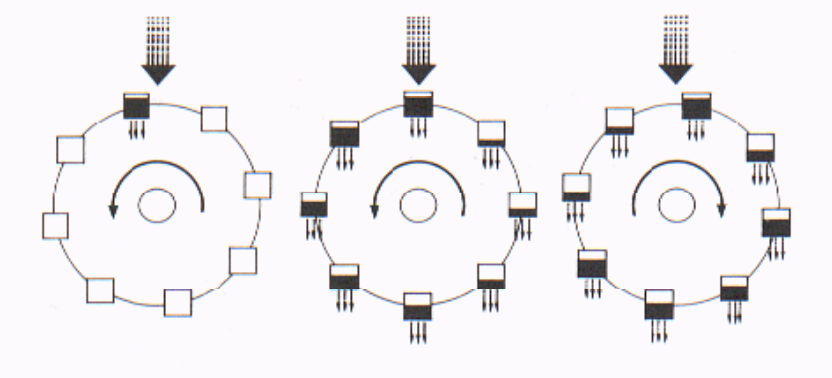


Figura 2 La noria de agua de Lorenz es un ejemplo sencillo de cómo un sistema puede realizar un movimiento caótico

El agua cae en la parte superior continuamente en cubos situados en el borde de la rueda, y cada uno se vacía por un agujerito. Si la corriente de agua es lenta, los cubos más elevados nunca se llenan con la plenitud necesaria para vencer la fricción; cuando es rápida los cubos se llenan rápidamente y el peso hace que la noria gira, y la rotación tiene la posibilidad de hacerse continua.

Si el caudal es muy veloz, los cubos llenos de agua dan la vuelta hasta el fondo y se remontan por el lado contrario, con lo que la rueda empieza a menearse despacio hasta detenerse e invertir su rotación, yendo primero en un sentido y luego en el otro. Aparentemente si la corriente de agua no varía, se creará un estado estable. La noria girará con regularidad u oscilará regularmente adelante y atrás, primero en un sentido y luego en el otro a intervalos constantes.

Al simular Lorenz con la computadora este sistema, éste dio un resultado sorprendente. Con el objeto de obtener una imagen con aquellos datos, Lorenz empleó cada variable, es decir, cada noria, como coordenadas en un espacio tridimensional. Así la secuencia numérica produjo una serie de puntos que trazaba una trayectoria continua mostrando una complejidad infinita. Permanecía siempre dentro de ciertos límites y nunca se repetía una misma trayectoria.

Reveló una configuración extraña, algo por el estilo a una espiral doble en tres dimensiones, como una mariposa con un par de alas que denotaba un desorden puro. Esta curva misteriosa se conoció como atractor de Lorenz:

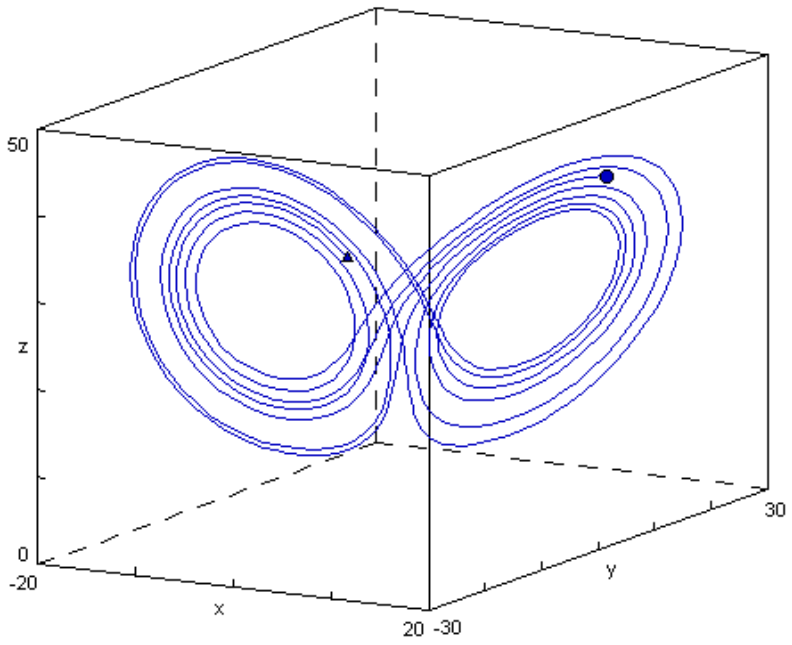


Figura 3 Atractor de Lorenz

En un instante dado, las tres variables señalan la situación de un punto en un espacio tridimensional. Como el sistema nunca se repite de modo exacto, la trayectoria jamás se corta a si misma. En lugar de ello, describe curvas una y otra vez para siempre. De esta manera el traslado de un ala del atractor a otra, corresponde a una inversión de la dirección del giro en la noria.

Los valores cambiantes de cualquier variable se pueden mostrar a través de la serie temporal:

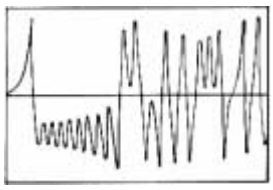


Figura 4 Serie de tiempo para el atractor de Lorenz

Examinando la gráfica, lo que obtenemos es una serie con una sucesión de picos sin ningún tipo de regularidad o periodicidad. Supongamos que un ala del atractor corresponde a la parte positiva de la serie temporal, y que la otra ala del atractor es la parte negativa. Si analizamos la gráfica de la Figura 4, cada vez que nos encontramos con una sucesión de picos, o bien positivos o negativos, implica que dentro del atractor estamos girando alrededor de un ala sin pasar a la otra. Cuando pasamos de un pico positivo a otro negativo o viceversa, es que hemos pasado de un ala del atractor a la otra.

Caos y criptografía

Las redes modernas de telecomunicaciones especialmente Internet y las redes de telefonía móvil han extendido enormemente los límites y posibilidades de comunicaciones y transmisión de la información. Asociado a este rápido crecimiento existe una creciente demanda de técnicas criptográficas, cuyo uso ha generado intensas actividades de investigación en el estudio de la criptografía [Stinson, 1995; Menezes *et al*., 1997]. Desde la década de los 90's muchos investigadores han notado que existe una importante relación entre el caos y la criptografía; muchas propiedades de los sistemas caóticos tienen su correspondiente contraparte en los criptosistemas tradicionales. La siguiente tabla contiene una lista parcial de estas propiedades.

Propiedad caótica	*Propiedad criptográfica*	*Descripción*
Ergodicidad[3]	**Confusión**[4]	**La salida tiene la misma distribución para cada entrada.**
Sensibilidad a las condiciones iniciales	**Difusión**[5] **con un pequeño cambio en el texto plano/llave secreta.** **Difusión con un pequeño cambio en un bloque de texto plano de todo el espacio de texto plano.**	**Una pequeña desviación en la entrada puede causar un cambio considerable en la salida.** **Una pequeña desviación en el área local puede causar un cambio considerable en el espacio completo.**
Dinámica determinista	**Seudo-aleatoriedad determinista.**	**Un proceso determinístico puede causar cierto comportamiento aleatorio (seudo-aleatorio).**
Estructura compleja	**Complejidad**[6] **del algoritmo (ataques).**	**Un proceso simple presenta una muy alta complejidad.**

Es interesante señalar que esta estrecha relación puede encontrarse en el documento clásico de Shannon sobre criptografía [1949]:

"Un buen proceso de transformación normalmente está compuesto por productos repetidos de dos simples operaciones no conmutativas. Hopof ha demostrado, por ejemplo, que la masa de un pastel se puede mezclar llevando a cabo una secuencia de operaciones. Primero se extiende la masa sobre una base, después se enrolla y luego se extiende nuevamente."

[3] *Ergodicidad* es la propiedad en la que una trayectoria en el espacio fase viene arbitrariamente cerca de sus estados anteriores. La trayectoria de un sistema caótico en su recorrido evolutivo también satisface esta propiedad. Esencialmente refleja que el sistema eventualmente se limita a un objeto espacial, un conjunto de puntos llamado atractor. La densidad de tales puntos es el tiempo invariante y esta propiedad es esencial en criptografía.

[4] La *confusión* oculta la relación entre el texto plano y el texto cifrado eliminando los intentos de obtener redundancia, así como patrones estadísticos.

[5] La *difusión* consiste en disipar la redundancia del texto plano distribuyéndola a lo largo del texto cifrado. Un criptanalista buscando dichas redundancias invertirá mucho tiempo y esfuerzo tratando de encontrarlas.

[6] La complejidad de un algoritmo está determinada por el poder computacional necesario para ejecutarlo. La complejidad computacional de un algoritmo comúnmente es medida por dos variables: *T* (para la complejidad del tiempo) y *S* (para la complejidad del espacio, o requerimientos de memoria). Tanto *T* como *S* comúnmente se expresan como funciones de *n*, donde *n* representa el tamaño de la entrada. (Existen otras medidas de complejidad: el número de bits aleatorios, las comunicaciones de banda ancha, la cantidad de datos, etc.)

Del párrafo anterior, es claro que Shannon en realidad discutió una típica ruta hacia el caos vía extendiendo y enrollando, la cual es bien conocida en la teoría actual del caos [Devaney, 1989]. En realidad, es una forma común de emplear uno o más transformaciones no lineales para diseñar un moderno sistema de cifrado, donde las transformaciones no lineales pueden ser consideradas en sus modalidades de tiempo discreto y valores discretos para algunos sistemas caóticos.

Como un esfuerzo útil, recientemente la dinámica caótica del algoritmo AES ha sido estudiada [Ruggiero *et al*., 2004; Kocarev *et al*., 2004]. Como resultado de esta investigación, ha surgido una gran variedad de criptosistemas basados en caos para las comunicaciones punto a punto [Hasler, 1998; Álvarez *et al*., 1999b; Silva & Young, 2000; Kocarev, 2001; Li, 2003; Yang, 2004; Li, 2005ª, b].

Existen dos enfoques principales para el diseño de criptosistemas basados en caos: analógicos y digitales. La mayoría de los criptosistemas analógicos basados en caos son esquemas seguros de comunicaciones diseñados para canales ruidosos, basados en la técnica de sincronización de caos [Pecora & Carroll, 1990]. La sincronización de caos es una técnica desarrollada desde la década de los años 90's. De manera muy estricta, significa que dos sistemas caóticos pueden sincronizarse uno respecto al otro bajo la guía de una (o más) señal escalar, la cual generalmente es enviada de un sistema a otro. Existen muchos tipos diferentes de sincronización de caos como son: sincronización completa, sincronización generalizada, sincronización impulsiva, sincronización por fase, sincronización proyectiva, sincronización por intervalos, sincronización por inducción de ruido, etc., esto debido a las diferentes definiciones matemáticas para la sincronización de caos [Boccaletti *et al*., 2002]. En los criptosistemas basados en la sincronización de caos, la información puede ser transmitida por una o más señales caóticas de varias maneras, incluyendo (no necesariamente) las siguientes;

- Enmascaramiento caótico [Kocarev *et al*., 1992; Wu & Chua, 1993; 18orgue & Feki, 1999; Cuomo *et al*., 1993; Shahruz *et al*., 2002; Memon, 2003], donde la señal analógica del mensaje *m(t)* se añade a las salida del generador caótico *x(t)*, dentro del trasmisor.

- Intercambio caótico o intercambio de llaves caótico (CSK) [Dedieu *et al*., 1993; Parlitz *et al*., 1992], donde una señal binaria del mensaje se usa para escoger la señal portadora de dos o más atractores caóticos diferentes.

- Modulación caótica [Halle *et al*., 1993; Cuomo & Openheim, 1993; Chen *et al*., 2003; Yang & Chua, 1996], donde el mensaje modula un parámetro del generador caótico ó cuando se usan técnicas de espectro disperso para multiplicar la señal del mensaje por la señal portadora caótica.

- Métodos de control de caos [Hayes *et al*., 1993, 1994; Lai *et al*., 1999], donde pequeñas perturbaciones causan la dinámica simbólica de un sistema caótico para realizar el seguimiento de una secuencia de símbolos prescrita.

- Sistema inverso de aproximación [Feldmann *et al*., 1996; Zhou & Ling, 1997b], donde el sistemas receptor es diseñado de manera inversa con el objetivo de asegurar la recuperación de la señal cifrada.

A pesar de los métodos usados para transmitir la señal del mensaje, el generador caótico receptor debe sincronizarse con el transmisor con el propósito de generar la señal portadora caótica *x(t)*, y así, recuperar el mensaje *m(t)* por medio de una separación de señales. Para consultar los trabajos más relevantes y recientes sobre criptosistemas analógicos basados en caos se puede consultar [Hasler, 1998; Álvarez *et al*., 1999b; Silva & Young, 2000; Yang, 2004; Li, 2005ª].

Por otro lado, los criptosistemas digitales basados en caos (también llamados cifrados caóticos digitales) están diseñados para las computadoras digitales donde uno o más transformaciones caóticos son implementados en informática de precisión finita para cifrar el mensaje de texto claro de varias maneras como son las siguientes:

- Cifrado por bloques basado en el uso de iteraciones caóticas ida/vuelta
  - Habutsu, *et al*., 1991;
  - Fridrich, 1998;
  - Uís *et al*., 1998;
  - Masuda & Aihara, 2007.

- Cifrados de flujo caóticos basados en la generación números seudo-aleatorios (PRNG)
  - Wolfram, 1985;
  - Matthews, 1989 ;
  - Berstein & Liberman, 1991 ;
  - Zhou & Ling, 1997a ;
  - Li *et al*., 2001 ;
  - Lee *et al*., 2007.

- Cifrado por bloques basado en la iteración de funciones caóticas o S-cajas
  - Kocarev *et al*., 1998 ;
  - Guo *et al*., 1999 ;
  - Jakimoski & Kocarev, 2001 ;
  - Papadimitriou *et al*., 2001 ;
  - Tang *et al*., 2007.

- Cifrados de flujo caóticos vía sistemas inversos de aproximación (con retroalimentación de texto cifrado)
    - Frey, 1993;
    - Zhou & Ling, 1997b;
    - Zhou & FENA, 2000;
    - Lü *et al*., 2007.

- Sistemas de cifrado caótico basados en la búsqueda de bits de texto plano en una secuencia caótica seudo-aleatoria
    - Baptista, 1998 ;
    - Álvarez *et al*., 1999a ;
    - Wong *et al*., 2001 ;
    - Li *et al*., 2004c ;
    - Huang & Guan, 2005 ;
    - Xiao *et al*., 2007.
    - Peng, J., 2008

El cifrado digital no requiere de la sincronización de caos en absoluto. En lugar de ello, normalmente usan uno o más transformaciones caóticas en los que las condiciones iniciales, así como los parámetros de control, juegan el papel de la llave secreta. Para una mejor comprensión de los criptosistemas digitales basados en caos ver [Álvarez *et al*., 1999b; Silva & Young, 2000; Kocarev, 2001; Li, 2003, 2004, 2005; Wei J., 2006; Aono, Shuichi., 2007; Peng, J., 2008].

CAPÍTULO II: MAPEO LOGÍSTICO

Resumen

En este capítulo se estudia el Mapeo Logístico, el cual se define como un sistema dinámico discreto determinista y unidimensional, es decir, el tiempo tiene valores discretos y sus valores son enteros positivos. Se muestran algunas propiedades importantes del Mapeo Logístico tales como sensibilidad a las condiciones iniciales y a la variación del parámetro que gobierna su comportamiento. Se describe dicho mapeo a través del uso de las herramientas de mecánica estadística tales como el diagrama de bifurcación, el diagrama de trayectorias y el exponente de Lyapunov. Finalmente, se muestra como este mapeo exhibe buenas propiedades estadísticas que se pueden aprovechar para procesos de cifrado.

Definición del Mapeo

Caos

El estudio de la dinámica caótica en los sistemas deterministas se ha vuelto muy popular en las últimas décadas, surgiendo del estudio de la dinámica no lineal. Esto debido a los asombrosos descubrimientos que su estudio ha arrojado. Por supuesto, sería natural pensar que si un sistema es determinista, su comportamiento sería fácil de predecir. Pero hay sistemas en los que su comportamiento parece no ser predecible, no por falta de determinismo, sino debido a que la complejidad de la dinámica requiere una precisión incapaz de implementarse en una computadora. Esto puede observarse en sistemas donde condiciones iniciales muy similares arrojan comportamientos totalmente diferentes. Así, supóngase que tenemos los estados iniciales 2.1234567890 y 2.1234567891, y después de algún tiempo el sistema tomará los valores 3.5 para el primer caso y 1.7 para el segundo caso. Entonces, no importa cuanta precisión tengamos ya que la más mínima diferencia tenderá, con el transcurso del tiempo, a arrojar resultados muy diferentes. Esto debido a que existe una divergencia exponencial entre las trayectorias del sistema (La demostración formal se puede obtener calculando los exponentes de Lyapunov como se verá en la sección correspondiente al tema análisis de estabilidad de este mismo capítulo.

Otra propiedad importante de los sistemas caóticos son atractores extraños[7]. Cuando se piensa en la dinámica caótica, es fácil asumir que la dinámica no sigue ningún patrón, pero si miramos cuidadosamente existe un patrón sorprendente, de tal manera que los estados no se repiten, sino que se concentrarán en un área determinada del espacio de los estados.

[7] Un atractor es una parte del espacio de estado (el cual es el conjunto de todos los posibles estados de un sistema) en el que se enfoca la dinámica. Por ejemplo, un simple péndulo con fricción tiene un atractor estable en la parte inferior del eje vertical, y no importando la ubicación del péndulo este terminara en algún momento en ese punto. Pero existe un atractor inestable en la parte superior del eje vertical, ya que si este es la condición inicial, en teoría el péndulo se quedaría ahí, pero la más mínima perturbación movería el péndulo fuera del punto estable o atractor estable. La inestabilidad del atractor repele la dinámica del sistema.

Estas propiedades de los atractores extraños son sorprendentes, ya que son auto afines (Mandelbrot, 1998) (una parte del atractor se asemeja al atractor), y por lo tanto fractales.

El Mapeo Logístico

Se pueden introducir los primeros conceptos de la teoría del caos, analizando el comportamiento de sistemas no lineales discretos mediante aplicaciones iteradas, ya que estas son el ejemplo más simple de un sistema no lineal. Consisten en elegir un número cualquiera como dato de entrada de una función determinada, y el resultado obtenido, utilizarlo como dato de entrada en la misma función en la siguiente iteración.

Una función iterada interesante y popular es la llamada función logística. La función logística surge como un modelo de crecimiento para algunas poblaciones de insectos basado en el modelo de crecimiento exponencial $X_{n+1} = \mu X_n$ pero con un factor adicional del lado derecho de la función $(1 - X_n)$. Fue popularizada en 1976 [May R. M., 1976]. La aplicación logística está dada por la siguiente expresión:

$$X_{n+1} = \mu X_n (1 - X_n), \ \mu \in [0,4], \ \ X \in [0,1] \qquad (1)$$

Donde μ es un parámetro, que en general varía entre 0-4, y que representa el suministro de alimento o fecundidad de una especie; en términos técnicos, una razón de crecimiento que puede situarse más alta o más baja. X_n representa el número de individuos de la población en la *n*-enésima generación.

Se Puede observar que esta función es una parábola que intersecta el eje x en los puntos (0,0) y (1,0).

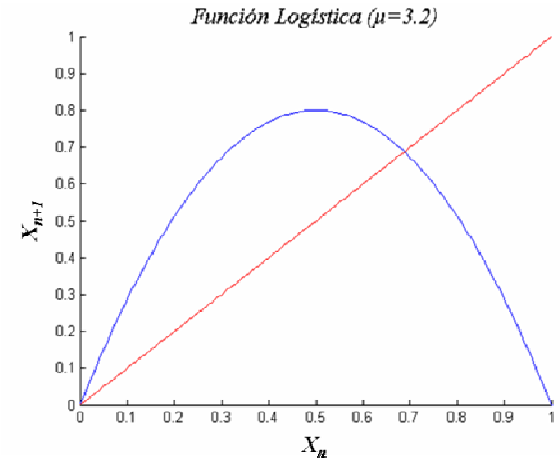


Figura 5 Parábola de la Función Logística

Por lo tanto, el Mapeo Logístico queda definido por la familia de parábolas que pasan por el origen y tienen su máximo en $X = ½$ con un valor de $μ/4$ de acuerdo a la expresión anterior.

Ahora, se aborda el comportamiento de este sistema, tenemos que si se toma un valor pequeño para X_n (un valor positivo muy cercano a cero), la cantidad $(1 - X_n)$ está cerca del 1, por lo que la ecuación (1) está muy cerca de μX_n, de este modo el crecimiento de la población se dará en proporción directa a μX_n. De manera similar, para valores relativamente grandes de X_n (valores cercanos al valor máximo) la cantidad $(1 - X_n)$ es pequeña (cercana a cero) o en otras palabras el crecimiento es pequeño.

La constante μ regula la pendiente y la altura de la parábola sobre la abscisa, ya que entre más grande sea μ, más alto será el pico de la parábola (por lo tanto más grande la población). La altura pico mide $μ/4$ unidades de X_{n+1}, por lo tanto como X_{n+1} se encuentra en el intervalo [0,1] y el valor de μ solo puede estar en el rango [0,4]. Si μ excede el valor de 4, las iteraciones de la función logística producirían valores que son imposibles, es decir, mayores que uno o menores que cero. Estas limitaciones en X_n, X_{n+1} y μ significan que para este modelo poblacional la parábola inicia suavemente del origen a un pico de máxima población en $X_n = ½$ y $X_{n+1} = μ/4$. Mientras X_n crece más allá de 0.5 la curva cae, es decir, la población retrocede.

El comportamiento del Mapeo Logístico como conjunto de curvas se muestra en la siguiente figura:

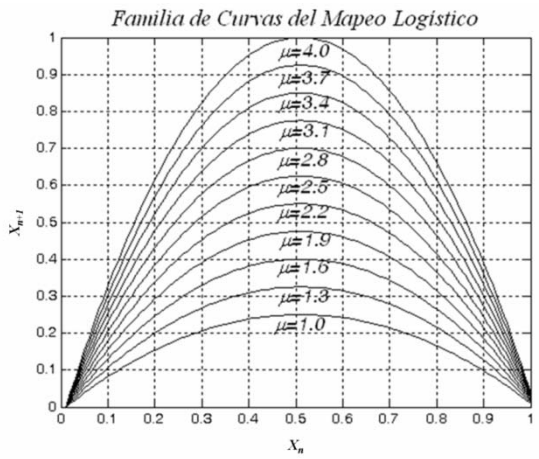


Figura 6 Familia de curvas del Mapeo Logístico

Diagrama de trayectorias

Para valores $0 \le \mu \le 4$, la altura de la parábola estará en el intervalo $[0..1]$. Si iteramos esta función observaremos la dinámica discreta de la población que modela la función.

Para iterar una función necesitamos un valor inicial x_0 y el resultado será la siguiente entrada de la función. Por lo tanto

$$X_1 = f(X_0)$$
$$X_2 = f(X_1) = f^2(X_0)$$
. . .
$$X_n = f(X_{n-1}) = f^n(X_0)$$

Donde x_n es la n-sima iteración de x_0. El conjunto de todas las iteraciones de una función es llamado el mapeo de la función.

Una manera fácil de visualizar la iteración de una función, es dibujando la línea recta $y = x$ (también llamada línea de identidad), y la función $f(x)$.
Si empezamos a dibujar líneas siguiendo los puntos (x_0, x_0), $(x_0, f(x_0) = x_1)$, (x_1, x_1), $(x_1, f(x_1) = x_2)$, (x_2, x_2), $(x_2, f(x_2) = x_3)$, … podemos ver que la iteración se puede hacer geométricamente trazando líneas desde $f(x)$ hasta la línea $y = x$ y de regreso a $f(x)$.

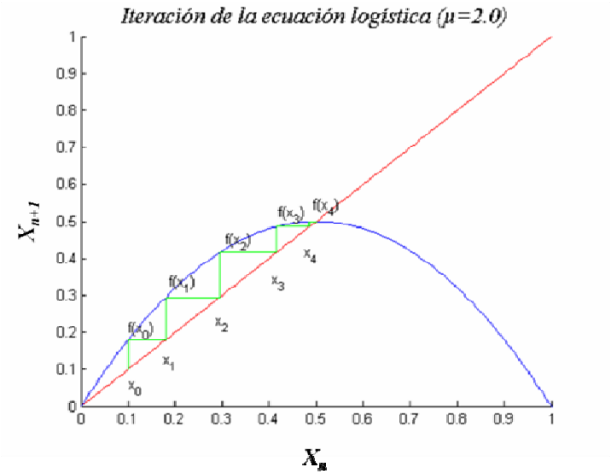


Figura 7 Iteración de la Función Logística

Para la función logística, si $0 \le \mu \le 4$ y $0 \le X_0 \le 1$ la dinámica estará delimitada en $[0..1]^2$. Observemos la dinámica en el Mapeo Logístico conforme incrementamos el valor de μ. Para $\mu = 0$, $f(x) = 0$, independientemente de x_0.

Si se aumenta un poco el valor de μ, se puede observar que después de algunas iteraciones, independientemente de x_0, la dinámica se ubicará en $x_n = 0$.

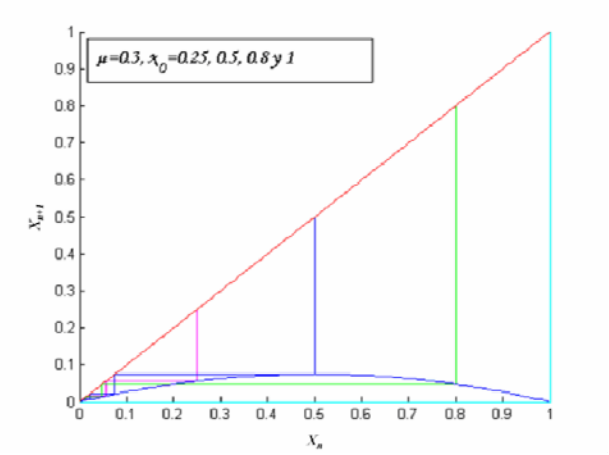


Figura 8 Iteración de la Función Logística

Para este caso, 0 es un atractor estable del mapeo. Si incrementamos un poco más el valor de μ, todavía observaremos este comportamiento.

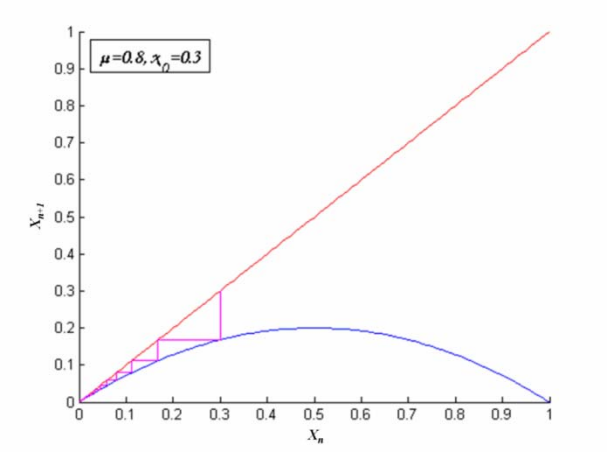


Figura 9 Iteración de la Función Logística

Incrementando aún más el valor, se puede observar que este comportamiento cambia en alguna parte.

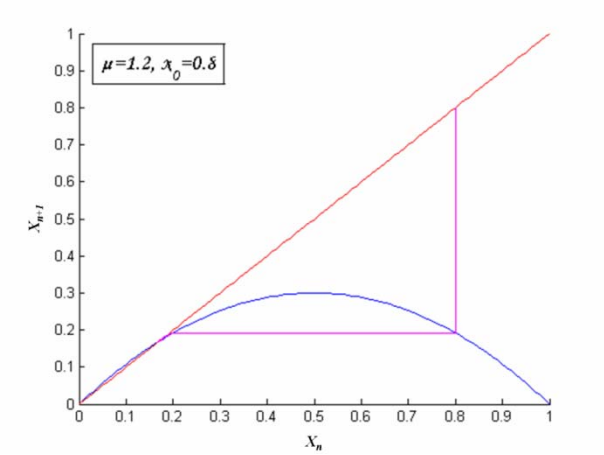


Figura 10 Iteración de la Función Logística

El valor del atractor ya no más es cero, pero si lo es la intersección de la parábola con la línea de identidad. En este punto surge una pregunta; ¿Donde ocurre este cambio?. Se puede observar que ocurre cuando la parábola intersecta la línea de identidad en otro punto que el orígen. Esta transición se da cuando $\mu = 1.0$.

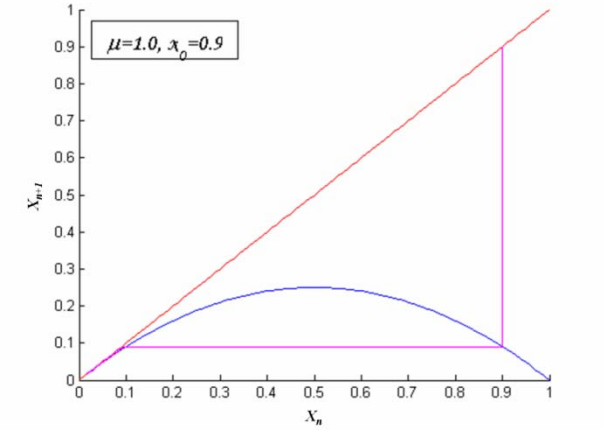


Figura 11 Iteración de la Función Logística

Para $\mu = 1$, el atractor 0 es asintótico. Esto significa que es alcanzado en el límite cuando $n \to \infty$. Pero incluso para los valores de $a > 1$, se pude observar que para valores de $x_0 = 0.0$ ó 1.0, 0 la dinámica se mantendrá en cero. Pero si x_0 es muy cercano a cero o uno, la dinámica será repelida. Se puede decir que cero se convierte en un atractor estable. Como un péndulo en posición vertical, si la condición inicial es el atractor instable, el sistema se mantendrá en ese estado, pero la más mínima perturbación alejará al sistema de esa fase inestable.

Si se incrementa el valor de μ, aún se percibirá el mismo comportamiento; la dinámica será atraída hacia la intersección de la parábola con la línea de identidad.

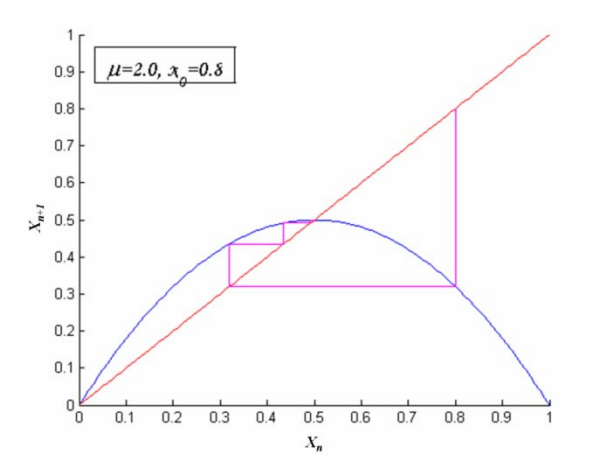


Figura 12 Iteración de la Función Logística

Pero el punto atractor está cambiando. ¿Cómo se puede calcular?. No es difícil. Un punto atractor satisface la ecuación $f(x_n) = x_n$, tan solo tenemos que encontrar las raíces de la ecuación

$$X_{n+1} = \mu X_n (1 - X_n)$$

Las cuales son $x = 0$, $x = 1 - 1/\mu$ Se puede observar que la primera es estable para $\mu \leq 1$ e inestable para $\mu > 1$. La segunda raíz esta fuera del intervalo $[0..1]$ cuando $\mu < 1$. Cuando $\mu = 1$, las dos raíces son las mismas, ¿Pero esta segunda raíz es siempre estable?

Si se incrementa μ, se puede ver que la dinámica converge al punto atractor $(1 - 1/\mu)$ más lentamente a medida que $\mu = 3.0$ y es precisamente en este punto que se observa algo similar que en $\mu = 1.0$

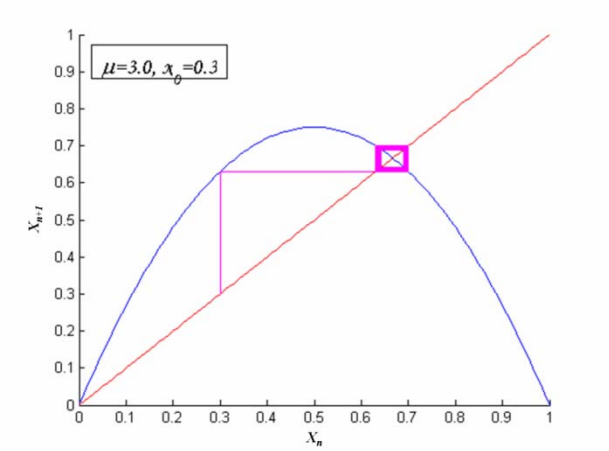


Figura 13 Iteración de la Función Logística

La dinámica tiende asintóticamente al atractor y lo alcanzará en el límite $n \to \infty$. ¿Qué pasa si se incrementa μ un poco más?

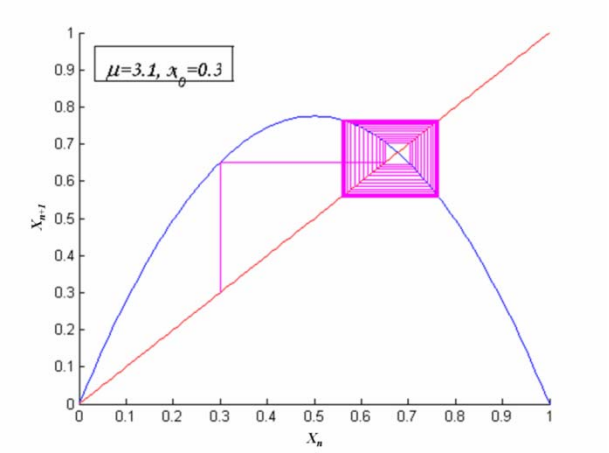


Figura 14 Iteración de la Función Logística

Se puede observar que el punto atractor se vuelve inestable tal como ocurrió con x_0 y ahora se tiene un atractor cíclico (también llamado órbita) de periodo 2. Esto implica que $x_n = x_{n-2}$ y puede ser más fácil de visualizar con $f^2(x_n)$:

$$f^2(x_n) = f(f(x_n)) = \mu(\mu x_n(1-x_n))(1-(\mu x_n(1-x_n)))$$

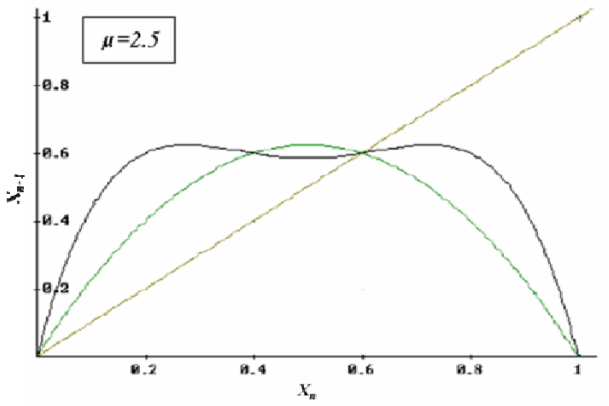


Figura 15 Periodo doble μ=2.5

Para $\mu < 3$, f^2 solo intersecta la línea de identidad en $(1-1/\mu)$, pero precisamente cuando $\mu = 3$ se puede observar un cambio

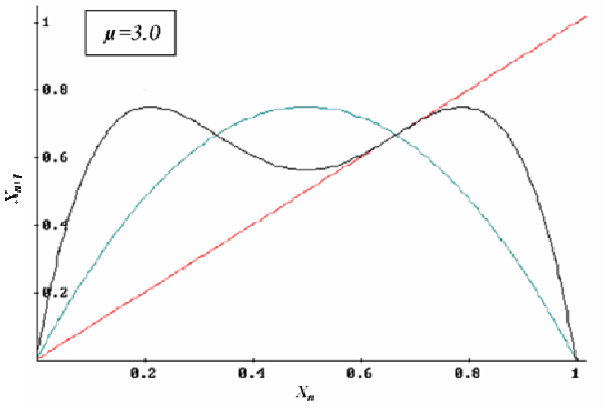


Figura 16 Periodo doble μ=3.0

Examinando los valores más grandes de μ se puede ver que lo que ocurre es que f^2 con $\mu > 3$ intersecta la línea de identidad en tres puntos.

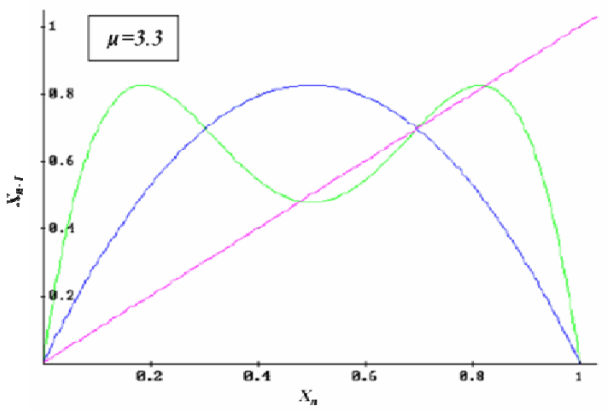


Figura 17 Periodo doble μ=3.3

Al iterar la función logística, se observa que el periodo 2 del atractor está dado precisamente por los puntos donde f^2 intersecta la línea de identidad.

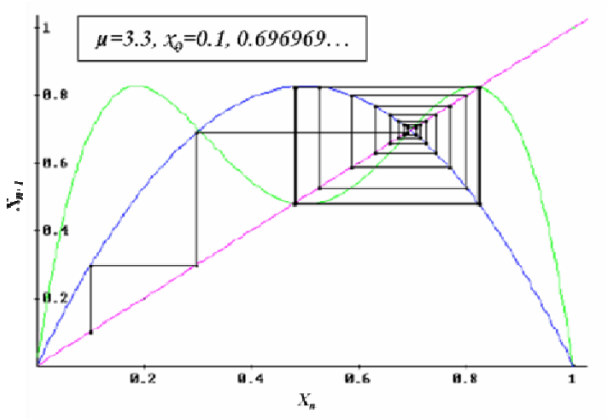


Figura 18 Periodo doble variando el parámetro x_0

Se puede observar que $(1-1/\mu)$ se convierte en un atractor inestable.

Se pueden encontrar nuevamente los puntos de los dos atractores cíclicos resolviendo la ecuación

$$f^2(x_n) = x_n$$

donde

$$f^2(x_n) = \mu(\mu x_n(1-x_n))(1-\mu x_n(1-x_n))$$

Se trata de una ecuación de cuarto grado y las raíces son

$x = 0$, $x = 1 - 1/\mu$, $x = \sqrt{(a+1)}\sqrt{(a-3)}/2a + 1/2a + 1/2$,
$x = -\sqrt{(a+1)}\sqrt{(a-3)}/2a + 1/2a + 1/2$

Se puede ver que las dos primeras son las mismas que para $f(x_n)$ y las dos últimas son las otras intersecciones. Pero, ¿Por tal motivo se vuelve inestable?[8] De hecho si, se vuelve inestable y al mismo tiempo, y por lo tanto se tienen ahora un atractor de periodo 4.

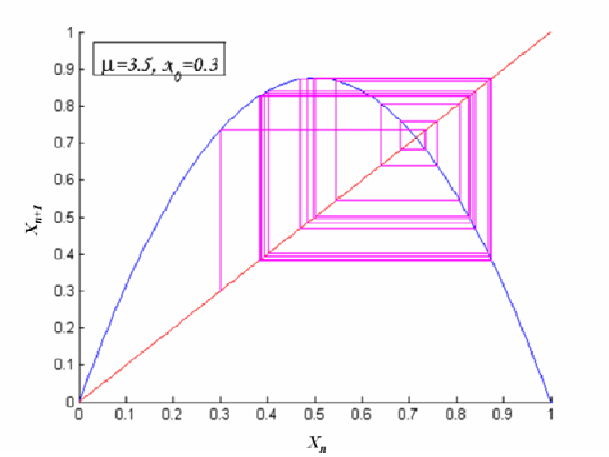


Figura 19 Atractor de periodo 4.

Nuevamente se puede encontrar los puntos del atractor de periodo 4 resolviendo la ecuación $x = f^4(x)$ pero comienza a ponerse un poco más complicado

[8] Se puede calcular si un punto es estable o inestable si pequeñas perturbaciones se incrementan o decrecen con cada iteración. Esto puede ser medido con la derivada $dx_{n+1}/dx_n = \mu(1-2x_n) = 2-a$. Si el valor absoluto excede uno, entonces el punto es inestable, si es más pequeño que uno entonces es un punto estable, si es igual a uno es un punto de bifurcación.

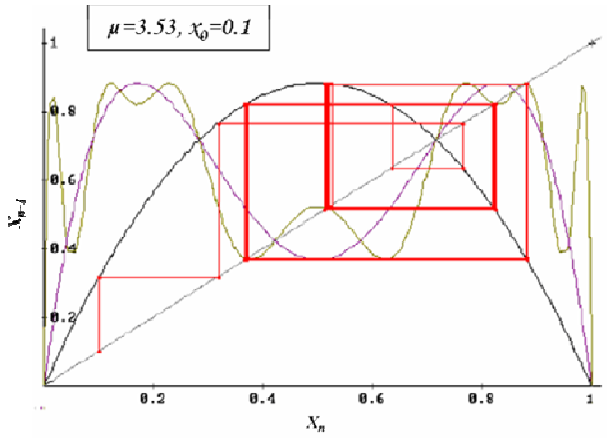


Figura 20 Ciclos atractores.

Se puede ver que los puntos atractores son los cuatro de f^4 que comienzan a intersectar la línea de identidad. Los anteriores se vuelven puntos inestables. Si se incrementa μ un poco más se encuentran ciclos atractores de periodo 8, 16, 32, 64, … Los valores de μ donde hay un cambio de periodo (el cual es doble) son llamados puntos de bifurcación.

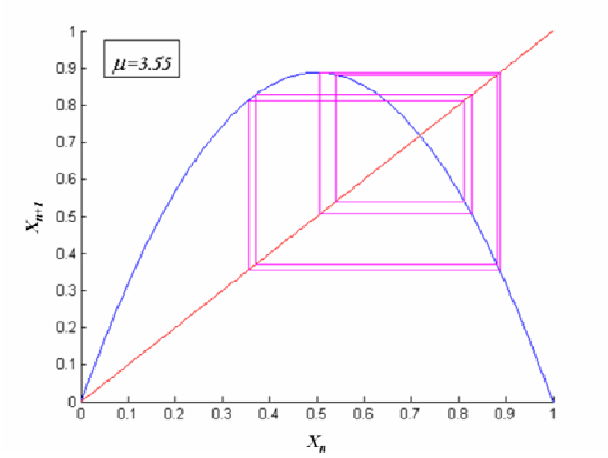


Figura 21 Se han omitido las primeras iteraciones. Ciclo atractor de periodo 8.

Se vuelve extremadamente difícil encontrar estos puntos analíticamente, por lo tanto, no tiene caso si quiere intentar encontrar las raíces de $f^{1024}(n)$, así que se utilizará otro método diferente. Por otra parte, si se incrementa el valor de μ aún más, se observa que la dinámica no parece repetirse, se ha alcanzado el **caos**. Se puede ver como una infinitud de puntos estables e inestables, cada uno atrayendo o repeliendo la dinámica provocando un comportamiento caótico.

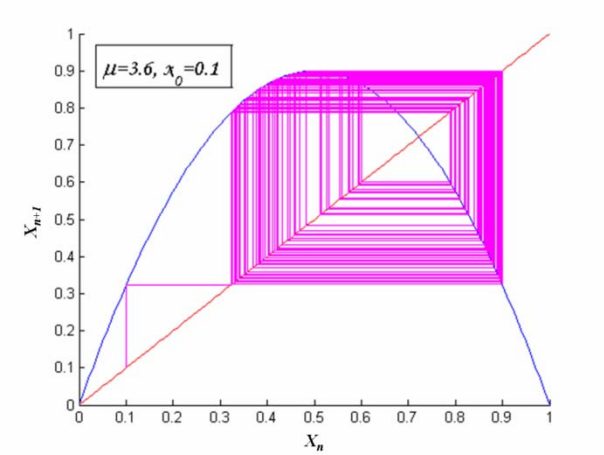


Figura 22 Punto atractor

Se puede observar que la dinámica del sistema no es aleatoria, no cubre todos los posibles espacios (para este caso, $[0..1]$). El área en la cual la dinámica está concentrada se conoce como un atractor extraño.

Diagrama de bifurcación

Los diagramas de bifurcación son una herramienta muy útil para poder observar que es lo que está pasando. Básicamente, se grafica el valor de μ contra los puntos donde la dinámica se ha concentrado después de algunas iteraciones iniciales, esto es, un atractor (punto, ciclo o extraño).

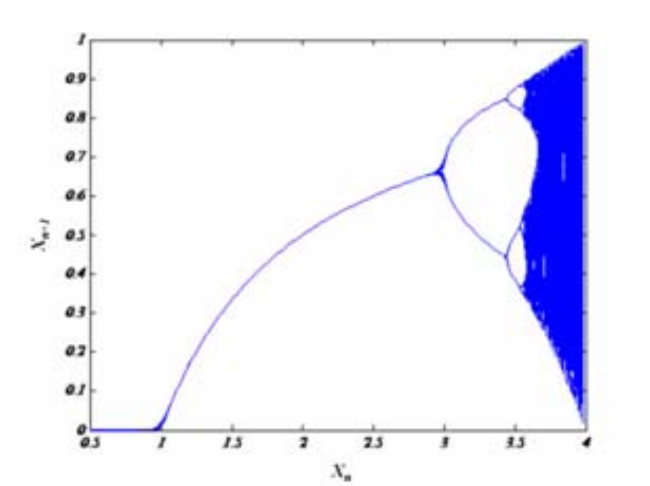


Figura 23 Mapeo Logístico como función del parámetro

Aquí se puede observar con cierta claridad todas las observaciones previas y mucho más: par valores de $0 < \mu < 1$, 0 es un punto atractor estable, $\mu = 1$ es un punto de bifurcación y $0 < \mu < 3$ contendrá el punto atractor $1 - 1/\mu$, $\mu = 3$ es otro punto de bifurcación y ahora hay un ciclo atractor de periodo 2 que posteriormente se incrementa a periodo 4, 8, 16, 32, … Observando más de cerca esta parte del diagrama de bifurcación.

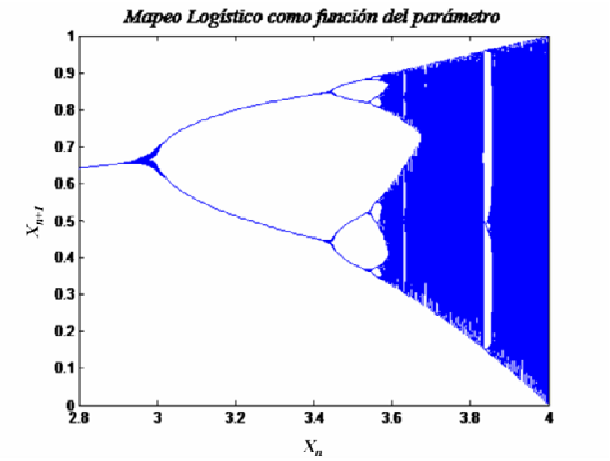


Figura 24 Diagrama de bifurcación.

Se puede observar que mientras el periodo del atractor se duplica, el siguiente punto de bifurcación es más parecido al anterior. Después de este límite $n \to \infty$ por lo tanto se cuenta con atractores de periodo más que infinito, es aquí donde comienza la región caótica. Cuando $\mu = 4$ el atractor extraño cubre todos los espacios $[0..1]$ (pero nótese que existen áreas con diferentes densidades…).

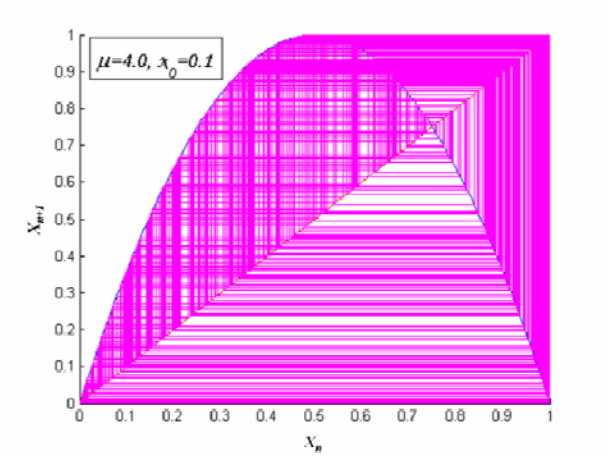


Figura 25 Región caótica.

Pero en el diagrama de bifurcación se observa algunas regiones donde hay otros periodos llamados **ventanas**. Nótese una gran ventana de periodo 3 cerca de $\mu = 3.83$. Estos periodos también se duplican a 6, 12, 24, … y nuevamente alcanzan una región caótica.

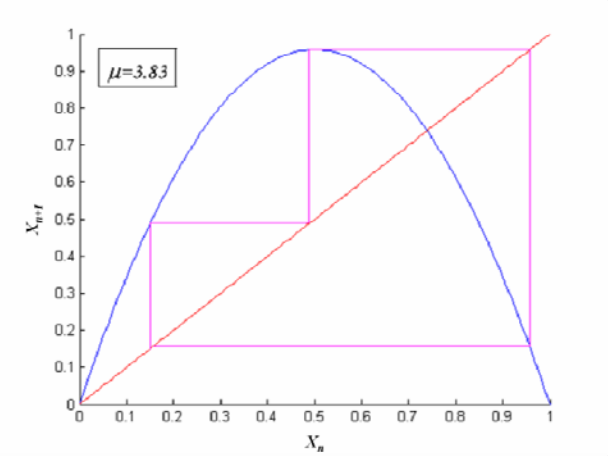


Figura 26 Periodo 3.

Pero se puede observar que existen muchas otras ventanas de diferentes periodos (surgiendo de los puntos de acumulación). De hecho se puede demostrar que en este diagrama de bifurcación existen orbitas de periodo *n* para todos los números naturales *n*.

Si se hace un acercamiento a la parte superior de la primera bifurcación se puede observar que es bastante similar al diagrama de bifurcación original, tal vez un poco a escala, pero también se puede observar una ventana de periodo 3 y muchas otras ventanas.

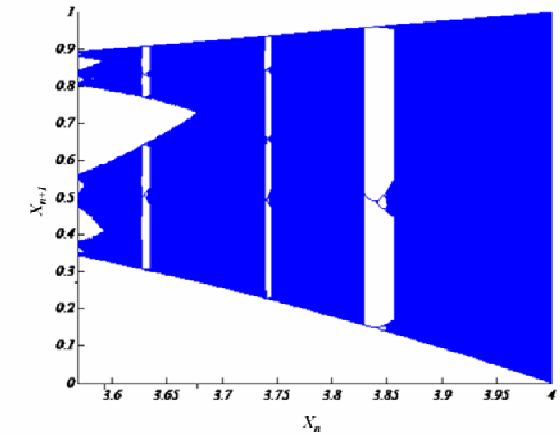


Figura 27 Ventana de periodo 3.

Si se continúa acercándose todavía más pareciera que los cambios de escala no cambiarían en nada el diagrama original y esto es se debe que este diagrama de bifurcación es un **fractal**, es auto afín.

Anteriormente se mencionó que el primer diagrama de bifurcación contenía órbitas de periodo de todos los números naturales y si se hace un acercamiento un poca más amplio,

no importa que tanto, encontraremos la órbitas de todos los números naturales nuevamente. Los números naturales se contienen a ellos mismos un número infinito de veces. De hecho son infinitos, pero esta conclusión no es evidente.

Universalidad

Alrededor de Octubre de 1975, Mitch Feigenbaum se encontraba estudiando las propiedades del mapeo logístico con la ayuda de una calculadora de bolsillo. Se dio cuenta que se requería de mucho esfuerzo tratar de encontrar los puntos de bifurcación calculando cada posible valor de μ (no existían muchas computadoras y las que habían no eran demasiado rápidas), estudió la convergencia geométrica de la duplicación de los periodos ya que parecían tener cierta regularidad.

Por tal motivo, se puede llamar el primer punto de bifurcación μ_0, el cual, para el mapeo logístico es 1, el segundo punto de bifurcación $\mu_1 = 3$, y así, $\mu_2 = 3.4495$, $\mu_3 = 3.5441...$, $\mu_4 = 3.5644...$ Así, la relación de cambio es

$$\frac{\mu_n - \mu_{n-1}}{\mu_{n+1} - \mu_n}$$

Así, Feigenbaum calculó el límite

$$\lim_{n\to\infty} = \frac{\mu_n - \mu_{n-1}}{\mu_{n+1} - \mu_n} = \delta = 4.6692016091029...$$

Feigenbaum encontró que el límite de los coeficientes de δ es el mismo que para el mapeo senoidal ($\mu * sen(x)$) y de hecho para cualquier otra función con periodos dobles. Así, la δ de Feigenbaum parece ser una constante universal. Independientemente de la función, δ siempre será la relación de las bifurcaciones que conducen al caos. La constante de Feigenbaum también está presente en el famoso conjunto de Mandelbrot.

Pero Faigenbaum también encontró otra constante universal. Si se calcula el límite de los coeficientes de la distancia desde $x_n = 0.5$ (el cual representa el punto crítico en el mapeo logístico), hasta el punto más cercano del ciclo atractor, también será una constante universal llamada α de Feigenbaum. Cuando un punto del ciclo atractor es igual a 0.5 se le llama órbita superestable, porque es cuando converge más rápidamente al atractor.

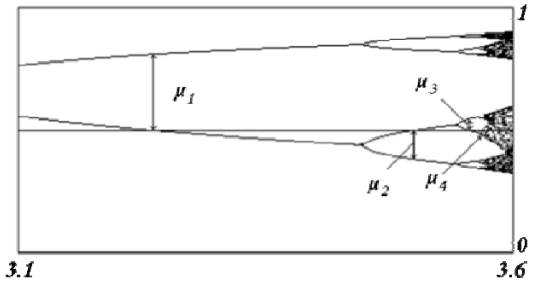


Figura 28 Bahías de estabilidad

Así que se tiene

$$\lim n \to \infty \frac{\mu_n}{\mu_{n+1}} = \alpha = 2.502907876...$$

La relación de las distancias entre los valores de μ, donde hay órbitas superestables es también δ

Distribución estadística

Con la finalidad de mostrar el efecto del parámetro μ, en las figuras 29 y 30 se muestran los diagramas de bifurcaciones para valores de $\mu \in (0.0, 4.0)$, y para $\mu \in (2.8, 4.0)$ respectivamente. Nótese que para valores de $\mu \in (3.0, 3.45)$ se genera una órbita de periodo 2 y a partir de $\mu=3.45$ se genera una órbita de periodo 4. Con las figuras 29 y 30 se puede estimar que el comportamiento caótico del mapeo aparece en dos regiones para valores de μ mayores a 3.6. Aproximadamente a partir de $\mu=3.7$ aparece el comportamiento caótico del mapeo pero en una sola región. Sin embargo, en el diagrama de bifurcación de ambas figuras, se notan unas ventanas en las que se puede apreciar un comportamiento discontinuo que será analizado en la siguiente sección de este capítulo, cuando se analicen las ventanas de estabilidad. Después, de esas ventanas la región se hace nuevamente única, conservándose esta característica hasta que μ alcanza el valor de 4.0.

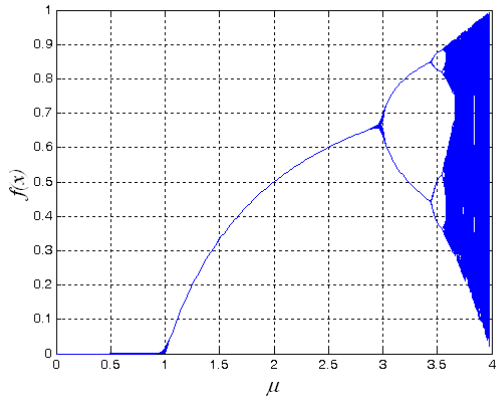


Figura 29 Mapeo Logístico como función del parámetro

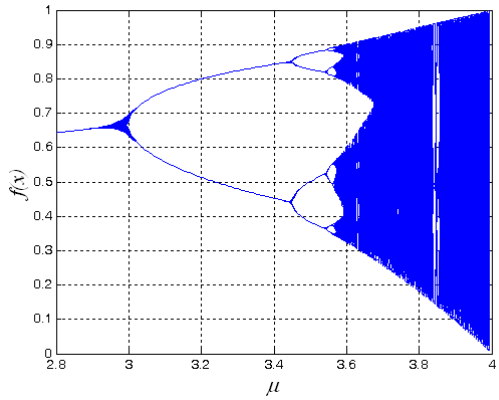


Figura 30 Diagrama de bifurcación del Mapeo Logístico para $\mu \in (2.8, 4.0)$

En ambas figuras puede notarse como, a medida que el parámetro tiende a 4.0, el mapeo cubre con mayor suficiencia el intervalo (0, 1). Nótese que cuando $\mu=4.0$, el mapeo cubre totalmente el intervalo.

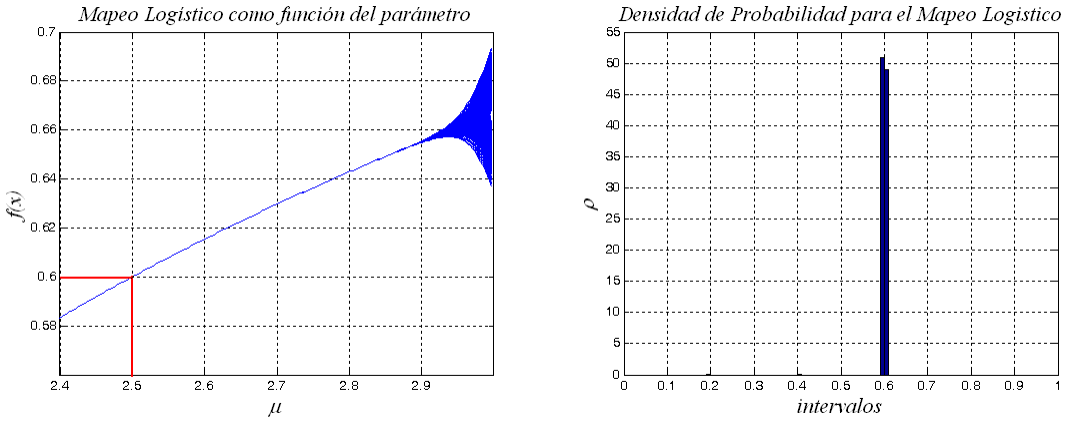
Cuando el sistema es caótico, no se puede determinar el comportamiento de la órbita a largo plazo. Entonces se requiere un análisis estadístico de la órbita, el cual consiste en averiguar qué tan frecuentemente la órbita visita diferentes regiones, dando lugar a un histograma asociado a esta órbita. El problema de este punto de vista radica en que se tendría que analizar una órbita infinita. Para obtener el histograma asociado a la órbita se hace uso del Teorema Ergódico, que dice que se debe estudiar la evolución de una distribución inicial, y a todos y cada uno de los puntos que la conforman se les aplique el mapeo, y cuando se obtenga una distribución que sea invariante ante la aplicación del mapeo, tal distribución corresponde a la que se encontraría en el análisis estadístico de la órbita infinita.

Así, aunque no se pueda predecir el valor que tome una órbita, si se puede saber el comportamiento estadístico del sistema.

Como puede observarse del diagrama de bifurcación del mapeo, la distribución invariante depende del valor del parámetro. Así, para μ menor que 3.0, la distribución invariante estará compuesta por una delta de Dirac; en tanto que, para valores de $\mu \in (3.0, 3.45)$ se presentarán dos deltas de Dirac. Finalmente, para valores de μ superiores a 3.6 la distribución cubre un solo intervalo, exceptuando las ventanas mencionadas anteriormente.

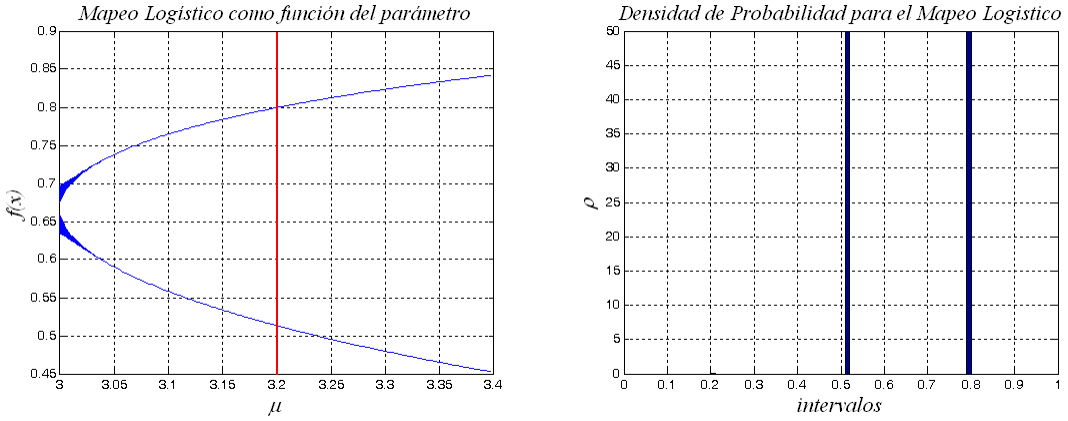
Esta situación se muestra en las figuras 31 a 34. La figura 31, se obtiene cuando $\mu=2.5$, y el histograma tiene una función delta de Dirac, la cual está alrededor de 0.6. En la figura 32, para $\mu=3.2$, el histograma muestra dos deltas de Dirac, una alrededor de 0.52 y la otra alrededor de 0.8. En la figura 33, para $\mu=3.5$ el histograma muestra cuatro deltas de Dirac, colocadas en 0.38, 0.5, 0.83 y 0.88 respectivamente. En la figura 34 para $\mu=3.56$ se presentan 8 deltas de Dirac, lo que habla de un fenómeno conocido como Doblamiento del Periodo. Para las figuras 31 a 33 se ha usado una partición de 100 intervalos para el intervalo (0, 4), mientras que para la figura 34, con la finalidad de apreciar las 8 deltas de Dirac, se ha usado una partición de 1000 intervalos.

Los resultados expresados en estas gráficas son consistentes con el diagrama de bifurcación, por ello, en cada caso, se ha colocado la porción correspondiente de dicho diagrama, ya que la forma del histograma en cada caso coincide con la densidad de puntos para el valor del parámetro usado.



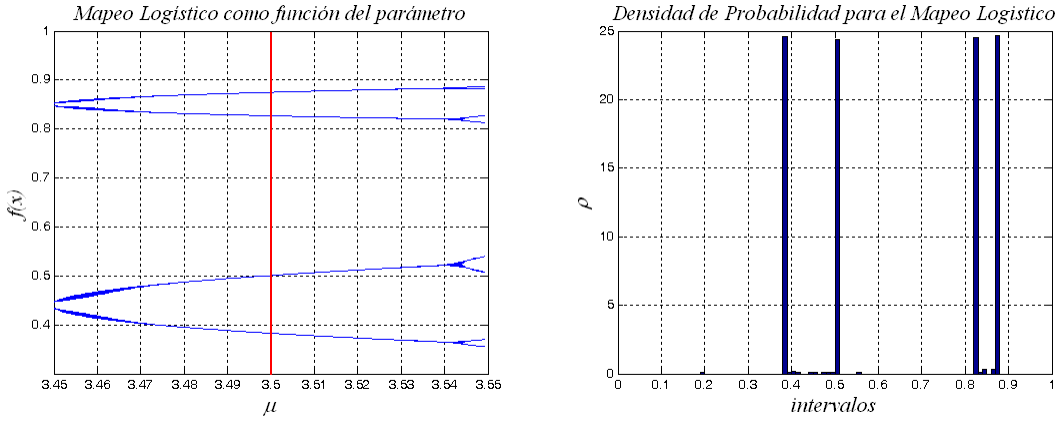
a) **b)**

Figura 31 Mapeo Logístico para μ=2.5 a) Diagrama de bifurcación. b) Densidad de probabilidad



a) **b)**

Figura 32 Mapeo Logístico para μ=3.2 a) Diagrama de bifurcación. b) Densidad de probabilidad



a) **b)**

Figura 33 Mapeo Logístico para μ=3.5 a) Diagrama de bifurcación. b) Densidad de probabilidad

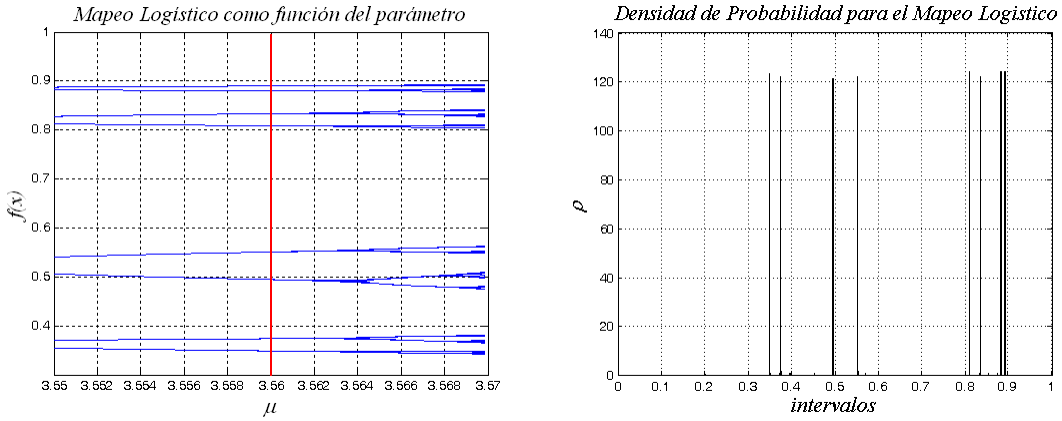
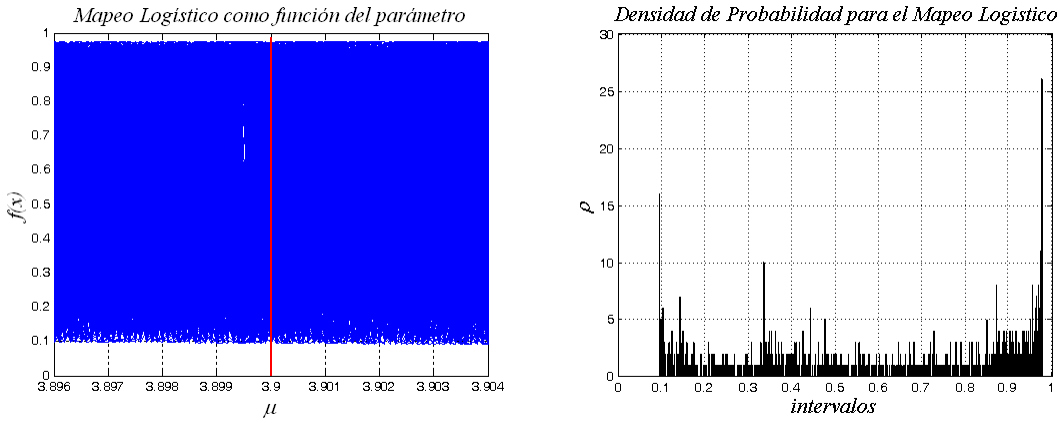


Figura 34 Mapeo Logístico para μ=3.56 a) Diagrama de bifurcación. b) Densidad de probabilidad

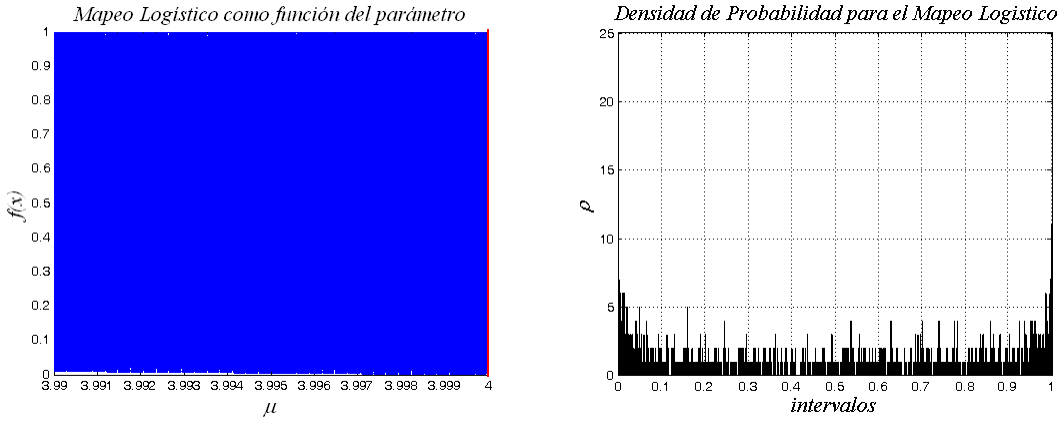
Por otro lado, para valores del parámetro cercanos a μ=4.0 se tiene una densidad de probabilidad con componentes prácticamente en todo el intervalo de interés (0, 4). Esto se muestra en las figuras 35 y 36, para valores de μ=3.9 y μ=4.0 respectivamente.

Nótese en la figura 35, tanto en el diagrama de bifurcación como en la densidad de probabilidad, que el mapeo genera órbitas entre 0.1 y 0.98. Para la densidad de probabilidad se ha usado una partición de 1000 intervalos para el intervalo (0, 4). En la figura 36, se puede apreciar que la densidad de probabilidad es muy parecida a la uniforme, exceptuando en los extremos límite del intervalo.



a) **b)**

Figura 35 Mapeo Logístico para *μ=3.9* a) Diagrama de bifurcación b) Densidad de probabilidad



a) **b)**

Figura 36 Mapeo Logístico para *μ=4.0* a) Diagrama de bifurcación. b) Densidad de probabilidad

Los histogramas que describen la función de densidad de probabilidad para este mapeo se han calculado aprovechando la propiedad de que el Mapeo Logístico es Ergódico, lo que significa que el comportamiento a largo plazo de una sola órbita, que se obtiene al dar una condición inicial e iterar el mapeo un número grande de veces es igual al comportamiento estadístico de un ensemble de condiciones iniciales.

Análisis de estabilidad

El mapeo logístico puede ser utilizado para el estudio de diferentes sistemas dinámicos. Por ejemplo, en 2007, Lev G., et al., discutió un método de Modulación por Ancho de Pulsos (PWM) basado en el mapeo logístico con el fin de reducir la interferencia electromagnética de los convertidores de nivel. En 1999, J. Tou. Et al., discutió cómo los sistemas caóticos proporcionan un simple medio para generar las señales deterministas que se asemejan al ruido blanco y utilizan el mapeo logístico para su labor. En 1997, la estadística característica de las secuencias caóticas generadas con el mapeo logístico mejorado, fueron analizadas por W. Hai y H. Jiandong. Ellos mostraron que las secuencias caóticas del mapeo logístico tienen una buena correlación y sugieren que estas secuencias pueden ser usadas en las comunicaciones de espectro disperso. En 2004, H. Tanaka et al reportó dos diseños de un generador compacto de ruido caótico para circuitos integrados grandes usando la tecnología CMOS, usando el mapeo logístico y el mapeo de la tienda de campaña. La apuesta (BET), es típicamente usada para encontrar la distribución estadística de la señal de ruido producida por los mapeo caóticos. BET implica que es equivalente a estudiar la evaluación de alguna distribución estadística, de modo que el mapeo caótico debe aplicarse a cada punto en la distribución estadística inicial y una vez que la distribución estadística invariante es obtenida, esa distribución corresponde a la distribución estadística de la órbita infinita producida por el mismo mapeo caótico. De esta manera, es posible conocer el comportamiento estadístico de sistema caótico, aunque el valor que toman algunas órbitas producidas por el sistema mencionado, no se podrían predecir. BET fue probado por primera vez en 1931 por G.D. Birkhoff. La prueba más reciente fue realizada por Rudolph en 1990.

Considérese $|\delta_n| = f^n(x_0+\delta_0)-f^n(x_0)$, siendo n el número de iteraciones del mapeo caótico, x_0 la condición inicial δ_0 algún número arbitrariamente pequeño, y considérese el límite cuando $\delta_0 \rightarrow 0$ una precisa y computacionalmente útil fórmula del exponente de Lyapunov se puede obtener.

$$\lambda = \frac{1}{n}\ln\left|\frac{\delta_n}{\delta_0}\right| = \frac{1}{n}\ln\left|\left(f^n\right)'(x_0)\right|$$

El término dentro del algoritmo puede ampliarse de tal manera que el exponente de Lyapunov se puede aproximar por la siguiente expresión:

$$\lambda \approx \frac{1}{n}\sum_{i=0}^{n-1}\ln\left|f'(x_i)\right|$$

Si la siguiente ecuación tiene sus límites cuando $n \rightarrow \infty$, entonces el exponente de Lyapunov para la órbita que comienza en x_0 es determinado por:

$$\lambda = \lim_{n\to\infty}\left\{\frac{1}{n}\sum_{i=0}^{n-1}\ln\left|f'(x_i)\right|\right\}$$

Nótese que x depende de x_0. Para los puntos y ciclos estables, λ es negativo y para atractores caóticos λ es positivo.

Cálculo numérico de λ

De la ecuación anterior, y considerando que para el exponente de Lyapunov $f'_{\mu}(x) = \mu(1-2x)$, λ puede ser calculado numéricamente usando BET:

$$\lambda = \int \rho_{est}(x)[\ln(\mu)+\ln(1-2x)] = \ln(\mu)+\int \rho_{est}(x)\ln(1-2x)$$

De cualquier manera, el segundo término de la integral en la ecuación evaluado numéricamente corresponde a los promedios de las funciones mediante la distribución invariante por medio de BET. Por lo tanto, es suficiente evaluar el promedio sobre la órbita del exponente de Lyapunov para aproximar numéricamente la siguiente integral:

$$\int \ln(1-2x)\rho_{est}(x)dx = \langle\ln(1-2x)\rangle \cong \frac{1}{N+1}\sum_{i=1}^{N}\ln(1-2x_i)$$

donde, x_i pertenece a la órbita del exponente de Lyapunov. De este modo, es posible calcular aproximaciones numéricas de λ y para ello ln|1-2x| debe ser evaluado usando la órbita del exponente de Lyapunov a medida que la distribución invariante (estacionaria) ha sido alcanzada. Así, el valor de λ puede ser estimado de acurdo a la siguiente expresión:

$$\lambda = \lim_{n\to\infty}\left\{\frac{1}{n}\sum_{i=0}^{n-1}\ln\left|f'(x_i)\right|\right\} = \ln(\mu)+\frac{1}{N+1}\sum_{i=1}^{N}\ln(1-2x_i)$$

Otra manera de poder estimar el valor de λ es considerando que para la ecuación (5) es posible calcular ρ_{est} suponiendo una partición de intervalos regulares, de modo que $\Delta x_i = x_{i+1} - x_i = \Delta x$

$$\rho_{est} = \sum_{i=1}^{M}\rho(x_i)D\left(\frac{x-x_i}{\Delta x}\right)$$

De esta manera, la expresión (6) se convierte en

$$\langle\ln(1-2x)\rangle = \sum_{i=1}^{M}\rho(x_i)\left\{\int \ln(1-2x)D\left(\frac{x-x_i}{\Delta x}\right)dx\right\}$$

Ahora, considerando que

$$\int \ln(1-2x) D\left(\frac{x-x_i}{\Delta x}\right) dx = \int_{x_i}^{x_i+\Delta x} \ln(1-2x)\,dx = \ln\left(1-2\overline{x_i}\right)\Delta x$$

Donde $\overline{x_i}$ es el valor central del intervalo *$(x_i, x_i + \Delta x)$* y

$$D\left(\frac{x-x_i}{\Delta x}\right) = \begin{cases} 1 & x \in (x_i, x_i + \Delta x) \\ 0 & x \notin (x_i, x_i + \Delta x) \end{cases}$$

El valor de λ también puede ser calculado de la siguiente manera:

$$\lambda = \lim_{n\to\infty}\left\{\frac{1}{n}\sum_{i=0}^{n-1} \ln\left|f'(x_i)\right|\right\} = \ln(\mu) + \sum_{i=1}^{M} \rho(x_i)\ln\left(1-2\overline{x_i}\right)\Delta x$$

Para concluir se debe entender que muchos de los sistemas no lineales no se pueden resolver analíticamente. En estos casos, se pueden obtener algunas soluciones por medio de aproximaciones, las cuales son muy útiles en la descripción de un fenómeno. La razón por la que las ecuaciones lineales son más sencillas de analizar se debe a que los sistemas lineales se pueden separar en partes, resolver cada una de ellas y después juntar las soluciones para obtener la solución final. Por otro lado, los mapeos unidimensionales 1-D exhiben una sensible dependencia a las condiciones iniciales. Esto significa que dos trayectorias que comienzan una cerca de la otra divergen y cada una tendrá un futuro totalmente diferente a la de la otra. Es aquí donde toma sentido e importancia calcular numéricamente el exponente de Lyapunov.

Discretización y escalamiento

Una diferencia importante entre el caos y la criptografía radica en el hecho de que los sistemas usados en caos están definidos solamente en números reales, mientras que la criptografía trabaja con sistemas definidos en números finitos de enteros.

La discretización es un proceso en el que el mapa **G:** $\mathcal{Y} \to \mathcal{Y}$ es remplazado por el mapa ***F:*** $\mathcal{X} \to \mathcal{X}$. La discretización no es un proceso único. De cualquier manera, en muchos casos uno puede identificar una forma natural de llevarlo a cabo. Así, por ejemplo, $\beta = \{C_0, \ldots, C_{2^m-1}\}$ es una partición finita del espacio fase $\mathcal{Y}$, entonces $X = \{0, \ldots, 2^m - 1\}$ y F es la restricción de **G** en $\mathcal{X}$ (asumiendo que existe tal restricción).

En esta tesis, y para la construcción de el algoritmo propuesto, el mapeo logístico discretizado queda definido como

$$F(x)=\begin{cases} floor\left[x(256-x)/64\right] & if \quad \tilde{x}<256 \\ 255 & if \quad \tilde{x}=256 \end{cases}$$

Donde $\tilde{x}=floor\left[x(256-x)/64\right]$ y $x\in\{0,...,255\}$. La función de transformación se obtiene a partir de la ecuación del mapeo logístico en dos pasos: primero el mapeo logístico es escalado de tal manera que los valores de entrada y salida del mapeo estén en el intervalo $[0,...,256]$; segundo, el mapeo logístico escalado es discretizado.

El proceso de escalamiento y discretización se da en función de los valores para la variable μ (3.6, 3.8, 3.9) usados como parámetros de entrada en el proceso de cifrado, pero también en función de los bits empleados.

Tomando en cuenta el párrafo anterior, a continuación se muestran las gráficas correspondientes a la parábola del mapeo, escalado y discretizado al universo de los caracteres ASCII, esto es, 2^n, siendo n el número de bits y μ en función de n.

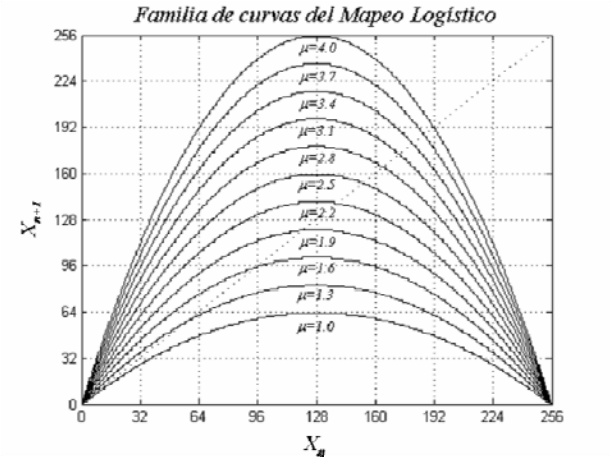


Figura 37 Familia de curvas del Mapeo Logístico escalado con valores de *μ=1* hasta *μ=4*.

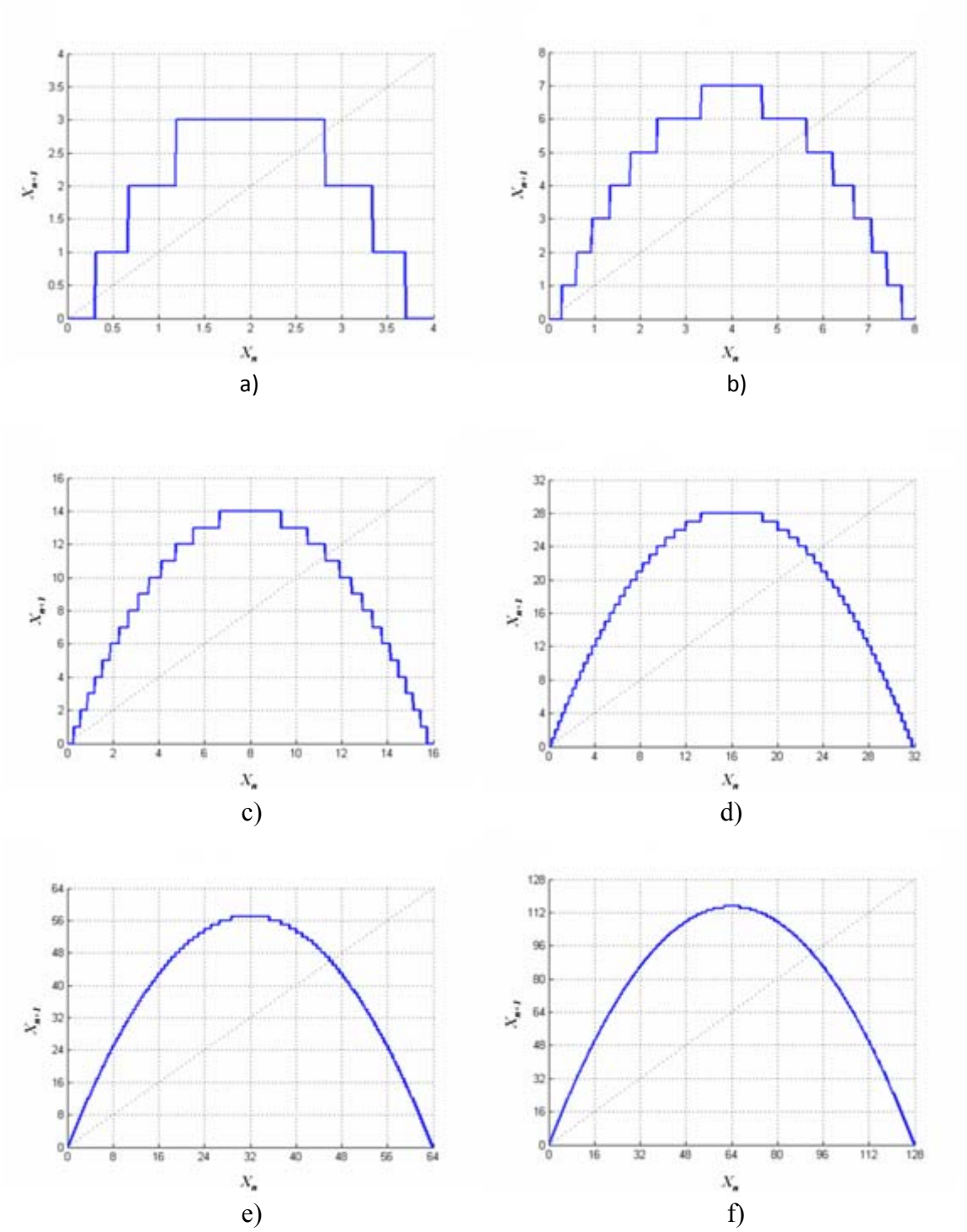


Figura 38 Curva del Mapeo Logístico escalado con valores de μ=3.6 y 100 puntos muestreados. a) 2^2 bits, b) 2^3 bits, c) 2^4 bits, d) 2^5 bits, e) 2^6 bits, f) 2^7 bits.

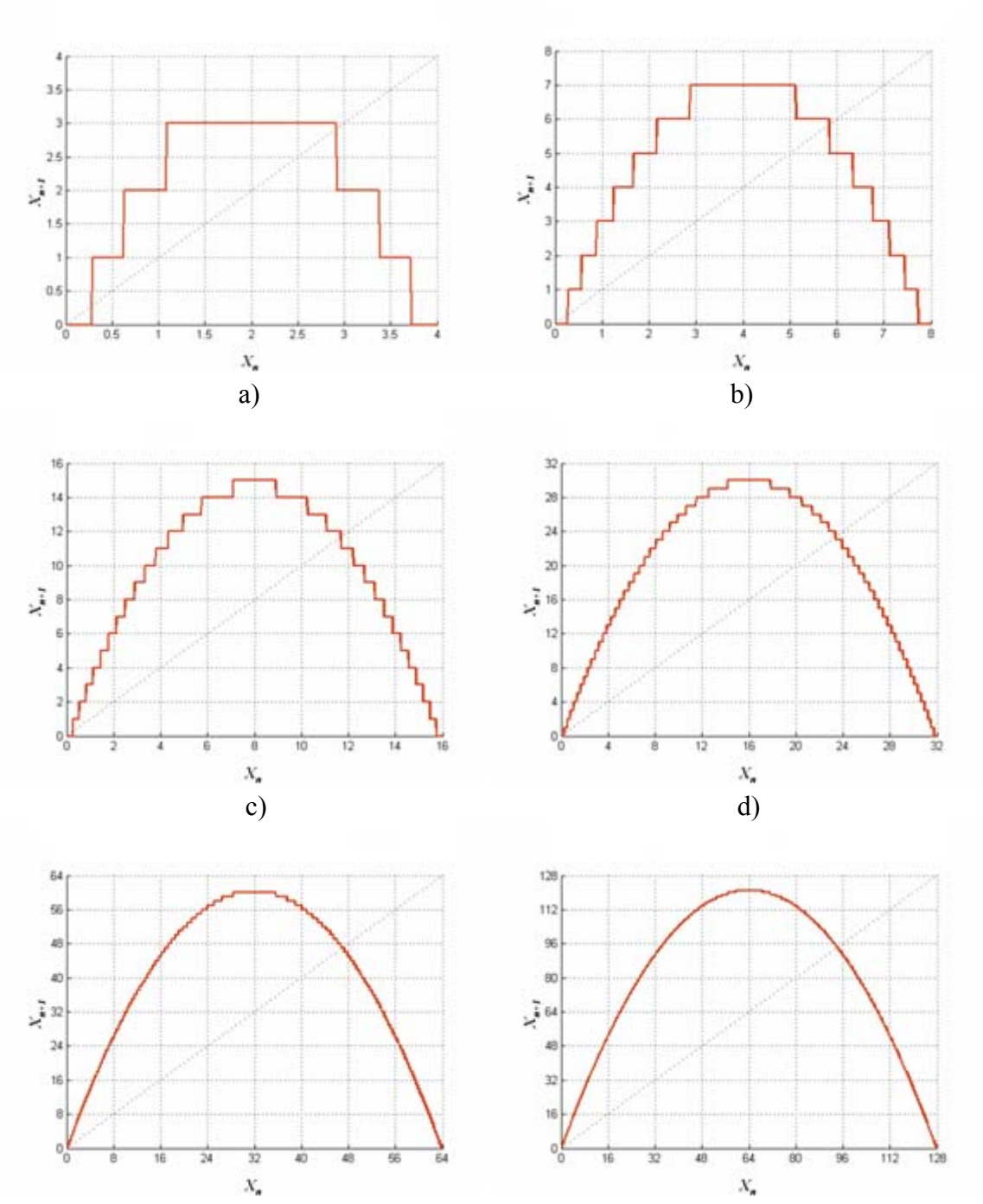


Figura 39 Curva del Mapeo Logístico escalado con valores de μ=3.8 y 100 puntos muestreados. a) 2^2 bits, b) 2^3 bits, c) 2^4 bits, d) 2^5 bits, e) 2^6 bits, f) 2^7 bits.

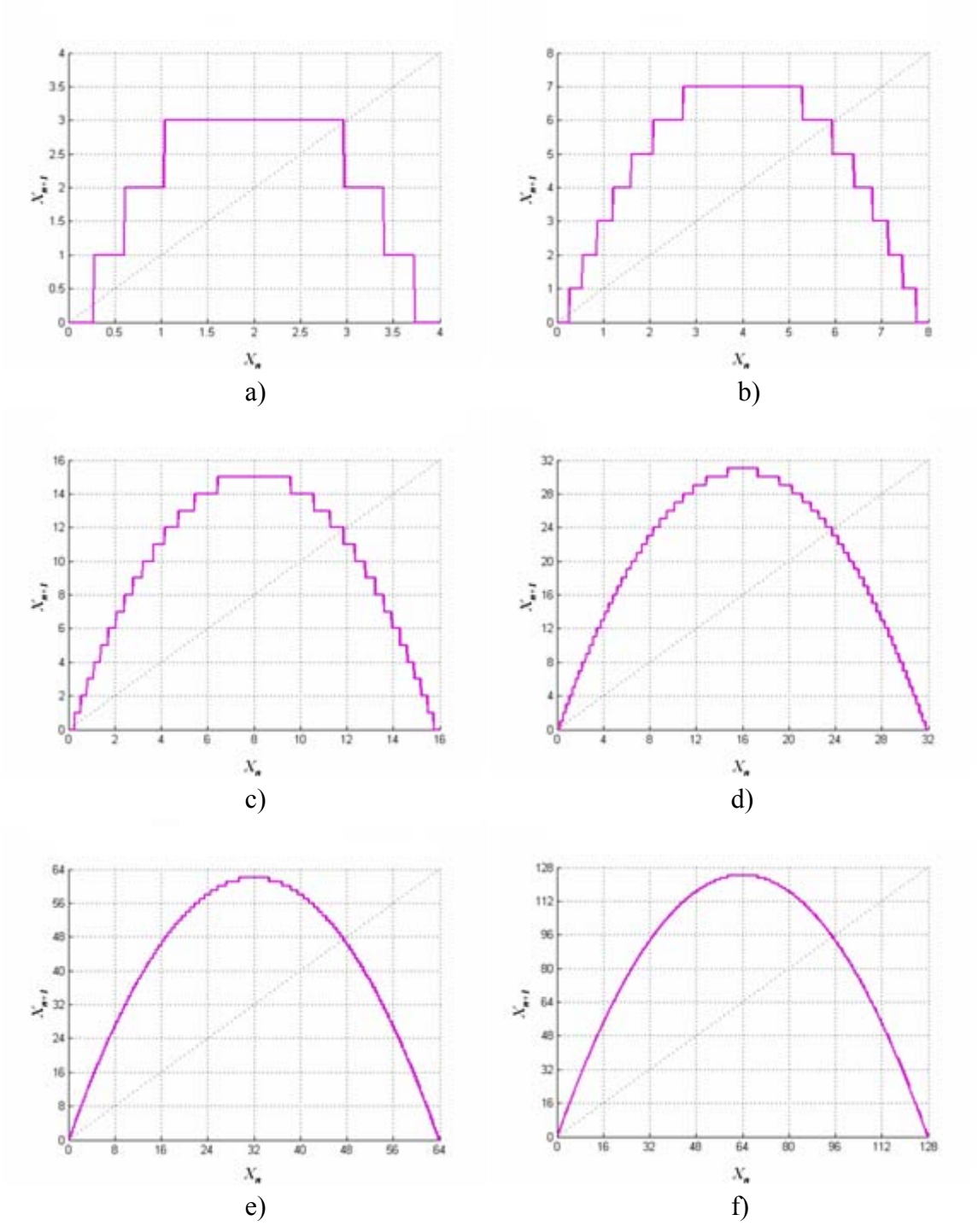


Figura 40 Curva del Mapeo Logístico con valores de μ=3.9 y 100 puntos muestreados. a) 2^2 bits, b) 2^3 bits, c) 2^4 bits, d) 2^5 bits, e) 2^6 bits, f) 2^7 bits.

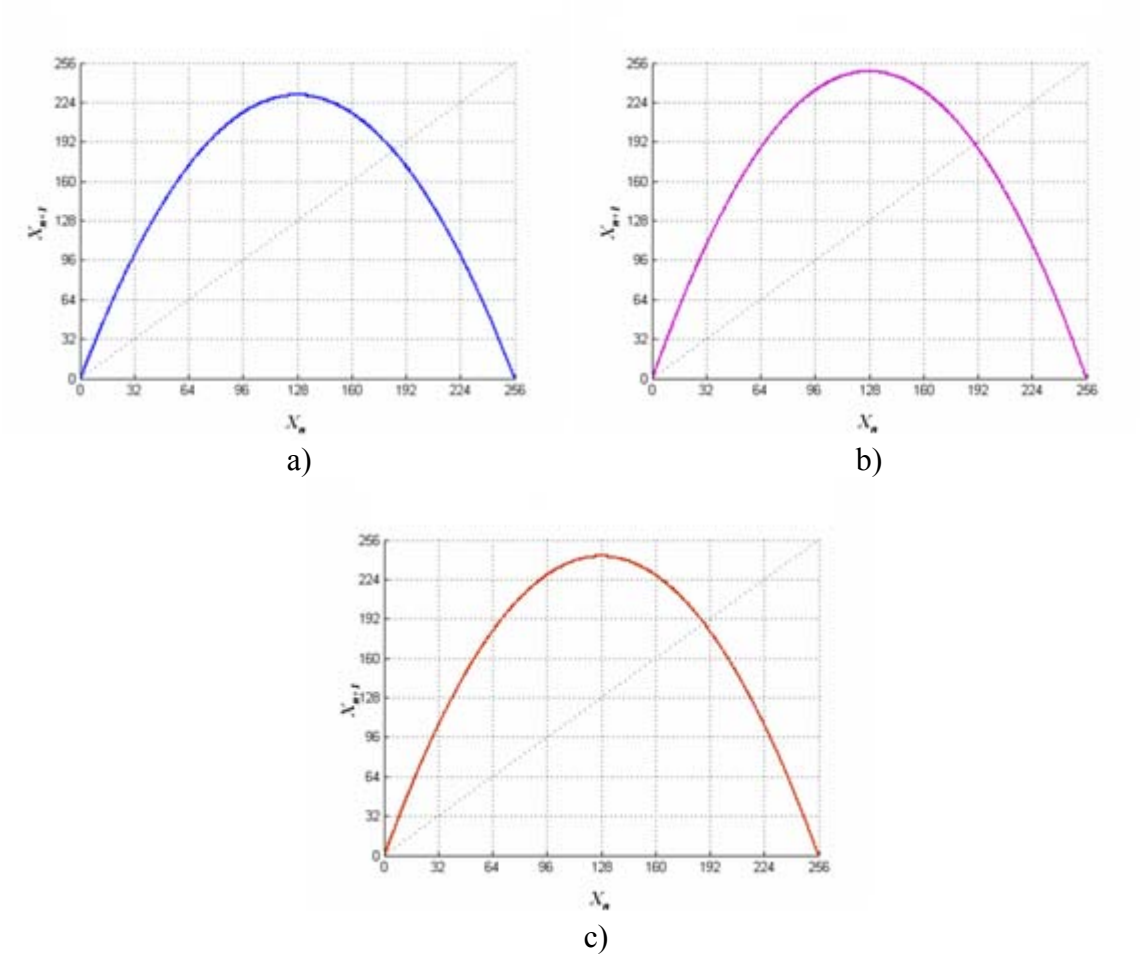
Como se mencionó anteriormente, el caos y la criptografía tienen algunas propiedades en común, siendo las más importantes la sensibilidad a las condiciones iniciales y el cambio en los parámetros de entrada. Sin embargo, así como existen que relacionan a estas dos ciencias, también existen diferencias entre una y otra. Por lo tanto, una importante diferencia entre el caos y la criptografía radica en el hecho de que los sistemas usados en caos están definidos en números reales, mientras que la criptografía trata con sistemas definidos en números enteros finitos.

Las figuras 38 - 40 muestran el proceso de escalamiento del mapeo logístico. Para cada valor del eje de las x corresponde un valor en el eje de las y, siendo x el valor de entrada e y el valor de salida. El propósito de estas figuras es demostrar el efecto de truncamiento ya que todos los números reales en el intervalo [0,1], correspondiente al mapeo logístico, son truncados pasando de un esquema basado en números reales a un esquema de números enteros, cuyo intervalo va de [0,n], siendo n, el número de bits utilizados en la muestra o construcción del esquema.

Tomando en cuenta que el proceso de truncamiento es llevado a cabo para la evaluación de la función logística, se obtiene la función escalera como una aproximación de la parábola como se observa en la figuras.

La siguiente tabla muestra la correspondencia entre los valores de entrada x y su correspondiente valor en y, o dicho de otra manera, la aproximación de la parábola para el inciso b), de la figura 39. Nótese que todos los valores están ubicados dentro del intervalo [0,256].

x	y	x	Y
0	0	4	7
1	3	5	7
2	5	6	5
3	7	7	3



a) b)

c)

Figura 41 Curva del Mapeo Logístico discretizada para μ=3.6, μ=3.8 y μ=3.9.

Finalmente de la curva del mapeo queda escalada y discretizada en el intervalo [0,256], abarcando así, el universo de los caracteres ASCII. Es importante mencionar que la discretización solo se alcanza cuando el número de bits empleados es 2^8. La altura máxima de la parábola está definida por el parámetro μ.

CAPÍTULO III: CIFRADO DE BLOQUES CAÓTICO

Resumen

En este Capítulo se presenta la realización de un algoritmo criptográfico de bloques basado en el mapeo logístico discretizado, escalado al universo del alfabeto ASCII y considerando una estructura de Feistel desbalanceada. La realización que aquí se presenta es congruente con la propuesta de Ljupco Kocarev y Goce Jakimoski y su distribución estadística se ha calculado numéricamente usando el Teorema Ergódico de Birkhoff, suponiendo un ensamble de condiciones iniciales con una distribución arbitraria.

Antecedentes

Algoritmos simétricos

En algunas aplicaciones modernas de criptografía simétrica se emplea el uso de algoritmos de cifrado por bloques. El mensaje se divide en partes (llamadas bloques) de longitud fija $n>1$ y se cifran bloque por bloque. El entero n es la longitud del bloque. En el caso del algoritmo DES, $n=64$, la longitud del bloque es 64 bits y la longitud de la llave es de 56. Así, para cada bloque, la entrada es de 64 bits, la salida es de 64 bits y la longitud de la llave es de 56, por lo que hay 2^{56} posibles llaves.

Desde un punto de vista muy general, el algoritmo DES es simplemente una combinación de las dos técnicas fundamentales empleadas para la construcción de cualquier algoritmo criptográfico adoptadas por Shannon en el año de 1949, conocidas como *Confusión* y *Difusión*.

Confusión. Esta técnica impide al criptanalista obtener patrones estadísticos y redundancias en el texto cifrado a partir del texto plano. Así, la dependencia estadística del texto cifrado en el texto plano es oculta. La manera más sencilla de llevar a cabo la *confusión* es a través de la substitución. En el caso de una cadena binaria, se substituyen los valores unos y ceros por ceros y unos respectivamente, de acuerdo a una formula predeterminada.

Difusión. Esta técnica disipa la redundancia del texto plano difundiéndola a lo largo del texto cifrado. La *difusión* implica que, si cambia una sola letra o caracter en el texto plano, causaría un gran cambio en el texto cifrado. Por lo tanto se necesitará una gran cantidad de texto cifrado para obtener redundancia en el texto plano.

Varios autores han sugerido que, de acuerdo a Shannon la *difusión* simplemente implica una permutación o reordenamiento de caracteres en la cadena del mensaje.

Sin embargo, esto no es totalmente cierto ya que una permutación aún conserva la frecuencia de los caracteres.
El proceder de Shannon respecto a la *difusión* también implica que actúe en una cadena con una función de difusión.

En las palabras del maestro mismo, Shannon [Shannon, C.E., 1949]: "En la difusión, la estructura estadística de M la cual da lugar a la redundancia se 'disipada' en rangos estadísticos amplios, es decir, en la estructura estadística implicando grandes combinaciones de letras en el criptograma. El efecto es que el enemigo debe interceptar una enorme cantidad de material para empatar esta estructura, ya que la estructura es evidente solo en bloques de muy pequeña probabilidad individual. Además, aún cuando el tenga suficiente material, el trabajo analítico requerido es mucho mayor, ya que la redundancia ha sido difundida a lo largo de un gran número de estadísticas individuales".

Un ejemplo de la *difusión* de las estadísticas opera sobre un mensaje $M = m_1, m_2, m_3, \ldots$ con una operación promedio, por ejemplo

$$y_n = \sum_{i=1}^{s} m_{n+i} \pmod{26}$$

Sumando s letras sucesivas del mensaje para obtener una letra y_n. Se puede observar que la redundancia de la secuencia *y* es la misma que la secuencia *m*, sin embargo la estructura ha sido disipada. Así, la frecuencia de las letras en la secuencia *y* estará muy cerca de ser igual a la secuencia *m*; el diagrama de frecuencias estará muy cerca de ser igual. Por lo tanto, cualquier operación reversible que produce una letra a la salida por cada letra a la entrada y no tiene una "memoria" infinita, tienen una salida con la misma redundancia que la entrada. Los patrones estadísticos nunca podrán ser eliminados sin un proceso de compresión, pero si pueden ser disipados.

Cifrados históricos de criptografía clásica como el cifrado Cesar y el Vigenere, no cuentan con las propiedades de *confusión* y *difusión*. Por otro lado, algoritmos como DES y AES utilizan la *confusión* y *difusión* para un mejor resultado.

Redes de Feistel y cifradores de bloque

En los algoritmos simétricos de cifrado por bloques se remplaza un bloque de N bits de texto plano, por un bloque de N bits de texto cifrado. Normalmente los bloques son de 64 bits de longitud. Esta idea general se ilustra en la siguiente figura:

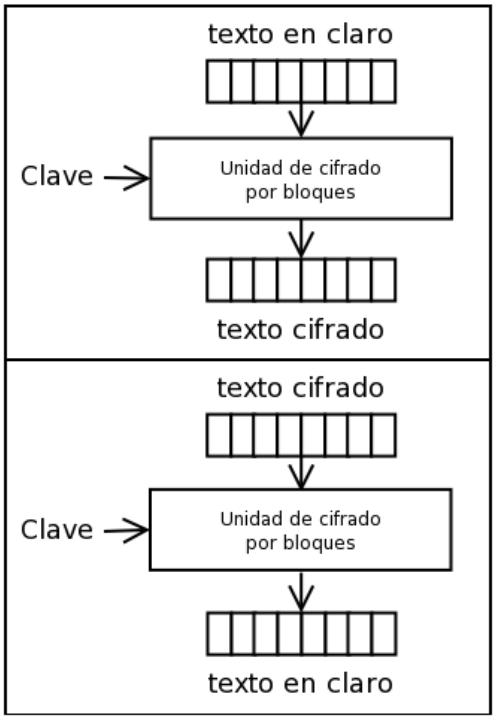


Figura 42 Esquema de cifrado de bloques

En un cifrado de bloques ideal la relación entre los bloques de entrada y los bloques de salida es completamente aleatoria, pero debe ser invertible para poder llevar a cabo el proceso de descifrado. Por lo tanto, tiene que existir un mapeo uno-a-uno, lo que significa que a cada bloque de entrada corresponde un bloque a la salida.

La figura 59 muestra un cifrado de bloques ideal que usa bloques de 8 bits de longitud. Cada bloque de 8 bits de texto plano se convierte en un bloque de 8 bits de texto cifrado.

Los algoritmos de producto usan las dos formas de cifrado clásicas: substitución y transposición alternativamente en múltiples rondas para implementar tanto la confusión como la difusión respectivamente. Shannon fue el primero en investigar los algoritmos de producto (también llamados redes de substitución-permutación) y mostró que algunos sistemas sofisticados de cifrado heurístico no eran más que producto de algunos sistemas más sencillos. Pero lo más importante es que Shannon identificó las condiciones necesarias para incrementar la fortaleza del cifrado utilizando algoritmos de cifrado simple en cascada.

Una manera posible de construir un algoritmo de llave secreta usando redes de substitución-permutación es partir ó dividir la entrada en trazos de tamaño razonable, hacer una substitución en cada pequeño trozo y posteriormente tomar las salidas para pasarlas a través de un sistema de permutación, el cual intercambia el orden de las letras. Este proceso se repite nuevamente.

Tomando en cuenta que los criptosistemas modernos se basan en el uso de computadoras, se asume que tanto el texto plano como el texto cifrado son cadenas de bits $(\{0,1\})$, en lugar de cadenas de letras $(\{a,b,c,\ldots,z\})$.

Una red de Feistel es un método general para transformar cualquier función (normalmente llamada función *F*) en una permutación. Fue inventada por Horst Feistel durante el diseño del algoritmo lucifer y ha sido usada en el diseño de muchos algoritmos de cifrado por bloques desde entonces, como por ejemplo: DES, FEAL, GOST, Fhufu y Khafre, LOKI, CAST, Blowfish, y RC5 por mencionar algunos.

La siguiente figura muestra la estructura de una red de Feistel que consiste de múltiples rondas para el procesamiento de texto plano, cada una constituyendo un proceso tanto de substitución como de permutación.

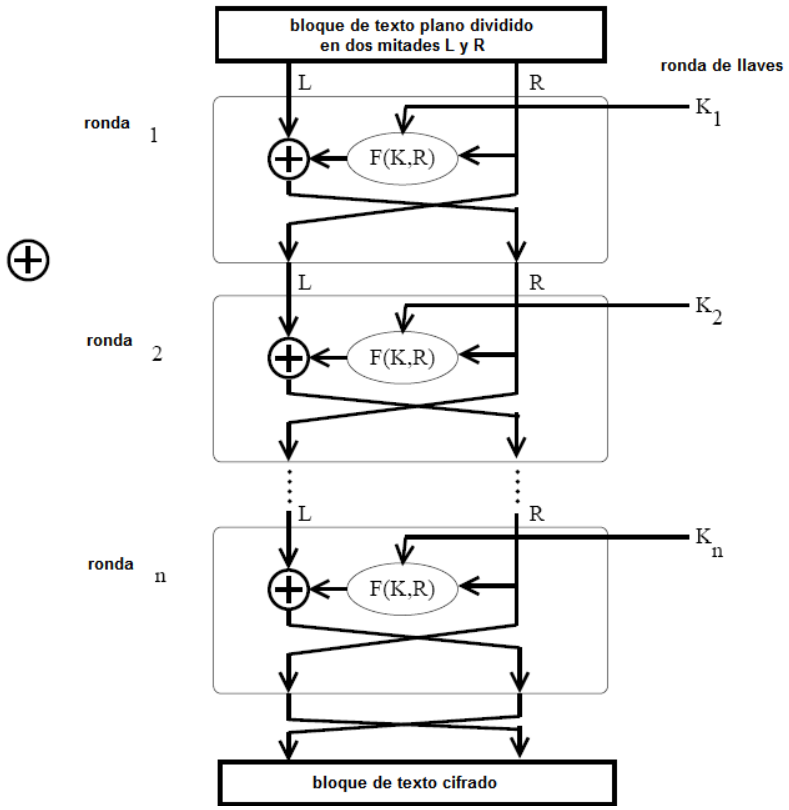


Figura 43 Red de Feistel

La red de Feistel que se muestra en la figura 60 es una forma particular de las redes de substitución-permutación. La entrada en una red de Feistel es un bloque de texto plano de *n* bits y una llave *K*. El bloque de texto plano se divide en dos mitades, L y R. Las dos mitades de los datos pasan a través de *r* rondas de transformación y se combinan para producir el bloque de texto cifrado.

El proceso de permutación al final de cada ronda, consiste en intercambiar las mitades ahora modificadas L y R, por lo tanto, el bloque L pasará a formar el bloque R en la siguiente ronda, y el bloque R pasará a formar el bloque L en la siguiente ronda.

Descripción matemática de cada ronda en una estructura de Feistel

Sea LE_i y RE_i los bloques de salida al final de la *i*-enésima ronda. La letra 'E' denota el proceso de cifrado. Se tiene entonces

$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus F\left(RE_{i-1}, K_i\right)$$

Donde $\oplus$ indica una operación lógica binaria. La letra *F* se refiere a la función de transformación que intercambia el bloque RE_{i-1} de la ronda anterior con la llave K_i correspondiente a la ronda actual.

Como ya se ha mencionado *F* representa la función de transformación propia de cada algoritmo y se le llama función de Feistel después del trabajo realizado por Horst Feistel.

Considerando una estructura de 8 rondas, la salida de la última ronda está dada por

$$LE_8 = RE_7$$
$$RE_8 = LE_7 \oplus F\left(RE_7, K_8\right)$$

El proceso de descifrado como se muestra en la siguiente figura, es exactamente el mismo que el proceso de cifrado, solo se tiene que invertir el orden de las llaves. La salida de cada ronda en el proceso de descifrado representa la entrada en la ronda correspondiente durante el proceso de cifrado. Esta propiedad se cumple independientemente de cual sea la función de transformación *F*.

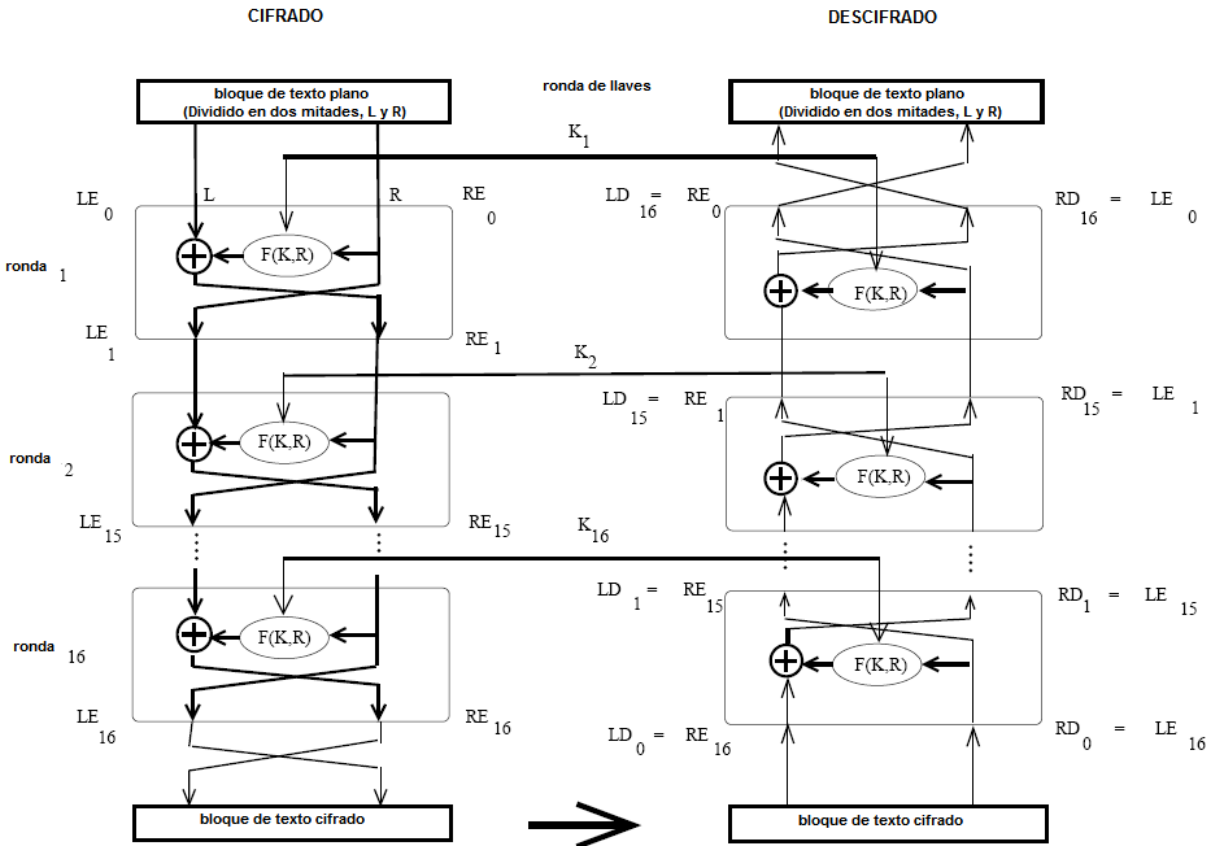


Figura 44 Proceso de cifrado – descifrado en una estructura de Feistel.

Para demostrar la afirmación anterior, sean LD_i y RD_i la mitad derecha y la mitad izquierda de la salida en la *i*-enésima ronda. Esto quiere decir que, la salida de la primer ronda de descifrado está formada por los bloques LD_1 y RD_1, por lo tanto, los bloques LD_0 y RD_0 representan la entrada en la primera ronda de descifrado. La relación que existe entre las dos mitades (bloques) que se introducen en la primera ronda de descifrado y las dos mitades que se obtienen a la salida, está dada por:

$$LD_0 = RE_8$$
$$RD_0 = LE_8$$

La salida de la primera ronda de descifrado se puede representar por las siguientes ecuaciones:

$$\begin{aligned} LD_1 &= RD_0 \\ &= LE_8 \\ &= RE_7 \end{aligned}$$

$$\begin{aligned} RD_1 &= LD_0 \oplus F(RD_0, K_8) \\ &= RE_8 \oplus F(LE_8, K_8) \\ &= [LE_7 \oplus F(RE_7, K_8)] \oplus F(RE_7, K_8) \\ &= LE_7 \end{aligned}$$

Se puede observar que la salida de la primera ronda de descifrado es la misma que la entrada en el último ciclo de la ronda de cifrado, ya que se obtienen los bloques

$$LD_1 = RE_7$$
$$RD_1 = LE_7$$

Las consideraciones anteriores se obtienen hacienda uso de las igualdades siguientes. Sean A, B y C tres arreglos de bits respectivamente, por lo tanto se tiene

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$
$$A \oplus A = 0$$
$$A \oplus 0 = A$$

El resultado anterior es independiente de la naturaleza exacta de la función *F*.

Redes de Feistel Desbalanceadas

Una red de Feistel desbalanceada por sus siglas en inglés (UFN; Unbalanced Feistel Networks), es una red de Feistel donde el bloque del lado izquierdo y el bloque del lado derecho no son del mismo tamaño.

Una ronda de s-sobre-t, o s:t, de una red de Feistel desbalanceada es de la forma:

$$X_{i+1} = (F(msb_s(X_i), k_i) \oplus lsb_t(X_i)) \| msb_s(X_i)$$

donde $msb_s(X_i)$ es conocido como source block, y $lsb_t(X_i)$ es conocido como target block, de ahí sus nombres s y t.

En la estructura de una red de Feistel desbalanceada encontraremos dos opciones que varían su funcionamiento. Cuando en una red de Feistel desbalanceada s > t se le llama source heavy, que equivale a decir que la mitad origen opera sobre la mitad destino y cuando s < t se le llama target heavy, que equivale a decir que la mitad destino opera sobre la mitad origen.

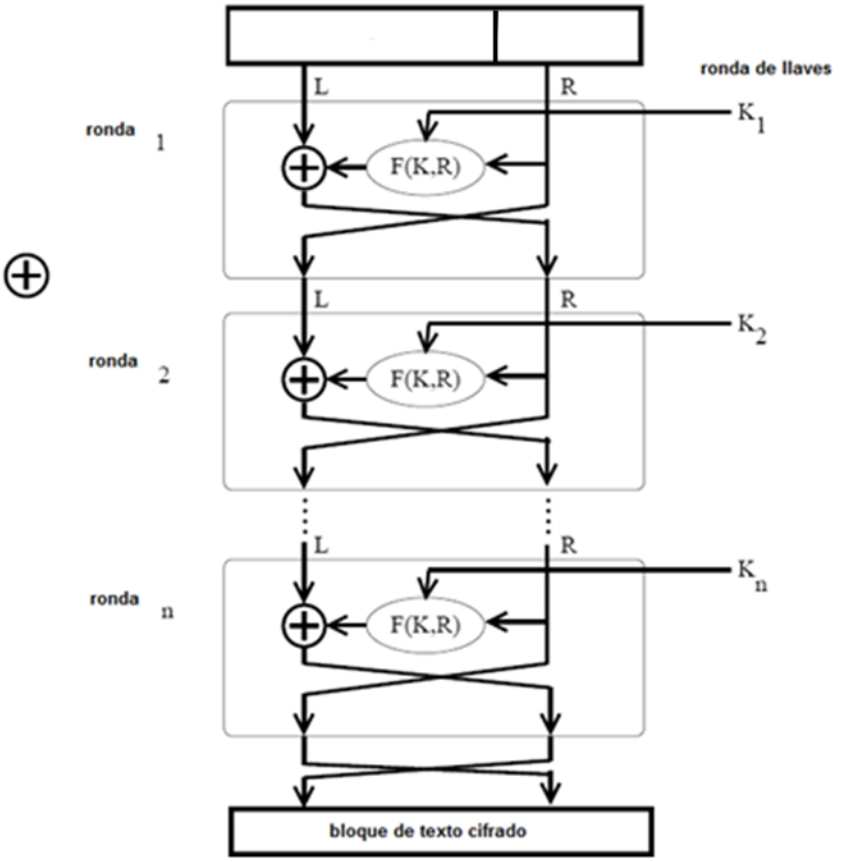


Figura 45 Red de Feistel desbalanceada.

Redes Homogéneas y Heterogéneas

A pesar de que en la mayoría de las estructuras de Feistel la función *F* es alterada solo por la ronda de las llaves ronda tras ronda, no hay razón por lo que debería de ser siempre así.

Una red de Feistel desbalanceada es homogénea cuando la función *F* es idéntica en cada ronda. Una red de Feistel desbalanceada es heterogénea cuando la función *F* es diferente en cada ronda

La ventaja de las redes heterogénea es que debido a que sus propiedades internas cambian de ronda en ronda, puede ser mucho más difícil encontrar algún tipo de característica que se propague de manera adecuada a través de los diferentes tipos de rondas que aparecen en la

estructura de cifrado. Los algoritmos de cifrado por bloques McGuffin [Matt Blaze and Bruce Schneier, 1994], MARS [C. Burwick, 1998] y SkipJack [Lars R. Knudsen and David Wagner, 2001] son ejemplos de redes de Feistel desbalanceadas.

Descripción del algoritmo de Ljupco Kocarev

Recuérdese que la mayoría de los algoritmos de cifrado tienen la forma

$$x_0 = B_0$$
$$x_i = E_Z[x_{i-1}] \quad i = 1,\ldots,r$$
$$B_r = x_r$$

Donde B_0, B_r representan el bloque de texto y plano y el bloque de texto cifrado, respectivamente, con una longitud L en bytes, x es un vector L dimensional y E_Z es la clave de cifrado que depende de la transformación.

En la literatura, se han estudiado tres tipos de funciones de transformación: Redes de Feistel [H. Feistel, 1973], redes de Feistel desbalanceadas siendo los ejemplos más comunes los algoritmos MacGuffin [M. Blaze and Schneir, 1995], BEAR/LION [R. Anderson and E. Biham, 1996] y Redes de sustitución-permutación (SP-Networks) también llamadas estructuras de transformación uniformes, como por ejemplo, IDEA [X. Lai and J.L. Massey, 1991] y SAFER [J.L. Massey, 1993].

En esta tesis, se estudia una clase particular de algoritmos de cifrado por bloques que se describe de la siguiente manera:

Sea B_0 un bloque de texto plano de 64 bits de longitud (L = 8 bytes). Los 8 bytes del bloque B_i están representados por los valores $x_{i,0},\ldots,x_{i,7}$, por lo tanto $B_i = x_{i,0},\ldots,x_{i,7}$. El cifrado consiste de r rondas de transformaciones idénticas aplicadas en una secuencia sobre el bloque de texto plano. La función de cifrado está dada por

$$x_{i,k+1} = x_{i-1,k} \oplus f_{k-1}\left(x_{i-1,1},\ldots,x_{i-1,k-1},z_{i-1,k-1}\right)$$

Donde $i = 1,\ldots,r$, $k = 1,\ldots,8$, $f_0 = z_{i,0}$, $x_8 \equiv x_0$, $x_9 \equiv x_1$ y $z_{i,0},\ldots,z_{i,7}$ son los 8 bytes de la subllave z_i, la cual controla la i-enésima ronda.

Las funciones $f,\ldots,f_7$ tienen la siguiente forma:

$$f_j = \left(x_1,\ldots,x_j,z_j\right)$$

Donde $j = 1,\ldots,7$ y $f : M \to M$, $M = \{0,\ldots,255\}$ es un mapeo derivado de un mapeo caótico. El bloque de salida $B_i = x_{i,0},\ldots,x_{i,7}$ es la entrada en la siguiente ronda, excepto en la última ronda. Por lo tanto, $B_r = x_{r,0},\ldots,x_{r,7}$ es el bloque de texto cifrado (la información cifrada). La longitud del bloque de texto cifrado es de 64 bits (8 bytes) y es igual a la longitud del bloque de texto plano. Cada ronda *i* es controlada por una sub-llave z_i de 8 bytes. Hay *r* sub-llaves en total las cuales se derivan de la llave en un procedimiento para la generación de rondas.

La estructura de descifrado deshace la transformación del proceso de cifrado. Se aplican *r* rondas de descifrado sobre el bloque de texto cifrado B_r para producir el bloque original de texto plano B_0. Las rondas de sub-llaves se aplican ahora en orden inverso, por lo tanto, la función de transformación está dada por

$$x_{i-1,k} = x_{i,k+1} \oplus f_{k-1}\left(x_{i-1,1},\ldots,x_{i-1,k-1},z_{i-1,k-1}\right)$$

Donde $k = 1,\ldots,8$, $f_0 = z_0$, $x_8 \equiv x_0$ y $x_9 \equiv x_1$.

Cifrador propuesto basado en el Mapeo Logístico

Descripción del Algoritmo

Se toma un bloque B_0 de texto claro de 64 bits de longitud dado por

$$B_0 = \left\langle x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3}, x_{0,4}, x_{0,5}, x_{0,6}, x_{0,7} \right\rangle$$

De modo que las $x_{0,j}$ con $j = 0,\ldots,7$ son los 8 bytes que conforman el bloque

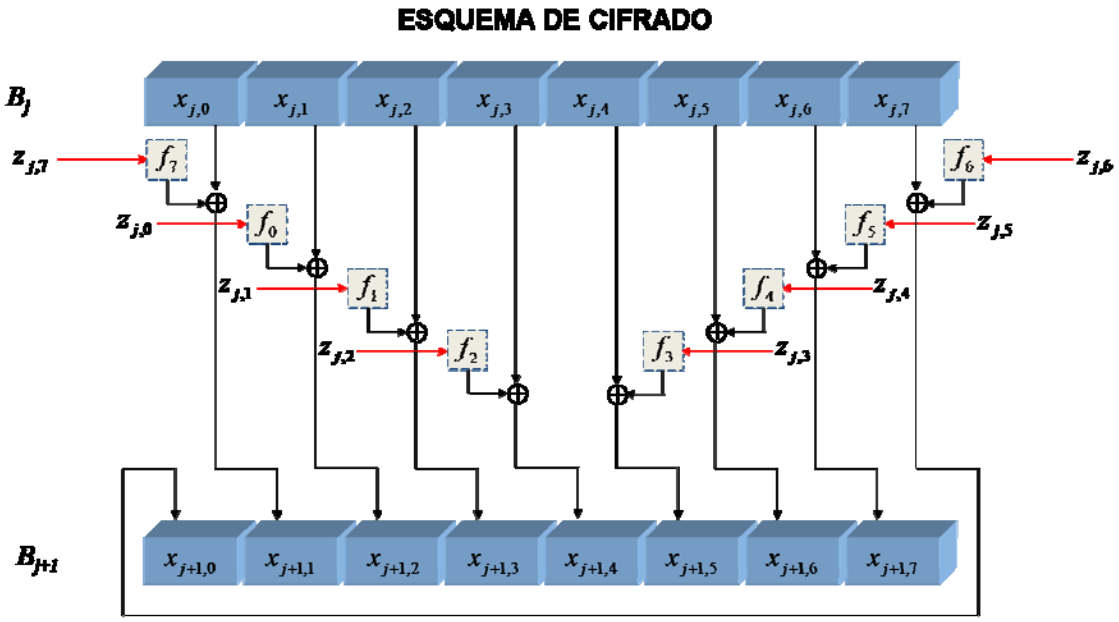


Figura 46 Estructura de cifrado

La función de transformación está dada por la siguiente expresión:

$$x_{i,k+1} = x_{i-1,k} \oplus f_{k-1}\left(x_{i-1,1}, \ldots, x_{i-1,k-1}, z_{i-1,k-1}\right)$$

Con esta expresión se calcula el valor de cada bloque, por lo tanto las funciones quedan definidas de la siguiente manera.

$$x_{j+1,2} = x_{j,1} \oplus f_0$$
$$x_{j+1,3} = x_{j,2} \oplus f_1(x_{j,1} \oplus z_{i-1,1})$$
$$x_{j+1,4} = x_{j,3} \oplus f_2(x_{j,1} \oplus x_{j,2} \oplus z_{i-1,2})$$
$$x_{j+1,5} = x_{j,4} \oplus f_3(x_{j,1} \oplus x_{j,2} \oplus x_{j,3} \oplus z_{i-1,3})$$
$$x_{j+1,6} = x_{j,5} \oplus f_4(x_{j,1} \oplus x_{j,2} \oplus x_{j,3} \oplus x_{j,4} \oplus z_{i-1,4})$$
$$x_{j+1,7} = x_{j,6} \oplus f_5(x_{j,1} \oplus x_{j,2} \oplus x_{j,3} \oplus x_{j,4} \oplus x_{j,5} \oplus z_{i-1,5})$$
$$x_{j+1,0} = x_{j,7} \oplus f_6(x_{j,1} \oplus x_{j,2} \oplus x_{j,3} \oplus x_{j,4} \oplus x_{j,5} \oplus x_{j,6} \oplus z_{i-1,6})$$
$$x_{j+1,1} = x_{j,8} \oplus f_7(x_{j,1} \oplus x_{j,2} \oplus x_{j,3} \oplus x_{j,4} \oplus x_{j,5} \oplus x_{j,6} \oplus x_{j,7} \oplus z_{i-1,6})$$

Los valores $x_{j,1}, x_{j,2}, \ldots x_{j,n}$ representan los bits del bloque de texto claro, y los valores $x_{j+1,1}, x_{j+1,2}, \ldots x_{j+1,n}$ representan los bits del bloque de texto cifrado. Para calcular el valor del bloque de texto cifrado se toma el valor anterior. Esto se puede apreciar en todos y cada uno de los esquemas de cifrado representados por las figuras 63-70. Así, para calcular $x_{j+1,2}$ se toma $x_{j,1}$ y se suma (suma lógica) con la llave z.

La función caótica utilizada para la construcción del cifrador está contenida en *F*, siendo *F* misma la función de transformación del algoritmo.

Una correcta programación de la función logística como función de transformación del algoritmo, depende de un adecuado proceso de discretización, ya que el intervalo sobre el cual varía ahora la función es de [0,256] abarcando así todos los caracteres del código ASCII extendido.

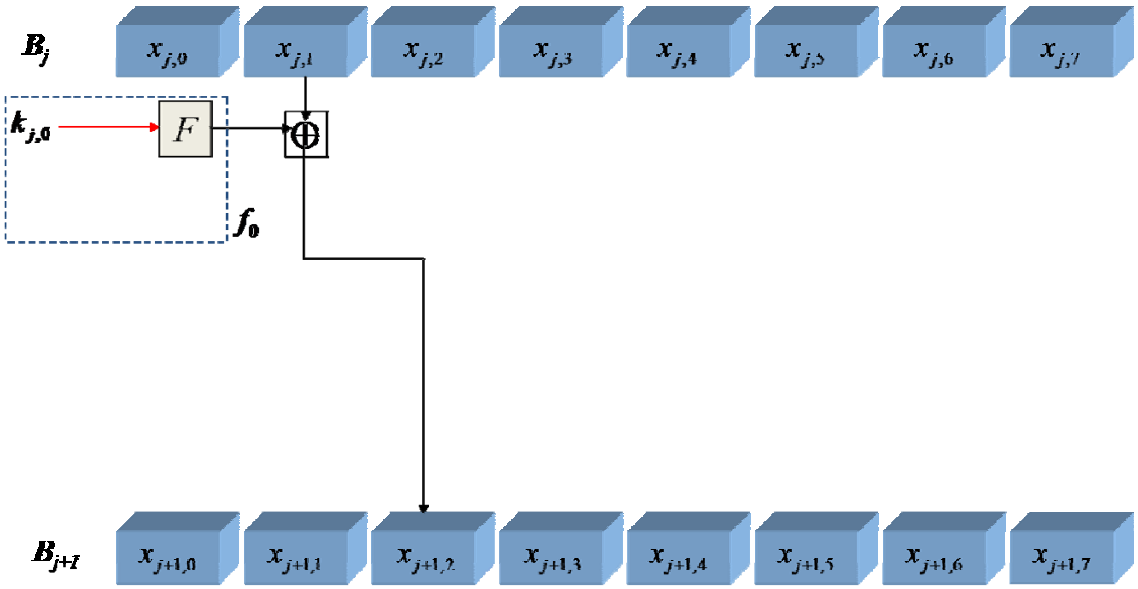


Figura 47 Estructura de cifrado para f_0

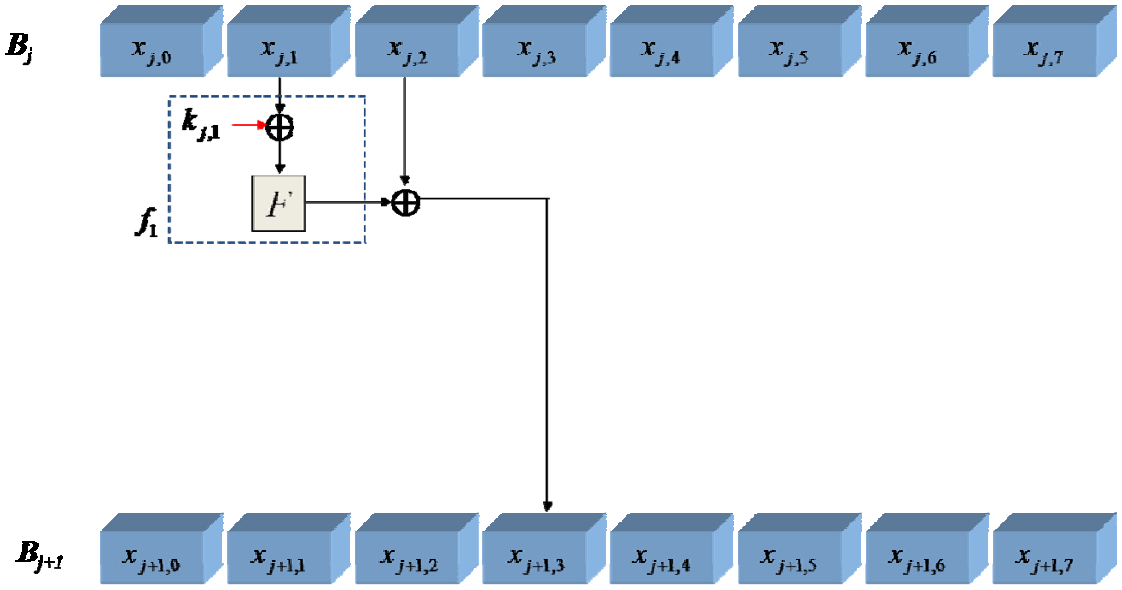


Figura 48 Estructura de cifrado para f_1

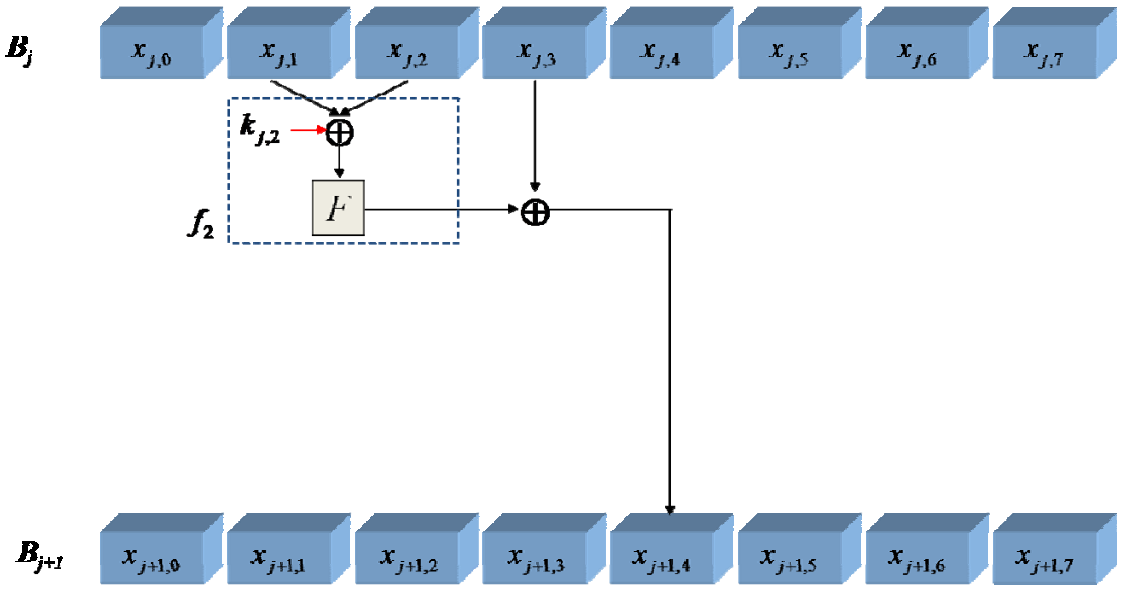


Figura 49 Estructura de cifrado para f_2

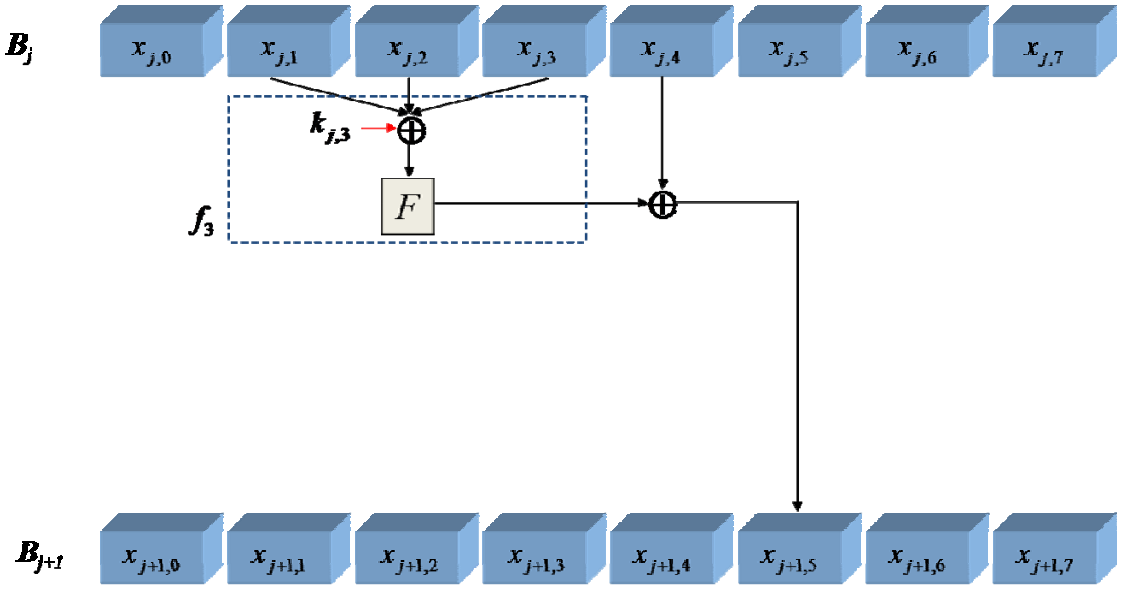


Figura 50 Estructura de cifrado para f_3

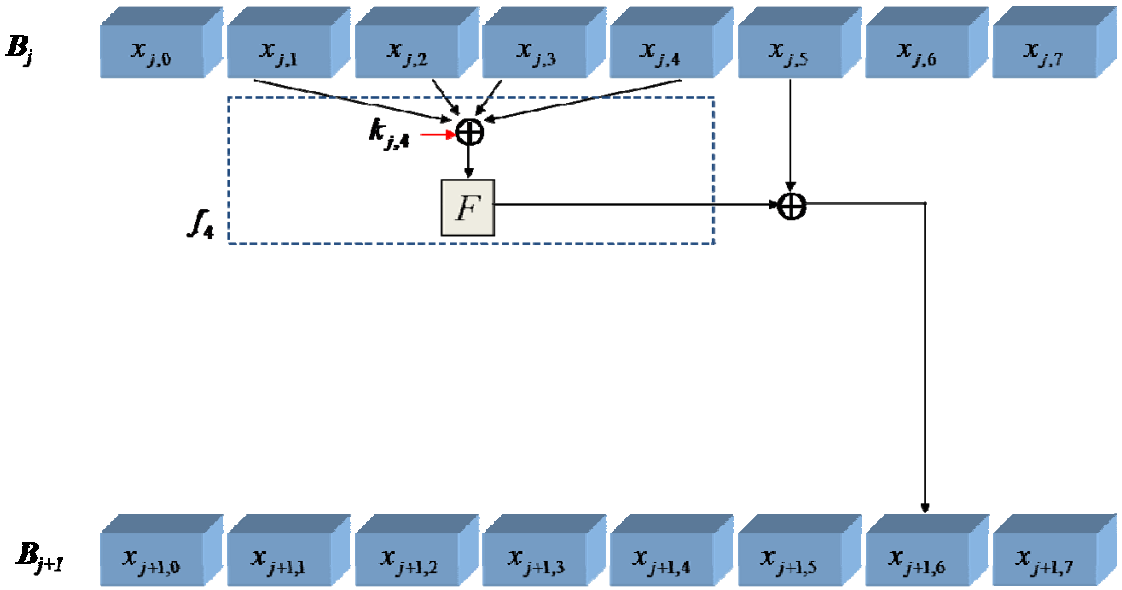


Figura 51 Estructura de cifrado para f_4

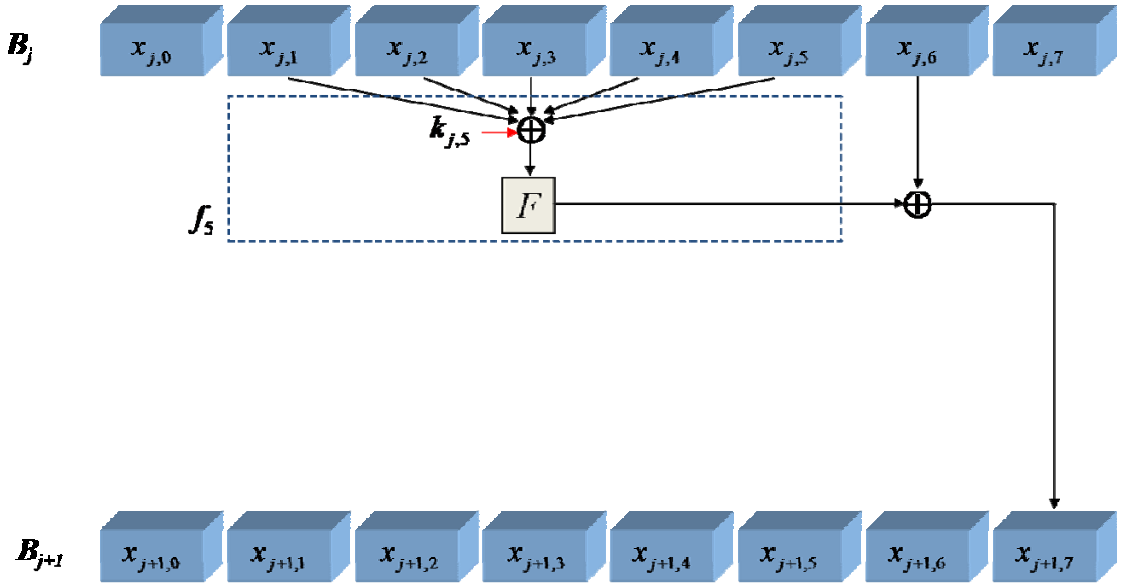


Figura 52 Estructura de cifrado para f_5

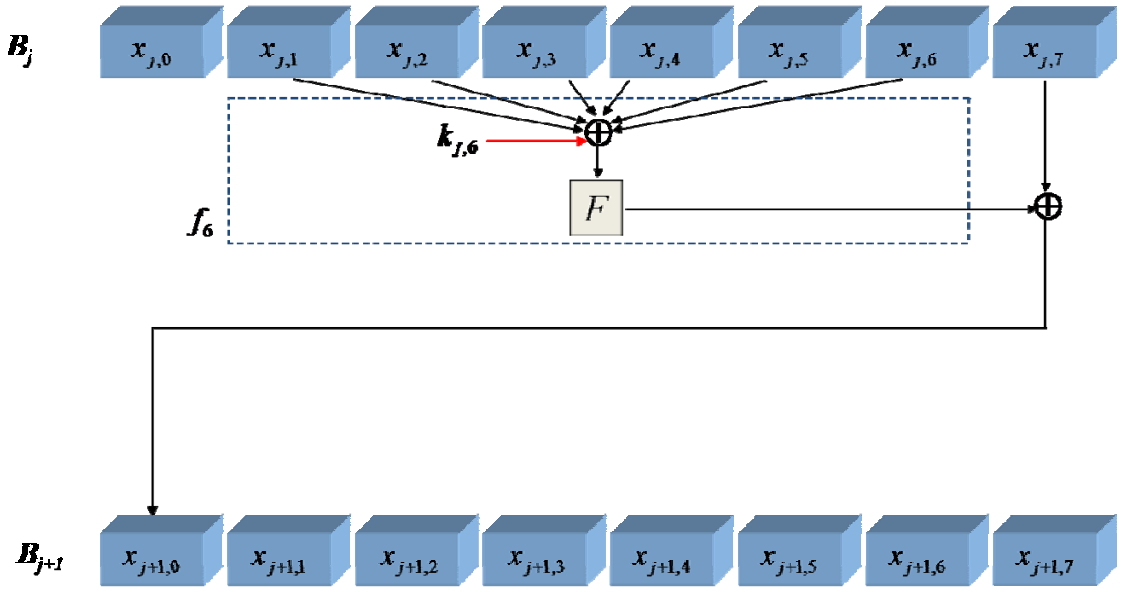


Figura 53 Estructura de cifrado para f_6

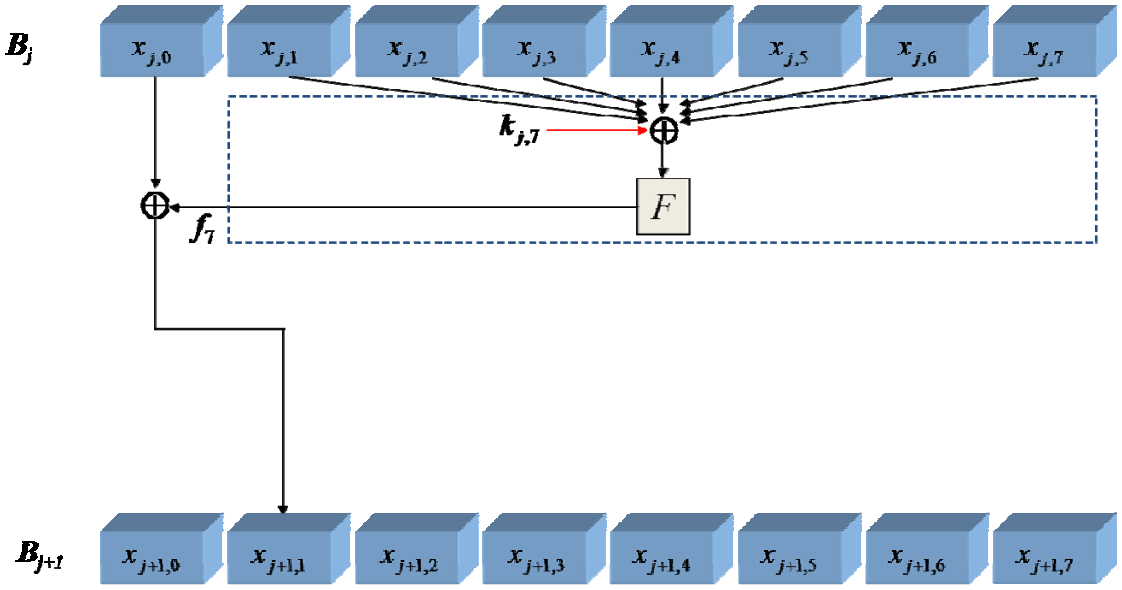


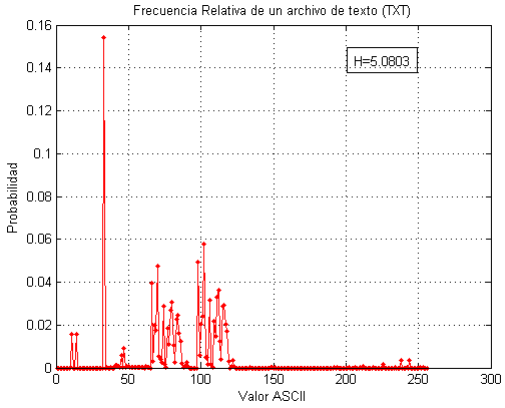
Figura 54 Estructura de cifrado para f_7

Pruebas de funcionalidad

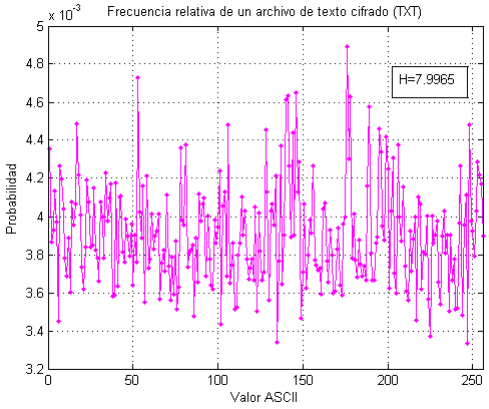
Todo proceso de transformación o cifrado tiene como objetivo afectar tres de las propiedades fundamentales del lenguaje natural. Estas propiedades son la sintáctica, la semántica y la estadística del lenguaje. La sintáctica se encarga de darle coherencia y sentido a los enunciados utilizados en el lenguaje, es decir, son las reglas gramaticales. La semántica se encarga de asignarle significado a los enunciados. Finalmente la estadística se puede asociar con el grado de redundancia de los caracteres utilizados en el lenguaje como se vio anteriormente.

Por otro lado, es importante mencionar que en el lenguaje natural está implícito un grado de redundancia ó repetición de caracteres. Por ejemplo, si se escribe una carta en español, el caracter que más se repetiría es la letra e, pero si se escribe la misma carta ahora en inglés el caracter que más se repetiría sería la letra w.

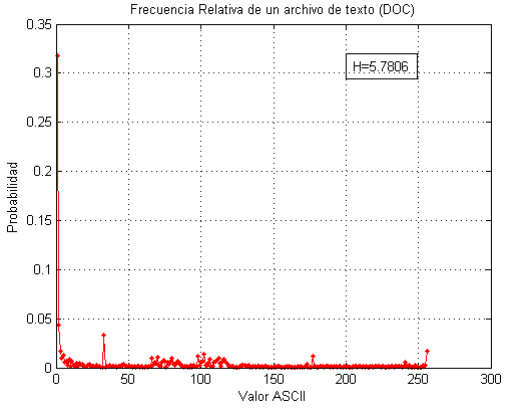
Partiendo del hecho anterior, se presentan una serie de figuras en las cuales se puede apreciar el comportamiento de los archivos considerando la frecuencia de aparición de los caracteres que contienen.



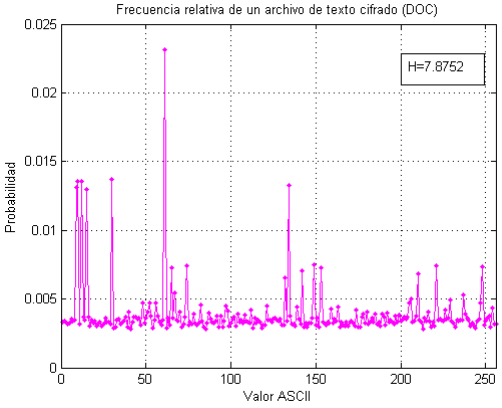
a)



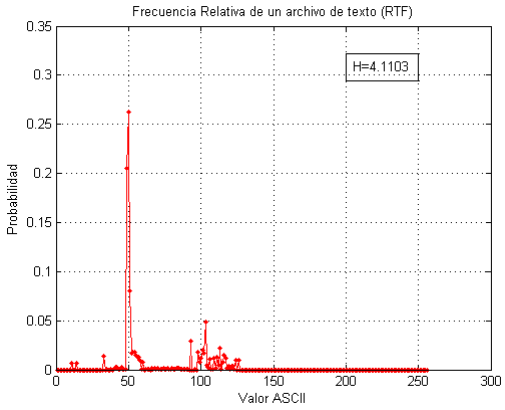
b)



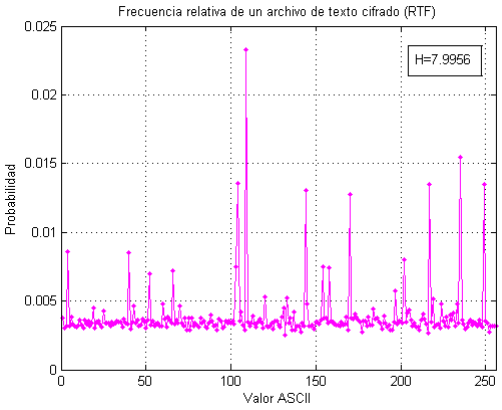
c)



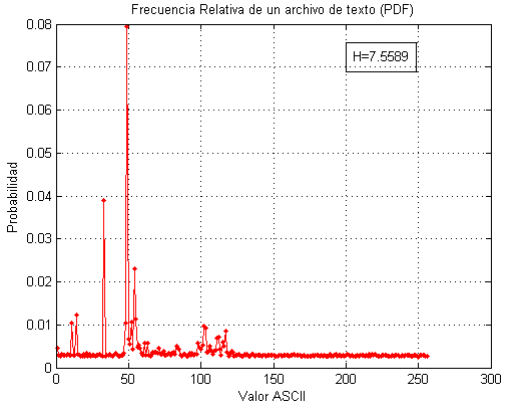
d)


Frecuencia Relativa de un archivo de texto (RTF)
H=4.1103
Probabilidad
Valor ASCII

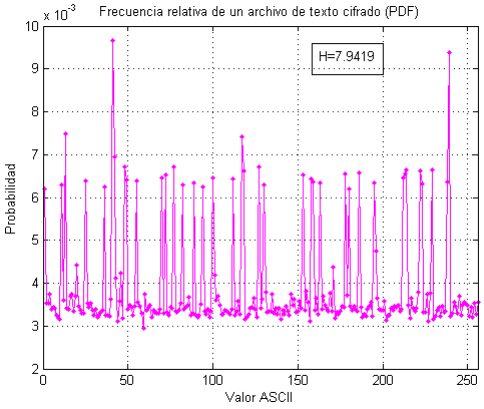
e)


Frecuencia relativa de un archivo de texto cifrado (RTF)
H=7.9956
Probabilidad
Valor ASCII

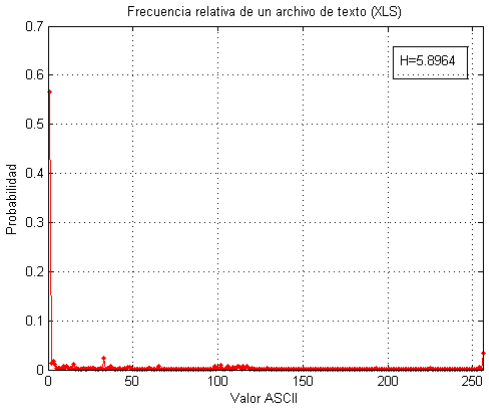
f)


Frecuencia Relativa de un archivo de texto (PDF)
H=7.5589
Probabilidad
Valor ASCII

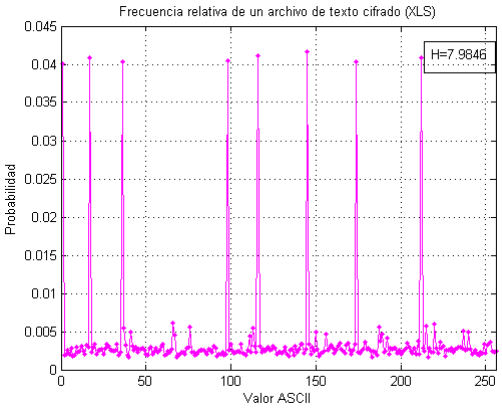
g)


x 10^-3
Frecuencia relativa de un archivo de texto cifrado (PDF)
H=7.9419
Probabilidad
Valor ASCII

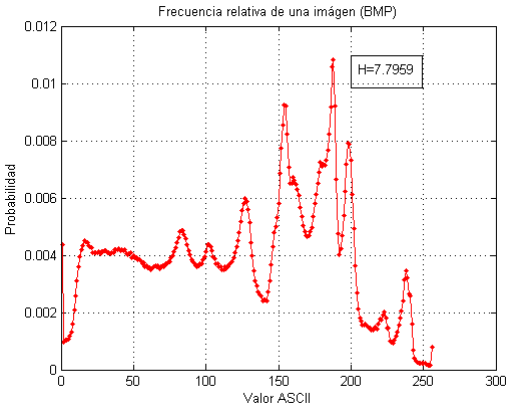
h)


Frecuencia relativa de un archivo de texto (XLS)
H=5.8964
Probabilidad
Valor ASCII

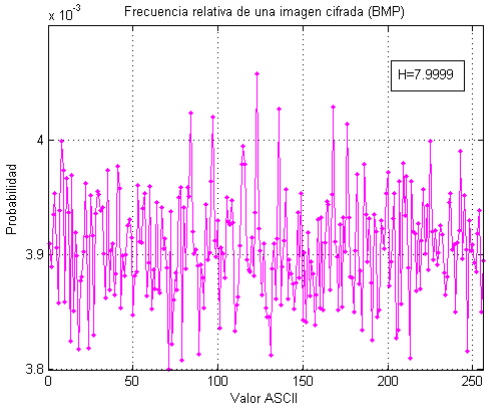
i)


Frecuencia relativa de un archivo de texto cifrado (XLS)
H=7.9846
Probabilidad
Valor ASCII

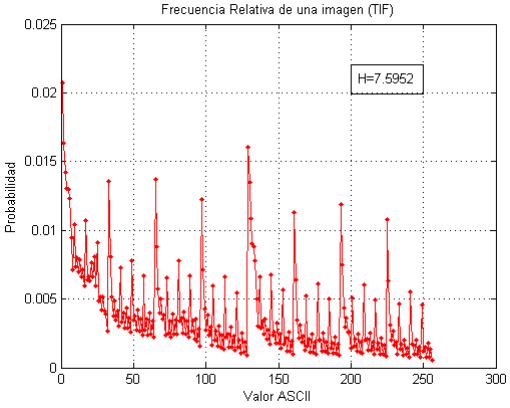
j)


Frecuencia relativa de una imágen (BMP)
H=7.7959
Probabilidad
Valor ASCII

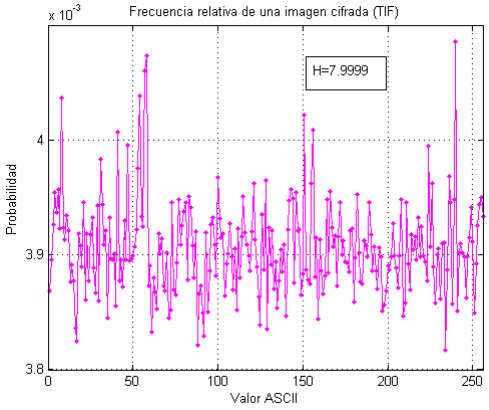
k)


x 10^-3
Frecuencia relativa de una imagen cifrada (BMP)
H=7.9999
Probabilidad
Valor ASCII

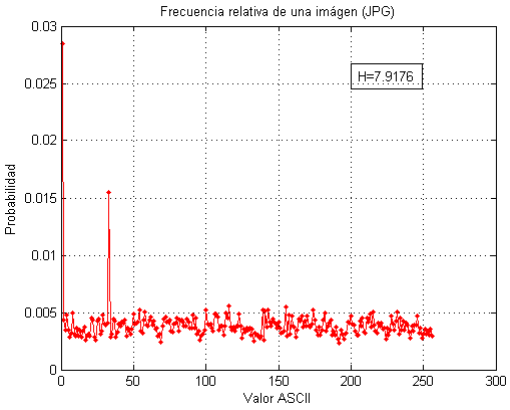
l)


Frecuencia Relativa de una imagen (TIF)
H=7.5952
Probabilidad
Valor ASCII

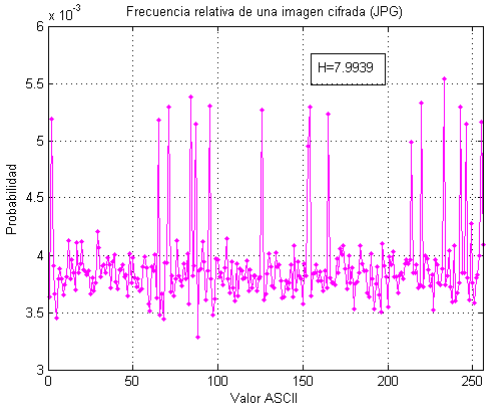
m)


x 10^-3
Frecuencia relativa de una imagen cifrada (TIF)
H=7.9999
Probabilidad
Valor ASCII

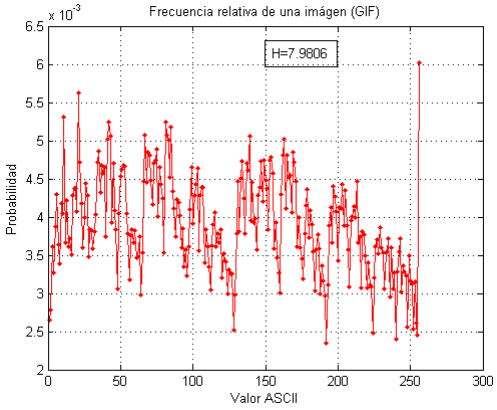
n)


Frecuencia relativa de una imágen (JPG)
H=7.9176
Probabilidad
Valor ASCII

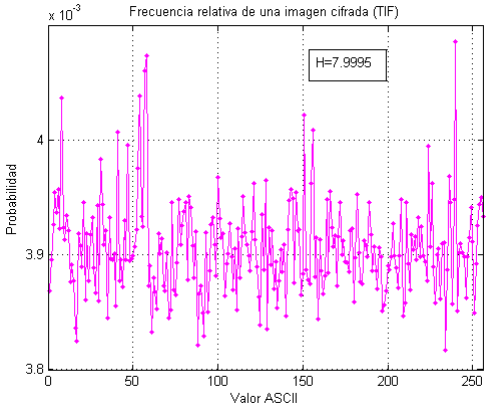
o)


x 10^-3
Frecuencia relativa de una imagen cifrada (JPG)
H=7.9939
Probabilidad
Valor ASCII

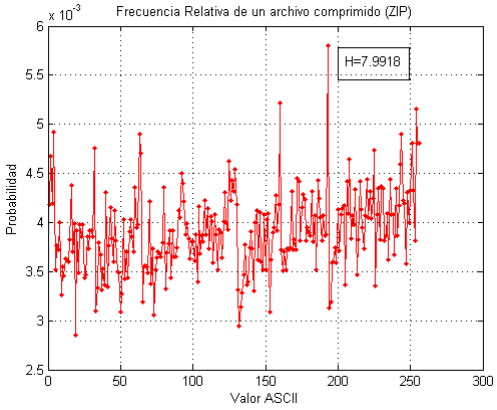
p)


Frecuencia relativa de una imágen (GIF)
H=7.9806
Probabilidad
Valor ASCII

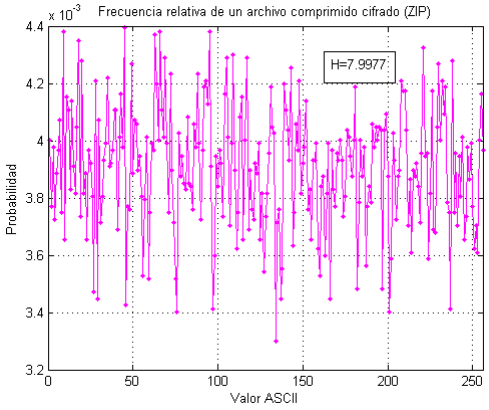
q)


Frecuencia relativa de una imagen cifrada (TIF)
H=7.9995
Probabilidad
Valor ASCII

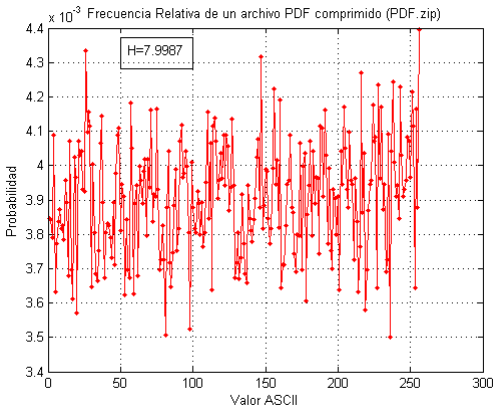
r)


Frecuencia Relativa de un archivo comprimido (ZIP)
H=7.9918
Probabilidad
Valor ASCII

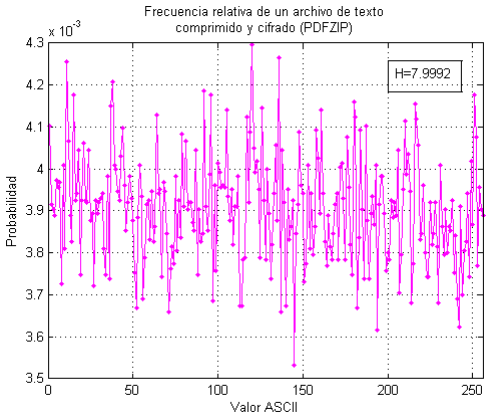
s)


Frecuencia relativa de un archivo comprimido cifrado (ZIP)
H=7.9977
Probabilidad
Valor ASCII

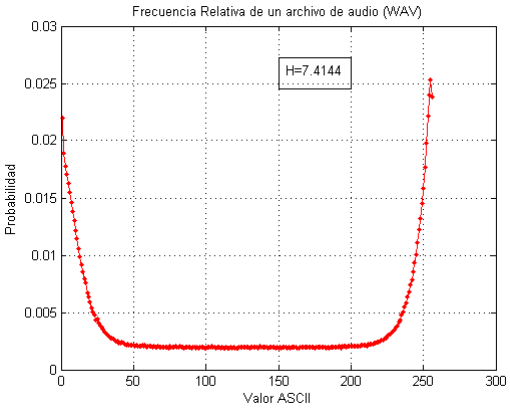
t)


Frecuencia Relativa de un archivo PDF comprimido (PDF.zip)
H=7.9987
Probabilidad
Valor ASCII

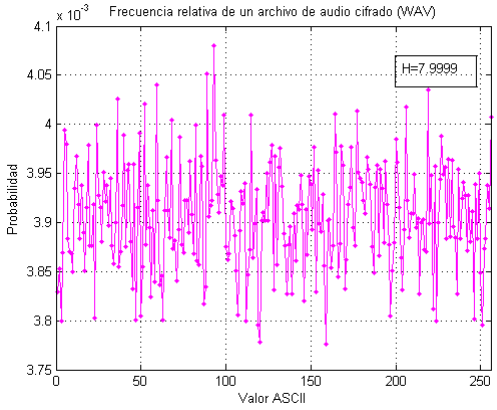
u)


Frecuencia relativa de un archivo de texto comprimido y cifrado (PDFZIP)
H=7.9992
Probabilidad
Valor ASCII

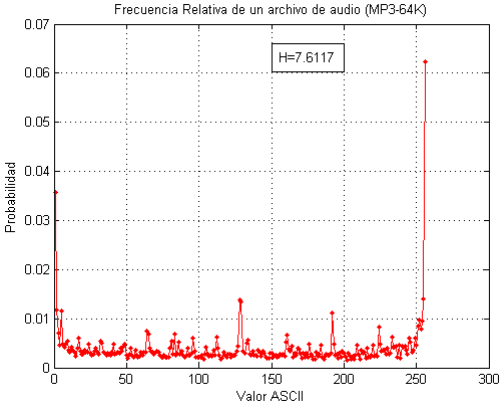
v)


Frecuencia Relativa de un archivo de audio (WAV)
H=7.4144
Probabilidad
Valor ASCII

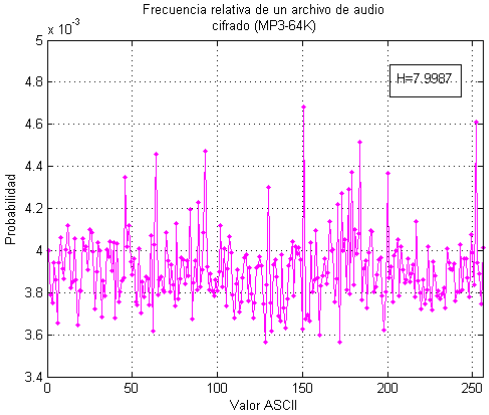
w)


Frecuencia relativa de un archivo de audio cifrado (WAV)
x 10^-3
H=7.9999
Probabilidad
Valor ASCII

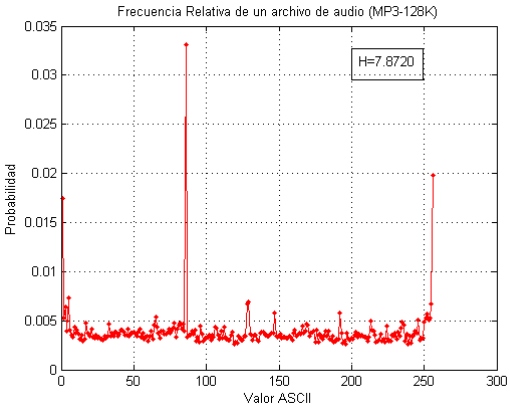
x)


Frecuencia Relativa de un archivo de audio (MP3-64K)
H=7.6117
Probabilidad
Valor ASCII

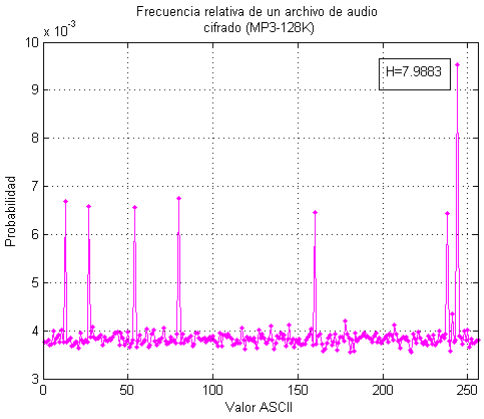
y)


Frecuencia relativa de un archivo de audio cifrado (MP3-64K)
x 10^-3
H=7.9987
Probabilidad
Valor ASCII

z)


Frecuencia Relativa de un archivo de audio (MP3-128K)
H=7.8720
Probabilidad
Valor ASCII

a.1)


Frecuencia relativa de un archivo de audio cifrado (MP3-128K)
x 10^-3
H=7.9883
Probabilidad
Valor ASCII

b.1)

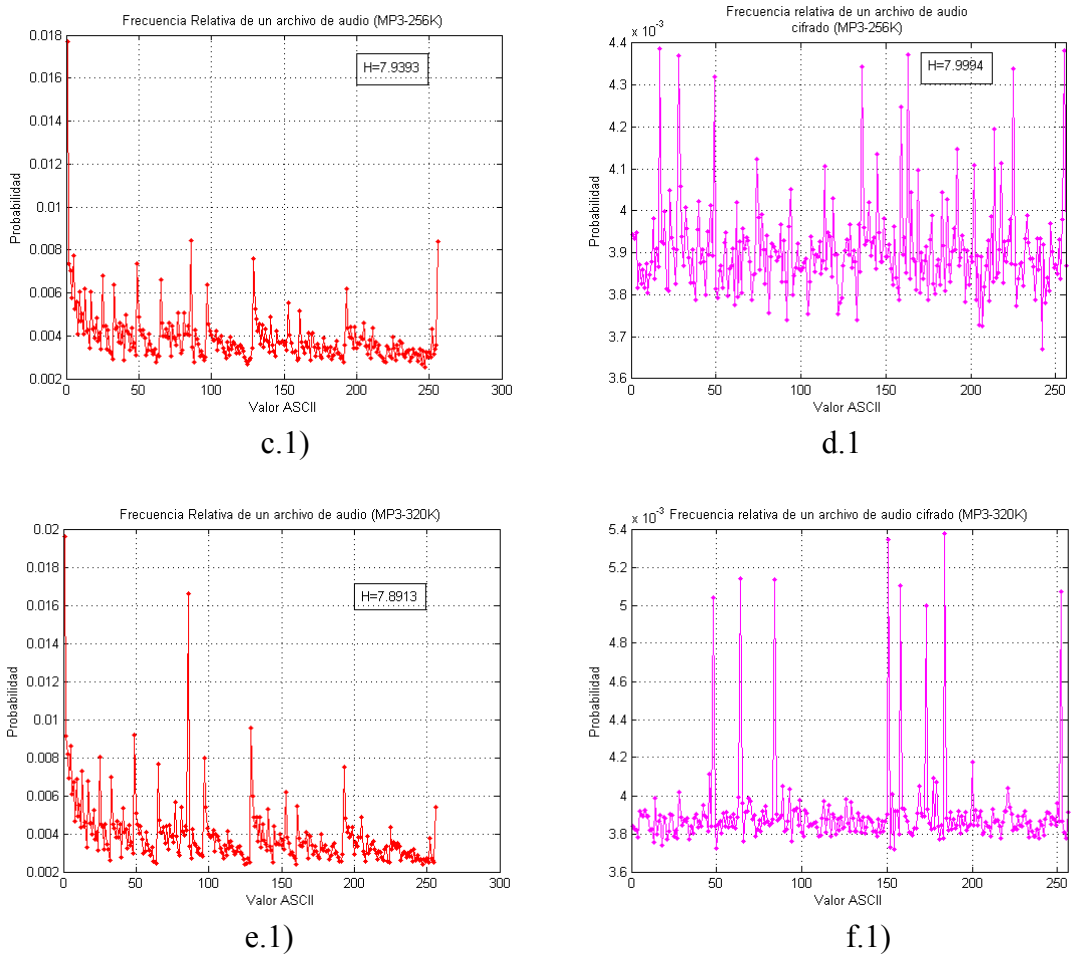


Figura 55 Análisis de la distribución estadística.

En las figuras se puede apreciar, desde un punto de vista estadístico, el comportamiento de los archivos d texto claro. Las figuras del lado izquierdo corresponden a los archivos de texto claro. Las figuras del lado derecho corresponden a los archivos de texto cifrados son el Cifrador caótico de bloques. Dichas figuras se pueden interpretar de la siguiente manera:

Las gráficas de los archivos de texto claro muestran una gran cantidad de picos, mismos que corresponden a la frecuencia de aparición ó repetición de algún carácter en especial.

Una vez que dichos archivos son afectados en sus tres propiedades básicas (sintáctica, semántica y estadística), se puede observar como la gráfica de la distribución tiene a ser más uniforme, esto es, muy parecida a una señal de ruido.

ARCHIVO ORIGINAL	TAMAÑO	ARCHIVO CIFRADO	TAMAÑO	ARCHIVO DESCIFRADO	TAMAÑO
TXT	213KB	213KB	213KB	213KB	213KB
DOC	356KB	356KB	356KB	356KB	356KB
RTF	356KB	356KB	356KB	356KB	356KB
PDF	245KB	245KB	245KB	245KB	245KB
XLS	26KB	26KB	26KB	26KB	26KB
BMP	3.51MB	3.51MB	3.51MB	3.51MB	3.51MB
TIF	3.64MB	3.64MB	3.64MB	3.64MB	3.64MB
JPG	168KB	168KB	168KB	168KB	168KB
GIF	402KB	402KB	402KB	402KB	402KB
ZIP	85KB	85KB	85KB	85KB	85KB
PDFZIP	187KB	187KB	187KB	187KB	187KB
WAV	1.22MB	1.22MB	1.22MB	1.22MB	1.22MB
MP2-64K	210KB	210KB	210KB	210KB	210KB
MP2-128K	420KB	420KB	420KB	420KB	420KB
MP2-256K	840KB	840KB	840KB	840KB	840KB
MP2-320K	1.02MB	1.02MB	1.02MB	1.02MB	1.02MB

En la tabla anterior se puede observar que el tamaño de los archivos se mantiene constante y que no importando el proceso de transformación al que es sometido, su longitud nunca cambia.

Para verificar la integridad de los archivos, se calcula la su función resumen (HASH) a la entrada y a la salida del Cifrador obteniendo los siguientes resultados:

	ARCHIVO TXT	TAMAÑO	FUNCIÓN HASH
Original		213KB	**8a6dcb9c9836d0a0749d9a5907dad6ade76a1c2c853764a2be23038b8928924b**
Recuperado		213KB	**8a6dcb9c9836d0a0749d9a5907dad6ade76a1c2c853764a2be23038b8928924b**

a)

	ARCHIVO DOC	TAMAÑO	FUNCIÓN HASH
Original		356KB	f54269bb7108e386b5ce496b129edcde2f43eb477982188bdcc18b69e08e704f
Recuperado		356KB	f54269bb7108e386b5ce496b129edcde2f43eb477982188bdcc18b69e08e704f

b)

	ARCHIVO RTF	TAMAÑO	FUNCIÓN HASH
Original		1.90MB	282e72b94704094266ea7029d272cdacff2bcf227799a3fed086fff05326611b
Recuperado		1.90MB	282e72b94704094266ea7029d272cdacff2bcf227799a3fed086fff05326611b

c)

	ARCHIVO PDF	TAMAÑO	FUNCIÓN HASH
Original		245KB	a410012acc3b35cffd775f8379edb48ae97a16ad8e67db8765decc31af41c587
Recuperado		245KB	a410012acc3b35cffd775f8379edb48ae97a16ad8e67db8765decc31af41c587

d)

	ARCHIVO XLS	TAMAÑO	FUNCIÓN HASH
Original		26KB	323e26413194c336fef33787148600a4b166f952325dc8f9569f87d5fb3f1d4e
Recuperado		26KB	323e26413194c336fef33787148600a4b166f952325dc8f9569f87d5fb3f1d4e

e)

	ARCHIVO BMP	TAMAÑO	FUNCIÓN HASH
Original		3.51MB	f85e1c10dd9c19087cb6c1370d7cb50ab8e5b31553014e85dc1819914dd8c81a
Recuperado		3.51MB	f85e1c10dd9c19087cb6c1370d7cb50ab8e5b31553014e85dc1819914dd8c81a

f)

	ARCHIVO TIF	TAMAÑO	FUNCIÓN HASH
Original		3.64MB	57fdfe2a4fe1ba75185ff2a672e25142f061dace6649e425b54a831579ec768c
Recuperado		3.64MB	57fdfe2a4fe1ba75185ff2a672e25142f061dace6649e425b54a831579ec768c

g)

	ARCHIVO JPG	TAMAÑO	FUNCIÓN HASH
Original		168KB	8f3c0b8403ec9a7086e5e8dc05600956bcad62f355c3916f87cb98739d698d21
Recuperado		168KB	8f3c0b8403ec9a7086e5e8dc05600956bcad62f355c3916f87cb98739d698d21

h)

	ARCHIVO GIF	TAMAÑO	FUNCIÓN HASH
Original		402KB	c30f8a2385d9c3112c1f7557f048d632899fb4c44356353981e348b0f057f8c2
Recuperado		402KB	c30f8a2385d9c3112c1f7557f048d632899fb4c44356353981e348b0f057f8c2

i)

	ARCHIVO ZIP	TAMAÑO	FUNCIÓN HASH
Original		85KB	e356921520ac420c86303a01fd05d2b095a64c46d2a74fb2273ea6d13a6794cd
Recuperado		85KB	e356921520ac420c86303a01fd05d2b095a64c46d2a74fb2273ea6d13a6794cd

j)

	ARCHIVO PDFZIP	TAMAÑO	FUNCIÓN HASH
Original		187KB	5515689ad38a3c5387865e668952d2be84d25ebf659f0cbf949c3e84da909e03
Recuperado		187KB	5515689ad38a3c5387865e668952d2be84d25ebf659f0cbf949c3e84da909e03

k)

	ARCHIVO WAV	TAMAÑO	FUNCIÓN HASH
Original		1.22MB	0731cf9a6722f65c40238b5ccb35a27d7ee744746b3aaec75a95573ea92cc4cb
Recuperado		1.22MB	0731cf9a6722f65c40238b5ccb35a27d7ee744746b3aaec75a95573ea92cc4cb

l)

	ARCHIVO MP3-64K	TAMAÑO	FUNCIÓN HASH
Original		210KB	c6c487685bdc949ae343a553e726fffdb827f4d58486dc39c22baf1559b877f5
Recuperado		210KB	c6c487685bdc949ae343a553e726fffdb827f4d58486dc39c22baf1559b877f5

m)

	ARCHIVO MP3-128K	TAMAÑO	FUNCIÓN HASH
Original		420KB	349c4af7b5110a8206f88f5636ac73b05762044554b687b3a8492b2121157247
Recuperado		420KB	349c4af7b5110a8206f88f5636ac73b05762044554b687b3a8492b2121157247

n)

	ARCHIVO MP3-256K	TAMAÑO	FUNCIÓN HASH
Original		840KB	c941c087c938a36aacbc79db56207ffc1b048bd8410f2d5fa13f229860caa727
Recuperado		840KB	c941c087c938a36aacbc79db56207ffc1b048bd8410f2d5fa13f229860caa727

o)

	ARCHIVO MP3-320K	TAMAÑO	FUNCIÓN HASH
Original		1.02MB	ec06291edb247b3e0012cdd8d64dcbb5e8b34d2f0c46c2737470df7803603374
Recuperado		1.02MB	ec06291edb247b3e0012cdd8d64dcbb5e8b34d2f0c46c2737470df7803603374

p)

Figura 56 Análisis de la distribución estadística.

Con esta demostración se puede concluir que a pesar de que se afecta las tres propiedades del lenguaje, no se afecta su tamaño o longitud.

CAPÍTULO IV: EVALUACIÓN DEL CIFRADOR PROPUESTO

Resumen

En este capítulo se muestran los criterios de evaluación para un criptosistema de acuerdo a los principios de shannon. Posteriormente, el algoritmo es evaluado empleando conceptos de la teoría de la información como; información mutua, entropía del mensaje de entrada y de salida, y su distribución estadística. Con el principal objetivo de evaluar la aleatoriedad en la distribución estadística se hace uso de la Suite de pruebas estadísticas planteadas el NIST. Finalmente, se compara el desempeño del algoritmo propuesto con los principales algoritmos de cifrado de bloques (DES, TRIPLE-DES, AES, BLOWFISH, etc.) usados actualmente para el cifrado de las comunicaciones.

Antecedentes

La necesidad de contar con números aleatorios y seudo-aleatorios se plantea en muchas aplicaciones criptográficas, por ejemplo, los criptosistemas comunes emplean llaves que deben ser generadas de manera aleatoria. Muchos protocolos criptográficos también requieren de diversas entradas aleatorias o pseudos-aleatorias en varios puntos, para cantidades auxiliares en la generación de firmas digitales, o para la generación de ataques en protocolos de autenticación.

Existen dos tipos básicos de generadores usados para producir secuencias aleatorias: *Generadores de Números Aleatorios* (RNG) y *Generadores de Números Seudo-Aleatorios* (PRNG) Para aplicaciones criptográficas, estos dos tipos de generadores producen una secuencia de ceros y unos que pueden dividirse en sub-secuencias o bloques de números aleatorios.

Aleatoriedad

Una secuencia de bits aleatorios se podría interpretar como el lanzamiento de una moneda no cargada con lados representados por los valores “1” y “0” y para cada lanzamiento teniendo una probabilidad de exactamente ½ de producir un “1” o “0”. Además, los lanzamientos son independientes unos de otros, por lo que el resultado del lanzamiento anterior no afecta los futuros lanzamientos. Una moneda no cargada, es por lo tanto un generador de flujo de bits aleatorios perfecto, tomando en cuenta que los valores “1” y “0” estarán distribuidos aleatoriamente. Todos los elementos de la secuencia son generados independientemente unos de otros y el valor del siguiente elemento en la secuencia, no se puede predecir, independientemente de la cantidad de elementos que ya se han producido.

Obviamente el uso de monedas no cargadas para aplicaciones criptográficas es impráctico, sin embargo, la salida hipotética de tal generador ideal de flujo de bits aleatorios sirve como punto de referencia para la evaluación de generadores de números aleatorios y seudo-aleatorios.

Imprevisibilidad

Los números aleatorios y seudo-aleatorios generados para aplicaciones criptográficas deben ser impredecibles. En el caso de los PRNG, si la semilla es desconocida, el siguiente número de salida en la secuencia debería ser impredecible a pesar de tener conocimiento alguno de los números anteriores. Esta propiedad se conoce como la imprevisibilidad futura. Tampoco debería ser factible determinar la semilla a partir del conocimiento de cualquier valor generado (la imprevisibilidad hacia atrás también es requerida). No debe existir ninguna correlación evidente entre la semilla y cualquier valor generado de esa semilla. Cada elemento de la secuencia debe parecer el resultado de un evento aleatorio independiente con probabilidad ½

Para garantizar la imprevisibilidad futura se debe tener cuidado en la obtención de las semillas. Los valores producidos por los generadores de números seudo-aleatorios son completamente predecibles si se conoce la semilla y el algoritmo para su generación. Considerando que en muchos casos el algoritmo generador se encuentra públicamente disponible, la semilla se debe mantener en secreto y no debe deducirse a partir de la secuencia seudo-aleatoria que este produce. Además, la semilla misma debe ser impredecible.

Generadores de números aleatorios (RNG)

El primer tipo de generador de secuencias es un Generador de Números Aleatorios (RNG por sus siglas en inglés). Un RNG usa una fuente no determinista (fuente de entropía) junto con una función de procesamiento (el proceso de obtención de le entropía) para producir aleatoriedad. El proceso de obtención es necesario para superar cualquier debilidad en la fuente de entropía. Que resulta en la producción de números no aleatorios (la existencia de cadenas largas de ceros y unos). La fuente de entropía típicamente consiste de alguna cantidad física como el ruido en un circuito eléctrico, el ritmo de procesos de un usuario (pulsación de teclas o movimientos del mouse) o los efectos cuánticos en un semiconductor. Se pueden usar diversas combinaciones de dichas entradas.

La salida de un RNG puede ser usada directamente como un número aleatorio o puede ser introducida en un Generador de Números Seudo-aleatorios (PRNG). Par poder ser usada directamente (sin procesos adicionales) la salida de cualquier RNG necesita satisfacer criterios estrictos de aleatoriedad empleados por diversas pruebas estadísticas con el propósito de determinar que las fuentes físicas de entrada de los RNG parecen aleatorias. Por ejemplo, una fuente física como el ruido eléctrico puede contener una superposición de estructuras regulares tales como ondas u otros fenómenos periódicos que pueden parecer aleatorios, pero que aun se consideran como no aleatorios usando pruebas estadísticas.

Par propósitos criptográficos, la salida de los RNG necesita ser impredecible. Sin embargo, algunas fuentes físicas son bastante predecibles. Estos problemas pueden ser eliminados combinando salidas de diferentes tipos de fuentes para usarlas como entradas en los RNG. De cualquier manera, las salidas resultantes de los RNG todavía pueden ser deficientes al momento de ser evaluadas por pruebas estadísticas. Por otro lado, la generación de números aleatorios de alta calidad puede consumir bastante tiempo, convirtiendo tal esfuerzo indeseable cuando se necesita una gran cantidad de números aleatorios. Para producir grandes cantidades de números aleatorios, los generadores de números seudo-aleatorios pueden ser preferibles.

Generadores de números Seudo-aleatorios (PRNG)

El segundo tipo de generador es un Generador de Números Seudo-aleatorios (PRNG). Un PRNG usa un o más entradas y generan múltiples números seudo-aleatorios. Las entradas de los PRNG reciben el nombre de semillas. En situaciones en las que se requiere la imprevisibilidad, la semilla misma debe ser aleatoria e impredecible, por lo tanto, un PRNG debe obtener la semilla de la salida generada por un RNG.

Las salidas de los PRNG normalmente son funciones deterministas de la semilla pero la verdadera aleatoriedad se limita a la generación de semillas. La naturaleza determinista del proceso conduce a la expresión “Seudo-aleatorio”. Cada elemento de una secuencia seudo-aleatoria es reproducible a partir de sus semillas, solamente es necesario contar con la semilla si se requiere la reproducción o validación de la secuencia seudo-aleatoria.

Irónicamente, los números seudo-aleatorios parecen ser más aleatorios que los números aleatorios obtenidos de fuentes físicas. Si una secuencia sudo-aleatoria es construida adecuadamente, cada valor en la secuencia se genera a partir del valor anterior a través de una trasformación la cual parece introducir aleatoriedad adicional. Una serie de tales transformaciones pueden eliminar la correlación entre la entrada y la salida. Por lo tanto, las salidas de un PRNG pueden tener mejores propiedades y por lo tanto generarse más rápido que las salidas de los RNG.

Criterios de evaluación

Existen diferentes herramientas que se pueden considerar para comprobar que una secuencia que se dice ser aleatoria, sea realmente aleatoria, o almenos tenga la apariencia de una señal de ruido, cuya distribución es muy parecida a una señal uniforme.

Como es bien sabido, la teoría de la información es un área de las matemáticas, cuyos conceptos pueden ser aplicados para la evaluación de dichas secuencias. La teoría de la información dispone de conceptos como entropía, información mutua y distribución estadística.

Sin embargo, no se puede hablar de teoría de la información sin mencionar las aportaciones hechas por C. E. Shannon, considerado por muchos como el padre de la teoría de la información. Por tal motivo, el primer punto a considerar al momento de evaluar las secuencias generadas con el Cifrador de bloques caótico, es lo referente a los principios enunciados por Shannon, los cuales definen los criterios de criptosistema seguro.

Información Mutua y Principio de Shannon

Para llevar a cabo una buena evaluación de todo proceso criptográfico, es necesario tener en cuenta los fundamentos teóricos de la criptografía, dando una serie de nociones básicas sobre Teoría de la Información, introducida por Claude E. Shannon a finales de los años cuarenta.

Sin lugar a dudas, esta disciplina permitirá efectuar una aproximación teórica al estudio de la seguridad de cualquier algoritmo criptográfico.

Cantidad de Información

Este concepto se puede introducir partiendo de su idea intuitiva. Para ello se plantea el siguiente ejemplo: supóngase que se tiene una bolsa con nueve bolas negras y una blanca. ¿Cuanta información se obtiene si alguien dice que ha sacado una bola blanca de la bolsa? y ¿cuánta se obtiene si después saca otra y dice que es negra?

Obviamente, la respuesta a la primera pregunta es que aporta bastante información, puesto que es casi seguro que la bola tenía que salir negra. Análogamente, si hubiera salido negra, entonces este suceso aporta poca información. En cuanto a la segunda pregunta, claramente se puede contestar que no aporta ninguna información, ya que al no quedar bolas blancas se sabe que iba a salir negra.

Se puede observar la cantidad de información como una medida de disminución de la incertidumbre acerca de un seceso. Por ejemplo, si al lanzar un dado, el número que ha salido es menor que dos, aporta más información que si el número que ha salido es par.

Se puede decir que la cantidad de información que aporta conocer un hecho es directamente proporcional al número posible de estados que éste tenía a priori. Si inicialmente se cuenta con diez posibilidades, conocer el hecho proporciona más información que si inicialmente se tuvieran dos. Por ejemplo, supone mayor información conocer la combinación ganadora del próximo sorteo de la lotería, que saber si una moneda lanzada al aire va a caer con la cara o cruz hacia arriba. Claramente es más fácil acertar en el segundo caso, puesto que el número de posibilidades a priori (y por tanto la incertidumbre, suponiendo sucesos equiprobables) es menor.

También la cantidad de información es proporcional a la probabilidad de un suceso. En el caso de las bolas se tienen dos sucesos: sacar bola negra, que es más probable y sacar bola blanca, que es menos probable. Sacar bola negra aumenta el grado de certeza inicial de un

90% a un 100%, proporcionando una ganancia del 10%. Sacar una bola blanca aumenta esa misma certeza en un 90% (puesto que se parte de un 10%). Se puede considerar la disminución de incertidumbre proporcional al aumento de certeza, por lo cual se dice que el primer suceso (sacar bola negra) aporta menos información.

Con el objeto de simplificar la notación, se emplea una variable aleatoria *X* para representar los posibles sucesos que se pueden encontrar. Se denota el suceso *i*-enésimo como x_i, $P(x_i)$ será la probabilidad asociada a dicho suceso y *n* será el número de sucesos posibles.

Supóngase ahora que se sabe con total seguridad que el único valor que puede tomar *X* es x_i. Saber el valor de *X* no aporta ninguna información (se conoce de antemano). Por el contrario, si se tiene una certeza del 99% sobre la posible ocurrencia del valor x_i, obtener un x_j aportará bastante información, como ya se ha visto. Este concepto de información es cuantificable y se puede definir de la siguiente forma:

$$I_i = -\log_2(P(x_i))$$

Entropía

Sumando ponderadamente las cantidades de información de todos los posibles estados de una variable aleatoria *X*, obtenemos:

$$H(X) = \sum_{i=1}^{n} p_i \log\left(\frac{1}{p_i}\right) = -\sum_{i=1}^{n} p_i \log(p_i)$$

La segunda igualdad se obtiene considerando el hecho de que $\log\left(\frac{1}{t}\right) = -\log(t)$ para todo número $t > 0$.

La sumatoria anterior se lleva a cabo sobre todos los valores positivos p_i. Adicionalmente, se tiene $0\log(\frac{1}{0})$ igual a cero con respecto a la definición $H(X)$ anterior. Una justificación para esto, es que la entropía debería ser continua y se sabe por cálculos que $\lim_{x\to 0} x\log(x) = 0$. La magnitud $H(X)$ se conoce como la entropía de la variable aleatoria *X*.

Entropía condicional

Supóngase que se tiene ahora una variable aleatoria bidimensional $H(X,Y)$. Recordando las distribuciones de probabilidad más usuales que se pueden definir sobre dicha variable teniendo *n* posibles casos para *X* y *m* para *Y* se tiene:

- Distribución conjunta de (X,Y):

$$p(x_i, y_j)$$

- Distribuciones marginales de *X* e *Y*:

$$p(x_i) = \sum_{j=1}^{m} p(x_i, y_j) \qquad p(y_j) = \sum_{i=1}^{n} p(x_i, y_j)$$

- Distribuciones condicionales de *X* sobre *Y*, y viceversa:

$$p(x_i / y_j) = \frac{p(x_i, y_j)}{p(y_j)} \qquad p(y_j / x_i) = \frac{p(x_i, y_j)}{p(x_i)}$$

La entropía de las distribuciones que se acaban de referir se define de la siguiente manera:

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j)\log_2(p(x_i, y_j))$$

$$H(X / Y = y_j) = -\sum_{i=1}^{n} p(x_i / y_j)\log_2(p(x_i / y_j))$$

Así, como existe una ley de probabilidad total, análogamente se define la ley de entropías totales:

$$H(X,Y) = H(X) + H(Y/X) = H(Y) + H(X/Y)$$

Cumpliéndose además, si *X* e *Y* son variables independientes:

$$H(X,Y) = H(X) + H(Y)$$

Teorema de disminución de la entropía: La entropía de una variable *X* condicionada por otra *Y* es menor o igual a la entropía de *X*, alcanzándose la igualdad si y solo si las variables *X* e *Y* son independientes.

Este teorema representa una idea intuitiva bien clara: conocer algo acerca de la variable *Y* puede ayudar a saber más sobre *X* (tener menos entropía), pero en ningún caso hará aumentar la incertidumbre.

La expresión que define la ley de las entropías totales también se conoce como regla de la cadena para las entropías. Se generaliza a cualquier número de variables. Por ejemplo, tenemos:

$$H(X,Y,Z) = H(X) + H(Y/X) + H(Z/X,Y) = H(Y) + H(Z/Y) + H(X/Y,Z)$$

Cantidad de información entre dos variables: Información mutua

Shannon propuso una medida para la cantidad de información que aporta sobre una variable el conocimiento de otra. Se define la cantidad de información de Shannon que la variable *X* contiene sobre *Y* como:

$$I(X:Y) = H(X) - H(X/Y) = H(X) + H(Y) - H(X,Y)$$

Considerando que $H(X) - H(X/Y) = H(X) + H(Y) - H(X,Y)$ y $H(X,Y) = H(Y,X)$ se tiene lo siguiente:

$$I(X:Y) = I(Y:X)$$

Por lo tanto la expresión $I(X:Y)$ es la información mutua de *X*, *Y*. Por lo tanto, la información que aporta *X* sobre *Y* es igual a la información que porta *Y* sobre *X*.

Criptosistema seguro de Shannon

Se dice que un criptosistema es seguro si la cantidad de información que nos aporta el hecho de conocer el mensaje cifrado *C* sobre la entropía del texto plano *M* vale cero. Es decir:

$$I(C,M) = 0$$

Esto significa que la distribución de probabilidad que inducen todos los posibles mensajes no cifrados no cambia si conocemos el mensaje cifrado. Para una mejor comprensión, supóngase que si se modifica dicha distribución: El hecho de conocer un mensaje cifrado, al variar la distribución de probabilidad sobre *M* haría unos mensajes más probables que otros y por consiguiente unas claves de cifrado (aquellas que permiten llegar de los *M* más probables al *C* concreto que tenga en cada momento) más probables que otras.

Repitiendo esta operación muchas veces con mensajes diferentes, cifrados con la misma clave, se podría ir modificando la distribución de probabilidad sobre la clave Empleada asta obtener un valor de clave mucho más probable que otros, permitiendo romper el criptosistema.

Si por el contrario el criptosistema cumpliera la condición anterior, jamás se podría romper, ni siquiera empleando una computadora con capacidad de proceso infinita. Por ello, los criptosistemas que cumplen la condición de Shannon se denominan también criptosistemas ideales.

Se puede considerar también que para que un criptosistema sea seguro según el criterio de Shannon, la cardinalidad del espacio de claves ha de ser al menos igual que la del espacio de mensajes. En otras palabras, la clave ha de ser al menos tan larga como el mensaje a cifrar. En la práctica, esto vuelve inútiles a este tipo de criptosistemas en la práctica.

Redundancia

Si alguna persona leyera un lenguaje en el que faltan algunas letras, normalmente puede reconstruirlo. Esto ocurre porque casi todos los símbolos de un lenguaje en lenguaje natural contienen información que se puede extraer de los símbolos de alrededor (información que, en la práctica, se está enviando dos veces), o en otras palabras, porque el lenguaje natural es redundante. Puesto que se tienen mecanismos para definir la cantidad de información que presenta un seceso, se puede intentar medir el exceso de información (redundancia) de un lenguaje. Pare ello, es necesario entender las siguientes definiciones:

- *Índice de un lenguaje.* Se define el índice de un lenguaje para mensajes de longitud *k* como:

$$r_k = \frac{H_k(M)}{k}$$

  Siendo $H_k(M)$ la entropía de todos los posible mensajes de longitud *k*. Se está midiendo el número de bits de información que aporta cada carácter en mensajes de una longitud determinada. Para idiomas como el inglés, r_k suele valer alrededor de 1.3 bis / letra para valores pequeños de *k*.

- *Índice absoluto de un lenguaje.* Es el máximo número de bits de información que pueden ser codificados en cada caracter, asumiendo que todas las combinaciones de caracteres son igualmente probables. Suponiendo *m* letras diferentes en el alfabeto (27 en el saco del español), este índice vale:

$$R = \log_2(m)$$

En el caso del español se podrían codificar 4.7 bits / letra aproximadamente, luego parece que el nivel de redundancia (asumiendo que su índice *r* sea parecido al del inglés) es alto.

- Finalmente, la redundancia de un lenguaje se define como la diferencia entre las dos magnitudes anteriores:

$$D = R - r$$

También se define el índice de redundancia como el siguiente cociente:

$$I = \frac{D}{R}$$

Una de las aplicaciones de la teoría de la información es la compresión de datos, que simplemente trata de eliminar la redundancia dentro de un archivo (considerando cada byte como un mensaje elemental y codificándolo con más o menos bits según su frecuencia d aparición).

Se pueden aplicar varias pruebas estadísticas a una secuencia para tratar de comparar y evaluar dicha secuencia con una secuencia realmente aleatoria. La aleatoriedad es una propiedad probabilística, es decir, las propiedades de una secuencia aleatoria que pueden ser caracterizadas y descritas en términos de probabilidad. Cuando se aplican pruebas estadísticas sobre una secuencia verdaderamente aleatoria, el resultado probable es conocido a priori y puede ser descrito en términos probabilísticos. Existe un número infinito de posibles pruebas estadísticas, cada una evaluando la presencia o ausencia de un 'patrón', el cual si es detectado, indicará si la secuencia no es aleatoria. Debido a que existen muchas pruebas estadísticas que se aplican para determinar si una secuencia es o no aleatoria, se considera que ningún conjunto finito específico es "completo." Además, los resultados de las pruebas estadísticas deben ser interpretados con cierto cuidado y precaución, a fin de evitar conclusiones incorrectas acerca de un generador específico.

Con el objetivo de evaluar la aleatoriedad de las secuencias obtenidas en el proceso de cifrado, en esta tesis se hace uso de una suite de pruebas estadísticas planteadas por el NIST, mismas que se describen en los párrafos siguientes.

NIST

En la práctica existen muchas estrategias distintas empleadas en el análisis estadístico de un generador de secuencias aleatorias. El NIST, ha adoptado la estrategia esbozada en la figura XX. La figura XX proporciona un ejemplo de arquitectura de las cinco etapas involucradas en la evaluación estadística de un generador de secuencias aleatorias.

ETAPA 1: SELECCIÓN DE UN GENERADOR

La selección de hardware o software (algoritmo criptográfico) basado en un generador para su evaluación. El generador debe producir una secuencia binaria de 0's y 1's de una longitud *n* determinada. Ejemplos de generadores seudo-aleatorios (PRNG) que pueden ser seleccionados incluyen al DES-basado en PRNG de ANSI X9.17, así como dos métodos más que se especifican en FIPS 186 y se basan en los algoritmos seguros SHA-1 y DES.

ETAPA 2: GENERACIÓN DE LA SECUENCIA BINARIA

Para una secuencia de longitud fija *n* y el generador preseleccionado, se construye un conjunto de *m* secuencias binarias y posteriormente se guardan en un archivo.

ETAPA 3: EJECUCIÓN DE LA SUITE DE PRUEBAS ESTADÍSTICAS

Se invoca la suite de pruebas estadísticas del NIST usando el archivo generado en la etapa 2 y la secuencia de la longitud deseada. Se seleccionan las pruebas estadísticas y los parámetros de entrada que deben aplicarse.

ETAPA 4: EXAMINAR LOS VALORES DE *P*

La suite de pruebas estadísticas genera un archivo de salida con los valores intermedios relevantes, como las pruebas estadísticas y los valores de *P* para cada prueba estadística. Basándose en los valores de *P*, se pueden hacer conclusiones respecto a la calidad de la secuencia.

ETAPA 5: EVALUACIÓN: ASIGNACIÓN DE LOS VALORES PASA/FRACASA

Por cada prueba estadística se produce un conjunto de valores *P* (correspondiente al conjunto de secuencias). Para un nivel de significación se espera que cierto porcentaje de valores *P* indiquen el fracaso. Por ejemplo, si el nivel de significación es 0.01 ($\alpha = 0.01$), entonces se espera que aproximadamente el 1% de las secuencias fracasen. Una secuencia aprueba una prueba estadística siempre y cuando el valor de $P \geq \alpha$ de lo contrario falla. En consecuencia, por cada prueba estadística, la proporción de secuencias que aprueban se calcula y analiza

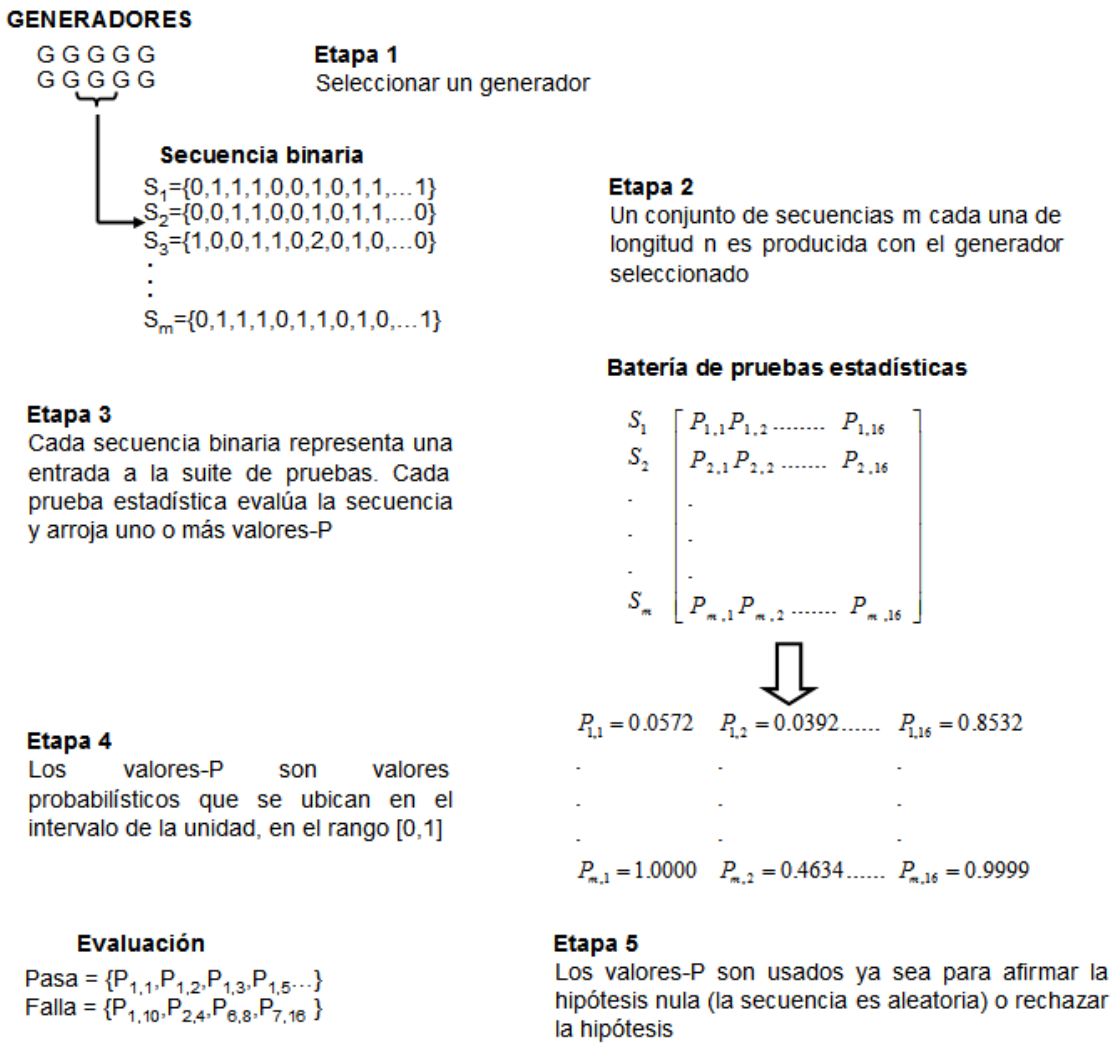


Figura 57 Arquitectura de la suite de pruebas estadísticas del NIST.

INTERPRETACIÓN DE LOS RESULTADOS EMPÍRICOS

Los tres escenarios representan eventos que pueden ocurrir debido a las pruebas empíricas. Caso 1: El análisis de los valores-P no indica una desviación de la aleatoriedad. Caso 2: El análisis indica claramente una desviación de la aleatoriedad. Caso 3: El análisis es concluyente.

La interpretación de los resultados empíricos puede realizarse en cualquier número de formas.
Los dos enfoques que el NIST ha adoptado incluyen (1) el examen de la proporción de las secuencias que aprueban un test estadístico y (2) la distribución de los valores-P para comprobar la uniformidad.

En caso de que cualquiera de estos dos enfoques falle (la hipótesis nula correspondiente debe ser rechazada), se deben llevar a cabo experimentos numéricos adicionales sobre diferentes muestras del generador para determinar si el fenómeno era una anomalía estadística o una clara evidencia de no aleatoriedad.

PROPORCIÓN DE LAS SECUENCIAS QUE SUPERAN UNA PRUEBA

Dados los resultados empíricos para una prueba estadística particular, calcular la proporción de las secuencias que pasan, por ejemplo, si 1000 secuencias binarias fueron probadas (m=1000), $\alpha = 0.01$ (el nivel de significancia) y 996 secuencias binarias arrojaron un valor-P $\geq$ .01 entonces la proporción es 996/1000=0.9960.

El rango de proporciones aceptables se determina usando el intervalo de confianza definido

$$\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}$$

Donde $\hat{p} = 1 - \alpha$ y m es el tamaño de la muestra. Si la proporción cae fuera de este intervalo entonces existe evidencia de que los datos no son aleatorios. Nótese que pudieron haber sido usados otros valores de la desviación estándar. Para el ejemplo de arriba, el intervalo de confianza es

$$.99 \pm 3\sqrt{\frac{.99(.01)}{1000}} = .99 \pm 0.0094392$$

La proporción se debe ubicar por encima de 0.9805607. Esto se puede ilustrar usando una gráfica como en la siguiente figura.

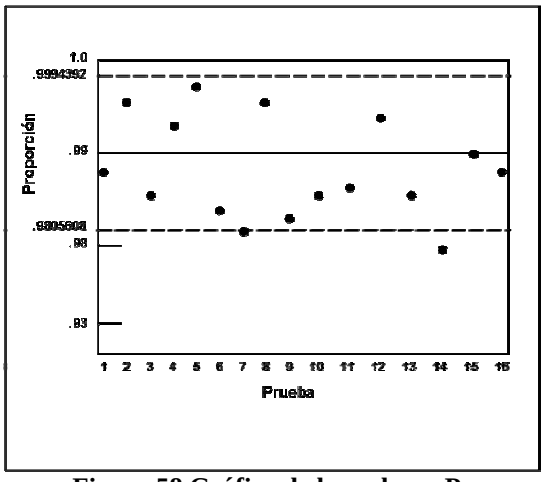


Figura 58 Gráfica de los valores-P.

El intervalo de confianza se calculó usando una distribución normal como una aproximación a la distribución binomial, la cual es considerablemente precisa para muestras de gran tamaño ($n \geq 1000$).

DISTRIBUCIÓN UNIFORME DE LOS VALORES-P

La distribución de los valores-P es examinada para garantizar uniformidad. Esto se puede ilustrar usando un histograma ver (figura) donde el intervalo entre 0 y 1 es dividido en 10 sub-intervalos y los valores-P que se ubican dentro de cada sub-intervalo son contados y mostrados.

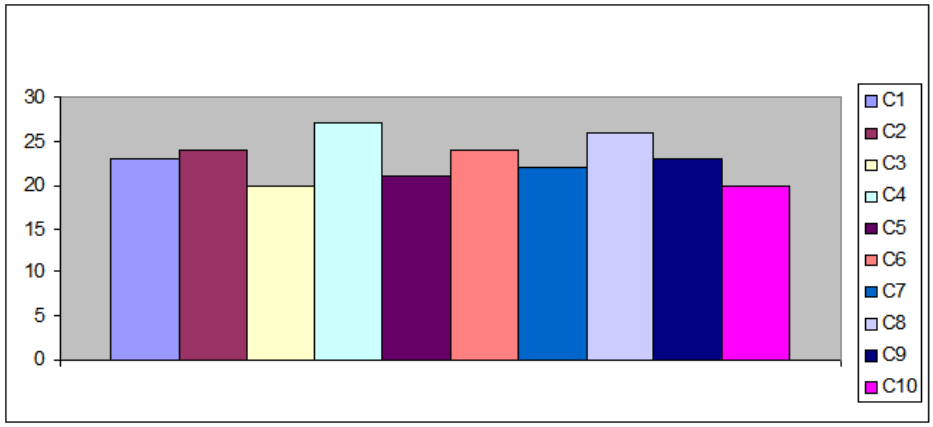


Figura 59 Histograma de los valores-P.

La uniformidad también se puede determinar aplicando la prueba de χ^2 y la determinación de un valor-P correspondiente a la propiedad de ajuste de la prueba de la distribución sobre los valores-P obtenidos para una prueba estadística arbitraria. (el valor-P de los valores-P). Esto se logra calculando

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - s/10)^2}{s/10}$$

Donde F_i es el número de valores-P en el sub-intervalo *i*, y *s* es el tamaño de la muestra. Un valor-P es calculado de tal manera que $valor_T - P = igamc(9/2, \chi^2/2)$. Si $valor_T - P \geq 0.0001$ entonces la secuencia se puede considerar uniformemente distribuida.

Pruebas estadísticas de aleatoriedad

La aleatoriedad de una secuencia de bits es caracterizada y descrita en términos de probabilidad. La Suite de pruebas estadísticas del NIST es altamente recomendada entre la lista de baterías de pruebas estadísticas y además se obtiene de manera gratuita. Esta Suite incluye más de una docena de pruebas estadísticas independientes y computacionalmente intensivas. Muchas de estas pruebas arrojan su correspondiente P-valor [Soto, J.] El P-valor es la probabilidad de obtener una prueba estadística tan impresionante como la observada si la secuencia es aleatoria. En otras palabras, el P-valor resume la fuerza de la evidencia en contra de la hipótesis de aleatoriedad perfecta.

Suite de pruebas estadísticas del NIST

La Suite del NIST cuenta con 16 pruebas estadísticas. Estas pruebas evalúan la presencia de un patrón, el cual, si es detectado indicaría que la secuencia no es aleatoria. En cada prueba, se calcula un P-valor. El nivel de significancia α para todas las pruebas de la suite se establece en 1%. Un P-valor de cero indicaría que la secuencia no es aleatoria en absoluto.

Un P-valor menor que α significaría que la secuencia no es aleatoria con una certeza de un 99%. Si el P-valor es mayor que α, concluimos que la secuencia es aleatoria con una certeza del 99%.

La siguiente tabla muestra los resultados obtenidos por la suite de pruebas estadísticas del NIST. En ella se puede observar el P-valor arrojado para cada prueba, siendo dichos resultados congruentes con los valores de α definidos por la suite.

* * * * * * * Pruebas de aleatoriedad del NIST * * * * * * *			
No.	**Prueba**	**P-valor**	**Conclusión**
1	Frequency	**0.282626**	***Pasa***
2	Block Frequency	**0.339271**	***Pasa***
3	Cusum-Forward	**0.881662**	***Pasa***
4	Cusum-Reverse	**0.224821**	***Pasa***
5	Runs	**0.336918**	***Pasa***
6	Long Runs of Ones	**0.343176**	***Pasa***
7	Rank	**0.407091**	***Pasa***
8	Spectral DFT	**0.455937**	***Pasa***
9	Non-Overlapping Templates	**0.798137**	***Pasa***
10	Overlapping Templates	**0.534146**	***Pasa***
11	Universal	**0.213309**	***Pasa***
12	Approximate Entropy	**0.717714**	***Pasa***
13	Random Excursions	**0.964295**	***Pasa***
14	Random Excursions Variant	**0.911413**	***Pasa***
15	Linear Complexity	**0.964295**	***Pasa***
16	Serial	**0.639202**	***Pasa***

Revisión de la proporción de secuencias aprobadas

En el reporte final de resultados generado por la suite, el valor conocido como proporción se enlista para cada prueba. La proporción es el número de secuencias que tienen un P-valor mayor que el nivel de significancia α, dividido entre el número total de bits de la secuencia probada. Este el porcentaje de pruebas aprobadas. Este hecho se muestra con más detalle en la parte de interpretación de los resultados empíricos correspondiente al tema Criterios de Evaluación de este mismo capítulo.

Comparación con otros procesos de cifrado

Como parte de las pruebas de desempeño y evaluación a que se sometió el cifrador caótico de bloques no solo se consideró la evaluación de las secuencias generadas desde un punto de vista estadístico, aplicando 16 pruebas de estadísticas de aleatoriedad consideradas internacionalmente para evaluar secuencias criptográficas (NIST SP 800-22rev1), sino que también se consideraron conceptos de teoría de la información como entropía e información mutua.

La tabla muestra una comparación de los valores de la entropía, obtenidos con el cifrador caótico de bloques, contra los valores de la entropía obtenidos con tres de los algoritmos criptográficos de bloques más usados actualmente en las comunicaciones. Estos algoritmos son AES, DES y una variante del algoritmo DES conocida como 3DES ó Triple-DES.

La estructura general de cualquier algoritmo criptográfico de bloques se basa en una red de Feistel. Partiendo de la premisa anterior y considerando el hecho de que el cifrador caótico de bloques, desarrollado y evaluado en estas tesis tiene como estructura general una red de Feistel, se decidió comparar su nivel de robustez con los algoritmos mencionados arriba.

* * * * * * * * * * * * Entropías * * * * * * * * * * * *				
Archivo	ALGORITMO			
	Logístico	**AES**	**DES**	**TRIPLE-DES**
TXT	**7.9965**	**7.9992**	**7.9992**	**7.9992**
DOC	**7.9752**	**7.9994**	**7.9990**	**7.9990**
RTF	**7.9955**	**7.9999**	**7.9999**	**7.9999**
PDF	**7.9419**	**7.9992**	**7.9992**	**7.9992**
XLS	**7.9548**	**7.9978**	**7.9451**	**7.9451**
BMP	**7.9999**	**7.9999**	**7.9999**	**7.9999**
TIF	**7.9999**	**8.0000**	**7.9999**	**7.9999**
JPG	**7.9939**	**7.9989**	**7.9976**	**7.9976**
GIF	**7.9995**	**7.9996**	**7.9995**	**7.9995**
ZIP	**7.9977**	**7.9979**	**7.9977**	**7.9977**
PDFZIP	**7.9992**	**7.9991**	**7.9991**	**7.9991**
WAV	**7.9999**	**7.9999**	**7.9999**	**7.9999**
MP3-64K	**7.9987**	**7.9990**	**7.9989**	**7.9989**
MP3-128K	**7.9883**	**7.9996**	**7.9995**	**7.9995**
MP3-256K	**7.9994**	**7.9998**	**7.9997**	**7.9997**
MP3-320K	**7.9976**	**7.9999**	**7.9998**	**7.9998**

El valor más alto que puede tener $H(X)=8$, ya que $\log_2(256)=8$, siendo 256 todos los caracteres del código ASCII extendido. Por lo tanto cuando $H(X)=8$ se considera que todos los valores de la secuencia cifrada son equiprobables, en otras palabras, todos los eventos tienen la misma probabilidad de ocurrencia. Una entropía con valores cercanos a 8 es un buen parámetro para considerar que el algoritmo bajo evaluación es robusto. Como se puede ver en la tabla, el cifrador caótico de bloques al igual que los algoritmos AES, DES Y 3DES exhibe una entropía muy cercana a la ideal.

Algo que hay que resaltar es que el valor de la entropía en las imágenes como .jpg, .gif, es mayor, esto se debe a que su estructura ya ha sido afectada por algún algoritmo de compresión antes de pasar por algún proceso de cifrado. Este hecho se puede ver en todos los archivos que hayan sufrido un proceso de compresión.

La siguiente tabla muestra el cálculo de la información mutua sobre los archivos cifrados con los algoritmos AES, DES 3DES y logístico o cifrador caótico de bloques, pero además se agregaron dos algoritmos de bloques más GOST y Skipjack. De acuerdo a la definición dada por Shannon para un criptosistema seguro, el valor ideal de la información mutua vale cero. En la tabla se puede ver que todos los valores calculados con cada uno de los algoritmos arrojan un valor muy cercano a cero, lo cual indica que dichos algoritmos son congruentes con la definición de Shannon.

* * * * * * * * * * * * * * Información Mutua * * * * * * * * * * * * * *						
Archivo	ALGORITMO					
	AES	**Logístico**	**DES**	**Triple-DES**	**GOST**	**Skipjack**
TXT	**0.0693**	**0.1777**	**0.0689**	**0.0698**	**0.0689**	**0.0688**
DOC	**0.1432**	**0.7356**	**0.1437**	**0.1443**	**0.1451**	**0.1429**
RTF	**0.0075**	**0.8744**	**0.0078**	**0.0078**	**0.0078**	**0.0078**
PDF	**0.2056**	**0.6712**	**0.2061**	**0.2067**	**0.2056**	**0.2098**
XLS	**0.5629**	**0.6998**	**0.5644**	**0.5682**	**0.5639**	**0.5681**
BMP	**0.0130**	**0.0219**	**0.0139**	**0.0138**	**0.0133**	**0.0130**
TIF	**0.0123**	**0.0135**	**0.0128**	**0.0128**	**0.0125**	**0.0127**
JPG	**0.0118**	**0.3774**	**0.0117**	**0.0117**	**0.0119**	**0.0117**
GIF	**0.0118**	**0.1180**	**0.0118**	**0.0118**	**0.0119**	**0.0118**
ZIP	**0.0111**	**0.6301**	**0.0112**	**0.0112**	**0.0119**	**0.0117**
PDFZIP	**0.0285**	**0.2728**	**0.0288**	**0.0288**	**0.0285**	**0.0289**
WAV	**0.0227**	**0.0376**	**0.0229**	**0.0237**	**0.0231**	**0.0228**
MP3-64K	**0.1253**	**0.2567**	**0.1292**	**0.1155**	**0.1153**	**0.1155**
MP3-128K	**0.1089**	**0.2049**	**0.1089**	**0.1101**	**0.1089**	**0.1114**
MP3-256K	**0.0291**	**0.0696**	**0.0297**	**0.0297**	**0.0298**	**0.0299**
MP3-320K	**0.0257**	**0.0810**	**0.0258**	**0.0298**	**0.0284**	**0.0259**

En la siguiente tabla se muestran los valores de la entropía de los archivos de texto claro, así como los valores de la entropía de los archivos de texto cifrado.

Valores correspondientes al cálculo de la Entropía e Información Mutua para los archivos cifrados con el cifrador caótico de bloques variando el parámetro μ.							
Archivo	$H(M) = -\sum_{i=1}^{n} p_i \log p_i$	$H(C) = -\sum_{i=1}^{n} p_i \log p_i$			**I(M,C) = 0**		
		μ=3.6	***μ=3.8***	***μ=3.9***	***μ=3.6***	***μ=3.8***	***μ=3.9***
TXT	5.0803	7.9965	7.9969	7.9965	0.1809	0.1821	0.1803
DOC	5.7806	7.8686	7.8814	7.8752	0.7374	0.7377	0.7346
RTF	5.4569	7.9956	7.9956	7.9955	0.6549	0.6599	0.6518
PDF	7.5589	7.9415	7.9430	7.9419	0.6683	0.6759	0.6712
XLS	5.8964	7.9867	7.9894	7.9846	0.6197	0.6091	0.6998
BMP	7.7959	7.9999	7.9999	7.9999	0.0222	0.0220	0.0219
TIF	7.5952	7.9999	7.9999	7.9999	0.0135	0.0134	0.0135
JPG	7.9176	7.9937	7.9918	7.9939	0.3780	0.3781	0.3774
GIF	7.9806	7.9996	7.9995	7.9995	0.1175	0.1186	0.1180
ZIP	7.9918	7.9980	7.9979	7.9977	0.6242	0.6255	0.6301
PDFZIP	7.9987	7.9989	7.9990	7.9992	0.2702	0.2683	0.2728
WAV	7.4144	7.9998	7.9999	7.9999	0.0372	0.0372	0.0376
MP3-64K	7.6117	7.9987	7.9987	7.9987	0.2580	0.2592	0.2567
MP3-128K	7.8720	7.9903	7.9895	7.9883	0.2010	0.1997	0.2049
MP3-256K	7.9393	7.9993	7.9995	7.9994	0.0694	0.0694	0.0696
MP3-320K	7.8913	7.9976	7.9977	7.9976	0.0818	0.0815	0.0810

Los valores de la entropía de los archivos cifrados se obtuvieron variando el parámetro μ, Así mismo, para determinar dichos valores se consideraron las regiones más densas del mapeo las cuales comienzan a partir de 3.53. Esto se puede ver más claro si se observa la gráfica correspondiente al exponente de Lyapunov para el mapeo logístico en la siguiente figura:

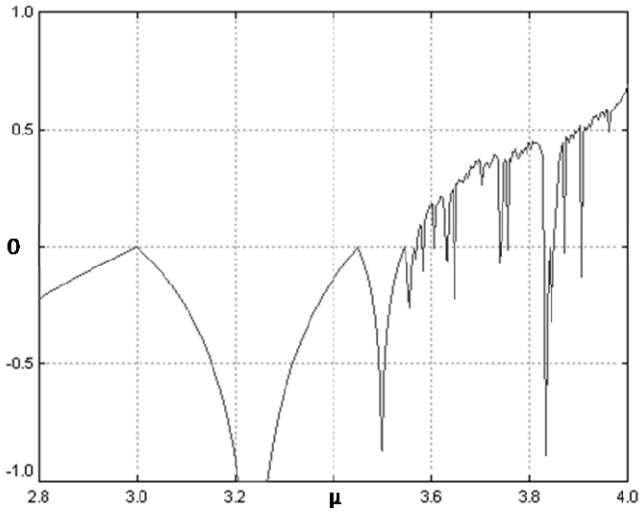


Figura 60 Exponente de Lyapunov del Mapeo Logístico.

El exponente se puede interpretar de la siguiente manera: Cuando los valores de μ están por debajo de cero el sistema presenta un comportamiento estable, pero cuando el valor de μ se vuelve positivo se ha alcanzado los puntos más densos del mapeo, esto es el caos.

Conclusiones

El algoritmo analizado y evaluado pertenece a una clase particular conocida como redes de Feistel. Una parte esencial de una red de Feistel son las funciones de transformación. Este algoritmo hace uso de una función de transformación, la cual utiliza operaciones no lineales de sustitución como parte fundamental en el proceso de cifrado. Cabe mencionar que es común que dichas funciones se construyan ya sea aleatoriamente y/ó logarítmicamente. Nótese que el algoritmo analizado emplea el uso de mapeos caóticos (Mapeo Logístico) como una manera alternativa en la construcción de la función de transformación. Partiendo de esta base, se pueden plantear las siguientes conclusiones:

En esta tesis se ha demostrado el uso de mapeos caóticos como una manera alternativa en la construcción de la función de transformación. Alternativamente, se demostró y por tal motivo se concluye que con la acertada elección de un mapeo caótico que exhiba las propiedades fundamentales de la teoría del caos (mezclado, impredecibilidad de la señal generada y sensibilidad a las condiciones iniciales) y un adecuado procedimiento de discretización se pueden generar funciones de transformación seguras.

El algoritmo fue evaluado usando conceptos fundamentales de Teoría de la información como Entropía, Información Mutua y distribución estadística. Se demostró que la información mutua de las secuencias generadas está altamente relacionada con la distribución estadística de la señal de salida, de manera que si la información mutua es cercana a cero (criterio de criptosistema seguro de Shannon), la distribución estadística del criptosistema es muy parecida a la distribución estadística de una señal de ruido. Finalmente y como parte fundamental en el trabajo de esta tesis, se evaluaron las secuencias generadas desde un punto de vista estadístico usando para ello la suite de pruebas estadísticas de aleatoriedad del NIST. Los resultados mostraron que las secuencias generadas por el cifrador caótico de bloques pasaron exitosamente todas las pruebas, concluyendo así, que el algoritmo diseñado, desarrollado y evaluado, puede ser implementado en sistemas de seguridad de la información.

Trabajos a futuro

Dentro de las aplicaciones más prometedoras de los sistemas caóticos esta su uso en el campo de la criptografía, donde la utilización de sus propiedades no lineales hacen de este tipo de sistemas una buena opción en el desarrollo de algoritmos criptográficos, como lo demostrado en esta tesis. Cabe resaltar que estas propiedades no lineales desencadenan un comportamiento caótico, mismo que es difícil predecir por métodos analíticos sin el conocimiento de la llave secreta (condiciones iniciales/o parámetros). En base a esto y a la continua necesidad de contar con algoritmos de cifrado más robustos, se plantean los siguientes problemas como para su estudio en los trabajos a futuro:

- Análisis de la convergencia fuerte y divergencia de Kullback Leibler como medidas de eficiencia del algoritmo de cifrado para poder comparar que tan parecida es la distribución de la señal de salida, a la distribución de una señal de ruido.

- Revisar el efecto que pudiera causar alguna modificación a la estructura de Feistel usada.

- Mejorar los valores de la Entropía e Información Mutua haciendo uso de algún algoritmo de compresión como Lempel Ziv, Huffman, etc., y así obtener una señal de salida lo más parecida posible a una señal de ruido.

- Comparación con otros esquemas de cifrado que haga uso de funciones no lineales para la construcción de la función de transformación.

Artículos publicados

- Congreso Internacional de Matemáticas Aplicadas AppliedMath III, CD. de México 2007 *"Algoritmo de Cifrado Usando Mapeos Caóticos"*

- **L** Congreso Nacional de Física, Boca del Río Veracruz 2007 *"Análisis de Series de Tiempo Para un Sistema de Partículas Cargadas"*

- Decimonovena Reunión de Otoño de Comunicaciones, Computación, Electrónica y Exposición Industrial ROC&C 2008 *"Algoritmo Cifrador de Bloques Usando Mapeos Caóticos"*

- ODESA Ucrania 2008 *"Numerical Calculation of The Lyapunov Exponent For the Logistic Map"*

Apéndices

PRUEBA DE FRECUENCIA (Monobits)

El propósito de esta prueba es determinar si el número unos y ceros en una secuencia es aproximadamente la misma que se podía esperar para una secuencia verdaderamente aleatoria. El examen evalúa la proximidad de la fracción de unos a ½, es decir, el número de ceros y unos en una secuencia deben de ser semejantes. *Todas las pruebas posteriores dependen de la aprobación de esta prueba.*

Descripción de la prueba

Para esta prueba se toman dos entradas:

n= longitud de la cadena de bits

ε= La secuencia de bits generada por el RNG (generador de números aleatorios)o PRNG(generador de números pseudoaleatorios).

Para la prueba estadística tenemos:

S_{obs}= El valor absoluto de la suma de X_i (donde $X_i=2\varepsilon-1=\pm1$) en la secuencia dividida por la raíz cuadrada de la longitud de la secuencia.

La distribución de referencia para la prueba estadística es la normal media.(Nota: Si Z(donde $z = \frac{S_{obs}}{\sqrt{2}}$) se distribuye como normal, entonces, |z| se distribuye como normal media). Si la secuencia es normal, entonces los mas y los menos unos se tienden a cancelar una contra otra salida para que sea la prueba estadística sobre 0. Si hay muchos unos o muchos ceros, entonces la prueba estadística tiende a ser más grande que cero.

Los pasos para realizar este test son:

1.- Conversión a ±1: Los ceros y los unos de la secuencia de entrada (ε) son convertidas a valores de -1 y +1 y son agregados juntos para producir $S_n = X_1 + X_2 + \cdots + X_n$, donde $X_i = 2\varepsilon_i-1$.

Por ejemplo, si ε=1011010101, entonces n=10 y S_n=1+(-1)+1+1+(-1)+1+(-1)+1+(-1)+1=2.

2.- Calcular la prueba estadística $S_{obs}=\frac{|S_n|}{\sqrt{n}}$

Siguiendo el ejemplo anterior tenemos $S_{obs} = \frac{|2|}{\sqrt{10}} = .632455532.$

3.- Calcular P-value=$erfc\left(\frac{S_{obs}}{\sqrt{2}}\right)$, donde **erfc** es la función error complementaria.

$$erfc(z) = \frac{2}{\sqrt{\pi}}\int_z^{\infty} e^{-u^2}\,du$$

$$P\text{-}value = \textbf{\textit{erfc}}\left(\frac{.632455532}{\sqrt{2}}\right) = 0.527089.$$

Si el cálculo de P-value es <0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia es aleatoria.

PRUEBA PARA FRECUENCIA DENTRO DE UN BLOQUE

El enfoque de la prueba es la proporción los ceros y unos dentro de bloques de M-bits. El propósito de esta prueba es determinar si la frecuencia de unos en un bloque de M-bits es aproximadamente M/2, como se esperaría en un supuesto de aleatoriedad.

Descripción de la prueba

Para esta prueba se toman tres entradas:
M= Longitud de cada bloque.
n= Longitud de la cadena de bits.
ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG(generador de números pseudoaleatorios).
Para la prueba estadística y la distribución de referencia tenemos:
$X^2(obs)$= Una medida de que tan bien la proporción de unos dentro un dado bloque de M-bits observada coincide con la proporción prevista (1/2).
La distribución de referencia para la prueba estadística es una distribución X^2.
Los pasos para realizar esta prueba son:

1.- Partición de la secuencia de entrada en $N = \left[\frac{n}{M}\right]$ bloques no superposicionados. Descartar cualquier bit no utilizado.
Por ejemplo, si n=10, M=3 y ε=0110011010, 3 bloques (N=3) se crearía, compuesta de 011, 001 y 101. El 0 final será descartado.
2.- Determinar la proporción π_i de unos en cada bloque M-bit usando la ecuación

$$\pi_i = \frac{\sum_{j=1}^{M} \varepsilon_{(i-1)M+j}}{M}$$

Para $1 \leq i \leq N$.
Para $\pi_1 = \frac{2}{3}, \pi_2 = \frac{1}{3}\ y\ \pi_3 = \frac{2}{3}$.
3.- Calcular la X^2 estadística: $X^2(obs) = 4M\sum_{i=1}^{N}(\pi_i - \frac{1}{2})^2$.

$$X^2(obs) = 4x3x\left[\left(\frac{2}{3}-\frac{1}{2}\right)^2 + \left(\frac{1}{3}-\frac{1}{2}\right)^2 + \left(\frac{2}{3}-\frac{1}{2}\right)^2\right] = 1$$

4.- Calcular $P - value = \boldsymbol{igamc}\left(\frac{N}{2}, \frac{X^2(obs)}{2}\right)$, donde ***igamc*** es la función gama incompleta para Q(a,x)

$$P(a,x) \equiv \frac{\gamma(a,x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)}\int_0^x e^{-t}t^{a-1}dt$$

donde P(a,0)=0 y P(a,∞)=1

$$P - value = \boldsymbol{igamc}\left(\frac{3}{2}, \frac{1}{2}\right) = 0.801252$$

Con lo anterior descrito si P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario, se concluye que la secuencia es aleatoria.
Para el caso del ejemplo se concluye que si aleatoria.

PRUEBA DE EJECUCION

El objetivo de esta prueba es el número total de ceros y unos ejecutándose en la secuencia completa, en donde una ejecución es una secuencia ininterrumpida de bits idénticos. Una ejecución de longitud k significa que una ejecución consta de exactamente k bits idénticos y que está limitada antes y después con un bit del valor opuesto. El propósito de la prueba de ejecución es determinar si el número de ejecuciones de unos y ceros de diferentes longitudes es como se esperaba para una secuencia aleatoria. En particular, en este examen se determina si la oscilación en tales subcadenas está demasiado rápida o demasiado lenta.

Descripción de la prueba

n= La longitud de la cadena bits
ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG (generador de números pseudoaleatorios).
$V_n(obs)$= El número total de ejecuciones (el número total de ceros corriendo + el número total de unos corriendo) a lo largo de todos los n bits.
La distribución de referencia para la prueba estadística es un distribución X^2 .
La prueba de ejecución consta de los siguientes pasos:
Nota: La prueba de ejecución realiza una prueba de frecuencia como requisito previo.

1.- Calcular π que proporciona la pre-prueba de unos en la secuencia de entrada. $\pi = \frac{\sum_j \varepsilon_j}{n}$
Por ejemplo, si ε=1001101011, entonces n=10 y π=6/10=3/5.

2.- Determinar si se pasa la prueba de frecuencia: Si puede ser demostrada $\left|\pi - 1/2\right| \geq \tau$, entonces la prueba de ejecución no se necesita realizar. Si la prueba no se aplica el valor de $\tau = \frac{2}{\sqrt{n}}$ P-value queda con 0.000. Note que para esta prueba, ha sido predefinida en el código de prueba.
Para el ejemplo tenemos que $\tau = \frac{2}{\sqrt{10}} \approx 0.63246$, entonces $\left|\pi - \frac{1}{2}\right| = \left|\frac{3}{5} - \frac{1}{2}\right| = 0.1 < \tau$, y la prueba no es ejecuta.
Dado que el valor observado π es dentro de los límites seleccionados, la prueba de ejecución es aplicable.

3.- Calcular la prueba estadística $V_n(obs) = \sum_{k=1}^{n-1} r(k) + 1$ donde *r(k)=0* si $\varepsilon_k = \varepsilon_{k+1}$, *r(k)=1* de lo contrario.
Desde ε=1001101011, entonces $V_{10}(obs) = (1+0+1+0+1+1+1+1+0)+1 = 7$.
4.- Calcular $P - value = \boldsymbol{erfc}\left(\frac{|V_n(obs) - 2n\pi(1-\pi)|}{2\sqrt{2n\pi(1-\pi)}}\right)$.
Para el ejemplo, $P - value = \boldsymbol{erfc}\left[\frac{7 - \left(2 \cdot 10 \cdot \frac{3}{5}\left(1 - \frac{3}{5}\right)\right)}{2 \cdot \sqrt{2 \cdot 10} \cdot \frac{3}{5} \cdot \left(1 \cdot \frac{3}{5}\right)}\right] = 0.147232.$
Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBAS PARA LA MÁS LARGA EJECUCIÓN DE UNOS EN UN BLOQUE
El propósito de este examen es determinar si la longitud de la más larga ejecución de unos dentro de la secuencia probada es coherente con la longitud de la más larga ejecución de unos de las que podría esperarse en una secuencia aleatoria. Note que una irregularidad en la longitud esperada de la más larga ejecutar de unos implica que hay también una irregularidad en la longitud esperada de la ejecución más larga de ceros. Por lo tanto, sólo una prueba es necesaria.

Descripción de la prueba

n= La longitud de la cadena bits
ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG (generador de números pseudoaleatorios).
M= la longitud de cada bloque. El código de prueba ha sido preestablecido para dar cabida a tres valores para M; M=8, M=128 y M=10^4 en conformidad con la siguiente tabla.

***n* mínima**	**M**
128	8
6272	128
750,000	10^4

N= El numero de bloques; seleccionando en conformidad con el valor de M.
Para la prueba estadística y distribución de referencia.
$X^2(obs)$= Una medida de que tan bien la longitud más larga de ejecución dentro de bloques de M-bits observada coinciden con la longitud más larga esperada dentro de bloques de M-bits.
La distribución de referencia para la prueba estadística es una distribución X^2.
La prueba de la más larga ejecución de unos en un bloque de M-bits consiste en lo siguiente:
1.- Dividir la secuencia en bloques de M-bits.
Por ejemplo, para el caso donde K=3 y M=8

ε = 11001100000101010110100010011001110000000000001001
00110101010001000100111101011010000000110101111100
11001110011011011000010110010

$n = 128$

Subblock	Max-Run	Subblock	Max-Run
11001100	(2)	00010101	(1)
01101100	(2)	01001100	(2)
11100000	(3)	00000010	(1)
01001101	(2)	01010001	(1)
00010011	(2)	11010110	(2)
10000000	(1)	11010111	(3)
11001100	(2)	11100110	(3)
11011000	(2)	10110010	(2)

2.- Tabular la frecuencia V_i de la más larga ejecución de unos en cada bloque entre categorías, donde cada celda contiene el número de ejecuciones de unos de una longitud dada.
Para el valor de M compatibles con el código de prueba, Las celdas V_i tendrá los siguientes datos:

v_i	M = 8	M = 128	$M = 10^4$
v_0	≤ 1	≤ 4	≤ 10
v_1	2	5	11
v_2	3	6	12
v_3	≥ 4	7	13
v_4		8	14
v_5		≥ 9	15
v_6			≥ 16

$v_0 = 4;\ v_1 = 9;\ v_2 = 3;\ v_4 = 0;$"

3.- Calcular $\chi^2(obs) = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i}$ donde el valor para π_i son proporcionados en la siguiente tabla:

$$K = 3, M = 8$$

classes	$\{\nu \leq 1\}$	$\{\nu = 2\}$	$\{\nu = 3\}$	$\{\nu \geq 4\}$
probabilities	$\pi_0 = 0.2148$	$\pi_1 = 0.3672$	$\pi_2 = 0.2305$	$\pi_3 = 0.1875$

$$K = 5, M = 128$$

classes	$\{\nu \leq 4\}$	$\{\nu = 5\}$	$\{\nu = 6\}$	$\{\nu = 7\}$
probabilities	$\pi_0 = 0.1174$	$\pi_1 = 0.2430$	$\pi_2 = 0.2493$	$\pi_3 = 0.1752$

$\{\nu = 8\}$	$\{\nu \geq 9\}$
$\pi_4 = 0.1027$	$\pi_5 = 0.1124$

$$K = 5, M = 512$$

classes	$\{\nu \leq 6\}$	$\{\nu = 7\}$	$\{\nu = 8\}$	$\{\nu = 9\}$
probabilities	$\pi_0 = 0.1170$	$\pi_1 = 0.2460$	$\pi_2 = 0.2523$	$\pi_3 = 0.1755$

$\{\nu = 10\}$	$\{\nu \geq 11\}$
$\pi_4 = 0.1015$	$\pi_5 = 0.1077$

$$K = 5, M = 1000$$

classes	$\{\nu \leq 7\}$	$\{\nu = 8\}$	$\{\nu = 9\}$	$\{\nu = 10\}$
probabilities	$\pi_0 = 0.1307$	$\pi_1 = 0.2437$	$\pi_2 = 0.2452$	$\pi_3 = 0.1714$

$$\begin{array}{cc} \{\nu = 11\} & \{\nu \geq 12\} \\ \pi_4 = 0.1002 & \pi_5 = 0.1088 \end{array}$$

$$K = 6, M = 10000$$

$$\begin{array}{cccccc} \text{classes} & \{\nu \leq 10\} & \{\nu = 11\} & \{\nu = 12\} & \{\nu = 13\} \\ \text{probabilities} & \pi_0 = 0.0882 & \pi_1 = 0.2092 & \pi_2 = 0.2483 & \pi_3 = 0.1933 \end{array}$$

$$\begin{array}{ccc} \{\nu = 14\} & \{\nu = 15\} & \{\nu \geq 16\} \\ \pi_4 = 0.1208 & \pi_5 = 0.0675 & \pi_6 = 0.0727 \end{array}$$

El valor de K y de N es determinado por el valor de M de acuerdo con la siguiente tabla:

M	**K**	**N**
8	3	16
128	5	49
10^4	6	75

Para el ejemplo tenemos;

$$\chi^2(obs) = \frac{(4-16(.2148))^2}{16(.2148)} + \frac{(9-16(.3672))^2}{16(.3672)} + \frac{(3-16(.2305))^2}{16(.2305} + \frac{(0-16(.1875))^2}{16(.1875)} = 4.882605$$

4.- Calcular.

Para elejemplo, $$P\text{-}value = \mathbf{igamc}\left(\frac{3}{2}, \frac{4.882605}{2}\right) = 0.180598.$$

Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBA DE RANGO DE MATRICES ALEATORIAS BINARIAS.

El objetivo de la prueba es el rango de sub-matrices separadas de la secuencia completa. El propósito de esta prueba es buscar la dependencia lineal en subcadenas de longitud fija de la secuencia original. Tenga en cuenta que este examen también aparece en la batería DIEHARD de pruebas.

Descripción de la prueba

n= La longitud de la cadena bits

$$P\text{-}value = \mathbf{igamc}\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right)$$

ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG (generador de números pseudoaleatorios).

M= El numero de filas en cada matriz. Para el conjunto de pruebas, M se ha establecido a 32. Si otro valor de M es usado, se necesita calcular una nueva aproximación.

Q= El numero de columnas en cada matriz. Para el conjunto de pruebas, Q se ha establecido a 32. Si otro valor de Q es usado, se necesita calcular una nueva aproximación.
Para la prueba estadística y la distribución de referencia tenemos:
$X^2(obs)$= Una medida de que tan bien los números observados del rango de varios ordenes coincide con el número esperado de rangos bajo una supuesta aleatoriedad.
La distribución de referencia para la prueba estadística es una distribución X^2.
La prueba de rango de matrices binarias aleatorias consiste en:
1.- Secuencialmente dividir la secuencia en M•Q-bit bloques separados; existirá $N = \left[\frac{n}{MQ}\right]$ en esos bloques. Los bits descartados se reportaran como no usados en el cálculo dentro de cada bloque. Recopilar los segmentos M•Q bits en M por Q matrices. Cada fila de la matriz está llena con sucesivos bloques Q-bits de la secuencia original ε.
Para el ejemplo, si n=20, M=Q=3, y ε=01011001001010101101, entonces se particiona la secuencia en $N = \left[\frac{n}{3\cdot 3}\right] = 2$ matrices de cardinalidad M•Q(3•3=9). Note que los últimos 2 bits (0 y 1) serán descartados. Las dos matrices son $\begin{vmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{vmatrix}$ y $\begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix}$. Note que la primera matriz de los primeros tres bits en la fila 1, la conjunto de tres bits en la fila 2, y el tercer conjunto de tres bits en la fila 3. La segunda matriz se construye similarmente usando los siguientes nueve bits en la secuencia.
2.- Determinar la rango binaria (R_ℓ) de cada matriz, donde ℓ=1,…,N.
Para el ejemplo, el rango de la primera matriz es 2(R_1=2), y la rango de la segunda matriz es 3(R_2=3).
3.- Tenemos F_M= el numero de matrices con R_ℓ=M (rango completo)
F_{M-J}= el numero de matrices con R_ℓ=M-1 (rango completo)
N-F_M-F_{M-1}=el numero de matrices restantes.
Para el ejemplo, F_M=F_3=1 (R_2 tiene el rango completo de 3), F_{M-J}=F_2=1 (R_1 tiene rango 2), y ninguna matriz tiene cualquier rango inferior.
4.- Calcular

$$\chi^2(obs) = \frac{(F_M - 0.2888N)^2}{0.2888N} + \frac{(F_{M-1} - 0.5776N)^2}{0.5776N} + \frac{(N - F_M - F_{M-1} - 0.1336N)^2}{0.1336N}$$

Para el ejemplo quedaría:

$$\chi^2(obs) = \frac{(1 - 0.2888 \bullet 2)^2}{0.2888 \bullet 2} + \frac{(1 - 0.5776 \bullet 2)^2}{0.5776 \bullet 2} + \frac{(2 - 1 - 1 - 0.1336 \bullet 2)^2}{0.1336 \bullet 2} = 0.596953.$$

5.- Calcular $P - value = e^{-\chi^2(obs)/2}$. Puesto que hay 3 clases en el ejemplo, la P-value para el ejemplo es igual a $igamc\left(1, \frac{\chi^2(obs)}{2}\right)$.

$$P\text{-}value = e^{0.596953/2} = 0.741948$$

Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBA DE LA TRANSFORMADA DISCRETA DE FOURIER (ESPECTRAL)

El objetivo de esta prueba son los picos altos en la transformación de Fourier discreta de la secuencia. El propósito de esta prueba es detectar características periódicas (es decir, repetitivos patrones que están cerca uno de otro) en la secuencia probada que se indica una desviación de supuesta aleatoriedad. La intención es detectar si el número de picos que excedan el umbral de 95 % es significativamente diferente de 5 %.

Descripción de la prueba

n= La longitud de la cadena bits

ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG (generador de números pseudoaleatorios).

d= La diferencia normalizada entre lo observado y lo esperado del número de componentes de frecuencia que están más allá del umbral de 95 %.

La distribución de referencia para la prueba estadística es una distribución normal.

Para la prueba de la transformada discreta de Fourier se tiene que:

1.- Conversión a ±1: Los ceros y los unos de la secuencia de entrada (ε) son convertidas a valores de -1 y +1 y son agregados juntos para producir $S_n = X_1, X_2, \ldots, X_n$ donde $X_i = 2\varepsilon_i - 1$

Por ejemplo, si n=10 y ε=1001010011, entonces X=1,-1,-1,1,-1,1,-1,-1,1,1.

2.- Aplicar una transformada discreta de Fourier (DFT) en X para producir: S=DFT(X). Una secuencia de variables complejas se produce la cual representa los componentes periódicos de la secuencia de bits a diferentes frecuencias.

3.- Calcular M=*modulus*(S')≡|S'|, donde S es la subcadena compuesta por los primeros N/2 elementos en S, y la función mudulus produce una secuencia de picos altos.

4.- Calcular $T = \sqrt{3n} = al\ 95\%\ del\ valor\ umbral\ del\ pico\ alto$. Bajo una suposición de aleatoriedad, 95% de los valores obtenidos de la prueba no debería exceder a T.

5.- Calcular N_0=.95n/2. N_0 es teóricamente el esperado (95%)numero de picos (bajo la suposición de aleatoridad) que es menor que T.

Para el ejemplo, N_0=4.75.

6.- Calcular N_1=al actual número de picos observado en M que es menor que T.

Para el ejemplo, N_1=4.

7.- Calcular $d = \dfrac{(N_1 - N_0)}{\sqrt{n(.95)(.05)/2}}$.

Para el ejemplo

$$d = \frac{(4 - 4.75)}{\sqrt{10(.95)(.05)/2}} = -1.538968$$

8.- Calcular $P\text{-}value = erfc\left(\dfrac{|d|}{\sqrt{2}}\right)$

Para el ejemplo,

$$P\text{-}value = erfc\left(\frac{1.538968}{\sqrt{2}}\right) = 0.123812$$

Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBA DE PATRONES DE PAREAMIENTO QUE NO SE SUPERPONEN

El objetivo de este examen es el número de apariciones de cadenas de destino pre-especificados. El propósito de esta prueba es detectar generadores que producen demasiadas repeticiones en un determinado patrón (aperiódico) no periódico. Para esta prueba y para la siguiente prueba, una ventana de m-bits se utiliza para buscar un patrón de m-bits específico. Si no se encuentra el patrón, la ventana se desliza un bit de posición. Si se encuentra el patrón, la ventana se restablece al bit después del patrón encontrado y reanuda la búsqueda.

Descripción de la prueba

m= La longitud en bits de cada patrón. El patrón es la cadena destino.

n= La longitud de la cadena de bits todo bajo prueba.

ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG (generador de números pseudoaleatorios).

B= El patrón m-bit para asociarse; B es una cadena de unos y ceros (de longitud m) cual está definida en una librería de patrones no periódicos contenidos dentro del código de prueba.

M= La longitud en bits de la subcadena de ε para ser probada. M se ha establecido en 131,072 en el código de la prueba.

N= El numero de bloques independientes. N se ha establecido en 8 en el código de la prueba.

Para la prueba estadística y la distribución de referencia tenemos:

$X^2(obs)$= Una medida de que tan bien el número observado de patrón “hits” coincide con el número esperado de patrón “hits” (bajo una supuesta aleatoriedad).

La distribución de referencia para la prueba estadística es una distribución X^2.

Para esta prueba se procede a lo siguiente:

1.- Particionar la secuencia en N bloques independientes de longitud M.

Por ejemplo, si ε=10100100101110010110, entonces n=20. Si N=2 y M=10, entonces los dos bloques deben ser 1010010010 y 1110010110.

2.- Tenemos $W_j(j=1,\ldots,N)$ que es el número de veces que B (el patrón)se encuentra dentro del bloque j. Note que j=1,…,N. La búsqueda de coincidencias continúa creando una ventana de m-bits sobre la secuencia, que compara los bits dentro de esa ventana contra el patrón. Si no hay coincidencias, la ventana se desliza un bit, por ejemplo si m=3 y la ventana actual contiene bits del 3 al 5, entonces la siguiente ventana contendrá bits del 4 al 6. Si hay una coincidencia, la ventana se desliza m bits, por ejemplo si la actual ventana contiene bits del 3 al 5 entonces la siguiente ventana contendrá bits del 6 al 8.

Siguiendo con el ejemplo, si m=3 y el patrón B=001, entonces el proceso de examinación es como sigue:

Posiciones de bits	Bloque 1		Bloque 2	
	Bits	**W_1**	**Bits**	**W_2**
1-3	101	0	111	0
2-4	010	0	110	0
3-5	100	0	100	0
4-6	001(hit)	Incrementa a 1	001(hit)	Incrementa 1
5-7	No examinado		No examinado	
6-8	No examinado		No examinado	
7-9	001	Incrementa a 2	011	1
8-10	010(hit)	2	110	1

Así, W_1=2 y W_2=1.

3.- Bajo una suposición de aleatoriedad, calcular teóricamente la media μ y la varianza σ^2:

$$\mu = (M-m+1)/2^m \qquad \sigma^2 = M\left(\frac{1}{2^m} - \frac{2m-1}{2^{2m}}\right)$$

Para el ejemplo, $\mu = \frac{10-3+1}{2^3} = 1, y\ \sigma^2 = 10 \cdot \left(\frac{1}{2^3} - \frac{2\cdot 3-1}{2^{2\cdot 3}}\right) = 0.26875$

4.- Calcular $\chi^2(obs) = \sum_{j=1}^{N} \frac{(W_j - \mu)^2}{\sigma^2}$.

Para el ejemplo tenemos, $\chi^2(obs) = \frac{(2-1)^2 + (1-1)^2}{0.46875} = \frac{1+0}{0.46875} = 2.133333$

5.- Calcular $P\text{-}value = \mathbf{igamc}\left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right)$. Note que los múltiples P-value serán calculados. Para m=9, hasta 148 P-values pueden ser calculados; para m=10, hasta 248 P-values pueden ser calculados.

Para el ejemplo tenemos que, $P\text{-}value = \mathbf{igamc}\left(\frac{2}{2}, \frac{2.133333}{2}\right) = 0.344154$

Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBA DE PATRONES DE PAREAMIENTO QUE SE SUPERPONEN

El enfoque de la prueba de patrones de pareamiento que se superponen es el número de apariciones de cadenas de destino pre-especificados. Tanto esta prueba y la prueba anterior utilice una ventana de m-bits para buscar de un patrón de m-bits específico. Como con la prueba anterior, si no se encuentra el patrón, la ventana se desliza un bit de posición. La diferencia entre este examen y la prueba anterior es que cuando se encuentra el patrón, la ventana se desliza sólo un poco antes de reanudar la búsqueda.

Después de la prueba

m= La longitud en bits del patrón- en esta caso, la longitud de la ejecución de unos.

n= La longitud de la cadena de bits.

ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG (generador de números pseudoaleatorios).
K= El numero de grados de libertad. K se ha establecido en 5 en el código de prueba.
B= El patrón m-bit para asociarse.
M= La longitud en bits de una subcadena de ε para ser probada. M se ha establecido en 1032 en el código de la prueba.
N= El numero de bloques independientes de n. N se ha establecido en 986 en el código de la prueba.
Para la prueba estadística y la distribución de referencia tenemos:
$X^2(obs)$= Una medida de que tan bien el número observado de patrón "hits" coincide con el número esperado de patrón "hits" (bajo una supuesta aleatoriedad).
La distribución de referencia para la prueba estadística es una distribución X^2.
1.- Particionar la secuencia en N bloques independientes de longitud M.
Por ejemplo, si ε=1011101111**00101101000**1110010111**0111111000**0101101001, entonces n=50. Si K=2, M=10 y N=5, entonces el bloque cinco son 1011101111, 0010110100, 0111001011, 1011111000, y 0101101001.
2.- Calcular el número de ocurrencias de B en cada uno de los bloques N. La búsqueda de coincidencias continúa creando una ventana de m-bits sobre la secuencia, comparar los bits dentro de esa ventana contra B e incrementar un contador cuando hay una coincidencia. La ventana se desliza un bit después de cada examen, por ejemplo si m=4 y la primera ventana contiende bits del 42 al 45, la siguiente ventana consiste de los bits 43 al 46. Registrar el número de repeticiones de B en cada bloque para incrementar una matriz v_i (donde i=0,…,5), tal que v_0 se incrementa cuando no hay repeticiones de B en una subcadena, v_1 es incrementado para una repetición de B,… y v_5 es incrementado para 5 o más repeticiones de B.

Si m=2 y B=11, entonces el examen del primer bloque (1011101111) procesado como sigue:

Posiciones de Bit	**Bits**	**No. De repeticiones de B=11**
1-2	10	0
2-3	01	0
3-4	11(hit)	Incrementa a 1
4-5	11(hit)	Incrementa a 2
5-6	10	2
6-7	01	2
7-8	11(hit)	Incrementa a 3
8-9	11(hit)	Incrementa a 4
9-10	11(hit)	Incrementa a 5

Por lo tanto, después de bloque 1, hay cinco repeticiones de 11, v_5 es incrementado, y v_0=0, v_1=0, v_2=0, v_4=0, y v_5=1.
De manera similar, los bloques 2-5 son examinados. En el bloque 2, hay 2 repeticiones de 11; v_2 es incrementado. En el bloque 3, hay 3 repeticiones de 11; v_3 es incrementado. En el

bloque 4, hay 4 repeticiones de 11; v_4 es incrementado. En el bloque 5, hay una repetición de 11; v_1 es incrementado.
Por lo tanto, v_0=0, v_1=1, v_2=1, v_3=1, v_4=1, v_5=1 después todos los bloques se han examinado.
3.- Calcular valores para λ y η que se usara para calcular la probabilidad teórica de π_i correspondiendo a la clase de v_0:

$$\lambda = (M-m+1)/2^m \qquad \eta = \lambda/2$$

Para el ejemplo, $\lambda=(10-2+1)/2^2=2.25$, y $\eta=\lambda/2=1.125$.
4.- Calcular $\chi^2(obs) = \sum_{i=0}^{5} \frac{(v_i - N\pi_i)^2}{N\pi_i}$, donde π_0=0.367879, π_1=0.183940, π_2=0.137955, π_3=0.099634, π_4=0.069935, y π_5=0.140657.
Para el ejemplo, los valores de π_i son recalculados, ya que el ejemplo no se ajusta a los requerimientos. Los valores de π_i son: π_0=0.324652, π_1=0.182617, π_2=0.142670, π_3=0.106645, π_4=0.0077147, y π_5=0.166269.

$$\chi^2(obs) = \frac{(0-5\bullet 0.324652)^2}{5\bullet 0.324652} + \frac{(1-5\bullet 0.182617)^2}{5\bullet 0.182617} + \frac{(1-5\bullet 0.142670)^2}{5\bullet 0.142670} + \frac{(1-5\bullet 0.106645)^2}{5\bullet 0.106645} + \frac{(1-5\bullet 0.077147)^2}{5\bullet 0.077147} + \frac{(1-5\bullet 0.166269)^2}{5\bullet 0.166269} = 3.167729.$$

5.- Calcular $P\text{-}value = \mathbf{igamc}\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right)$
Para el ejemplo,

$$P\text{-}value = \mathbf{igamc}\left(\frac{5}{2}, \frac{3.167729}{2}\right) = 0.274932.$$

Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

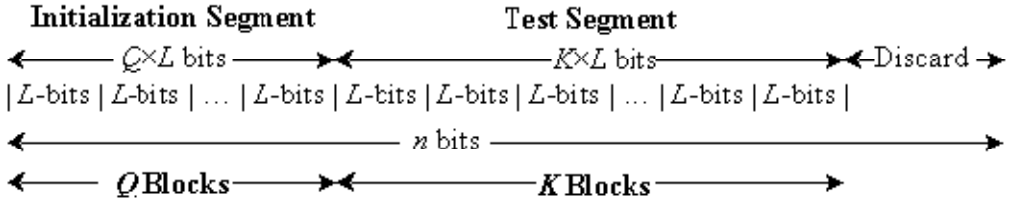
PRUEBA DE "ESTADÍSTICA UNIVERSAL" DE MAURER

El objetivo de este examen es el número de bits entre patrones coincidentes (una medida que está relacionada con la longitud de una secuencia comprimida). El propósito de la prueba es detectar si o no puede comprimirse significativamente la secuencia sin pérdida de información. Una secuencia significativamente comprimible se considera que no es aleatoria.

Descripción de la prueba

n= La longitud de la cadena de bits.
ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG (generador de números pseudoaleatorios).
L= la longitud de cada bloque. Note que el uso de la L como tamaño del bloque no es coherente con la notación del tamaño de bloque (M) usado para otras pruebas. Sin embargo, el uso de L como el tamaño del bloque fue especificado en el código original de la prueba de Maurer.
Q= El numero de bloques en la secuencia de inicialización.
Para la prueba estadística y la distribución de referencia tenemos:
f_n= La suma de la distancia log_2 entre el pareamiento de patrones de L-bits.

La distribución de referencia para la prueba estadística es una distribución media normal como es también el caso de la prueba de frecuencia.
El procedimiento de esta prueba es:
1.- La secuencia n-bit (ε) es particionada en dos segmentos; un segmento de inicialización consiste compuesto de Q bloques no superpuestos de L-bit., y un segmento de prueba compuesta de K bloques no superpuestos de L-bit. Los bits restantes al final de la secuencia que no forman un bloque completo de L-bits son descartados.



Los primeros Q bloques son usados para inicializar la prueba. Los restantes K bloques son los bloques de prueba (K=|n/L|-Q).
Por ejemplo, si ε=01011010011101010111, entonces n=20. Si L=2 y Q=4, entonces $K = [n/L] - Q = [20/2] - 4 = 6$. El segmento de inicialización es 01011010; el segmento de prueba es 011101010111.

Los bloques de L-bits son mostrados en la siguiente tabla:

Bloque	**Tipo**	**Contenido**
1	Segmento de inicialización	01
2		01
3		10
4		10
5	Segmento de prueba	01
6		11
7		01
8		01
9		01
10		11

2.- usar el segmento de inicialización, una tabla es creada para cada posible valor L-bit. El numero de bloque de la ultima repetición de cada bloque L-bit es anotado en la tabla.
Para el ejemplo la siguiente tabla es creada usando los 4 bloques de inicialización.

	Posible valor de L-bit			
	00 (salvado en T_0)	01 (salvado en T_1)	10 (salvado en T_2)	11 (salvado en T_3)
Inicialización	0	2	4	0

3.- Examinar cada uno de los bloques K en el segmento prueba y determinar el número de bloques desde la última aparición del mismo bloque de L-bit. Remplazar el valor en la

tabla con la locación del bloque actual. Agregar el cálculo de la distancia en las repeticiones del mismo bloque L-bit para un acumulativo de la suma de log_2 de todas las diferencias detectadas en los bloques K.
Para el ejemplo, la tabla y la suma acumulada son desarrolladas como sigue:
Para el bloque 5 (el 1er bloque prueba); 5 es colocado en el "01" filas de la tabla y suma=log_2 (5-2)=1.584962501.
Para el bloque 6; 6 es colocado en el "11" filas de la tabla y suma=1.584962501+ log_2(6-0)=4.169925002.
Para el bloque 7; 7 es colocado en el "01" filas de la tabla y suma=4.169925002+log_2 (7-5)=5.169925002.
Para el bloque 8; 8 es colocado en el "01" filas de la tabla y suma=5.169925002+log_2 (8-7)= 5.169925002
Para el bloque 9; 9 es colocado en el "01" filas de la tabla y suma=5.169925002+log_2 (9-8)= 5.169925002.
Para el bloque 10; 10 es colocado en el "11" filas de la tabla y suma=5.169925002+log_2 (10-6)=7.169925002.

Los estados de la tabla son:

Iteración de bloque	**Posible valor L-bit**			
	00	**01**	**10**	**11**
4	0	2	4	0
5	0	5	4	0
6	0	5	4	6
7	0	7	4	6
8	0	8	4	6
9	0	9	4	6
10	0	9	4	10

4.- Calcular el prueba estadística: $f_n = \frac{1}{K}\sum_{i=Q+1}^{Q+K} log_2(i - T_j)$ donde T_j es la entrada de la tabla asociada a la representación decimal de los contenidos de el i^mo bloque de L-bit.

Para el ejemplo tenemos, $f_n = \frac{7.169925002}{6} = 1.1949875.$

5.- Calcular $P\text{-}value = erfc\left(\left|\frac{f_n - expectedValue(L)}{\sqrt{2}\sigma}\right|\right)$ y expectedValue(L) y σson tomados de una tabla de valores[2] precalculados. Bajo la suposición de aleatoriedad, la semejante igualdad, expectedValue(L), es el esperado valor teórico del cálculo estadístico para dar la longitud L-bit. La derivada estándar teórica es dada por $\sigma = c\sqrt{\frac{variance(L)}{K}}$ donde $c = 0.7 - \frac{0.8}{L} + \left(4 + \frac{32}{L}\right)\frac{K^{-3/L}}{15}$

L	expectedValue	Varianza
6	5.2177052	2.954
7	6.1962507	3.125
8	7.1836656	3.238
9	8.1764248	3.311
10	9.1723243	3.356
11	10.170032	3.384

L	expectedValue	Varianza
12	11.168765	3.401
13	12.168070	3.410
14	13.167693	3.416
15	14.167488	3.419
16	15.167379	3.421

Para el ejemplo, $$P\text{-}value = erfc\left(\left|\frac{1.1949875 - 1.5374383}{\sqrt{2}\sqrt{1.338}}\right|\right) = 0.767189$$

Note que el valor esperado y la varianza para L=2 no son provistos en la tabla, des un bloque de longitud dos no se recomienda para pruebas. Sin embargo, este valor para L es fácil para usarse en un ejemplo.

Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBA DE COMPRESION LEMPEL-ZIV

El objetivo de este examen es el número de patrones acumulativamente distintos (palabras) en la secuencia. El propósito de la prueba es determinar cuánto se puede comprimir la secuencia probada. La secuencia es considera no-aleatorios si puede comprimir significativamente. Una secuencia aleatoria tendría un número característico de patrones distintos.

Descripción de la prueba.

n= La longitud de la cadena de bits.

ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG (generador de números pseudoaleatorios).

$W_{(obs)}$= El número de palabras disjuntas y acumulativamente distintos en la secuencia.

La distribución de referencia para la prueba estadística es una distribución normal.

1.- Analizar la secuencia en palabras consecutivas, disjuntas y distintas que formarán un diccionario de palabras en la secuencia. Esto se consigue mediante la creación de subcadenas de bits consecutivos de la secuencia hasta que se crea una subcadena que no se ha encontró anteriormente en la secuencia. La subcadena resultante es una palabra nueva en el diccionario.

Tenemos W_{obs}= El número de palabras acumulativamente distintas.

Por ejemplo, si ε=010110010, entonces el examen continúa como sigue:

Posición del Bit	Bit	¿Nueva palabra?	La palabra es:
1	0	Si	0(Bit 1)
2	1	Si	1(Bit 2)
3	0	No	
4	1	Si	01(3-4 bits)
5	1	No	
6	0	Si	10(5-6 bits)
7	0	No	
8	1	No	
9	0	Si	010(7-9 bits)

Hay 5 palabras en el "diccionario" : 0, 1, 01, 010. Por lo tanto, W_{obs}=5.

2.- Calcular $P\text{-}value = \frac{1}{2}\, erfc\left(\frac{\mu - W_{obs}}{\sqrt{2\sigma^2}}\right)$ donde η=69586.25 y $\sigma = \sqrt{70.448718}$ cuando $n=10^6$. Para otros valores de n, los valores de μ y σ necesitan ser calculados. Tenga en cuenta que no hay teoría conocida disponible para determinar los valores exactos de μ y σ, estos valores son calculados usando SHA-1. El generador de Blum-Blum-Shub dará resultados similares para μ y σ^2.

Dado que el ejemplo es mucho menor que la longitud recomendada, los valores para μ y σ^2 no son válidos. En su lugar, supongamos que la prueba se realizó en una secuencia de bits de un millón, y se obtuvo el valor W_{obs}=69600, entonces $P\text{-}value = \frac{1}{2}\, erfc\left(\frac{69586.25 - 69600}{\sqrt{2 \bullet 70.448718}}\right) = 0.949310$.

Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBA DE COMPLEJIDAD LINEAL

El objetivo de este examen es la longitud de un registro de desplazamiento con retroalimentación lineal (LFSR). El propósito de este examen es determinar si o no la secuencia es lo suficientemente compleja para ser considerada aleatoria. Las secuencias aleatorias se caracterizan por LFSRs largos. Un LFSR que es demasiado corto implica no aleatoriedad.

Descripción de la prueba

n= La longitud de la cadena de bits
ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG (generador de números pseudoaleatorios).
M= La longitud en bits de un bloque.
K= el numero de grados de libertad; K=6.
Para la prueba estadística y la distribución de referencia tenemos:
$X^2(obs)$= Una medida de que tan bien el número observado de repeticiones de longitud fija LFSRs coincide con el número esperado de repeticiones en un supuesto de aleatoriedad.

La distribución de referencia para la prueba estadística es una distribución X^2.
El procedimiento consiste en:
1.- Particionar la secuencia n-bit en N bloques independientes de M bits, donde n=MN.
2.- Usar el algoritmo Berlekamp-Massey, determinar la complejidad lineal L_i de cada uno de los bloques N (i=1,…,N). L_i es la longitud de la más corta secuencia del registro de desplazamiento con retroalimentación lineal que genera todos los bits en el bloque i. En cualquier secuencia L_i-bit, algunas combinaciones de los bits, cuando se agrega junto al módulo 2, se produce el siguiente bit en la secuencia (bit L_i+1).
Por ejemplo, si M=13 y el bloque que se va a probar es 1101011110001, entonces L_i=4, y la secuencia es producida agregando el 1er y 2do bits dentro de una subsecuencia del bit 4 para producir el siguiente bit (el 5to bit). El examen procedió como sigue:

Los primeros 4 bits y el 5to bit resultante:
2-5 bits y el 6to bit resultante:
3-6 bits y el 7mo bit resultante:

9-12 bits y el 13mo bit resultante:

Bit 1	Bit 2	Bit 3	Bit 4	Bit 5
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	0
1	1	1	0	0
1	1	0	0	0
1	0	0	0	1

Para este bloque, el algoritmo de desplazamiento de prueba funciona. Si este no fuera el caso, otros algoritmos de desplazamiento se intentaran para el bloque.
3.- Bajo una suposición de aleatoriedad, calcular la media teórica μ:

$$\mu = \frac{M}{2} + \frac{\left(9+(-1)^{M+1}\right)}{36} - \frac{\left(M/3+2/9\right)}{2^M}$$

Para el ejemplo, $$\mu = \frac{13}{2} + \frac{\left(9+(-1)^{13+1}\right)}{36} - \frac{\left(13/3+2/9\right)}{2^{13}} = 6.777222$$
4.- Para cada subcadena, calcular un valor de T_i, donde $T_i=(-1)^M \bullet (L_i-\mu)+2/9$.
Para el ejemplo, $$T_i = (-1)^{13}(4-6.777222)+2/9 = 2.999444$$
5.- Registrar los valores T_i en $v_0,\ldots,v_6$ como sigue:
Si:

$$\begin{aligned} T_i \le -2.5 &\quad incrementa\ v_0\ por\ uno \\ -2.5 < T_i \le -1.5 &\quad incrementa\ v_1\ por\ uno \\ -1.5 < T_i \le -0.5 &\quad incrementa\ v_2\ por\ uno \\ -0.5 < T_i \le 0.5 &\quad incrementa\ v_3\ por\ uno \\ 0.5 < T_i \le 1.5 &\quad incrementa\ v_4\ por\ uno \\ 1.5 < T_i \le 2.5 &\quad incrementa\ v_5\ por\ uno \\ T_i > 2.5 &\quad incrementa\ v_6\ por\ uno \end{aligned}$$

6.- Calcular $\chi^2(obs) = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i}$ donde π_0=0.01047, π_1=0.03125, π_2=0.125, π_3=0.5, π_4=0.25, π_5=0.0625, π_6=0.02078.

7.- Calcular $P\text{-}value = \mathbf{igamc}\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right)$

Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBA DE SERIE

El enfoque de este examen es la frecuencia de todos los posibles patrones de m-bits superpuestos a través de toda la secuencia. El propósito de este examen es determinar si el número de apariciones de los patrones superpuestos de 2^m m-bits es aproximadamente la misma que se podía esperar para una secuencia aleatoria. Las secuencias aleatorias tienen uniformidad; es decir, todo patrón de m-bits tiene las mismas posibilidades de aparecer como todo patrón de m-bits de otro. Note que para m=1, la prueba de serie es equivalente a la prueba de frecuencia.

Descripción de la prueba

m= La longitud en bits de cada bloque.
n= La longitud en bits de la cadena de bits.
ε= La secuencia de bits generada por el RNG (generador de números aleatorios) o PRNG (generador de números pseudoaleatorios).
Para la prueba estadística y la distribución de referencia tenemos:
$\nabla\Psi_m^2(obs)$ y $\nabla^2\Psi_m^2(obs)$ =Una medida de que tan bien las frecuencias observadas de patrones de m-bits coinciden con las frecuencias de los patrones de m-bits esperados.
La distribución de referencia para la prueba estadística es una distribución X^2.
Para la prueba se tiene que hacer:
1.- Formar una secuencia aumentada de ε': Ampliar la secuencia anexando los primeros m-1 bits al final de la secuencia para distintos valores de n.
Por ejemplo, dado n=10 y ε=0011011101. Si m=3, entonces ε'=001101110100. Si m=2, entonces ε'=00110111010. Si m=1, entonces ε'=a la secuencia original 0011011101.
2.- Determinar la frecuencia de todos los posibles bloques de m-bits superpuestas, todos los posibles bloques de (m-1)-bits superpuestos y todos los posibles bloques de (m-2)-bits superpuestos. Tenemos $v_{i_1 \ldots i_m}$ denota la frecuencia del patrón de m-bit $i_1 \ldots i_m$; tenemos $v_{i_1 \ldots i_{m-1}}$ denota la frecuencia del patrón (m-1)-bit $i_1 \ldots i_{m-1}$; y tenemos $v_{i_1 \ldots i_{m-2}}$ denota la frecuencia del patrón (m-2)-bit $i_1 \ldots i_{m-2}$.
Para el ejemplo, donde m=3, entonces (m-1)=2, y (m-2)=1. La frecuencia de todos los bloques de 3-bits es: v_{000}=0, v_{001}=1, v_{010}=1, v_{011}=2, v_{100}=1, v_{101}=2, v_{110}=2, v_{111}=0. La frecuencia de todos los posibles bloques (m-1)-bit es: v_{00}=1, v_{01}=3, v_{10}=3, v_{11}=3. La frecuencia de todos los bloques de (m-2)-bit es: v_0=4, v_1=6.
3.- Calcular:

$$\psi_m^2 = \frac{2^m}{n}\sum_{i_1\ldots i_m}\left(v_{i_1\ldots i_m} - \frac{n}{2^m}\right)^2 = \frac{2^m}{n}\sum_{i_1\ldots i_m} v_{i_1\ldots i_m}^2 - n$$

$$\psi_{m-1}^2 = \frac{2^{m-1}}{n}\sum_{i_1\ldots i_{m-1}}\left(v_{i_1\ldots i_{m-1}} - \frac{n}{2^{m-1}}\right)^2 = \frac{2^{m-1}}{n}\sum_{i_1\ldots i_{m-1}} v_{i_1\ldots i_{m-1}}^2 - n$$

$$\psi_{m-2}^2 = \frac{2^{m-2}}{n}\sum_{i_1\ldots i_{m-2}}\left(v_{i_1\ldots i_{m-2}} - \frac{n}{2^{m-2}}\right)^2 = \frac{2^{m-2}}{n}\sum_{i_1\ldots i_{m-2}} v_{i_1\ldots i_{m-2}}^2 - n$$

Para este ejemplo

$$\psi^2{}_3 = \frac{2^3}{10}(0+1+1+4+1+4+4+1) - 10 = 12.8 - 10 = 2.8$$

$$\psi^2{}_2 = \frac{2^2}{10}(1+9+9+9) - 10 = 11.2 - 10 = 1.2$$

$$\psi^2{}_1 = \frac{2}{10}(16+36) - 10 = 10.4 - 10 = 0.4$$

4.- Calcular

$$P\text{-}value1 = \mathbf{igamc}\left(2^{m-2}, \nabla\psi_m^2\right) \text{ and}$$

$$P\text{-}value2 = \mathbf{igamc}\left(2^{m-3}, \nabla^2\psi_m^2\right).$$

Para el ejemplo tenemos:

$$P\text{-}value1 = \mathbf{igamc}\left(2, \frac{1.6}{2}\right) = 0.9057$$

$$P\text{-}value2 = \mathbf{igamc}\left(1, \frac{0.8}{2}\right) = 0.8805.$$

Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

TEST DEL APROXIMADO DE LA ENTROPIA

Como con la prueba de serie, el enfoque de este examen es la frecuencia de todos los posibles patrones de m-bits superpuestos a través de toda la secuencia. El propósito de la prueba es comparar la frecuencia bloques superpuestos de dos longitudes consecutivos/adyacentes (m y m+1) contra el resultado esperado para una secuencia aleatoria.

Descripción de la prueba

m= La longitud de cada bloque en este caso, la longitud del primer bloque usado en la prueba. m+1 es la longitud del segundo bloque usado.
n= La longitud de toda la secuencia de bits.

ε= La secuencia de bits generada por el RNG (generador de números aleatorios)o PRNG(generador de números pseudoaleatorios).
Para la prueba estadística y la distribución de referencia tenemos:
$X^2(obs)$= Una medida de que tan bien el valor de ApEn(m) observado coincide con el valor esperado.
La distribución de referencia para la prueba estadística es una distribución X^2.
Para esta prueba se procede a lo siguiente:
1.- Aumentar la secuencia de n-bits para crear n secuencias de m-bits superpuestos para anexar m-1 bits desde el principio de la secuencia para el final de la secuencia.
Por ejemplo, si ε=0100110101 y m=3, entonces n=10. Agregar el 0 y 1 del principio de la secuencia al final de la secuencia. La secuencia que se va a probar se convierte en 010011010101. (Nota: Esto se realiza para cada valor de m.)
2.- Un conteo de frecuencia es hecho de los n bloques superpuestos. Permite que el conteo de los valores posibles de m-bits ((m+1)-bit) se representa como C_i^m, donde i es el valor m-bits.
Para el ejemplo, los bloques de m-bits superpuestos (donde m=3) convertido 010, 100, 001, 011, 101, 010, 101, 010 y 101. Los conteos calculados para $2^m=2^3=8$ posibles l cadenas de m-bits son:

$$\#000 = 0,\ \#001 = 1,\ \#010 = 3,\ \#011 = 1,\ \#100 = 1,\ \#101 = 3,\ \#110 = 1,\ \#111 = 0$$

3.- Calcular $C_i^m = \frac{\#i}{n}$ para cada valor de i.
Para el ejemplo $C^3_{000} = 0,\ C^3_{001} = 0.1,\ C^3_{010} = 0.3,\ C^3_{011} = 0.1,\ C^3_{100} = 0.1,\ C^3_{101} = 0.3,\ C^3_{110} = 0.1,\ C^3_{111} = 0.$
4.- Calcular $\varphi^{(m)} = \sum_{i=0}^{2^m-1} \pi_i log\, \pi_i$ donde $\pi_i=C_j^3$, donde $j=log_2 i$.
Para el ejemplo se tiene:

$$\varphi^{(3)} = 0(\log 0) + 0.1(log 0.1) + 0.3(\log 0.3) + 0.1(\log 0.1) + 0.1(log 0.1) + 0.3(\log 0.3) + 0.1(\log 0.1) + 0(\log 0) = -1.64341772.$$

5.- Repita los pasos del 1 al 4, reemplazar m por m+1.
6.- Calcular la prueba estadística: $\chi^2 - 2n[log\ 2 \quad ApEn(m)]$ donde $ApEn(m) = \varphi^{(m)} - \varphi^{(m+1)}$
Para el ejemplo tenemos

$$ApEn(3) = -1.643418 - (-1.834372) = 0.190954$$
$$\chi^2 = 2 \bullet 10(0.693147 - 0.190954) = 0.502193$$

7.- Calcular $P\text{-}value = \mathbf{igamc}(2^{m-1}, \frac{\chi^2}{2})$
Para el ejemplo $P\text{-}value = \mathbf{igamc}\left(2^2, \frac{0.502193}{2}\right) = 0.261961.$
Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBA DE LA SUMA ACUMULATIVA

El propósito del examen es determinar si la suma acumulativa de las secuencias parciales que ocurren en la secuencia probada es demasiado grande o demasiado pequeño relativo al comportamiento esperado de esa suma acumulativa para secuencias aleatorios. Esta suma acumulativa puede considerarse como un paseo aleatorio. Para una secuencia aleatoria, las excursiones del paseo de aleatoria deben de estar cerca de cero. Para ciertos tipos de secuencias no aleatorias, las excursiones de este paseo aleatorio desde cero serán grandes.

Descripción de la prueba

n= La longitud de la secuencia de bits.
ε= La secuencia de bits generada por el RNG (generador de números aleatorios)o PRNG(generador de números pseudoaleatorios).
mode= Un interruptor para aplicar la prueba ya sea por delante de la secuencia de entrada(mode=0) o por atrás de la secuencia (mode=1).
Para la prueba estadística y la distribución de referencia tenemos:
z= La excursión más grande desde el origen de las sumas acumulados en la correspondiente (-1, 1) secuencia.
La distribución de referencia para la prueba estadística es una distribución normal.
Para esta prueba se procede a:
1.- Formar una secuencia normalizada: los ceros y los uno de la secuencia de entrada (ε) son convertidos a valores X_i de -1 y +1 usando $X_i=2\varepsilon_i-1$.
Por ejemplo, si ε=1011010111, entonces X=1,-1, 1, 1, -1, 1, -1, 1, 1, 1.
2.- Calcular la suma parcial S_i de las subsecuencias sucesivamente más grandes, cada inicio con X_1 (si mode=0)o X_n(si mode=1).

Mode=0 (delante)	Mode=1(atras)
$S_1 = X_1$	$S_1 = X_n$
$S_2 = X_1 + X_2$	$S_2 = X_n + X_{n-1}$
$S_3 = X_1 + X_2 + X_3$	$S_3 = X_n + X_{n-1} + X_{n-2}$
.	.
.	.
$S_k = X_1 + X_2 + X_3 + \ldots + X_k$	$S_k = X_n + X_{n-1} + X_{n-2} + \ldots + X_{n-k+1}$
.	.
.	.
$S_n = X_1 + X_2 + X_3 + \ldots + X_k + \ldots + X_n$	$S_n = X_n + X_{n-1} + X_{n-2} + \ldots + X_{k-1} + \ldots + X_1$

Esto es $S_k=S_{k-1}+X_k$ para mode 0, y $S_k=S_{k-1}+X_{n-k+1}$ para mode 1.
Para el ejemplo, cuando mode=0 y X=1,-1, 1, 1, -1, 1, -1, 1, 1, 1, entonces:

$S_1 = 1$
$S_2 = 1 + (-1) = 0$
$S_3 = 1 + (-1) + 1 = 1$
$S_4 = 1 + (-1) + 1 + 1 = 2$
$S_5 = 1 + (-1) + 1 + 1 + (-1) = 1$
$S_6 = 1 + (-1) + 1 + 1 + (-1) + 1 = 2$
$S_7 = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) = 1$
$S_8 = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 = 2$
$S_9 = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + 1 = 3$
$S_{10} = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + 1 + 1 = 4$

3.- Calcular la prueba estadística $z = max_{1\le k\le n}|S_k|$, donde $max_{1\le k\le n}|S_k|$ es el más grande de los valores absolutos de la suma parcial S_k.
Para el ejemplo tenemos que el más grande valor de S_k es 4, y z=4.
4.- Calcular

$$P\text{-}value = 1 - \sum_{k=\left(\frac{-n}{z}+1\right)/4}^{\left(\frac{n}{z}-1\right)/4}\left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}}\right)\right] + \sum_{k=\left(\frac{-n}{z}-3\right)/4}^{\left(\frac{n}{z}-1\right)/4}\left[\Phi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right)\right]$$

Donde φ es la Función Normal Estándar de la Distribución de la Probabilidad Acumulativa
En el caso del ejemplo P-value=0.4116588.
Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBA DE EXCURSIONES ALEATORIOS

El objetivo de esta prueba es el número de ciclos teniendo exactamente K visitas en un paseo de aleatoriedad de la suma acumulativa. El paseo de aleatoriedad de la suma acumulativa se deriva de sumas parciales después de la secuencia (0,1) se transfiere a la adecuada secuencia (-1, 1). Un ciclo de un paseo aleatorio consiste en una secuencia de pasos de unidad de longitud tomados al azar que comienza y vuelve al origen. El objetivo de esta prueba es determinar si el número de visitas a un estado particular dentro de un ciclo se desvía de lo que se espera para una secuencia aleatoria. Esta prueba consiste en realidad, de una serie de ocho pruebas (y conclusiones), una prueba y la conclusión para cada uno de los Estados:-4, -3, -2, -1 y 1, 2, 3, 4.

Descripción de la prueba

n= La longitud de la secuencia de bits.
ε= La secuencia de bits generada por el RNG (generador de números aleatorios)o PRNG(generador de números pseudoaleatorios).
Para la prueba estadística y la distribución de referencia tenemos:
$X^2(obs)$= Para un dado estado x, una medida de que tan bien el número observado de visitas de estado dentro de un ciclo coinciden con el número esperado de visitas de estado dentro de un ciclo, en un supuesto de aleatoriedad
La distribución de referencia para la prueba estadística es una distribución X^2.
Para esta prueba se procede a lo siguiente:

1.- Formar una secuencia normalizada (-1,+1) X: los ceros y los uno de la secuencia de entrada (ε) son convertidos a valores de -1 y +1 usando $X_i=2\varepsilon_i-1$.
Por ejemplo, si ε=0110110101, entonces X= -1, 1, 1, -1, 1, 1, -1, 1, -1, 1.
2.- Calcular la suma parcial S_i de las subsecuencias sucesivamente más grandes, cada inicio con X_1. Forma el sistema S={S_i}.

$$S_1 = X_1$$
$$S_2 = X_1 + X_2$$
$$S_3 = X_1 + X_2 + X_3$$
.
.
$$S_k = X_1 + X_2 + X_3 + \ldots + X_k$$
.
.
$$S_n = X_1 + X_2 + X_3 + \ldots + X_k + \ldots + X_n$$

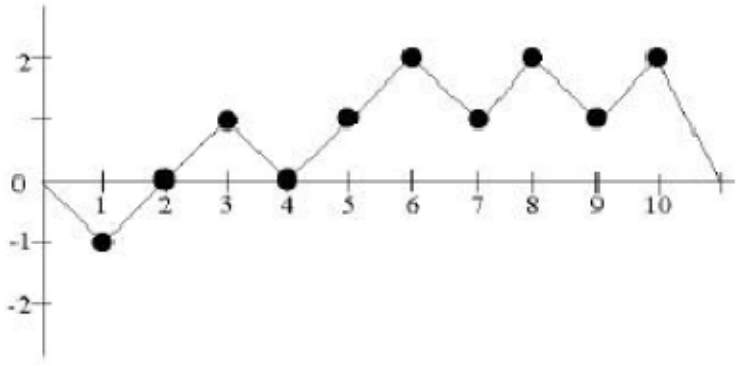
Para el ejemplo de esta sección tenemos:

$S_1 = -1$ $\quad$ $S_6 = 2$
$S_2 = 0$ $\quad$ $S_7 = 1$
$S_3 = 1$ $\quad$ $S_8 = 2$
$S_4 = 0$ $\quad$ $S_9 = 1$
$S_5 = 1$ $\quad$ $S_{10} = 2$

El sistema S={-1, 0, 1, 0, 1, 2, 1, 2, 1, 2}.

3.- Formar una nueva secuencia S' adjuntando ceros antes y después del sistema S. Eso es, S'=0, s_1, s_2,…, s_n, 0.

Para el ejemplo tenemos, S'=0, -1, 0, 1, 0, 1, 2, 1, 2, 1, 2, 0. El paseo aleatorio resultante es mostrado a continuación.



Ejemplo paseo aleatorio (S')

4.- Tenemos J= al numero total de cruces cero en S', donde un cruce cero es un valor de cero en S' esto sucede después del cero inicial. J es también el numero de ciclos en S', donde un ciclo de S' es una subsecuencia de S' compuesta de una ocurrencia de cero, seguido por los valores no ceros, y terminando con otro cero. El cero final en un ciclo puede ser el cero principiando en otro ciclo. El numero de ciclos en S' es el numero de cruces cero. Si J<500, discontinua la prueba.

Para el ejemplo, si S'={0, -1, 0, 1, 0, 1, 2, 1, 2, 1, 2, 0}, entonces J=3(hay ceros en la posición 3, 5 y 12 de S'). Los cruces cero son fácilmente observados por encima del trazado. Desde J=3, hay 3 ciclos, consistiendo de {0, -1, 0}, {0, 1, 0} y {0, 1, 2, 1, 2, 1, 2, 0}.

5.- Para cada ciclo y para cada valor de estado distinto de cero x tiene valores $-4 \leq x \leq -1$ y $1 \leq x \leq 4$, calcular la frecuencia de cada x dentro de cada ciclo.

Para el ejemplo, en el paso 3, el primer ciclo tiene una recurrencia de -1, el segundo ciclo tiene una recurrencia de 1 y el tercer ciclo tiene tres recurrencia cada una de 1 y 2. Esto se puede ver en la siguiente tabla.

Estado X	**Ciclos**		
	Ciclo 1 (0, -1, 0)	**Ciclo 2 (0, 1, 0)**	**Ciclo 3 (0, 1, 2, 1, 2, 1, 2, 0)**
-4	0	0	0
-3	0	0	0
-2	0	0	0
-1	1	0	0
1	0	1	3
2	0	0	3
3	0	0	0
4	0	0	0

6.- Para cada uno de los ocho estados de x, el cálculo de $V_k(x)$= el número total de ciclos en cual estado x se presenta exactamente k veces entre todos los ciclos, para k=0, 1, …, 5 (para k=5, todas las frecuencias ≥ 5 son almacenadas en $v_5(x)$). Note que $\sum_{k=0}^{5} v_k(x) = J$

Para el ejemplo:

- $v_0(-1)$= 2 (el estado -1 aparece exactamente 0 veces en dos ciclos),
  $v_1(-1)$=1 (el estado -1 aparece solo una vez en 1 ciclo), y
  $v_2(-1)=v_3(-1)=v_4(-1)=v_5(-1)=0$ (el estado -1 aparece exactamente {2, 3, 4≥5} veces en 0 ciclos).

- $v_0(1)$= 1 (el estado 1 aparece exactamente 0 veces en un ciclo),
  $v_1(1)$=1 (el estado 1 aparece solo una vez en 1 ciclo),
  $v_3(1)$=1 (el estado 1 aparece exactamente tres veces en un ciclo), y
  $v_2(1)=v_4(1)=v_5(1)=0$ (el estado 1 aparece exactamente {2, 4, ≥5} veces en 0 ciclos).

- $v_0(2)= 2$ (el estado 2 aparece exactamente 0 veces en 2 ciclos),
  $v_3(2)=1$ (el estado 2 aparece exactamente tres veces en un ciclo), y
  $v_1(2)=v_2(2)=v_4(2)=v_5(2)=0$ (el estado 1 aparece exactamente {1, 2, 4, ≥5} veces en 0 ciclos).

- $v_0(-4)=3$ (el estado -4 aparece exactamente 0 veces en un ciclo), y
  $v_1(-4)=v_2(-4)=v_3(-4)=v_5(-4)=0$ (el estado -4 aparece exactamente {1, 2, 3, 4, ≥5} veces en 0 ciclos).

Esto se puede mostrar usando la siguiente tabla:

Estado	**Numero de ciclos**					
x	**0**	**1**	**2**	**3**	**4**	**5**
-4	3	0	0	0	0	0
-3	3	0	0	0	0	0
-2	3	0	0	0	0	0
-1	2	1	0	0	0	0
1	1	1	0	1	0	0
2	2	0	0	1	0	0
3	3	0	0	0	0	0
4	3	0	0	0	0	0

7.- Para cada uno de los ocho estados de x, calcular la prueba estadística $\chi^2(obs) = \sum_{k=0}^{5} \frac{(v_k(x) - J\pi_k(x))^2}{J\pi_k(x)}$ donde $\pi_k(x)$ es le probabilidad que el estado x aparezca k veces en una distribución aleatoria.
Para el ejemplo cuando x=1,

$$\chi^2 = \frac{(1-3(0.5))^2}{3(0.5)} + \frac{(1-3(0.25))^2}{3(0.25)} + \frac{(0-3(0.125))^2}{3(0.125)} + \frac{(1-3(0.0625))^2}{3(0.0625)} + \frac{(0-3(0.0312))^2}{3(0.0312)} + \frac{(0-3(0.0312))^2}{3(0.0312)}$$
$$= 4.333033$$

8.- Para cada estado de x, calcular $P\text{-}value = \mathbf{igamc}(5/2,\ \chi^2(obs)/2)$. Ocho P-value serán producidas.

$$P\text{-}value = \mathbf{igamc}\left(\frac{5}{2}, \frac{4.333033}{2}\right) = 0.502529$$

Para el ejemplo cuando x=1,
Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

PRUEBA DE LA VARIANTE DE EXCURSIONES ALEATORIAS
El objetivo de este examen es el número total de veces que un estado particular es visitado (es decir, ocurre) en un paseo aleatorio de suma acumulativa. El propósito de esta prueba es detectar desviaciones del número esperado de visitas a varios estados en el paseo aleatorio.

Esta prueba consiste en realidad una serie de la dieciocho pruebas (y conclusiones), de una prueba y la conclusión para cada uno de los estados:-9,-8,..., -1 y 1, 2,..., 9.

Descripción de la prueba

n= La longitud de la secuencia de bits.

ε= La secuencia de bits generada por el RNG (generador de números aleatorios)o PRNG(generador de números pseudoaleatorios).

Para la prueba estadística y la distribución de referencia tenemos:

ξ= Para un dado estado x, el número total de veces que el estado dado es visitado durante el paseo aleatorio entero como se determina en el paso cuatro de la prueba anterior.

La distribución de referencia para la prueba estadística es la media normal. (Nota: si ξ es distribuido como normal, entonces $|\xi|$ es distribuido como la media normal.) Si la secuencia es aleatoria, entonces la prueba estadística será grande.

Para esta prueba se procede a lo siguiente:

1.- Formar una secuencia normalizada (-1,+1) X en la cual los ceros y los uno de la secuencia de entrada (ε) son convertidos a valores de -1 y +1 usando $X=X_1, X_2,\ldots,X_n$, donde $X_i=2\varepsilon_i-1$.

Por ejemplo, si ε=0110110101, entonces X= -1, 1, 1, -1, 1, 1, -1, 1, -1, 1.

2.- Calcular la suma parcial S_i de las subsecuencias sucesivamente más grandes, cada inicio con X_1. Forma el sistema $S=\{S_i\}$.

$$S_1 = X_1$$
$$S_2 - X_1 + X_2$$
$$S_3 = X_1 + X_2 + X_3$$
.
.
$$S_k = X_1 + X_2 + X_3 + \ldots + X_k$$
.
.
$$S_n = X_1 + X_2 + X_3 + \ldots + X_k + \ldots + X_n$$

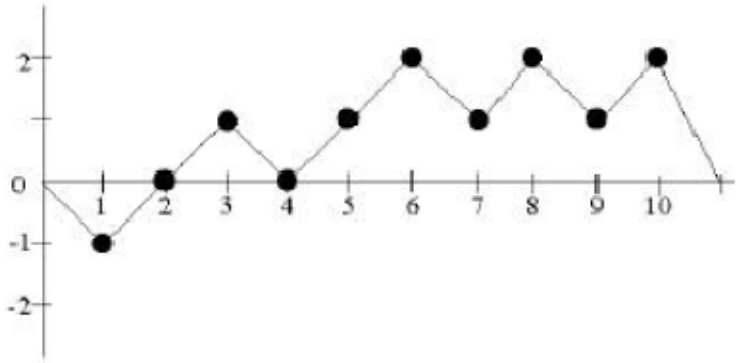
Para el ejemplo de esta sección tenemos:

$S_1 = -1$ $\quad$ $S_6 = 2$
$S_2 = 0$ $\quad$ $S_7 = 1$
$S_3 = 1$ $\quad$ $S_8 = 2$
$S_4 = 0$ $\quad$ $S_9 = 1$
$S_5 = 1$ $\quad$ $S_{10} = 2$

El sistema S={-1, 0, 1, 0, 1, 2, 1, 2, 1, 2}.

3.- Formar una nueva secuencia S' adjuntando ceros antes y después del sistema S. Eso es, $S'=0, s_1, s_2,\ldots, s_n, 0$.

Para el ejemplo tenemos, S'=0, -1, 0, 1, 0, 1, 2, 1, 2, 1, 2, 0. El paseo aleatorio resultante es mostrado a continuación.



Ejemplo paseo aleatorio (S')

4.- Para cada uno de los dieciocho estados de x, calcular $\xi(x)$= el numero total de veces que el estado x ocurre a través de todos los ciclos J.
Para el ejemplo, $\xi(-1) = 1, \xi(1) = 4, \xi(2) = 3$, y todos los otros $\xi(x)$=0.

5.- Para cada $\xi(x)$, calcular $P\text{-}value = \boldsymbol{erfc}\left(\frac{|\xi(x) - J|}{\sqrt{2J(4|x| - 2)}}\right)$. Dieciocho P-values son calculados.

Para el ejemplo cuando x=1, $P\text{-}value = \boldsymbol{erfc}\left(\frac{|4 - 3|}{\sqrt{2 \bullet 3(4|1| - 2)}}\right) = 0.683091$

Si el cálculo de P-value es < 0.01, entonces se concluye que la secuencia no es aleatoria. De lo contrario se concluye que la secuencia si es aleatoria.

Referencias

A.L. Goldberg. [2008] "Caos y fractales en la fisiología Humana," Investigación y Ciencia, Vol. 163.

Álvarez, G., Montoya, F., Romera, M. & Pastor, G. [1999b] "Chaotic cryptosystems," in L. D. Sanson, ed., *Proc. 33rd Annual 1999 International Carnahan Conference on Security Technology*, 332-338 (IEEE).

Aono, Shuichi., Nishio, Yoshifumi. [2007] "A Chaotic Cryptosystem Using Lyapunov Exponent," The 15th IEEE International Workshop on Nonlinear Dynamics of Electronic Systems
NDES'07, Tokushima, Japan, July 23-26.

B. Rubén, C. Isaac, Campos. Eric. [2006] "Transmisión y Recepción de Voz Empleando Caos," Encuentro de Investigación en Ingeniería Eléctrica, Zacatecas, Zac, Abril 5-7.

Boccaletti, S., Kurths, J., Osipov, G., Valladares, D. & Zhou, C. [2002] "The synchronization of chaotic systems," *Phys. Rep*. 366, 1-101.

C. Burwick, D. Coppersmith, E.D' Avignon, R. Gennaro, S. Halevi, C. Juttla, S. M. Matyas, L. O' Connor, M. Peyravian, D. Safford, and N. Zunic, "MARS: A candidate cipher for AES, NIST AES proposal," June 1998.

Devaney, R. L. [1989] *An introduction to Chaotic Dynamical Systems* (Addison-Wesley, Redwood City, California, USA).

H. Feistel, "Cryptography and computer privacy," *Scientific American*, Vol. 228, No. 5, pp. 15-33, 1973.

Hasler, M. [1998] "Synchronization of chaotic systems and transmission of information," *Int. J. Bifurc. Chaos* 8, 647-659.

J.L. Massey, "SAFER K-64: A byte orientated block-ciphering algorithm," in *Fast Software Encryption,* R. Anderson, Ed. Berlin, Germany: Springer, 1993, (LNCS 809), pp. 1-17.

Kocarev, L. [2001] "Chaos-based cryptography: A brief overview," *IEEE Circuits and Systems Magazine* 1, 6-21.

Lars R. Knudsen and David Wagner, "On the structure of Skipjack," *Discrete Applied Mathematics*, 111 (1-2): 103-116, 2001.

Li, C., Li, S., Chen, G., Chen, G. & Hu, L. [2005a] "Crytanalisys of a new signal security system for multimedia data transmission," *EURASIP J. Appl. Signal Process*. 2005, 1277-1288.

Li, C., Li, S., Zhang, D. & Chen, G. [2005b] "cosen-plaintext cryptanalysis of a clipped-neural-network-based chaotic cipher," in *Advances in Neural Networks – ISNN 2005: Second International Symposium on Neural Networks, Chongging, China, May 30 – June1, 2005, Proceedings, Part II, Lecture Notes in Computer Science*, vol. 3497, 630-636 (Springer-Verlag).

Li, S. [2003] *Analyese and New Designs of Digital Chaotic Ciphers,* PhD thesis, School of Electronics and Information Engieering, Xi'an Jiaotong University, Xi'an, China, available on line at http://www.hooklee.com/pub.html

Li, S. [2004] "Digital Chaotic Ciphers," Center for Chaos Control and Synchronization (CCCS) Department of Electronic Engineering, City University of Hong Kong, HK SAR, China.

M. Blaze and B. Schneier, "The MacGuffin Block Cipher Algorithm," in *Fast Software Encryption Second Int. Workshop Proc,.* Berlin Germany; Springer-Verlag, 1995. Pp. 97-110.

May, R. M. [1976] "Theorical Ecology: principles and applications" Blackwell Scientific Publishers

Matt Blaze and Bruce Schneier, "The MacGuffin block cipher algorithm," In Bart Preneel, editor, *Fast Software Encryption: Second International Workshop*, volume 1008 in *Lecture Notes in Computer Science*, pages 97-110 Springer-Verlag, 14-16 December 1994.

Menezes, A. J., van Oorschot, P. C. & Vanstone, S. A. [1997] *Handbook of Applied Cryptography* (CRC Press).

Nieto de Alba, Ubaldo., [2008] "Predicción y Caos en Economía" Universidad Complutense.

Pecora, L. M. & Carroll, T. L. [1990] "Synchronization in chaotic systems," *Phys. Lett. A* 64, 821-824.

Peng, J. [2008] "Research on A Block Encryption Cipher Based on Chaotic Dynamical System," in *Proc. IEEE Third International Conference On Natural Computation (ICNC)*.

R. Anderson and E. Biham, "Two Practical and provably secure block ciphers: BEAR and LION," *in Fast Software Encryption, Third Int. Workshop Proc.* Berlin, Germany: Springer-Verlag, 1996, pp. 113-120.

Ruggiero, D., Pedaci, I., Amato, P. & Kocarev, L. [2004] *"Analysis of the chaotic dynamic of Rijndael block cipher,"* in *Proc*. *RISP Int. Workshop on Nonlinear Circuit and Signal Processing* (NCSP'04), 77-80.

Shannon, C.E., 1949, Communication Theory of Secrecy Systems, Bell Technical Journal, vol. 28-4, 1949, pp. 656 – 715.

Silva, C. P. & Young, A. M. [2000] "Introduction to chaos-based communications and signal processing," in *Proc. IEEE Aerospace Conference*, 279-299.

Soto, J., Statistical Testing of Random Number Generators, http://csrc.nist.gov/rng/nissc-paper.pdf.

Wei J, Liao X, Wong KW, Xiang T. A new chaotic cryptosystem. Chaos, Solitons & Fractals 2006;30:1143-52.

X. Lai and J.L. Massey, "A proposal for a new block encryption standard," in *Advance in Cryptology-EUROCRYPT'90.* Berlin: Springer-Verlag, 1991, pp. 389-404.
Yang, T. [2004] "A survey of chaotic secure communications systems," *Int. J. Comp. Cognition* 2, 81-130.